Atefeh Farzindar
Vladimir Kešelj (Eds.)

LNAI 6085

Advances in Artificial Intelligence

23rd Canadian Conference
on Artificial Intelligence, Canadian AI 2010
Ottawa, Canada, May/June 2010
Proceedings

Springer

Lecture Notes in Artificial Intelligence 6085

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

Subseries of Lecture Notes in Computer Science

Atefeh Farzindar   Vlado Kešelj (Eds.)

Advances in Artificial Intelligence

23rd Canadian Conference
on Artificial Intelligence, Canadian AI 2010
Ottawa, Canada, May 31 – June 2, 2010
Proceedings

Springer

Series Editors

Randy Goebel, University of Alberta, Edmonton, Canada
Jörg Siekmann, University of Saarland, Saarbrücken, Germany
Wolfgang Wahlster, DFKI and University of Saarland, Saarbrücken, Germany

Volume Editors

Atefeh Farzindar
NLP Technologies Inc.
1255 University Street, Montreal, Quebec H3B 3W9, Canada
E-mail: farzindar@nlptechnologies.ca

Vlado Kešelj
Dalhousie University, Faculty of Computer Science
6050 University Ave, Halifax, Nova Scotia B3H 1W5, Canada
E-mail: vlado@cs.dal.ca

Library of Congress Control Number: 2010926372

CR Subject Classification (1998): I.3, H.3, I.2.7, H.4, F.1, H.5

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-642-13058-5 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-13058-8 Springer Berlin Heidelberg New York

springer.com

Printed in Germany

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

Preface

This volume contains the papers presented at the 23rd Canadian Conference on Artificial Intelligence (AI 2010). The conference was held in Ottawa, Ontario, from May 31 to June 2, 2010, and was collocated with the 36th Graphics Interface Conference (GI 2010), and the 7th Canadian Conference on Computer and Robot Vision (CRV 2010).

The Program Committee received 90 submissions for the main conference, AI 2010, from across Canada and around the world. Each submission was reviewed by up to four reviewers. For the final conference program and for inclusion in these proceedings, 22 regular papers, with allocation of 12 pages each, were selected. Additionally, 26 short papers, with allocation of 4 pages each, were accepted.

The papers from the Graduate Student Symposium are also included in the proceedings: six oral (four pages) and six poster (two pages) presentation papers.

The conference program featured three keynote presentations by Dekang Lin (Google Inc.), Guy Lapalme (Université de Montréal), and Evangelos Milios (Dalhousie University). The one-page abstracts of their talks are also included in the proceedings.

Two pre-conference workshops, each with their own proceedings, were held on May 30, 2010. The Workshop on Intelligent Methods for Protecting Privacy and Confidentiality in Data was organized by Khaled El Emam and Marina Sokolova. The workshop on Teaching AI in Computing and Information Technology (AI-CIT 2010) was organized by Danny Silver, Leila Kosseim, and Sajid Hussain.

This conference would not have been possible without the hard work of many people. We would like to thank all Program Committee members and external reviewers for their effort in providing high-quality reviews in a timely manner. We thank all the authors of submitted papers for submitting their work, and the authors of selected papers for their collaboration in preparation of the final copy. Many thanks to Ebrahim Bagheri and Marina Sokolova for organizing the Graduate Student Symposium, and chairing the Program Committee of the symposium.

We are in debt to Andrei Voronkov for developing the EasyChair conference management system and making it freely available to the academic world. It is an amazingly elegant and functional Web-based system, which saved us much time.

The conference was sponsored by the Canadian Artificial Intelligence Association (CAIAC), and we thank the CAIAC Executive Committee for the constant support. We would like to express our gratitude to Robert Laganière, the AI/GI/CRV General Chair, and Diana Inkpen, the AI Local Organizing Chair, as well as the other Organizing Chairs, for making AI/GI/CRV 2010 an enjoyable experience.

March 2010

Atefeh Farzindar
Vlado Kešelj

Organization

AI/GI/CRV 2010 General Chair

Robert Laganière	Univeristy of Ottawa

AI Program Committee Chairs

Atefeh Farzindar	NLP Technologies Inc. and Université de Montréal
Vlado Kešelj	Dalhousie University

AI Local Organizing Chair

Diana Inkpen	University of Ottawa

Graduate Student Symposium Chairs

Ebrahim Bagheri	National Research Council Canada and Athabasca University
Marina Sokolova	Children's Hospital of Eastern Ontario (University of Ottawa)

AI 2010 Program Committee

Esma Aïmeur	Université de Montréal
Massih-Reza Amini	National Research Council of Canada
Aijun An	York University
Dirk Arnold	Dalhousie University
Ebrahim Bagheri	National Research Council of Canada
Sabine Bergler	Concordia University
Scott Buffett	National Research Council of Canada
Cory Butz	University of Regina
Maria Fernanda Caropreso	University of Ottawa
Nick Cercone	York University
Yllias Chali	University of Lethbridge
Collin Cherry	National Research Council of Canada
Robin Cohen	University of Waterloo
Lyne Da Sylva	Université de Montréal

Douglas Dankel	University of Florida
Chrysanne DiMarco	University of Waterloo
Christopher Drummond	National Research Council of Canada
Atefeh Farzindar	NLP Technologies Inc. and UdeM
Yong Gao	University of British Columbia Okanagan
Dragan Gasevic	Simon Fraser University
Cyril Goutte	National Research Council of Canada
Robert Hilderman	University of Regina
Graeme Hirst	University of Toronto
Jimmy Huang	York University
Frank Hutter	University of British Columbia
Diana Inkpen	University of Ottawa
Nathalie Japkowicz	University of Ottawa
Igor Jurisica	University of Toronto
Froduald Kabanza	Université de Sherbrooke
Vlado Kešelj	Dalhousie University
Ziad Kobti	University of Windsor
Grzegorz Kondrak	University of Alberta
Leila Kosseim	Concordia University
Adam Krzyzak	Concordia University
Philippe Langlais	Université de Montréal
Guy Lapalme	Université de Montréal
Oscar Lin	Athabasca University
Hongyu Liu	National Research Council of Canada
Alejandro Lopez-Ortiz	University of Waterloo
Alan Mackworth	University of British Columbia
Yannick Marchand	National Research Council of Canada
Joel Martin	National Research Council of Canada
Stan Matwin	University of Ottawa
Gordon McCalla	University of Saskatchewan
Robert Mercer	University of Western Ontario
Evangelos Milios	Dalhousie University
Malek Mouhoub	University of Regina
David Nadeau	National Research Council of Canada
Eric Neufeld	University of Saskatchewan
Jian-Yun Nie	Université de Montréal
Gerald Penn	University of Toronto
Fred Popowich	Simon Fraser University
Doina Precup	McGill University
Robert Reynolds	Wayne State University
Denis Riordan	Dalhousie University
Mahdi Shafiei	Acadia University
Mohak Shah	McGill University
Weiming Shen	National Research Council of Canada
Daniel Silver	Acadia University

Marina Sokolova	Children's Hospital of Eastern Ontario
Bruce Spencer	National Research Council of Canada
Ahmed Tawfik	French University in Egypt
Choh Man Teng	Florida Inst. for Human & Machine Cognition
Thomas Tran	University of Ottawa
Thomas Trappenberg	Dalhousie University
André Trudel	Acadia University
Peter van Beek	University of Waterloo
Herna Viktor	University of Ottawa
Xin Wang	University of Calgary
Harris Wang	Athabasca University
Dunwei Wen	Athabasca University
Dan Wu	University of Windsor
Yang Xiang	University of Guelph
Nur Zincir-Heywood	Dalhousie University

External Reviewers

Magdy Aboul-Ela
Connie Adsett
Aditya Bhargava
Pierre-Etienne Genest
Diman Ghazi
Qinmin Hu
Yeming Hu
Sittichai Jiampojamarn
Fazel Keshtkar
Yael Kollet
Marek Lipczak
Guohua Liu
Haibin Liu
Bardia Mohabbati
Zeinab Noorian
Majid Razmara
Maxim Roy
Jona Schuman
Damon Sotoudeh
Milan Tofiloski
Davide Turcato
Chonghai Wang
Shengrui Wang
Yuefeng Wang
Wen Yan
Qian Yang
Ozge Yeloglu
Jessie Zhao
Xinghui Zhao

Graduate Symposium Program Committee

Ebrahim Bagheri	National Research Council of Canada
Julien Bourdaillet	Université de Montréal
Scott Buffett	National Research Council of Canada
Maria Fernanda Caropreso	University of Ottawa
Kevin Cohen	University of Colorado
Evgeniy Gabrilovich	Yahoo! Research
Liqiang Geng	National Research Council of Canada
Ali Ghorbani	University of New Brunswick

Arvind Gupta	MITACS
Svetlana Kiritchenko	National Research Council of Canada
Guy Lapalme	Université de Montréal
Hugo Larochelle	University of Toronto
Elliot Ludvig	University of Alberta
Bradley Malin	Vanderbilt University
Jonathan Schaeffer	University of Alberta
Mohak Shah	McGill University
Marina Sokolova	Children's Hospital of Eastern Ontario
Bruce Spencer	National Research Council of Canada
Stan Szpakowicz	University of Ottawa
Jo-Anne Ting	University of British Columbia

Sponsoring Institutions and Companies

Canadian Artificial Intelligence Association/Association pour l'intelligence artificielle au Canada (CAIAC)
http://www.caiac.ca

University of Ottawa
http://www.uottawa.ca

NLP Technologies Inc.
http://nlptechnologies.ca

MultiCorpora R&D Inc.
http://www.multicorpora.com

Palomino System Innovations Inc.
http://www.palominosys.com

AILIA.ca Association de l'industrie de la langue/Language Industry Association
http://www.ailia.ca

Table of Contents

Invited Talks

Regular Papers

Text Classification

Text Summarization and IR

Reasoning and E-Commerce

Probabilistic Machine Learning

Neural Networks and Swarm Optimization

Machine Learning and Data Mining

Short Papers

Natural Language Processing

Text Analytics

Reasoning and Planning

E-Commerce

Semantic Web

Machine Learning

Data Mining

CAI-GS 2010 Graduate Student Symposium

CAI-GS Presentation Papers

CAI-GS Poster Papers

Acquisition of 'Deep' Knowledge from Shallow Corpus Statistics

Dekang Lin

Google Inc.

Abstract. Many hard problems in natural language processing seem to require knowledge and inference about the real world. For example, consider the referent of the pronoun 'his' in the following sentences:

(1) John needed his friends
(2) John needed his support
(3) John offered his support

A human reader would intuitively know that 'his' in (1) and (3) is likely to refer to John, whereas it must refer to someone else in (2). Since the three sentences have exactly the same syntactic structure, the difference cannot be explained by syntax alone. The resolution of the pronoun references in (2) seem to hinges on the fact that one never needs one's own support (since one already has it).

I will present a series of knowledge acquisition methods to show that seemingly deep linguistic or even world knowledge may be acquired with rather shallow corpus statistics. I will also discuss the evaluation of the acquired knowledge by making use of them in applications.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, p. 1, 2010.

University and Industry Partnership in NLP, Is It Worth the “Trouble”?

Guy Lapalme

Laboratoire de Recherche Appliquée en Linguistique Informatique (RALI)
Laboratory for Applied Research in Computational Linguistics
Départment d’informatique et de recherche opérationnelle
Computer Science and Operational Research Department
Université de Montréal
lapalme@iro.umontreal.ca
http://www.iro.umontreal.ca/~lapalme

Abstract. We will present the research and products developed by members of the RALI for more than 15 years in many areas of NLP: translation tools, spelling checkers, summarization, text generation, information extraction and information retrieval. We will focus on projects involving industrial partners and will point out what we feel to be the benefits and the constraints in these types of projects for both parties. We will not describe in details the contents of each project but we will report some global lessons that we learned from these experiences.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, p. 2, 2010.

Corpus-Based Term Relatedness Graphs in Tag Recommendation

Evangelos Milios

Faculty of Computer Science
Dalhousie University, Halifax, Canada
eem@cs.dal.ca
http://www.cs.dal.ca/~eem

Abstract. A key problem in text mining is the extraction of relations between terms. Hand-crafted lexical resources such as Wordnet have limitations when it comes to special text corpora. Distributional approaches to the problem of automatic construction of thesauri from large corpora have been proposed, making use of sophisticated Natural Language Processing techniques, which makes them language specific, and computationally intensive. We conjecture that in a number of applications, it is not necessary to determine the exact nature of term relations, but it is sufficient to capture and exploit the frequent co-occurrence of terms. Such an application is tag recommendation.

Collaborative tagging systems are social data repositories, in which users manage web resources by assigning to them descriptive keywords (tags). An important element of collaborative tagging systems is the tag recommender, which proposes a set of tags to a user who is posting a resource. In this talk we explore the potential of three tag sources: resource content (including metadata fields, such as the title), resource profile (the set of tags assigned to the resource by all users that tagged it) and user profile (the set of tags the user assigned to all the resources she tagged). The content-based tag set is enriched with related tags in the tag-to-tag and title-word-to-tag graphs, which capture co-occurrences of words as tags and/or title words. The resulting tag set is further enriched with tags previously used to describe the same resource (resource profile). The resource-based tag set is checked against user profile tags - a rich, but imprecise source of information about user interests. The result is a set of tags related both to the resource and user. The system participated in the ECML/PKDD Discovery Challenge 2009 for the "content-based", "graph-based", and "online" recommendation tasks, in which it took first, third and first place respectively.

Joint work with Marek Lipczak, Yeming Hu, and Yael Kollet.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, p. 3, 2010.

Improving Multiclass Text Classification with Error-Correcting Output Coding and Sub-class Partitions*

Baoli Li and Carl Vogel

School of Computer Science and Statistics
Trinity College Dublin, Ireland
{baoli.li,vogel}@tcd.ie

Abstract. Error-Correcting Output Coding (ECOC) is a general framework for multiclass text classification with a set of binary classifiers. It can not only help a binary classifier solve multi-class classification problems, but also boost the performance of a multi-class classifier. When building each individual binary classifier in ECOC, multiple classes are randomly grouped into two disjoint groups: positive and negative. However, when training such a binary classifier, sub-class distribution within positive and negative classes is neglected. Utilizing this information is expected to improve a binary classifier. We thus design a simple binary classification strategy via multi-class categorization (2vM) to make use of sub-class partition information, which can lead to better performance over the traditional binary classification. The proposed binary classification strategy is then applied to enhance ECOC. Experiments on document categorization and question classification show its effectiveness.

Keywords: Text Classification, Error Correcting Output Coding, Binary Classification.

1 Introduction

Text classification aims at assigning one or more predefined categories to a textual segment. As an implicit reasoning mechanism, text classification is widely used in Natural Language Processing and Information Retrieval, such as document categorization, spam filtering, sentiment analysis, question classification, textual entailment recognition, named entity recognition, and so forth. In the past years, many algorithms, which include Naïve Bayes and Support Vector Machines, have been proposed and successfully used in dealing with different text classification tasks [1].

According to whether one textual segment can belong to more than one class, text classification problems are divided into two main categories: single-label multi-class categorization (in which any item may have only a single label; often called just *multi-class* categorization) and multi-label multi-class categorization (in which items may be

[*] This research is supported by the Science Foundation Ireland (Grant 07/CE/I1142) as part of the Centre for Next Generation Localisation (www.cngl.ie) at Trinity College Dublin.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 4–15, 2010.

cross-classified with multiple labels; AKA *multi-label* categorization). Herein we focus on multi-class categorization in which each segment only belongs to one class.

Error Correcting Output Coding (ECOC) provides a general framework to transform a multi-class problem into a set of binary classification problems [2, 3]. It can help a binary classifier solve multiclass classification problems, and, moreover, boost the performance of a multiclass classifier for dealing with this kind of problem. In ECOC, each class is assigned a unique codeword – a binary string of length *L*. With each bit *i* of these codewords, the original multi-class dataset will be split into two mixed classes: one contains all samples of the classes that have value 1 at bit *i* of their codewords, and the other has all the remaining samples. *L* binary classifiers (one for each bit) are learned for classifying a new sample and producing a codeword for it. The predicted class is the one whose codeword is closest to that produced by the classifiers according to a distance metric[1]. Figure 1 shows an example of ECOC classification (5 classes and *L*=10).

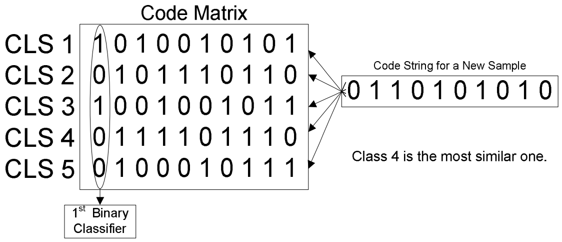

Code Matrix
CLS 1 1 0 1 0 0 1 0 1 0 1
CLS 2 0 1 0 1 1 1 0 1 1 0
CLS 3 1 0 0 1 0 0 1 0 1 1
CLS 4 0 1 1 1 1 0 1 1 1 0
CLS 5 0 1 0 0 0 1 0 1 1 1
Code String for a New Sample
0 1 1 0 1 0 1 0 1 0
Class 4 is the most similar one.
1st Binary Classifier

Fig. 1. An example of ECOC classification

Reconsidering the generation of each binary classifier in the ECOC framework, a problem is evident: when we build a binary classifier corresponding to a column of the codeword matrix, several heterogeneous classes may be combined together as a positive or negative class, and a relatively homogeneous class may be separated into different classes (one positive and one negative). Effectively, the sub-class partition information within positive and negative classes is ignored. *A priori* we do not expect an optimal classifier to emerge. ECOC, itself, has a strong correcting capacity, but why not use a better classifier if one exists? It is possible to improve binary classifiers on each bit and thereby boost ECOC.

In this study, we propose a simple strategy to improve binary text classification via multi-class categorization (dubbed 2vM) for applications where sub-class partitions of positive and/or negative classes are available. As multi-class categorization may implicitly capture the interactions between sub-classes, we expect that detailed subclasses will help differentiating the positive and negative classes with high accuracy. With carefully designed experiments, we empirically verified this hypothesis. After that, we integrate this strategy into the ECOC framework with hill-climbing local search. Experiments on document categorization and question classification demonstrate the effectiveness of the enhanced ECOC framework.

[1] For example, hamming distance counts the number of bit differences in codes.

The rest of this paper is organized as follows: in section 2, we investigate empirically whether we can improve binary text classification strategy when we use known sub-class partition information within positive and/or negative classes. Then, in section 3, our 2vM strategy with hill-climbing search is integrated into an ECOC framework. Experiments on document categorization and question classification, in section 4, demonstrate that the enhanced ECOC framework can lead to better performance. Related work and conclusions are given in section 5 and section 6, respectively.

2 Binary Classification via Multi-class Categorization

We present a binary classification strategy that utilizes sub-class partition information within positive and/or negative classes.

2.1 Method

Our proposed binary classification strategy targets solving a special kind of binary classification problem, where positive and/or negative classes may consist of several sub-classes. Suppose that the positive and negative classes in a binary classification problem contain $|P|$ and $|N|$ sub-classes, respectively, where $P=\{p_1, p_2, \ldots, p_{|P|}\}$ and $N=\{n_1, n_2, \ldots, n_{|N|}\}$. Our strategy then works as follows:

a). Build a multi-class classifier C_m, which considers $|P|+|N|$ sub-classes.
b). Classify a new item α with the learned classifier C_m outputting prediction c.
c). If c belongs to P, then label α as positive; otherwise, label α as negative.

If the multi-class classifier C_m supports probability outputs, the probability sums of sub-classes within P and N will be used for final decision. This binary classification strategy is expected to work with any multi-class categorization algorithm.

2.2 Experiments

2.2.1 Dataset

To evaluate the effectiveness of the proposed strategy, we experiment with the 20 Newsgroups dataset. This dataset is nearly evenly partitioned across 20 different newsgroups, each corresponding to a different topic. Among the different versions of this dataset, we use the *bydate* version[2] which is sorted by date and divided into a training set (60%) and a test set (40%), without cross-posts (duplicates or multi-labeled documents) nor newsgroup-identifying headers. The total number of documents in the “bydate” version is 18,846, with 11,314 for training and 7,532 for testing. We choose this dataset for experiments because it is almost balanced and without multi-label cases. We hope to remove the effects caused by these two factors.

2.2.2 Experimental Design

To obtain binary datasets, we randomly choose one or more original classes, combine those into a positive class and take the rest to form its complementary negative class.

[2] `http://www.ai.mit.edu/~jrennie/20Newsgroups/-`
last verified January 2010.

With the 20 newsgroups dataset, we have in total $C_{20}^{1}+C_{20}^{2}+...+C_{20}^{10}$ possible separations. They can be classified into ten types: 1vs19 (1 class as positive and the rest as negative), 2vs18, 3vs17, …, and 10vs10. The number of possible separations is huge (616,665). To make our experiments tractable, we randomly choose 100 separations from different types. Obviously, we can only have 20 different separations of type 1vs19. Totally we experiment with 920 different separations. Separations from the same type roughly have the same class distribution.

We compare our proposed strategy with the traditional binary classification strategy that doesn't consider sub-class partition information even though it is available. We label the traditional strategy BIN and call our proposal considering sub-class information 2vM.

We experiment with two widely used text categorization algorithms: Naïve Bayes with a multinomial model (NBM) [4] and Support Vector Machines (SVM). For SVM, we use a robust SVM implementation, LIBSVM[3], with a linear kernel and default values for other parameters. LIBSVM employs 1-against-1 strategy to extend the original binary SVM classifier to deal with multi-class categorization.

In preprocessing, we remove stop words without stemming. Words with document frequency below 2 are ignored. This leaves in total 49,790 words as features. For SVM, we use TFIDF weighting schema as (log(*TF*)+1)*log(*N*/*DF*), where *TF* is the frequency of a feature in a document, *DF* is the document frequency of a feature in a dataset, and *N* is the total number of documents. Micro-averaging and macro-averaging F-1 measures are used to evaluate performance, and paired t-test (two tailed) is used for significance analysis.

2.2.3 Results and Discussions

Figures 2 and 3 show the performance of the two binary text classification strategies with Naïve Bayes algorithm and linear SVM algorithm, respectively. The values are the averages of all separations of the same type. Figure 1 also shows the performance of a variant of 2vM (label as 2vMp), which uses the probability output for decision. 2vMp and 2vM have very close performance, although the former is statistically significantly better than the latter (P-values are 1.345E-142 and 0 for Mic-F1 and Mac-F1, respectively).

Both figures demonstrate the effectiveness of our simple strategy, binary text classification with sub-class information. The difference of Mic-F1 between BIN and 2vM grows larger as the dataset becomes more balanced, while the difference of Mac-F1 keeps stable. With the extremely imbalanced separation (1vs19), the Mic-F1 of the 2vM strategy is just a little higher than that of the traditional BIN strategy (0.9789 vs 0.9684 with Naïve Bayes, and 0.9842 vs 0.9810 with SVM). As the dataset changes from imbalanced to balanced, Mic-F1 sharply worsens, while Mac-F1 steadily improves.

For Mic-F1, this is not unexpected, as we can easily get a higher accuracy with an extremely imbalanced dataset by simply outputting the major one of the two classes. If the two classes within a dataset have more equal size the problem will become harder because the uncertainty of such a dataset becomes higher. As Mac-F1 score is more influenced by the performance on rare classes, the overall average scores are poor on imbalanced datasets because classifiers often perform poorly on rare classes.

[3] `http://www.csie.ntu.edu.tw/~cjlin/libsvm` – last verified January 2010.

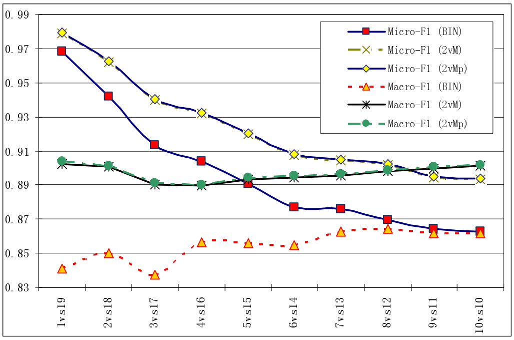


Fig. 2. F-1 measures of the two strategies on the 20 newsgroups dataset (Naïve Bayes)

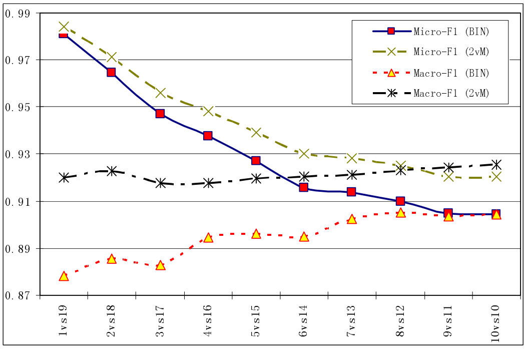


Fig. 3. F-1 measures of the two strategies on the 20 newsgroups dataset (linear SVM)

Imbalance of datasets also underlies the deviation between Mic-F1 and Mac-F1 of BIN and 2vM, which is larger for imbalanced separations than for balanced separations. As the dataset approaches balance, Mic-F1 and Mac-F1 grow very close.

The above analyses are based on the overall tendency: the values are averages of 100 runs. The performance difference between 2vM and BIN is statistically significant (e.g., with NBM, P-values of t-tests for Mic-F1 and Mac-F1 are 4.3E-269 and 8.2E-116, respectively), but this doesn't mean that on every possible separation 2vM beats BIN. With 2vM, better results are more likely: an oracle absent, it is sensible to choose our 2vM strategy.

3 ECOC with 2vM

With 2vM, we can get a better binary classifier in most cases, but, as pointed out in the previous section, there are cases in which BIN does beat 2vM. Therefore, our intuitive solution for improving ECOC is to consider each bit individually and choose the better one for each bit, either 2vM or BIN. To determine which strategy to use on each bit, we do *n*-fold cross validation on the original training data. However, this solution doesn't consistently work well in our experiments. The error correcting mechanism of ECOC is a double-edged sword: a set of binary classifiers can jointly correct errors on some bits; at the same time, they can also produce errors if they are not properly complementary even though each may perform well, individually.

Because it does not work to locally change binary classifiers for each bit due to the correcting mechanism of ECOC, we have to explore some global solution. Our problem can be formulated as a search problem as follows:

1. The search space is determined by *L* (the codeword length) and the binary choice at each bit: either 2vM or BIN. Thus, 2^L possible combinations exist.
2. Our goal is to find an optimal combination.
3. We choose to use accuracy as the evaluation metric.

As the length of codeword increases, the search space becomes huge, and exhaustively considering each possible combination is impossible. Therefore, we resort to a local optimal solution. A greedy hill-climbing algorithm is a simple but effective solution, as in many AI problems. The algorithm works as follows:

1. Start with all bits using BIN strategy or 2vM strategy, depending on which combination obtains better results with *n*-fold cross validation---Let's suppose all bits use BIN---this is regarded as the Base solution with accuracy *Acc*; we are trying to find a set of bits *C*, which includes the bits that should use another strategy; We use another set, *I,* to record a set of candidate bits. Initially, *C*=*I*={ }.
2. Individually consider each bit *b*:
   a) Change bit *b* with the alternative, e.g. 2vM, and evaluate the new combination (bit *b* with 2vM strategy, and other bits with BIN strategy) with *n*-fold cross validation.
   b) If the change in step 2.a) leads to improvement over the Base solution (all bits using BIN strategy), store it into set *I* , i.e. $I = I \cup \{b\}$.
3. Iteratively do the following until *I*={ }:
   a) Based on *C*, incrementally change a bit *t* in *I* with the other alternative, e.g. 2vM, evaluate the new combination ($C \cup \{t\}$) with *n*-fold cross validation, find the bit *o* that achieves the best performance *Acc_t*;
   b) If *Acc_t* <= *Acc*, break
   c) *Acc* = *Acc_t*
   d) Remove *o* from *I* and add *o* to *C*, i.e. $I = I\text{-}\{o\}$, $C = C \cup \{o\}$
4. Return an indication of which bits use 2vM and which bits use BIN.

In the above algorithm, we iterate choice of bits whose strategy is swapped, until no improvement is achieved on the cross-validation datasets.

Compared to the original ECOC with traditional binary classification, our proposed solution requires extra computation in the training stage: 1) we need to train a multiclass classifier for using the 2vM binary classification strategy; 2) *n*-fold cross validation on the original training data and hill-climbing search over a potentially huge space are used to find the best possible combination of using either BIN or 2vM strategy at each bit. With hill-climbing local search and calculating prediction value at each bit for each sample before local search, the extra training cost required by our solution can be greatly reduced.

During the running or testing stage, our proposed ECOC variant simply uses either BIN or 2vM strategy (which has been fixed after the training stage) to predict the value of each bit for a new sample. It works just like the original ECOC, and does not bring extra computation cost.

4 Experiments in Application Scenarios

We experiment with the proposed 2vM enhanced ECOC algorithm on two text classification tasks: document categorization and question classification. Longer textual segments are considered in document categorization, while shorter segments need to be processed in question classification.

4.1 Document Categorization

Document categorization aims at assigning one or more predefined classes to a document. As we mentioned earlier, in this research we focus on single-label multi-class problem, where one document belongs to only one class.

4.1.1 Datasets and Settings

We use two document categorization datasets: R52[4] and 20 Newsgroup. The 20 Newsgroup dataset has been introduced in section 2.2.1. R52 is a single-label dataset derived from Reuters-21578 with 90 classes by Ana Cardoso-Cachopo [5]. Documents with multiple labels in the original Reuters-21578 (90 classes) dataset are discarded and finally the R52 dataset contains 52 categories, 6,532 documents for training, and 2,568 documents for test. The dataset is imbalanced and some categories only have a few documents, e.g. classes *cpu* and *potato*. We use the "all-terms" version without stemming.

Naive Bayes with multinomial model algorithm is used as base classifiers. We experimented with two different kinds of codes: random codes and BCH codes. BCH codes are obtained from http://www.cs.cmu.edu/~rayid/ecoc/ecoc-codes.tar.gz. We also tried codes with different length, e.g. 15, 31, and 63.

In text classification with ECOC, how to assign codewords to classes is still an open question. Different assignments may result in different performance. Random assignment is widely used in practice. Therefore, it does not make sense to run just one experiment with a special assignment, deriving conclusions that may not generalize. Our experiments tried 100 different assignments for each setting. Accuracy averaged over 100 runs differentiates algorithms, with t-tests used to assess significance.

To determine a good combination of BIN and 2vM, we apply 3-fold cross validation on the original training data (i.e. n=3).

With the Naïve Bayes algorithm, we can get the probability output of each binary classifier at a bit. For each bit, we keep the probability that its corresponding bit in the

[4] Available at `http://web.ist.utl.pt/~acardoso/datasets/` – last verified January 2010.

codeword is one. Then we get a probability vector of length L (the length of codeword). Following [2], we compute as the distance metric the L^1 distance between this probability vector and a codeword, which can be regarded a 0/1 vector of length L. The L^1 distance is simply defined as the sum of absolute difference values between two corresponding elements in two vectors.

Table 1. Averaged accuracies on the R52 dataset

Code Type / Length	Random Codes			BCH Codes		
	All_BIN	**All_2vM**	**Mixed**	**All_BIN**	**All_2vM**	**Mixed**
15	**0.82458**	**0.83746**	**0.83894**	**0.84604**	**0.84941**	**0.85458**
31	**0.86558**	**0.84936**	**0.87234**	**0.86182**	**0.84939**	**0.87253**
63	**0.88448**	**0.84940**	**0.88799**	**0.89539**	**0.84937**	**0.89757**

4.1.2 Results and Analysis

Table 1 compares the averaged accuracies for three algorithms on the R52 dataset:

1. *All_BIN*: every bit uses the traditional binary classification algorithm;
2. *All_2vM*: every bit uses the proposed 2vM binary classification algorithm;
3. *Mixed*: the proposed hill-climbing solution;

From table 1, we can see that:

1) The longer the length of codes, the better performance we can get.

2) It is not always true that using BCH codes can get better results than using random codes. For example, BCH codes of length 31 didn't exhibit advantages over random codes when using the traditional binary strategy as the base classifier in ECOC, although BCH codes perform better elsewhere. This may explain why contradictory conclusions appear in the literature about BCH and random codes.

3) When using the 2vM binary strategy as the base classifier in ECOC, we obtain quite stable results. Actually, all the values are around the performance of a multi-class classifier (0.8493). If we use a hard decision function, *All_2vM* will behave as the multi-class classifier used in 2vM. There may be some small deviation due to the randomness when choosing classes in tie situations. In our experiments, we used a soft decision function with probability outputs, but the deviation is still very small.

4) *Mixed* shows advantages over *All_BIN*. The difference between *All_BIN* and *Mixed* is statistically significant (all P-values << 0.001), and it will decrease as the code length becomes larger. The proposed hill-climbing solution can derive better results, because it tries to find a good combination of BIN and 2vM. In other words, it can reach a reasonable configuration that at some bits, BIN binary classifiers will be used, where at other bits, we choose 2vM binary classifier.

5) With longer enough codes, ECOC can achieve better performance than a multi-class classifier that uses the same algorithm as the base classifier in ECOC.

Table 2. Averaged accuracies on the 20 Newsgroup dataset

Multi-Class Naïve Bayes	All_BIN	Mixed	Mixed (optimal)
0.78903	0.80998	0.81461	0.81840

Because BCH codes of length 63 achieve the highest performance, we use these codes in the following experiments. Table 2 gives the results on the 20 newsgroup dataset. The first column of table 2 shows the accuracy of the classifier using the multi-class Naïve Bayes algorithm, a baseline that we want to improve upon with ECOC. The last column of table 2 gives an optimal value with the proposed hill-climbing strategy derived by using the test data as a validation set. We can regard it as a reasonable upper bound for our proposed strategy, which determines the combination of BIN and 2vM by *n*-fold cross validation on part of the original training data. With this dataset, *Mixed* is also significantly better than *All_BIN* (P-value = 2.34E-12).

With ECOC, we can get better results than the traditional multi-class classifier. Our proposed solution to find a good combination of binary classifiers to use at each bit could further boost the results.

4.2 Question Classification

Question classification for determining the type of a question is an important step in question answering systems: the type of a question may narrow the answer search.

4.2.1 Dataset and Setting

We experiment with the dataset created by Li and Roth [6] at UIUC. 5,500 labeled questions are used for training and 500 questions from TREC-10 for testing. The total number of question types is 50. The dataset including the question taxonomy can be downloaded at http://l2r.cs.uiuc.edu/~cogcomp/Data/QA/QC.

Hacioglu and Ward [7] used this dataset and tried a solution combing Support Vector Machines and Error Correcting Codes. To compare our results with those reported by them, we use SVM as the algorithm for the base classifiers in this experiment.

Due to the higher computational training cost of SVM, we run 50 times for each setting rather than 100 times as we did in the document categorization experiments.

SVM originally does not provide probability output. Although some researchers have come up with strategies to extend SVM and generate probability output, the quality of such derived probabilities may not be good enough. Therefore, we use a hard decision function rather than a soft decision function in this experiment. Accordingly, hamming distance is taken as the distance metric in this experiment. We can assume that each hard binary classifier outputs either 0 or 1 as a probability estimate, and then we can use the L^1 distance discussed in section 4.1.1.

We use all the words and symbols in the questions as features, without stemming or feature selection. BCH codes of length 63 are used. We employ t-tests for significance analysis.

Table 3. Averaged accuracies (over 50 runs) on the question classification dataset

Multi-Class SVM	All_BIN	Mixed	Mixed (optimal)
0.794	0.81608	0.82113	0.82656

4.2.2 Results and Analysis

Table 3 gives the results on this question classification problem. We used LIBSVM in our experiments as in section 2, and it can handle multi-class classification problems via 1-against-1 strategy. With this algorithm, we obtained accuracy of 0.794 on the question classification dataset. With ECOC and the traditional binary classification strategy, we got accuracy of 0.8161 (column 2), which increases by 2.78%. Our proposed hill-climbing strategy (column 3) to find a good combination of BIN and 2vM can improve the baseline by 3.42%. Mixed is significantly better than All_BIN (P-value < 0.001).

Hacioglu and Ward [7] reported the highest accuracy of 0.82 with feature selection and named entity recognition, whereas our solution can reach this level without any additional processing. They did not indicate whether the reported results are averaged over many runs. As discussed in section 4.1.1, different codeword assignments to classes lead to different results: it does not make sense to show the results of one run or a few runs.

4.3 General Discussion

A performance improvement of 0.2% to 1% is small, but usually hard won. The differences on 20 newsgroup dataset and QC dataset are ~0.5% (with 63 bits BCH codes); 0.351% and 0.218% improvements occur on R52 datasets (63 bits random codes and BCH codes, respectively). Improvement approaching or exceeding 1% obtains in other cases. The gain on R52 is relatively larger than on the other two datasets. We think this is due to the fact that the relatively high inter-class heterogeneity in R52 (which lacks the internal structure inherent in the others) reduces the dependence between the constructed binary classifiers. Further detailed scrutiny is necessary to find the true cause. It is an instance of the general problem of being able to predict how system performance will vary with the profile of the dataset. Our results on R52 show decreasing performance difference with increasing code length. The reason is that hill-climbing search reaches a local optimality quickly (on average, with 63 bits random codes on R52, only 9.52 bits select 2vM) and only small part of a huge space (longer codes, larger space) is explored before it stops. In the future, we plan to explore other local search algorithms.

5 Related Work

Error-correcting output coding was originally introduced by Dietterich and Bakiri [8] for solving multiclass categorization problems. They [2] demonstrate with extensive experiments that ECOC improves both decision trees and neural networks.

Berger explored use of ECOC to improve Naïve Bayes and Decision Tree for text categorization [9]. In that research, Berger provided some theoretical evidence for the use of random codes rather than error-correcting codes. Ghani [3] explored the use of different kinds of codes, namely Error-Correcting Codes, Random Codes, Domain and Data-specific codes. His experiments showed that using error-correcting codes can help to obtain better performance than random codes.

Rennie and Rifkin [10] compared the performance of ECOC on the task of multi-class text classification based on Naive Bayes and Support Vector Machines algorithms. They found that ECOC with Support Vector Machines performs better than ECOC with Naïve Bayes. The accuracy difference between their work and our results follows mainly from the difference in the proportion of the dataset allocated for training vs. testing (us: 60%-40%; Rennie: 80%-20%).

Tan et al. [11] used improved Centroid binary classifiers to boost ECOC with Centroid algorithm for multi-class text categorization problems. The model refinement strategy is expected to improve the original Centroid algorithm, but as we have shown empirically, simply replacing all the binary classifiers with possibly better classifiers does not guarantee overall improvement.

Crammer and Singer [12] relax discrete codes to continuous codes, while Pujol et al. [13] and Zhou et al. [14] consider how to design problem-dependent ECOC code matrix. Luo and Xiong [15] try to improve ECOC by a kernel-based decoding strategy. They propose a new scheme of defining an optimal decoding function via supervised learning.

6 Conclusion and Future Work

For years researchers have sought improved ECOC performance in three directions: 1) different base classification algorithms; 2) different kinds of codes and strategies for code assignments; 3) more effective decoding strategy (i.e. determining final predictions based on the outputs of binary classifiers). Following the first direction, we seek to improve ECOC with a better binary classification strategy, motivated by two observations: a) in ECOC, sub-class partition information of positive and negative classes is available but ignored even though it has value for binary classification; b) no one algorithm can win on every dataset and situation (in ECOC, binary classifiers corresponding to each bit are trained from datasets generated by different binary divisions of the original multi-class datasets). The immediate practical consequences of (b) are daunting because of the additional computation, but our innovation in response is a systematic scheme, using a local search method to find an optimal global configuration that stipulates methods for individual bits, rather than determining each bit locally, and yet, without a uniform method for all bits, as is current practice. Moreover, (a) constitutes a very important distinction which we feel has not yet been adequately capitalized upon in the literature.

In this paper, we empirically demonstrate the effectiveness of a simple strategy for improving binary text classification via multi-class categorization when sub-class information is available. We then apply this strategy to enhance the general multi-class classification framework Error Correcting Output Coding with hill-climbing search. Experiments on document categorization and question classification exhibit the effectiveness of the enhanced ECOC framework.

In the future, we plan to conduct more experiments on more datasets. Our approach is expected to be useful for non-text applications, but it needs to be verified with extensive experiments. We also plan to explore other local search algorithms to further improve the proposed strategy.

References

1. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys 34(1), 1–47 (2002)
2. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. J. of Artificial Intelligence Research 2, 263–286 (1995)
3. Ghani, R.: Using Error-Correcting Codes for Text Classification. In: The Seventeenth International Conference on Machine Learning, ICML 2000 (2000)
4. Mccallum, A., Nigam, K.: A Comparison of Event Models for Naive Bayes Text Classification. In: The AAAI/ICML 1998 Workshop on Learning for Text Categorization (1998)
5. Cardoso-Cachopo, A.: Improving Methods for Single-label Text Categorization. PhD Thesis, Instituto Superior Técnico, Portugal (2007)
6. Li, X., Roth, D.: Learning question classifiers. In: The 19th International Conference on Computational Linguistics (COLING 2002), pp. 556–562 (2002)
7. Hacioglu, K., Ward, W.: Question Classification with Support Vector Machines and Error Correcting Codes. In: Proceedings of HLT-NAACL 2003 (2003) (short papers)
8. Dietterich, T.G., Bakiri, G.: Error-correcting output codes: A general method for improving multiclass inductive learning programs. In: The Ninth National Conference on Artificial Intelligence (AAAI 1991), pp. 572–577 (1991)
9. Berger, A.: Error-correcting output coding for text classification. In: IJCAI 1999 Workshop on Machine Learning for Information Filtering (1999)
10. Rennie, J., Rifkin, R.: Improving Multiclass Text Classification with the Support Vector Machine. Massachusetts Institute of Technology, AI Memo, AIM-2001-026 (2001)
11. Tan, S., Wu, G., Cheng, X.: Enhancing the Performance of Centroid Classifier by ECOC and Model Refinement. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) ECML PKDD 2009, Part II. LNCS, vol. 5782, pp. 458–472. Springer, Heidelberg (2009)
12. Crammer, K., Singer, Y.: Improved Output Coding for Classification Using Continuous Relaxtion. In: Neural Information Processing Systems (NIPS 2000), pp. 437–443 (2000)
13. Pujol, O., Radeva, P., Vitria, J.: Discriminant ECOC: A Heuristic Method for Application Dependent Design of Error Correcting Output Codes. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 1007–1012 (2006)
14. Zhou, J., Peng, H., Suen, C.Y.: Data-driven Decomposition for Multi-class Classification. Pattern Recognition 41, 67–76 (2008)
15. Luo, D., Xiong, R.: An improved error-correcting output coding framework with kernel-based decoding. Neurocomputing 71, 3131–3139 (2008)

Offensive Language Detection Using Multi-level Classification

Amir H. Razavi[1], Diana Inkpen[1], Sasha Uritsky[2], and Stan Matwin[1,3]

[1] School of Information Technology and Engineering (SITE),
University of Ottawa, Ottawa, ON, Canada, K1N 6N5
[2] Natural Semantic Modules co. 5 Tangreen Court, Suite 510
Toronto, ON, M2M 4A7
[3] Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland
{araza082,diana,stan}@site.uottawa.ca, sasha@nsemodules.com

Abstract. Text messaging through the Internet or cellular phones has become a major medium of personal and commercial communication. In the same time, flames (such as rants, taunts, and squalid phrases) are offensive/abusive phrases which might attack or offend the users for a variety of reasons. An automatic discriminative software with a sensitivity parameter for flame or abusive language detection would be a useful tool. Although a human could recognize these sorts of useless annoying texts among the useful ones, it is not an easy task for computer programs. In this paper, we describe an automatic flame detection method which extracts features at different conceptual levels and applies multi-level classification for flame detection. While the system is taking advantage of a variety of statistical models and rule-based patterns, there is an auxiliary weighted pattern repository which improves accuracy by matching the text to its graded entries.

Keywords: Flame Detection; Filtering; Information Extraction; Information Retrieval; Multi-level Classification; Offensive Language Detection.

1 Introduction

Recently, pattern recognition and machine learning algorithms are being used in a variety of Natural Language Processing applications. Everyday we have to deal with texts (emails or different types of messages) in which there are a variety of attacks and abusive phrases. An automatic intelligent software for detecting flames or other abusive language would be useful and could save its users time and energy.

Offensive phrases could mocks or insult somebody or a group of people (attacks such as aggression against some culture, subgroup of the society, race or ideology in a tirade). Here are several types of offensive language in this category:

Taunts: These phrases try to condemn or ridicule the reader in general.

References to handicaps: These phrases attack the reader using his\her shortcomings (i.e., “IQ challenged”).

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 16–27, 2010.

Squalid language: These phrases target sexual fetishes or physical filth of the reader.

Slurs: These phrases try to attack a culture or ethnicity in some way.

Homophobia: These phrases are usually talking about homosexual sentiments.

Racism: These phrases intimidate race or ethnicity of individuals [10].

Extremism: These phrases target some religion or ideologies.

There are also some other kinds of flames, in which the flamer abuses or embarrasses the reader (not an attack) using some unusual words/phrases like:

Crude language: expressions that embarrass people, mostly because it refers to sexual matters or excrement.

Disguise: expressions for which the meaning or pronunciation is the same as another more offensive term.

Four-letter words: there are five or six words which consist of only four letters.

Provocative language: expressions that may cause anger or violence.

Taboos: expressions which are forbidden in a certain society/community. There are lots of expressions that are forbidden because of what they refer to, not necessarily there is some particular taboo words used in the expression.

Unrefined language: some expressions that lack polite manners and the speaker is harsh and rude [12].

Based on the above definitions, when we say flame detection, implicitly we are talking about every context that falls into one or more of the defined cases.

Sometime, internet users searching or browsing in some specific sites are frustrated as they encounter offensive, insulting or abusive messages. It occasionally happens even in frequently-used websites like Wikipedia.

Therefore an automatic system for discriminating between regular texts and flames would save time and energy during our browsing on the web or in our everyday emails or text messages. At this stage, when we take a look at the literature on attempts to discriminate between acceptable contexts and the flames, we observe considerable percentage of disagreement between human expert annotators having the same definition of flames [1,2,3]. Therefore, it becomes evident that we cannot provide a rigid product for flame detection for all purposes. Hence in this paper we will define a tolerance margin for abusive language, based on certain conditions or applications (different sites and usages), so that the user could have an acceptable interaction with the computer.

The literature on offensive language detection and specifically on natural language analysis describes flames as exhibiting extreme subjectivity [3], depending on the context. These kinds of subjectivity are either speculative or evaluative [2]. Speculative expressions include any doubtful phrases, whereas for evaluative expressions we are dealing with emotions (such as hate, anger), judgments or opinions [9]. Therefore, any sign of extremity in such subjectivities could be considered as an effective feature for evaluation and possibly, flame detection.

However, computer software does not have the ability of capturing the exact concept of a flame context; yet, there are some useful features that we could point out, such as:

- The frequency of phrases which fall into one of the graded (weighted) flaming patterns (for each grade/weight separately);
- The frequency of graded/weighted words or phrases with abusive/extremist load, in each grade;
- The highest grade (maximum weight) which occurs in a context;
- The normalized average of the graded/weighted words or phrases.

These highlights led us to design and implement a fuzzy gauge of flame detection, and implement it in a software that could be modified regarding the acceptable tolerance margin, based on training data, manual adjustment, or even instant labeled contexts.

In section 2 of this paper we introduce some related works in this area, then we describe the flame-annotated data (section 3), the system features (sections 4), the methodology (section 5), the results (section 6), discussion (section 7), and conclusion and future work (section 8).

2 Related Work

Although there are few papers on computerized flame detection methods (which we review in this section), recently many researchers in Artificial Intelligence and Natural Language Processing have been working on different kinds of opinion extraction or sentiment analysis, e.g., Pang et al. [15], Turney and Littman [16], Gordon et al. [17], Yu and Hatzivassiloglou [18], Riloff and Wiebe [19], Yi et al. [20], Dave et al. [21], Riloff et al. [22] and Razavi and Matwin [23, 24]. In many cases detecting the level of intensity of moods or attitudes (Negative/Positive) could be an effective attribute of some specific opinion exploration for offensive language detection. Furthermore, subjective language recognition could also be useful in flame detection [1,9]. Hence, the subjective language detection is a task for which flame detection could be considered an offspring. In this area, we mention the work of Wiebe and her group: after tagging the contexts (as subjective or non subjective) using three expert judges, they applied machine learning algorithms for classifying texts based on some of their constituent words and expressions [13, 14]. This study led to similar, but more sophisticated work on evaluative and speculative language extraction [9]. Systematic subjectivity detection could be helpful in flame recognition or email classification as well [3, 5]

Swearing as a class of offensive language has been studied by Thelwall [25] which is mostly focused on the distribution by age considering their genders.

In addition to parts of speech, a corpus can be annotated with demographic features such as age, gender and social class, and textual features such as register, publication medium and domain. However some abusive languages may be related to religion (e.g. “Jesus”, “heaven”, “hell” and “damn”), sex (e.g. “fuck”), racism (e.g. “nigger”), defecation (e.g. “shit”), homophobia (e.g. “queer”) and other matters; [26, 27] try to examine only the pattern of uses of “fuck” and its morphological variants, because this is a typical swear-word that occurs frequently in the British National Corpus (BNC). Also McEnery et. Al. in this article try to build and expand upon the examination of “fuck” [28, 29] by examining the distribution pattern of “fuck” within and across spoken and written registers.

Specifically as flame detection systems, we should name *Smokey* [1] which probably is still being used by Microsoft in commercial applications. Smokey not only considers the insulting or abusing words, but also tries to recognize some structure of patterns through the flames. Smokey is equipped with a parser for syntactic analysis, which is a preliminary step for going through a semantic rule-base analysis process. Eventually, Smokey applies a C4.5 decision tree classifier for recognizing each context as a flame or not. The system, at the time of publication, used 720 message as its training set and 460 messages as testing set, and achieved 64% true-positive rate for the *flame* labeled messages and 98% true-positive rate for the *okay* labeled messages.

As another method for flame and insult detection, we can name Dependency Structure analysis which tries to detect any extreme subjectivity in texts [8].

Unfortunately, no flame detection software is freely available for trail or research purposes; therefore we cannot directly compare our results to results of other systems on our dataset.

3 Flame Annotated Data

In this study, we consider a message as a flame if either the main intention is *attack* (as we described above) or it contains *abusive* or *hostile* words, phrases or language, considering the desired tolerance margin.

We used two different sources of messages. The first set of data was provided by the NSM (Natural Semantic Module) company log files. This group of data contains 372 sentences in which the company’s users ask for some kind of information, services, or fun activities, in an interactive manner. An example of offensive statement is: “Do you have plans for this smelly meeting that is supposed to take place today?”

The second set of data that we used consists of 1288 Usenet newsgroup messages which were already annotated and used for flame recognition task by Martin *et al.* [2]. This dataset is balanced among the alt, sci, comp, and rec categories from the Usenet hierarchy. An example message, annotated as “flame”, is: “Feudalist has a new name. How many is that now? Feudalist. Quonster. Backto1913. That’s four with BacktoTheStoneAge. I have never met anyone this insecure before. Actually, I think that BacktoTheStoneAge is intended as a parody. If not, he vastly miscalculated, because I have been laughing hysterically at these posts.” Another example, also a “flame” is: “Do you find joy pouncing on strangers I have never found her doing this. Eric, have you?”. After deleting the messages longer than 2500 characters and two messages in French, we obtained with 1153 usable messages. The first dataset is composed mostly small of sentences using abusive language, and the second one contains rather long sentences full of sarcasms and ironic phrases; therefore we decided to combine them together in order to see the performance over a generic and typical offensive language detection task, rather a specific category.

We used a total number of 1525 messages (1038 *(68%) Okay* and 487 *(32%)* Flame), from the two datasets together, from which 10% was used as a test set, and the rest was used as training set for our multi-level classifier.

4 Methodology

After data preprocessing[1], we run a three-level classification for flame detection. Considering the attributes of each level we tried most of the applicable machine learning algorithms implemented in Weka (the standard machine learning software developed at the University of Waikato) [11]. We considered factors like time efficiency and updatability for online applications that determined the choice of classifier used (e.g., for the first level we needed to use fast algorithms which could work with a large number of attributes in acceptable time). After determining which algorithms satisfy these requirements, we chose the one that achieved the highest level of performance among the varieties of simple and combined complex methods available in Weka. This process for classifier selection was applied for the other levels as well. The classifiers discussed in this paper provided the highest discriminative power, compared to the other classifiers that we tried. In the third level of classification we use our Insulting and Abusing Language Dictionary which contains some word, phrase, and expression patterns for corresponding pattern recognition.

4.1 Insulting and Abusing Language Dictionary

We have collected about 2700 words, phrases, and expressions, with different degrees of manifestation of flame varieties. All the entries of this dictionary have considerable load of either *abusing / insulting* impact or *extreme subjectivity* in some of the above listed categories. We initially assigned all the entries weights in the range of 1 to 5, based on the potential impact level of each entry on the classification of the containing context. The weights that accompany this data can be used for setting the tolerance margin on flame detection for different applications. Then, in several steps of adaptive learning (on training data), we performed modifications on the weights to address the task for a most generic purpose. (However the process of the adaptive leaning could be performed based on any targeted specific domain in the field of the flame detection.) We achieved stability for the weighs with the highest level of discrimination on flames/non-flames. The result is our Insulting or Abusive Language Dictionary (IALD), a fundamental resource for our system.

At the beginning, some of these phrases or expressions contained up to five words including some wild-cards like *Somebody* or *Something (i.e. "chew Somebody's ass out" Or "Ball Somebody or Something up").* These entries are actually raw texts which in the next stage became patterns; they help the software to estimate the

[1] In preprocessing, first all the different headers, internet addresses, email addresses and tags were filtered out. Then all the delimiters such as spaces, tabs or new line characters, in addition to the following characters: "\ \r : ( ) ` 1 2 3 4 5 6 7 8 9 0 \ ' , ; = \ [ ] ; / < > { } | ~ @ # $ \ % ^ & * _ + " were removed from each message, whereas expressive characters (Punctuations) like: " - . ' ' ! ? " were kept. Punctuations (including " ") could be useful for determining the scope of speaker's messages. This step prevents the system from coming up with a lot of useless tokens as features for our first-level classifier.

probability of being a flame for each context. At this level we make a pattern for each of the entries that match a variety of word sequences (Replacing Somebody or Something wild cards for the above example). In this way each pattern could be matched with any sequence of words in which we have a few (not more than three) tokens in place of wild cards. The patterns also could match series using different types of verbs (ending in *ing*, *ed*, *d*, *es*, *s*) or nouns (ending in *es*, *s*)[2]. Hence, the original patterns in the repository entries were generalized, achieving considerable flexibility; now they could match tens of thousands word sequences in everyday contexts.

At this level, after pattern matching for each message/sentence we could supply another resource for flame probability estimation for the main task, which is flame detection.

4.2 Multilevel Classification

As part of the machine learning core of our package, we run three-level classifications on training data, using the IAL Dictionary.

In the first level of classification, considering the high degree of feature sparsity, we use the Complement Naïve Bayes classifier [11] for selecting the most discriminative (~1700) features[3] as the new training feature space and pass them to the next level of classification. (The initial raw data resulted after tokenization contained 15636 features, after preliminary feature trimming, i.e., removal of stop-words and terms that occurred only once).

In the second level, we chose the Multinomial Updatable Naïve Bayes classifier [11] in order to efficiently update its model (Model 2), based on new labeled sentences which could be added to the system after the initial training process in order to do adaptive learning. This classifier was run on the best feature space extracted from the previous level of classification. The outputs of this classification level are new aggregated features extracted from the previous level feature space, with the following attributes as the input for our last-level classification task, using IALD:

- Frequency of IALD word/phrase/expression patterns which are matched in the current instance, in each weight level (five attributes);
- Maximum weight of IALD entries that have been matched in the current message;
- Normalized average weight of IALD entries which have been matched in the current message;
- The probability that the current instance is *Okay,* based on the previous level classification applying Model 2;
- The probability that the current instance is a *Flame,* based on the previous level classification applying Model 2;
- The prediction of the previous level classification on the current instance, applying Model 2 (*Okay or Flame*);

[2] In addition to matching the wild cards, any word, phrase or expression which has any special character (leading or tailing) in the message would be tested and matched with the corresponding IADL entry.

[3] We used Wrapper Supervised Feature Selection algorithm with "RankSearch" method as our search method in Weka [11].

In the last level, we run a rule-based classifier named DTNB (Decision Table/Naive Bayes hybrid classifier [6]) on the output of the second level (the features described above and label assigned in the previous level), which makes the final decision upon the current instance (*Okay or Flame*).[4]

5 Results

After preprocessing and before performing the feature selection, we ran the Complement Naïve Bayes classifier on the whole feature space (15,636); applying 10-fold cross-validation on the above described data we got the results depicted in the first row of Tables 1 and 2.

At this level, the accuracy was about 16% better than the baseline. The baseline that we use for comparison always chooses the most frequent class (it reflects the class distribution) and has an accuracy of 68%. As shown in Table 1, there were 936 Okay texts classified correctly classified as Okay, and 349 Flames corrected classified as Flames. The others are classification errors: 102 Flames classified as Okay, and 138 Okay texts classified as Flames.

Since the 10-fold cross-validation works on features selected from the entire dataset, this is different from the operation of a deployed package where the test instances will not participate in the feature selection process. To evaluate the performance in such more realistic situation, we have trained separately, then tested on a held-out (10%) randomly selected test file for system stability verification: at the same level we applied the method on 10% test set (same baseline) and trained the method based on the rest of the data, and we achieved the results shown in the second row in Tables 1 and 2.

Table 1. Flattened confusion matrices for all 6 classification results – True Pos. shows the number of texts which correctly classified as Okay; False Pos. shows the number of texts which falsely classified as Okay; True Neg. shows the number of texts which correctly classified as Flame and the False Neg. shows the number of texts which falsely classified as Flame

True Pos.	False Pos.	True. Neg.	False Neg.	Classification#
936	102	349	138	1
89	16	36	11	2
999	39	385	102	3
84	3	27	8	4
1022	16	454	33	5
86	0	32	4	6

At the second classification level, we used the most expressive selected features (~1700 features selected by classification); the results of the Naïve Bayes Multinomial Updateable Classifier, applied with 10-folds cross-validation are shown

[4] As most parts of the computation are run prior to the final detection, the system could be applied easily in online interactive applications.

in the third row of the Tables 1 and 2. This results show that the second level of classification increased the software performance about 7%.

As above, we applied the method on 10% test set (same baseline) and trained the system based on the rest and we achieved the results shown in the fourth row in Tables 1 and 2.

At this stage, raising the system's discriminative power and going beyond the previous-level accuracy (~91%) was pretty tough task. The software needed lots of consideration and going through the structural details of IALD entries in order to increase the detection power beyond 91%. Hence, we applied the DTNB (Decision Table/Naive Bayes hybrid classifier) rule-based classifier based on extra added information extracted from IALD and its built-in semantic rules (pattern matching modules).

The third level results, using 10-fold cross-validation are in row 5 of the Tables 1 and 2. This result shows that performing the last level improves the accuracy by a valuable extra 6%.

Table 2. Performance comparison along the three levels of classifications, for cross-validation (C.V.) on the training data, and on the test set

Results → Experiments ↓		Correctly Classified	Incorrectly Classified	Okay Precision	Flame Precision	Row No
First level Classification	10 old C.V.	84.26%	15.73%	87.2%	77.4%	1
	10% Test Size	81.37%	18.62%	86.0%	56.3%	2
Second level Classification	10 Fold C.V.	90.75%	9.24%	90.7%	90.8%	3
	10% Test Size	90.98%	9.01%	9.13%	90.0%	4
Third level Classification	10 Fold C.V.	96.78%	3.21%	96.9%	96.6%	5
	10% Test Size	96.72%	3.27%	95.6%	100%	6

As with the previous levels, we tried to verify the stability of the achieved results, so we applied the method on a test set with size of 10% of the data, and obtained the results shown in row 6 of Tables 1 and 2.

When we considered the above results and the results of other numerous experiments that we run, we clearly observed that the stability of the system after each level rose, and at the last level, the results on cross-validation and on the test set were quite similar.

If we consider the pair-wise agreement of judges, on the data from the previous annotation project [2] (which was part of our data), we see that the pair-wise agreement between human judges (based on the same definition of a flame message) on average is 92%, whereas if we take a look at other survey results (on similar but different data), we can see that although the agreement rate is 98% for non-flammatory messages, this rate diminished to 64% consensus for flame messages [1]. One important issue for human annotation which should be taken into account is that the distribution of the data (balanced/unbalanced) does not have any considerable influence on human judgments, unlike for the machine learning classifiers.

Hence, our higher percentage of agreement with the labels shows that the current software has a high level of adaptivity, based on the training dataset, and the IALD patterns and weights. Therefore, we can conclude that our method has a high capacity of being customized for any specific application.

The reasons for discrepancies between human judges (with the same problem definition) could be their different sensitivity, mood, background and some other subjective conditions. Human judgment is subjective and it is not necessarily the same among different people. It is thus helpful to have a standard detection system that can pass judgments based on some constant predefinitions, patterns and rules.

Unfortunately, no flame detection software is freely available for trial or research purposes, therefore we cannot directly compare our results to results of other systems on our dataset.

6 Discussion

Many of our IALD entries are applied as semantic classification rules. In the third level of classification, we attempt to match each of the corresponding patterns that have been built regarding the entry's *wild cards* or some additional prefix, suffix or special characters (leading or trailing), which help to distinguish whether the containing instance is a *Flame* or an *Okay* instance.

The advantages of the method could be listed as:

- The software can be used for message level or sentence level classification application in real-time applications (a fraction of a second for each new context).
- Our system benefits from both statistical models and rule-based patterns, in addition to specific semantic patterns inside the IALD, and does not rely on only one of them.
- Our software is not very sensitive to punctuation and grammatical mistakes.
- The method could be adapted in time, based on user feedback.

Among the limitations of our system is the fact that it does not consider the syntactical structure of the messages explicitly and could be equipped with some modules designed for subjectivity detection based on their lexicons (in this case we have to take into account that the length of each message would be a limitation for the method).

As we apply some patterns from IALD, as well as classifier models for flame detection, it is important to prevent training the classifiers based on some instances in which the assigned labels are opposed to some of IALD built-in weighted patterns and vice versa. Otherwise, the system will suffer from a considerable level of noise in the data.

7 Conclusion and Future Work

We designed and implemented novel and very efficient flame detection software. It applies models from multi-level classifiers, boosted by an Insulting and Abusing Language Dictionary. We built two rule-based auxiliary systems; one of them is the last level of our classifiers and the other is used for building patterns out of the IALD repository. The software performs with a high level of accuracy for both normal text and for flames.

Our flame detection method can be modified based on any accumulative training data and applied on any collaborative writing web site in which people can add or modify content, in the style of Wikipedia. It could also be handy for some web-logs or some specialist forums. The software could also be adapted for some kinds of spam detection for any type of text messaging services, suc as cellular phone SMS. It also could be useful over text chat services, as well as any comment acceptance posts in social networking sites like *Orkut* and *Facebook*.

In future work, we could apply second order co-occurrence features (Pedersen et. al. [30]) in order to extract more semantic information by processing surrounding terms and contexts of each preliminarily detected flame. We could add a synchronized adaptive weight modifier module to the IALD accessory, based on further provided training data.

Acknowledgements. We gratefully acknowledge the contribution of Dr. Melanie J. Martin for sharing the 1288 annotated messages; this data was crucial for the progress of this research. Natural Sciences and Engineering Research Council of Canada supports the research of the second and fourth author.

References

1. Spertus, E.S.: Automatic recognition of hostile messages. In: Proceedings of the Eighth Annual Conference on Innovative Applications of Artificial Intelligence (IAAI), pp. 1058–1065 (1997)
2. Martin, M.J.: Annotating flames in Usenet newsgroups: a corpus study. For NSF Minority Institution Infrastructure Grant Site Visit to NMSU CS department (2002)
3. Wiebe, J., Wilson, T., Bruce, R., Bell, M., Martin, M.: Learning Subjective Language. Computational Linguistics 30(3), 277–308 (2004)
4. Gyamfi, y., Wiebe, J., Mihalcea, R., Akkaya, C.: Integrating Knowledge for Subjectivity Sense Labeling. In: Joint Conference of the North American Chapter of the Association for Computational Linguistics and the Human Language Technologies Conference, NAACL-HLT 2009 (2009)
5. Wiebe, J., Wilson, T., Cardie, C.: Annotating expressions of opinions and emotions in language. Language Resources and Evaluation 39(2-3), 165–210 (2005)
6. Hall, M., Frank, E.: Combining Naive Bayes and Decision Tables. In: FLAIRS Conference, pp. 318–319 (2008)
7. Wiebe, J., Wilson, T., Bell, B.: Identifying Collocations for Recognizing Opinions. In: Proc. ACL 2001 Workshop on Collocation, Toulouse, France (2001)
8. Mahmud, A., Ahmed, K.Z., Khan, M.: Detecting flames and insults in text. In: Proc. of 6th International Conference on Natural Language Processing (ICON 2008), CDAC Pune, India, December 20-22 (2008)
9. Wiebe, J., Bruce, R., Bell, M., Martin, M., Wilson, T.: A Corpus Study of Evaluative and Speculative Language. In: Proceedings of 2nd ACL SIGdial Workshop on Discourse and Dialogue, Aalborg, Denmark (2001)
10. Kaufer, D.: Flaming: A White Paper (2000)

11. Witten, I., Frank, E., Gray, J.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations (2008) ISBN13: 9781558605527
12. Spears, R.A.: Forbidden American English (1991) ISBN: 9780844251493
13. Bruce, R.F., Wiebe, J.: Recognizing subjectivity: a case study in manual tagging. Natural Language Engineering 5(2) (1999)
14. Wiebe, J., Bruce, R.F., O'Hara, T.: Development and use of a gold standard data set for subjectivity classifications. In: Proc. 37th Annual Meeting of the Assoc. for Computational Linguistics (ACL 1999), pp. 246–253 (1999)
15. Pang, B., Lee, L., Vaithyanathan, S.H.: Thumbs up? Sentiment classification using machine learning techniques. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 79–86 (2002)
16. Turney, P., Littman, M.: Measuring praise and criticism: Inference of semantic orientation from association. ACM Transactions on Information Systems (TOIS) 21(4), 315–346 (2003)
17. Gordon, A., Kazemzadeh, A., Nair, A., Petrova, M.: Recognizing expressions of commonsense psychology in English text. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003), pp. 208–215 (2003)
18. Yu, H., Hatzivassiloglou, V.: Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 129–136 (2003)
19. Riloff, E., Wiebe, J.: Learning extraction patterns for subjective expressions. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2003), pp. 105–112 (2003)
20. Yi, J., Nasukawa, T., Bunescu, R., Niblack, W.: Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In: Proceedings of the 3rd IEEE International Conference on Data Mining, ICDM 2003 (2003)
21. Dave, K., Lawrence, S., Pennock, D.M.: Mining the peanut gallery: Opinion extraction and semantic classification of produce reviews. In: Proceedings of the 12th International World Wide Web Conference (2003)
22. Riloff, E., Wiebe, J., Wilson, T.: Learning subjective nouns using extraction pattern bootstrapping. In: Proceedings of the 7th Conference on Natural Language Learning (CoNLL), pp. 25–32 (2003)
23. Razavi, A.H., Amini, R., Sabourin, C., Sayyad Shirabad, J., Nadeau, D., Matwin, S., De Koninck, J.: Classification of emotional tone of dreams using machine learning and text analyses. Paper presented at the Meeting of the Associated Professional Sleep Society in Baltimore. Sleep, vol. 31, pp. A380–A381 (2008)
24. Razavi, A.H., Amini, R., Sabourin, C., Sayyad Shirabad, J., Nadeau, D., Matwin, S., De Koninck, D.: Evaluation and Time Course Representation of the Emotional Tone of dreams Using Machine Learning and Automatic Text Analyses. In: 19th Congress of European Sleep Research Society; ESRS-Glasgow Journal of Sleep Research (2008) (in press)
25. Thelwall, M.: Fk yea I swear: Cursing and gender in a corpus of MySpace pages. Corpora 3(1), 83–107 (2008)
26. McEnery, A.M.: Swearing in English: Bad Language, Purity and Power from 1586 to the Present. Routledge, London (2005) (in press)
27. McEnery, A.M., Xiao, Z.: Swearing in modern British English: the case of fuck in the BNC. Language and Literature 13(3), 235–268 (2004)

28. McEnery, A.M., Baker, J.P., Hardie, A.: Swearing and abuse in modern British English. In: Lewandowska-Tomaszczyk, B., Melia, P.J. (eds.) Practical Applications of Language Corpora, Peter Lang, Hamburg, pp. 37–48 (2000)
29. McEnery, A.M., Baker, J.P., Hardie, J.: Assessing claims about language use with corpus data – swearing and abuse. In: Kirk, J. (ed.) Corpora Galore, Rodopi, Amsterdam, pp. 45–55 (2000)
30. Pedersen, T., Kulkarni, A.K., Angheluta, R., Kozareva, Z., Solorio, T.: An Unsupervised Language Independent Method of Name Discrimination Using Second Order Co-occurrence Features. In: Gelbukh, A. (ed.) CICLing 2006. LNCS, vol. 3878, pp. 208–222. Springer, Heidelberg (2006)

A Three-Way Decision Approach to Email Spam Filtering

Bing Zhou, Yiyu Yao, and Jigang Luo

Department of Computer Science, University of Regina
Regina, Saskatchewan, Canada S4S 0A2
{zhou200b,yyao,luo226}@cs.uregina.ca

Abstract. Many classification techniques used for identifying spam emails, treat spam filtering as a binary classification problem. That is, the incoming email is either spam or non-spam. This treatment is more for mathematical simplicity other than reflecting the true state of nature. In this paper, we introduce a three-way decision approach to spam filtering based on Bayesian decision theory, which provides a more sensible feedback to users for precautionary handling their incoming emails, thereby reduces the chances of misclassification. The main advantage of our approach is that it allows the possibility of rejection, i.e., of refusing to make a decision. The undecided cases must be re-examined by collecting additional information. A loss function is defined to state how costly each action is, a pair of threshold values on the posterior odds ratio is systematically calculated based on the loss function, and the final decision is to select the action for which the overall cost is minimum. Our experimental results show that the new approach reduces the error rate of classifying a legitimate email to spam, and provides better spam precision and weighted accuracy.

Keywords: Spam filter, three-way decision, naive Bayesian classification, Bayesian decision theory, cost.

1 Introduction

Email spam filtering is a growing concern on the Internet. A popular approach is to treat spam filtering as a classification problem. Many classification algorithms from machine learning were employed to automatically classify incoming emails into different categories based on the contents of emails [2,6,9,11,14,15]. Among these algorithms, Bayesian classifier achieved better results by reducing the classification error rates. The naive Bayesian classifier [6,11,14], along with many other classification algorithms, treat spam filtering as a binary classification problem, that is, the incoming email is either spam or non-spam. In reality, this simple treatment is too restrict and could result in losing vital information for users by misclassifying a legitimate email to spam. For example, a user could miss an important job offer just because the email contains "congratul" (i.e., a common word in email spam filter word list) in its header. On the other hand, misclassifying a spam email to non-spam also brings unnecessary costs and waste of resources.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 28–39, 2010.

In this paper, we introduce a three-way decision approach to spam filtering based on Bayesian decision theory, that is, to *accept*, *reject*, or *further-exam* an incoming email. The emails waiting for *further-exam* must be clarified by collecting additional information. The idea of three-way decision making can be found in some early literatures and has been applied to many real world problems [5,7]. For example, the three-way decisions are often used in clinical decision making for a certain disease, with options of treating the conditional directly, not treating the condition, or performing a diagnose test to decide whether or not to treat the condition [12]. Yao et al. [16,17] introduced decision theoretic rough set model (DTRS) based on three-way decisions. The ideas of DTRS have been applied to information retrieval by dividing the dynamic document stream into three states instead of the traditional relevant and irrelevant states [8]. More recently, Zhao et al. [18] introduced an email classification schema based on DTRS by classifying the incoming email into three categories instead of two. The main differences between their work and our approach are the interpretations of the conditional probabilities and the values of the loss functions. In their approach, the conditional probability was estimated by the rough membership function [13], which is only one of the possible ways and is impractical for real applications. They have simply defined the loss function that all errors are treated equally, which is not the case in many real applications. For instance, misclassifying a legitimate email to spam is usually considered more costly than misclassifying a spam email to legitimate. In our approach, the conditional probability is interpreted based on the naive Bayesian classification. The posterior odds is used a monotonic increasing transformation of the conditional probability to compare with the threshold values. A threshold value on the probability can indeed be interpreted as another threshold value on the odds. The naive independence assumptions are added to calculate the likelihood by assuming that each feature of an email is unrelated to any other features. After the transformations, all the related factors used to interpret the conditional probability are easily derivable from data. We consider the different cost associated for taking each action, which is more general than the zero-one loss function.

The main advantage of three-way decision making is that it allows the possibility of rejection, i.e., of refusing to make a decision. The undecided cases must be forwarded for re-examination. A loss function is defined to state how costly each action is, and the final decision is to select the action for which the overall cost is minimum. A pair of threshold values are estimated based on the loss function. The first threshold value determines the value of the probability necessary for a re-examination, and the second value determines the value of the probability necessary to reject an email. These settings provide users a fairly high degree of control over their incoming emails, thereby reduce the chances of misclassification. Our experimental results show that the new approach reduces the error rate of classifying a legitimate email to spam, and provides a better spam precision and weighted accuracy.

2 The Naive Bayesian Spam Filtering

The naive Bayesian spam filtering is a probabilistic classification technique of email filtering [14]. It is based on Bayes' theorem with naive (strong) independence assumptions [6,11,14].

Suppose each email can be described by a feature vector $\mathbf{x} = (x_1, x_2, ..., x_n)$, where $x_1, x_2, ..., x_n$ are the values of attributes of emails. Let C denote the *legitimate* class, and C^c denote the *spam* class. Based on Bayes' theorem and the theorem of total probability, given the vector of an email, the conditional probability that this email is in the *legitimate* class is:

$$Pr(C|\mathbf{x}) = \frac{Pr(C)Pr(\mathbf{x}|C)}{Pr(\mathbf{x})}, \tag{1}$$

where $Pr(\mathbf{x}) = Pr(\mathbf{x}|C)Pr(C) + Pr(\mathbf{x}|C^c)Pr(C^c)$. Here $Pr(C)$ is the *prior* probability of an email being in the *legitimate* class. $Pr(\mathbf{x}|C)$ is commonly known as the *likelihood* of an email being in the *legitimate* class with respect to $\mathbf{x}$.

The likelihood $Pr(\mathbf{x}|C)$ is a joint probability of $Pr(x_1, x_2, ..., x_n|C)$. In practice, it is difficult to analyze the interactions between the components of $\mathbf{x}$, especially when the number n is large. In order to solve this problem, an independence assumption is embodied in the naive Bayesian classifier [6,11] which assumes that each feature x_i is conditionally independent of every other features, given the class C, this yields,

$$\begin{aligned} Pr(\mathbf{x}|C) &= Pr(x_1, x_2, ..., x_n|C) \\ &= \prod_{i=1}^{n} Pr(x_i|C), \end{aligned} \tag{2}$$

where $Pr(x_i|C)$ can be easily estimated as relative frequencies from the training data set. Thus equation (1) can be rewritten as:

$$Pr(C|\mathbf{x}) = \frac{Pr(C)\prod_{i=1}^{n} Pr(x_i|C)}{Pr(\mathbf{x})}. \tag{3}$$

Similarly, the corresponding probabilities $Pr(C^c|\mathbf{x})$ of an email being in the *spam* class given vector $\mathbf{x}$ can be reformulated as:

$$Pr(C^c|\mathbf{x}) = \frac{Pr(C^c)\prod_{i=1}^{n} Pr(x_i|C^c)}{Pr(\mathbf{x})}. \tag{4}$$

Note that $Pr(\mathbf{x})$ in equation (3) and (4) is unimportant with regard to making a decision. It is basically a scale factor that assures $Pr(C|\mathbf{x}) + Pr(C^c|\mathbf{x}) = 1$. This scale factor can be eliminated by taking the ratio of $Pr(C|\mathbf{x})$ and $Pr(C^c|\mathbf{x})$:

$$\frac{Pr(C|\mathbf{x})}{Pr(C^c|\mathbf{x})} = \prod_{i=1}^{n} \frac{Pr(x_i|C)}{Pr(x_i|C^c)} \frac{Pr(C)}{Pr(C^c)}. \tag{5}$$

$Pr(C|\mathbf{x})/Pr(C^c|\mathbf{x})$ is called the *posterior odds* of an email being in the *legitimate* class against being in the *spam* class given $\mathbf{x}$. It is a monotonic increasing transformation of $Pr(C|\mathbf{x})$. $Pr(x_i|C)/Pr(x_i|C^c)$ is called the *likelihood ratio*. Thus, the conditional probability $Pr(C|\mathbf{x})$ can be easily calculated from $Pr(C|\mathbf{x})/Pr(C^c|\mathbf{x})$ based on the observation that $Pr(C|\mathbf{x}) + Pr(C^c|\mathbf{x}) = 1$. Finally, an incoming email can be classified as *legitimate* if $\frac{Pr(C|\mathbf{x})}{Pr(C^c|\mathbf{x})}$ (i.e., the posterior odds) exceeds a threshold value, otherwise it is *spam*.

3 Bayesian Decision Theory

Bayesian decision theory is a fundamental statistical approach that makes decisions under uncertainty based on probabilities and costs associated with decisions. Following the discussions given in the book by Duda and Hart [3], the basic ideas of the theory are reviewed.

Let $\Omega = \{w_1, \ldots, w_s\}$ be a finite set of s states and let $\mathcal{A} = \{a_1, \ldots, a_m\}$ be a finite set of m possible actions. Let $\lambda(a_i|w_j)$ denote the loss, or cost, for taking action a_i when the state is w_j. Let $Pr(w_j|\mathbf{x})$ be the conditional probability of an email being in state w_j given that the email is described by $\mathbf{x}$. For an email with description $\mathbf{x}$, suppose action a_i is taken. Since $Pr(w_j|\mathbf{x})$ is the probability that the true state is w_j given $\mathbf{x}$, the expected loss associated with taking action a_i is given by:

$$R(a_i|\mathbf{x}) = \sum_{j=1}^{s} \lambda(a_i|w_j) Pr(w_j|\mathbf{x}). \tag{6}$$

The quantity $R(a_i|\mathbf{x})$ is also called the conditional risk.

Given a description $\mathbf{x}$, a decision rule is a function $\tau(\mathbf{x})$ that specifies which action to take. That is, for every $\mathbf{x}$, $\tau(\mathbf{x})$ takes one of the actions, $a_1, \ldots, a_m$. The overall risk $\mathbf{R}$ is the expected loss associated with a given decision rule. Since $R(\tau(\mathbf{x})|\mathbf{x})$ is the conditional risk associated with action $\tau(\mathbf{x})$, the overall risk is defined by:

$$\mathbf{R} = \sum_{\mathbf{x}} R(\tau(\mathbf{x})|\mathbf{x}) Pr(\mathbf{x}), \tag{7}$$

where the summation is over the set of all possible descriptions of emails. If $\tau(\mathbf{x})$ is chosen so that $R(\tau(\mathbf{x})|\mathbf{x})$ is as small as possible for every $\mathbf{x}$, the overall risk $\mathbf{R}$ is minimized. Thus, the optimal Bayesian decision procedure can be formally stated as follows. For every $\mathbf{x}$, compute the conditional risk $R(a_i|\mathbf{x})$ for $i = 1, \ldots, m$ defined by equation (6) and select the action for which the conditional risk is minimum. If more than one action minimizes $R(a_i|\mathbf{x})$, a tie-breaking criterion can be used.

4 A Three-Way Decision Approach to Email Spam Filtering

In the naive Bayesian spam filter, an incoming email is classified as legitimate if the posterior odds ratio exceeds a certain threshold value. In our approach, a pair of threshold values is used to make a three-way decision of an incoming email. The first threshold value determines the probability necessary for a re-examination, and the second value determines the probability necessary to reject an email. There are different ways to acquire the required threshold values. One may directly supply the threshold values based on an intuitive understanding of the levels of tolerance for errors [19]. A more rational way is to infer these threshold values from a theoretical and practical basis. One such solution was given in DTRS [16,17] based on the well known Bayesian decision theory [3]. A pair of threshold values on the conditional probability is systematically calculated based on the loss function. In our approach, the posterior odds is used a monotonic increasing transformation of the conditional probability to compare with the threshold values. A new pair of threshold values is defined and calculated based on the prior odds ratio and the loss functions with the naive independence assumptions. This transformation ensures the easy estimation of all the related factors.

With respect to a set of emails to be approximated, we have a set of two states $\Omega = \{C, C^c\}$ indicating that an email is in C (i.e., *legitimate*) or not in C (i.e., *spam*), respectively. The incoming emails can be divided into three regions, namely, the positive region $\mathrm{POS}(C)$ includes emails being *legitimate*, the boundary region $\mathrm{BND}(C)$ includes emails that need *further-exam*, and the negative region $\mathrm{NEG}(C)$ includes emails that are *spam*. With respect to these three regions, the set of actions is given by $\mathcal{A} = \{a_P, a_B, a_N\}$, where a_P, a_B, and a_N represent the three actions in classifying an email x, namely, deciding $x \in \mathrm{POS}(C)$, deciding $x \in \mathrm{BND}(C)$, and deciding $x \in \mathrm{NEG}(C)$, respectively. The loss function is given by the 3 × 2 matrix:

	C (P)	C^c (N)
a_P	$\lambda_{PP} = \lambda(a_P \mid C)$	$\lambda_{PN} = \lambda(a_P \mid C^c)$
a_B	$\lambda_{BP} = \lambda(a_B \mid C)$	$\lambda_{BN} = \lambda(a_B \mid C^c)$
a_N	$\lambda_{NP} = \lambda(a_N \mid C)$	$\lambda_{NN} = \lambda(a_N \mid C^c)$

In the matrix, λ_{PP}, λ_{BP} and λ_{NP} denote the losses incurred for taking actions a_P, a_B and a_N, respectively, when an email belongs to C, and λ_{PN}, λ_{BN} and λ_{NN} denote the losses incurred for taking these actions when the email does not belong to C.

The expected losses associated with taking different actions for emails with description $\mathbf{x}$ can be expressed as:

$$\begin{aligned} R(a_P|\mathbf{x}) &= \lambda_{PP} Pr(C|\mathbf{x}) + \lambda_{PN} Pr(C^c|\mathbf{x}), \\ R(a_B|\mathbf{x}) &= \lambda_{BP} Pr(C|\mathbf{x}) + \lambda_{BN} Pr(C^c|\mathbf{x}), \\ R(a_N|\mathbf{x}) &= \lambda_{NP} Pr(C|\mathbf{x}) + \lambda_{NN} Pr(C^c|\mathbf{x}). \end{aligned} \tag{8}$$

The Bayesian decision procedure suggests the following minimum-risk decision rules:

(P) If $R(a_P|\mathbf{x}) \leq R(a_B|\mathbf{x})$ and $R(a_P|\mathbf{x}) \leq R(a_N|\mathbf{x})$, decide $x \in \mathrm{POS}(C)$;

(B) If $R(a_B|\mathbf{x}) \le R(a_P|\mathbf{x})$ and $R(a_B|\mathbf{x}) \le R(a_N|\mathbf{x})$, decide $x \in \mathrm{BND}(C)$;

(N) If $R(a_N|\mathbf{x}) \le R(a_P|\mathbf{x})$ and $R(a_N|\mathbf{x}) \le R(a_B|\mathbf{x})$, decide $x \in \mathrm{NEG}(C)$.

Tie-breaking criteria should be added so that each email is put into only one region.

Since $Pr(C|\mathbf{x}) + Pr(C^c|\mathbf{x}) = 1$, we can simplify the rules based only on the probabilities $Pr(C|\mathbf{x})$ and the loss function λ. Consider a special kind of loss functions with:

$$\begin{aligned}\text{(c0).}\quad & \lambda_{PP} \le \lambda_{BP} < \lambda_{NP},\\ & \lambda_{NN} \le \lambda_{BN} < \lambda_{PN}.\end{aligned} \tag{9}$$

That is, the loss of classifying an email x being in C into the positive region $\mathrm{POS}(C)$ is less than or equal to the loss of classifying x into the boundary region $\mathrm{BND}(C)$, and both of these losses are strictly less than the loss of classifying x into the negative region $\mathrm{NEG}(C)$. The reverse order of losses is used for classifying an email not in C. Under condition (c0), we can simplify decision rules (P)-(N) as follows. For the rule (P), the first condition can be expressed as:

$$\begin{aligned} & R(a_P|\mathbf{x}) \le R(a_B|\mathbf{x})\\ \Longleftrightarrow\ & \lambda_{PP}Pr(C|\mathbf{x}) + \lambda_{PN}Pr(C^c|\mathbf{x}) \le \lambda_{BP}Pr(C|\mathbf{x}) + \lambda_{BN}Pr(C^c|\mathbf{x})\\ \Longleftrightarrow\ & \lambda_{PP}Pr(C|\mathbf{x}) + \lambda_{PN}(1 - Pr(C|\mathbf{x})) \le \lambda_{BP}Pr(C|\mathbf{x}) + \lambda_{BN}(1 - Pr(C|\mathbf{x}))\\ \Longleftrightarrow\ & Pr(C|\mathbf{x}) \ge \frac{(\lambda_{PN} - \lambda_{BN})}{(\lambda_{PN} - \lambda_{BN}) + (\lambda_{BP} - \lambda_{PP})}. \end{aligned} \tag{10}$$

Similarly, the second condition of rule (P) can be expressed as:

$$R(a_P|\mathbf{x}) \le R(a_N|\mathbf{x}) \Longleftrightarrow Pr(C|\mathbf{x}) \ge \frac{(\lambda_{PN} - \lambda_{NN})}{(\lambda_{PN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{PP})}. \tag{11}$$

The first condition of rule (B) is the converse of the first condition of rule (P). It follows,

$$R(a_B|\mathbf{x}) \le R(a_P|\mathbf{x}) \Longleftrightarrow Pr(C|\mathbf{x}) \le \frac{(\lambda_{PN} - \lambda_{BN})}{(\lambda_{PN} - \lambda_{BN}) + (\lambda_{BP} - \lambda_{PP})}. \tag{12}$$

For the second condition of rule (B), we have:

$$R(a_B|\mathbf{x}) \le R(a_N|\mathbf{x}) \Longleftrightarrow Pr(C|\mathbf{x}) \ge \frac{(\lambda_{BN} - \lambda_{NN})}{(\lambda_{BN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{BP})}. \tag{13}$$

The first condition of rule (N) is the converse of the second condition of rule (P) and the second condition of rule (N) is the converse of the second condition of rule (B). It follows,

$$\begin{aligned} R(a_N|\mathbf{x}) \le R(a_P|\mathbf{x}) &\Longleftrightarrow Pr(C|\mathbf{x}) \le \frac{(\lambda_{PN} - \lambda_{NN})}{(\lambda_{PN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{PP})},\\ R(a_N|\mathbf{x}) \le R(a_B|\mathbf{x}) &\Longleftrightarrow Pr(C|\mathbf{x}) \le \frac{(\lambda_{BN} - \lambda_{NN})}{(\lambda_{BN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{BP})}. \end{aligned} \tag{14}$$

To obtain a compact form of the decision rules, we denote the three expressions in these conditions by the following three parameters:

$$\begin{aligned}\alpha &= \frac{(\lambda_{PN} - \lambda_{BN})}{(\lambda_{PN} - \lambda_{BN}) + (\lambda_{BP} - \lambda_{PP})},\\ \beta &= \frac{(\lambda_{BN} - \lambda_{NN})}{(\lambda_{BN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{BP})},\\ \gamma &= \frac{(\lambda_{PN} - \lambda_{NN})}{(\lambda_{PN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{PP})}.\end{aligned} \tag{15}$$

The decision rules (P)-(N) can be expressed concisely as:

$$\begin{aligned}&\text{(P) If } Pr(C|\mathbf{x}) \geq \alpha \text{ and } Pr(C|\mathbf{x}) \geq \gamma, \text{ decide } x \in \text{POS}(C);\\ &\text{(B) If } Pr(C|\mathbf{x}) \leq \alpha \text{ and } Pr(C|\mathbf{x}) \geq \beta, \text{ decide } x \in \text{BND}(C);\\ &\text{(N) If } Pr(C|\mathbf{x}) \leq \beta \text{ and } Pr(C|\mathbf{x}) \leq \gamma, \text{ decide } x \in \text{NEG}(C).\end{aligned}$$

Each rule is defined by two out of the three parameters.

The conditions of rule (B) suggest that $\alpha > \beta$ may be a reasonable constraint; it will ensure a well-defined boundary region. By setting $\alpha > \beta$, namely,

$$\frac{(\lambda_{PN} - \lambda_{BN})}{(\lambda_{PN} - \lambda_{BN}) + (\lambda_{BP} - \lambda_{PP})} > \frac{(\lambda_{BN} - \lambda_{NN})}{(\lambda_{BN} - \lambda_{NN}) + (\lambda_{NP} - \lambda_{BP})}, \tag{16}$$

we obtain the following condition on the loss function [17]:

$$\text{(c1).} \qquad \frac{\lambda_{NP} - \lambda_{BP}}{\lambda_{BN} - \lambda_{NN}} > \frac{\lambda_{BP} - \lambda_{PP}}{\lambda_{PN} - \lambda_{BN}}. \tag{17}$$

The condition (c1) implies that $1 \geq \alpha > \gamma > \beta \geq 0$. In this case, after tie-breaking, the following simplified rules are obtained [17]:

$$\begin{aligned}&\text{(P1)} \quad \text{If } Pr(C|\mathbf{x}) \geq \alpha, \text{ decide } x \in \text{POS}(C);\\ &\text{(B1)} \quad \text{If } \beta < Pr(C|\mathbf{x}) < \alpha, \text{ decide } x \in \text{BND}(C);\\ &\text{(N1)} \quad \text{If } Pr(C|\mathbf{x}) \leq \beta, \text{ decide } x \in \text{NEG}(C).\end{aligned}$$

The parameter γ is no longer needed.

From the rules (P1), (B1), and (N1), the (α, β)-probabilistic positive, negative and boundary regions are given, respectively, by:

$$\begin{aligned}\text{POS}_{(\alpha,\beta)}(C) &= \{x \in U \mid Pr(C|\mathbf{x}) \geq \alpha\},\\ \text{BND}_{(\alpha,\beta)}(C) &= \{x \in U \mid \beta < Pr(C|\mathbf{x}) < \alpha\},\\ \text{NEG}_{(\alpha,\beta)}(C) &= \{x \in U \mid Pr(C|\mathbf{x}) \leq \beta\}.\end{aligned} \tag{18}$$

The threshold parameters can be systematically calculated from a loss function based on the Bayesian decision theory.

The conditional probability $Pr(C|\mathbf{x})$ is difficult to directly derive from data. Recall that in naive Bayesian spam filter, the ratio of $Pr(C|\mathbf{x})$ and $Pr(C^c|\mathbf{x})$

(i.e., the posterior odds) can be used as a monotonic increasing transformation of the conditional probability $Pr(C|\mathbf{x})$. A threshold value on the probability can indeed be interpreted as another threshold value on the odds. For the positive region, we have:

$$P(C|\mathbf{x}) \geq \alpha \Longleftrightarrow \frac{Pr(C|\mathbf{x})}{Pr(C^c|\mathbf{x})} \geq \frac{\alpha}{1-\alpha} = \frac{\lambda_{PN} - \lambda_{BN}}{\lambda_{BP} - \lambda_{PP}}. \tag{19}$$

According to equation (5), we can re-expressed the above equation as:

$$\prod_{i=1}^{n} \frac{Pr(x_i|C)}{Pr(x_i|C^c)} \frac{Pr(C)}{Pr(C^c)} \geq \frac{\lambda_{PN} - \lambda_{BN}}{\lambda_{BP} - \lambda_{PP}}. \tag{20}$$

This computation can be further simplified by taking the logarithm of both side of equation (20):

$$\sum_{i=1}^{n} \log \frac{Pr(x_i|C)}{Pr(x_i|C^c)} \geq \log \frac{Pr(C^c)}{Pr(C)} + \log \frac{\lambda_{PN} - \lambda_{BN}}{\lambda_{BP} - \lambda_{PP}}. \tag{21}$$

Here $\log \frac{Pr(C^c)}{Pr(C)}$ is independent of the description of emails, we treat it as a constant. Similar expression can be obtained for the negative region as:

$$\sum_{i=1}^{n} \log \frac{Pr(x_i|C)}{Pr(x_i|C^c)} \leq \log \frac{Pr(C^c)}{Pr(C)} + \log \frac{\lambda_{BN} - \lambda_{NN}}{\lambda_{NP} - \lambda_{BP}}. \tag{22}$$

A new pair of threshold values α' and β' can be defined as:

$$\begin{aligned} \alpha' &= \log \frac{Pr(C^c)}{Pr(C)} + \log \frac{\lambda_{PN} - \lambda_{BN}}{\lambda_{BP} - \lambda_{PP}}, \\ \beta' &= \log \frac{Pr(C^c)}{Pr(C)} + \log \frac{\lambda_{BN} - \lambda_{NN}}{\lambda_{NP} - \lambda_{BP}}, \end{aligned} \tag{23}$$

where $Pr(C)/Pr(C^c)$ can be easily estimated from the frequencies of the training data by putting:

$$Pr(C) = \frac{|C|}{|U|} \text{ and } Pr(C^c) = \frac{|C^c|}{|U|}. \tag{24}$$

We can then get the (α', β')-probabilistic positive, negative and boundary regions written as:

$$\begin{aligned} \text{POS}_{(\alpha',\beta')}(C) &= \{x \in U \mid \sum_{i=1}^{n} \log \frac{Pr(x_i|C)}{Pr(x_i|C^c)} \geq \alpha'\}, \\ \text{BND}_{(\alpha',\beta')}(C) &= \{x \in U \mid \beta' < \sum_{i=1}^{n} \log \frac{Pr(x_i|C)}{Pr(x_i|C^c)} < \alpha'\}, \\ \text{NEG}_{(\alpha',\beta')}(C) &= \{x \in U \mid \sum_{i=1}^{n} \log \frac{Pr(x_i|C)}{Pr(x_i|C^c)} \leq \beta'\}. \end{aligned} \tag{25}$$

All the factors in equation (25) are easy to derive from data.

Table 1. Three-way decision results with $\lambda = 1$

	Actually legitimate	Actually spam	Total
accept	465	28	493
further-exam	22	13	35
Reject	12	227	239
Total	499	268	767

Table 2. Naive Bayesian results with $\lambda = 1$

	Actually legitimate	Actually spam	Total
Classified legitimate	476	32	508
Classified spam	23	236	259
Total	499	268	767

5 Experimental Results and Evaluations

Our experiments were performed on a spambase data set from UCI Machine Learning Repository [10]. The data set consists of 4601 instances, with 1813 instances as *spam*, and 2788 instances as *legitimate*, each instance is described by 58 attributes. Our goal is to compare our approach with the original naive Bayesian spam filter in terms of the error rate that a legitimate email is classified as spam, the precision and recall for both legitimate and spam emails, and the cost-sensitive measure suggested by Androutsopoulos et al. [1].

We split the spambase data set into a training set of 3834 instances, and a testing set of 767 instances. Since the attributes in the input data set have continuous values, entropy-MDL [4] is used as the discretization method applied to both the training and testing data sets before the calculations of probabilities. For the cost-sensitive evaluations, we assume that misclassifying a legitimate email as spam is λ times more costly than misclassifying a spam email as legitimate. We considered three different λ values ($\lambda = 9$, $\lambda = 3$, and $\lambda = 1$) for the original naive Bayesian spam filter. Three sets of loss functions for the three-way decision approach are set up accordingly with the same cost ratios. For instance, when we use $\lambda = 9$ for the naive Bayesian spam filter, $\lambda_{NP}/\lambda_{PN} = 9$ is used in the three-way decision approach.

Table 1 and Table 2 show the prediction results of the three-way decision and the naive Bayesian approach when $\lambda = 1$, respectively. Note that in this case, the cost of misclassifying a legitimate email as spam is the same as the cost of misclassifying a spam email as legitimate. Table 3 and Table 4 show the prediction results when $\lambda = 3$. Table 5 and Table 6 show the prediction results when $\lambda = 9$. From the above tables, we can easily find that the error rates of misclassifying a legitimate email into spam by using the three-way decision approach are lower than the original naive Bayesian spam filter in all three experiments. Since reducing this error rate is the most important factor to users. Although the accuracy of correctly classifying a legitimate email has slightly dropped, but we consider this as a reasonable trade off.

Table 3. Three-way decision results with $\lambda = 3$

	Actually legitimate	Actually spam	Total
accept	476	32	508
further-exam	12	10	22
Reject	11	226	237
Total	499	268	767

Table 4. Naive Bayesian results with $\lambda = 3$

	Actually legitimate	Actually spam	Total
Classified legitimate	483	38	521
Classified spam	16	230	246
Total	499	268	767

Table 5. Three-way decision results with $\lambda = 9$

	Actually legitimate	Actually spam	Total
accept	465	28	493
further-exam	29	36	65
Reject	5	204	209
Total	499	268	767

Table 6. Naive Bayesian results with $\lambda = 9$

	Actually legitimate	Actually spam	Total
Classified legitimate	491	46	537
Classified spam	8	222	230
Total	499	268	767

To further evaluate these results, we compare the precision, recall and weighted accuracy of both approaches. The legitimate precision and recall are defined as:

$$\textit{legitimate precision} = \frac{n_{L\to L}}{n_{L\to L} + n_{S\to L}}, \quad \textit{legitimate recall} = \frac{n_{L\to L}}{n_{L\to L} + n_{L\to S}},$$

where $n_{L\to L}$ denotes the number of emails classified as legitimate which truly are, $n_{L\to S}$ denotes the number of legitimate emails classified as spam, and $n_{S\to L}$ denotes the number of spam emails classified as legitimate. Similarly, we define:

$$\textit{spam precision} = \frac{n_{S\to S}}{n_{S\to S} + n_{L\to S}}, \quad \textit{spam recall} = \frac{n_{S\to S}}{n_{S\to S} + n_{S\to L}}.$$

Clearly, spam precision is the most important factor to users. The comparison results are shown in Table 7. We can easily find that the three-way decision approach provides a better spam precision than the naive Bayesian spam filter in all three experiments. For the cost-sensitive evaluations, we used weighted accuracy suggested by Androutsopoulos et al. [1], which is defined as:

$$\textit{weighted accuracy} = \frac{\lambda \cdot n_{L\to L} + n_{S\to S}}{\lambda \cdot N_L + N_S},$$

Table 7. Comparison between three-way decision and naive Bayesian approaches

Cost	Approaches	Spam		Legitimate		weighted accuracy
		precision	recall	precision	recall	
$\lambda = 1$	Three-way decision	94.98%	84.70%	94.32%	93.19%	94.54%
	Naive Bayesian	91.12%	88.06%	93.70%	95.39%	92.83%
$\lambda = 3$	Three-way decision	95.36%	84.33%	93.70%	90.71%	96.22%
	Naive Bayesian	93.50%	85.82%	92.70%	96.79%	95.13%
$\lambda = 9$	Three-way decision	97.61%	76.12%	94.32%	93.19%	98.36%
	Naive Bayesian	96.52%	82.84%	91.43%	98.40%	97.52%

where N_L and N_S are the number of legitimate and spam emails to be classified by the spam filter. From Table 7, we can find that the weighted accuracy of the three-way decision approach is higher than the original naive Bayesian approach in all three experiments. We also find that when λ changed to a bigger value, the performances of both approaches are increased, but the three-way decision approach performs out the naive Baysian spam filter in all three settings.

6 Conclusion

In this paper, we present a three-way decision approach to email spam filtering. In addition to the most commonly used binary classification for spam filtering, a third action is added to allow users make further examinations for undecided cases. The main advantage of our approach is that it provides a more sensible feedback to users for handling their emails, thus reduces the misclassification rate. A pair of threshold values are used. The first threshold value determines the point necessary for a re-examination, and the second value determines the point to reject an email. Instead of supplying the threshold values based on try and error, or intuitive understandings of the levels of tolerance for errors. We provide a systematically calculation of the threshold values based on Bayesian decision theory. A loss function is defined in association with each action. The final decision making is to select the action for which the overall cost is minimum. Our experimental results show that the new approach reduces the error rate of classifying a legitimate email to spam, and provides a better spam precision and weighted accuracy.

Acknowledgements

The first author is supported by an NSERC Alexander Graham Bell Canada Graduate Scholarship. The second author is partially supported by an NSERC Canada Discovery grant.

References

1. Androutsopoulos, I., Koutsias, J., Chandrinos, K.V., Spyropoulos, C.D.: An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages. In: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 160–167 (2000)
2. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, Cambridge (2000)
3. Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis. Wiley, New York (1973)
4. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence, pp. 1022–1029 (1993)
5. Forster, M.R.: Key concepts in model selection: performance and generalizability. Journal of Mathematical Psychology 44, 205–231 (2000)
6. Good, I.J.: The Estimation of Probabilities: An Essay on Modern Bayesian Methods. MIT Press, Cambridge (1965)
7. Goudey, R.: Do statistical inferences allowing three alternative decision give better feedback for environmentally precautionary decision-making. Journal of Environmental Management 85, 338–344 (2007)
8. Li, Y.F., Zhang, C.Q.: Rough set based decision model in information retrieval and filtering. In: Third World Multiconference on Systemics, Cybernetics and Informatics (SCI 1999) and Fifth International Conference on Information Systems Analysis and Synthesis (ISAS 1999), vol. 5, pp. 398–403 (1999)
9. Masand, B., Linoff, G., Waltz, D.: Classifying news stories using memory based reasoning. In: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 59–65 (1992)
10. http://www.ics.uci.edu/mlearn/MLRepository.html
11. Mitchell, T.: Machine Learning. McGraw-Hill, New York (1997)
12. Pauker, S.G., Kassirer, J.P.: The threshold approach to clinical decision making. New England Journal of Medicine (1980)
13. Pawlak, Z., Skowron, A.: Rough membership functions. In: Yager, R.R., Fedrizzi, M., Kacprzyk, J. (eds.) Advances in the Dempster-Shafer Theory of Evidence, pp. 251–271. John Wiley and Sons, New York (1994)
14. Sahami, M., Dumais, S., Heckerman, D., Horvitz, E.: A Bayesian approach to filtering junk e-mail. In: AAAI Workshop on Learning for Text Categorization, Madison, Wisconsin. AAAI Technical Report WS-98-05 (1998)
15. Schapire, E., Singer, Y.: BoosTexter: A boosting-based system for text categorization. Machine Learning 39(2/3), 135–168 (2000)
16. Yao, Y.Y., Wong, S.K.M., Lingras, P.: A decision-theoretic rough set model. In: Ras, Z.W., Zemankova, M., Emrich, M.L. (eds.) Methodologies for Intelligent Systems 5, New York, pp. 17–24. North-Holland, Amsterdam (1990)
17. Yao, Y.Y.: Decision-theoretic rough set models. In: Yao, J., Lingras, P., Wu, W.-Z., Szczuka, M.S., Cercone, N.J., Ślęzak, D. (eds.) RSKT 2007. LNCS (LNAI), vol. 4481, pp. 1–12. Springer, Heidelberg (2007)
18. Zhao, W.Q., Zhu, Y.L.: An email classification scheme based on decision-theoretic rough set theory and analysis of email security. In: Proceeding of 2005 IEEE Region 10 TENCON, pp. 1–6 (2005)
19. Ziarko, W.: Variable precision rough sets model. Journal of Computer and Systems Sciences 46, 39–59 (1993)

Hierarchical Approach to Emotion Recognition and Classification in Texts

Diman Ghazi[1], Diana Inkpen[1], and Stan Szpakowicz[1,2]

[1] School of Information Technology and Engineering, University of Ottawa
[2] Institute of Computer Science, Polish Academy of Sciences
{dghaz038,diana,szpak}@site.uottawa.ca

Abstract. We explore the task of automatic classification of texts by the emotions expressed. We consider how the presence of neutral instances affects the performance of distinguishing between emotions. Another facet of the evaluation concerns the relation between polarity and emotions. We apply a novel approach which arranges neutrality, polarity and emotions hierarchically. This method significantly outperforms the corresponding "flat" approach which does not take into account the hierarchical information. We also compare corpus-based and lexical-based feature sets and we choose the most appropriate set of features to be used in our hierarchical classification experiments.

Keywords: Sentiment analysis, emotion in text, emotion recognition, text classification, hierarchical classification.

1 Introduction

In recent years there has been a growing interest in automatic identification and extraction of opinions, emotions, and sentiment in text. Motivations for this task include the desire to provide tools for information analysts in government, commercial, and political domains, who want to automatically track attitudes and feelings in on-line forums [1]. Emotions, an important element of human nature, have also been widely studied in psychology and behavioral sciences. They have also attracted the attention of researchers in computer science and particularly in computational linguistics.

This paper looks at the categorization of a sentence into six basic emotions (defined as emotions with universally accepted distinctive facial expressions). Those are *happiness*, *sadness*, *fear*, *anger*, *disgust*, and *surprise* [2]. We added a class *non-emotional* for the sentences which bear no emotion. These seven classes are common in much of the previous work [3, 4, 5, 6].

There has been progress in research on polarity and sentiment analysis, but little work has been done in automatic recognition of emotion in text. We assume that emotions carried by a sentence are not independent of their polarity; therefore we try to find a link between them and we want to apply classification methods, which consider these connections.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 40–50, 2010.

Our proposed hierarchical classification is a novel method in emotional analysis, which considers the relation between polarity and emotion of a text. The main idea is to order these categories and their relations in a hierarchical form and perform classification based on this hierarchy.

We have two forms of hierarchy for classification. Using *two-level classification*, we explore how neutral instances affect the emotional analysis. In *three-level classification*, the first step is to define whether the instances are emotional. Next, the instances defined as emotional in the previous step are classified on their polarity. In the third step we assume that, among the six emotions, the instances of *happiness* have positive polarity, while the other five emotions are regarded as negative polarity. That is why we take the negative instances from the second step and classify them into the five negative emotion classes.

Our experiments on data annotated with emotions show that this approach significantly outperforms the corresponding "flat" approach. We note significantly improved precision, recall and F-measure of all emotional classes.

The remainder of this paper is organized as follows. Section 2 briefly presents previous work. Section 3 gives an overview of the dataset, lexicons and different feature sets which we compare. In Section 4 we describe the hierarchical classification method and we evaluate it by comparing it to previous flat classification results. We discuss future work and present a few conclusions in Section 5.

2 Previous Work

Computational approaches to emotional analysis have focused on various emotion modalities, but only limited work has been done in the direction of automatic recognition of emotion in text [3]. In SemEval 2007, one of the tasks was carried out in an unsupervised setting and the emphasis was on the study of emotion in lexical semantics [7-10].

The participants in the SemEval 2007 workshop took a linguistic approach, using enriched lexical resources such as SentiWordNet and WordNetAffect [8]. They also used statistics gathered from Web search engines [9] based on the hypothesis that groups of words which co-occur with a given emotion word are highly likely to express the same emotion. Mihalcea and Strapparava [7] exploited the co-occurrence of words in the text with the words which have explicit affective meaning. They used the WordNet affect list as the direct affective words, and implemented a variation of Latent Semantic Analysis to yield a vector space model which allows for a homogeneous representation of words [7]. These tasks were tested on the emotion classification of news headlines extracted from news Web sites.

The existence of an annotated corpus with rich information about opinions and emotions would support the development and evaluation of NLP systems which exploit such information. In particular, statistical and machine learning approaches have become the method of choice for constructing a wide variety of practical NLP applications [1]. Emotional classification is not exemplified.

Aman and Szpakowicz [3, 11] have shown that the best results on a dataset annotated with emotions are achieved by a combination of corpus-based unigram features and lexical-based features using Support Vector Machine (SVM). In general, two supervised machine learning algorithms, SVM and Naïve Bayes, have long been a method of choice for sentiment recognition at the text level [3, 5, 12, 13].

So far most of the research has been concentrated on the feature selections and applying lexical semantics rather than focusing on different learning schemes. In this work, in addition to comparing two different sets of features, corpus-based and lexically-based, we would like to focus more on a new learning approach, namely hierarchical classification.

Koller and Sahami [14] carried out the first proper study of a hierarchical text categorization problem in 1997. Afterwards more work in hierarchical text categorization has been reported [13, 15]. The work in [13] is more related to this paper. The authors applied a hierarchical approach in mood classification – classifying blog posts into 132 moods. Even though moods and emotions may seem similar, their classification is quite different. In mood classification we talk about a large number of mood classes. Also, the hierarchical structure applied is based on a definition of moods and relations among them which is completely different than our hierarchical structure for considering polarity in classifying emotions.

For these reasons, we propose a new method, called hierarchical classification, in our machine-learning approach to classifying blog sentences into Ekman's six emotion classes and one non-emotional class.

3 Resources and Feature Sets

In this section, we will first explain the resources, namely data set and lexicons, which have been used in our experiments. Next, we will compare three set of features to find the most proper one for hierarchical classification.

3.1 Resources

The statistical methods typically require training and test corpora, manually annotated with respect to each language-processing task to be learned [1].

The main consideration in the selection of data for the emotional classification task is that the data should be rich in emotion expressions in order to contain numerous learning instances. Another consideration is that the data should comprise enough instances of all the emotion categories. That is why personal texts such as diaries and blogs have received more attention recently [3, 4, 13]. As a result, we chose for our experiments a corpus of blog sentences annotated with emotion labels, discussed in [3].

Each sentence is tagged by a dominant emotion in the sentence, or as *non-emotional* if it does not include any emotion. The dataset contains 173 weblog posts annotated by two judges. It is annotated based on Eckman's six emotions [2] at the sentence level. The dataset contains 4090 annotated sentences, 68% of which were annotated as non-emotional. The highly unbalanced dataset with 68% of non-emotional sentences as the highest class and 3% of the *fear* and *surprise* classes prompted us to reduce the number of non-emotional sentences to 38% of all the sentences by removing 2000 of the non-emotional sentences, to reduce the unbalance. Table 1 shows the details of the chosen dataset.

Table 1. Data set specifications

	Domain	*Size*	*# classes*	*Min-Max%*
Data set 1	Weblogs	2090	7	6-38 %

We now compare three sets of features. The first set is corpus-based, so we need no other external resources; the other two sets are lexically-based. The lexically-based approach requires lexical-semantic resources. In our experiments, we use three *emotional lexicons*: Prior-Polarity subjectivity lexicon[1] [16], WordNet Affect[2] [17], and an emotion lexicon derived from *Roget's Thesaurus*[3] [18].

The prior-polarity subjectivity lexicon contains over 8000 subjectivity clues collected from a number of sources. To create this lexicon, the authors began with the list of subjectivity clues extracted from [19]. The list was expanded by using a dictionary and a thesaurus, and added positive and negative word lists from the General Inquirer[4] [16]. The words are grouped into strong subjective and weak subjective clues; Table 2 presents the distribution of their polarity.

The WordNet Affect lexicon contains six lists of words corresponding to the six basic emotion categories. It is the result of assigning a variety of affect labels to each synset in WordNet [17]. Table 3 shows the distribution of words in WordNet Affect.

The emotional lexicon derived from *Roget's Thesaurus* was created automatically finding the emotion-related words [11]. The total number of words selected for inclusion in this lexicon is 2622. The distribution of words in different emotion classes is shown in Table 4.

Table 2. Distribution of Prior-Polarity clues

neutral	*positive*	*negative*	*both*
6.9%	33.1%	59.7%	0.3%

Table 3. Distribution of the WordNet Affect emotional lexicon

happiness	*sadness*	*anger*	*disgust*	*surprise*	*fear*	*total*
398	201	252	53	71	141	1116

Table 4. Distribution of the *Roget's*-derived emotional lexicon

happiness	*sadness*	*anger*	*disgust*	*surprise*	*fear*	*total*
643	262	265	401	499	552	2622

[1] http://www.cs.pitt.edu/mpqa

[2] http://www.cse.unt.edu/~rada/affectivetext/data/WordNetAffectEmotionLists.tar.gz

[3] Several versions of Roget's Thesaurus are available. In the experiments reported here, the 1987 Penguin's Roget's Thesaurus was used [3].

[4] http://www.wjh.harvard.edu/~inquirer/'p

3.2 Feature Sets

We want to evaluate the performance of different lexicons by comparing them with unigram features. We also want to find the most appropriate set of features for our main experiments in the next section. Here, we perform three different experiments, all based on the flat classification approach.

The first experiment is a corpus-based classification which uses unigrams. The unigram models have been widely used in text classification, and shown to provide good results in sentiment classification tasks [12]. In this experiment, we use unigrams from the corpus, selected using feature selection methods from Weka[5].

The second experiment consists in the classification with features derived from the polarity lexicon. The features here are the tokens that are common between the prior polarity lexicon and the chosen dataset – 796 tokens in total.

In the last experiment, we use a combination of the emotional lists of words from *Roget's Thesaurus* and WordNet Affect. All the common tokens between the combined lexicons and dataset's tokens comprise 1264 words.

Text classification quite often deals with high dimensionality of the feature space. Many learning algorithms do not scale to a high-dimensional feature space. SVM has been shown to give good performance in text classification experiments: it scales well to the large numbers of features [3, 12]. We also experimented with other classifiers: Naïve Bayes and Decision Tree. As expected, the results for our task were lower, so we do not present them here. Consequently, we use SVM as a machine-learning algorithm to be applied to the chosen features.

As a result of the emotional classification experiments using SVM (the SMO algorithm in Weka) [20] and setting "10-fold cross validation" as a testing option, we get the accuracy of 65.55% for the first set of features, 62.67% for the second set and 57.3% for the last feature set.

Based on the accuracy of the three experiments, unigram features outperform the other two types. Even though it is significantly better than the third one, its difference with the second feature set merits a discussion. Here, we would like to point out parameters – other than the accuracy of the results – which should be considered in choosing the list of features. The number of features is one of the main parameters. We are not interested in having better results by adding more features. What is more interesting is to find a list of fewer but more meaningful features which could contribute more to learning. In our experiments, the size of polarity features is quite smaller than the unigrams. Also, by checking the features manually we noticed that they appear to be more meaningful. For example, among the unigram features we have proper nouns such as names of people and countries. It is also possible to have misspelled tokens in unigrams, while the prior-polarity lexicon features are well-defined words usually considered as polar. Besides, lexical features are known to be more domain and corpus-independent. For these reasons, polar lexically-based features will be chosen for the main experiments in the next section.

[5] Machine-learning software available at `http://www.cs.waikato.ac.nz/ml/weka/`

4 Hierarchical Classification

Hierarchical categorization deals with categorization problems in which categories are organized in hierarchies. For most text categorization tasks the category hierarchies have been carefully composed by humans and represent our knowledge of the subject [15].

In this work, we use the hierarchical categories to impart an additional knowledge to our classification method. We will convey the information in two forms of hierarchy. The first one is a two-level hierarchy which represents the relation of emotion and neutrality in text. The second form is a three-level hierarchy which addresses the relation between polarity and emotions in addition to the relation between emotion and neutrality. That is based on the assumption that, among the six chosen emotions, *happiness* belongs to the positive polarity class, while the other five emotions are regarded as having negative polarity.

In the remainder of this section, we will give a detailed explanation of both proposed hierarchical forms. The experiments and results of applying them to our chosen data set will be presented.

4.1 Two-Level Classification

The main goal of this task is to find out how the presence of neutral instances affects the performance of features for distinguishing between emotional classes. This was motivated by a similar work in polarity classification [16].

In the two-level classifications, the first level, emotional versus non-emotional classification, tries to determine whether an instance is neutral or emotional. The second step takes all instances which level 1 classified as emotional, and tries to classify them into one of the six emotions. The two levels of this hierarchy are as follows:

- **Emotional versus non-emotional:** We change the annotated data set by keeping all the non-emotional instances as is, and changing the class of all the other six emotional classes to “Emo” (for *emotional*). The result of this experiment and its comparison with the flat classification shows that the recall of non-emotional instances decreases while the precision increases. This happens to non-emotion class that used to be the dominant class in flat classification but it no longer dominates in hierarchical classification. Classifiers tends to give priority to a dominant class, so more instances are placed in this class; thus, classification achieves low precision and high recall. Hierarchical method tends to produce higher precision in this case.
- **Six-class classification:** After classifying the instances based on whether they are emotional, now we would like to classify the instances which level 1 classified as emotional. That will be done in two ways. In the first experiment, we assume that all the non-emotional instances are correctly classified, so we would only need to be concerned about the mistakes in level 2 (this is called *gold standard*). The second experiment considers the misclassified non-emotional instances of the first level as well. Both sets of results and the flat classification results are shown in Table 5.

Table 5. Two-level emotional classification (the highest precision, recall, and F-measure values for each class are shown in bold)

		Two-level classification			Flat classification		
		Precision	Recall	F-measure	Precision	Recall	F-measure
1st level	*Emo*	0.88	0.85	0.86	--	--	--
	Non-emo	**0.88**	0.81	**0.84**	0.54	**0.87**	0.67
2nd level gold standard	*happiness*	0.59	**0.95**	**0.71**	**0.74**	0.60	0.66
	sadness	**0.77**	**0.49**	**0.60**	0.69	0.42	0.52
	fear	**0.91**	**0.49**	**0.63**	0.82	**0.49**	0.62
	surprise	**0.75**	**0.32**	**0.45**	0.64	0.27	0.37
	disgust	0.66	**0.35**	**0.45**	**0.68**	0.31	0.43
	anger	**0.72**	**0.33**	**0.46**	0.67	0.26	0.38
2nd level based on the 1st level result	*happiness*	0.59	**0.96**	**0.73**	**0.74**	0.60	0.66
	sadness	**0.79**	**0.48**	**0.60**	0.69	0.42	0.52
	fear	**0.92**	**0.54**	**0.68**	0.82	0.49	0.62
	surprise	**0.66**	0.25	**0.37**	0.64	**0.27**	**0.37**
	disgust	0.65	**0.33**	**0.44**	**0.68**	0.31	0.43
	anger	0.65	**0.30**	**0.41**	**0.67**	0.26	0.38

By comparing the results of the second level of *gold standard* classification with the flat classification, we can see that the F-measure of all the emotional classes in the two-level experiment is higher than the F-measure of the emotional classes in the flat classification. In two emotion classes, however, the precision of the flat approach is higher. In *disgust* the difference is insignificant but the difference between precision and recall of the *happiness* class in the flat approach and the two-level approach cannot be ignored. This can be explained by the fact that at the second level of the two-level classification we do not have the non-emotional instances any more. The *happiness* class is the dominant class, with 42% of all the instances. This makes the classifier consider most all the instances it is not sure about as *happiness*. This results in high recall and low precision for the *happiness* class. We hope to address this big gap between precision and recall of the *happiness* class in the next experiments, three-level classification, which separates *happiness* from the other five emotions. So, it makes the number of instances of each level more balanced.

At the second level, when we consider the mistakes of the first level, we have to ignore both the false-negative emotional instances and false-positive non-emotional instances in order to obtain the set of instances which we use in the second level; therefore the number of instances will drop to 1101 from 1290. Because of the different number of instances, in this case the results are not quite comparable with the results of the original one-level task. Despite having to deal with the noise from the first level, the hierarchical approach still gets higher F-measures for five of the emotion classes and equal F-measure for the *surprise* class.

4.2 Three-Level Classification

In this approach, we go even further: we break the seven-class classification task into three levels. We add an assumption that the *happiness* class is positive and the remaining five emotions are negative.

Table 6. Three-level emotional classification (the highest precision, recall, and F-measure values for each class are shown in bold)

		Three-level Classification			Flat Classification		
		Precision	Recall	F-measure	Precision	Recall	F-measure
1st level	*Emo*	0.88	0.85	0.86	--	--	--
	Non-emo	**0.88**	0.81	**0.84**	0.54	**0.87**	0.67
2nd level gold standard	*positive*	**0.89**	**0.65**	**0.75**	0.74	0.60	0.66
	negative	0.79	0.94	0.86	--	--	--
3rd level gold standard	*sadness*	0.63	**0.54**	**0.59**	**0.69**	0.42	0.52
	fear	**0.88**	**0.52**	**0.65**	0.82	0.49	0.62
	surprise	**0.79**	**0.37**	**0.50**	0.64	0.27	0.38
	disgust	0.42	**0.38**	0.40	**0.68**	0.31	**0.43**
	anger	0.38	**0.71**	**0.49**	**0.67**	0.26	0.38
2nd level based on the 1st level result	*positive*	**0.90**	**0.66**	**0.76**	0.74	0.60	0.66
	negative	0.78	0.94	0.85	--	--	--
3rd level based on the 2nd level result	*sadness*	0.64	**0.64**	**0.59**	**0.69**	0.42	0.52
	fear	**0.90**	**0.53**	**0.67**	0.82	0.49	0.62
	surprise	**0.83**	**0.35**	**0.50**	0.64	0.27	0.37
	disgust	0.37	**0.45**	0.40	**0.68**	0.31	**0.43**
	anger	0.35	**0.56**	**0.43**	**0.67**	0.26	0.38

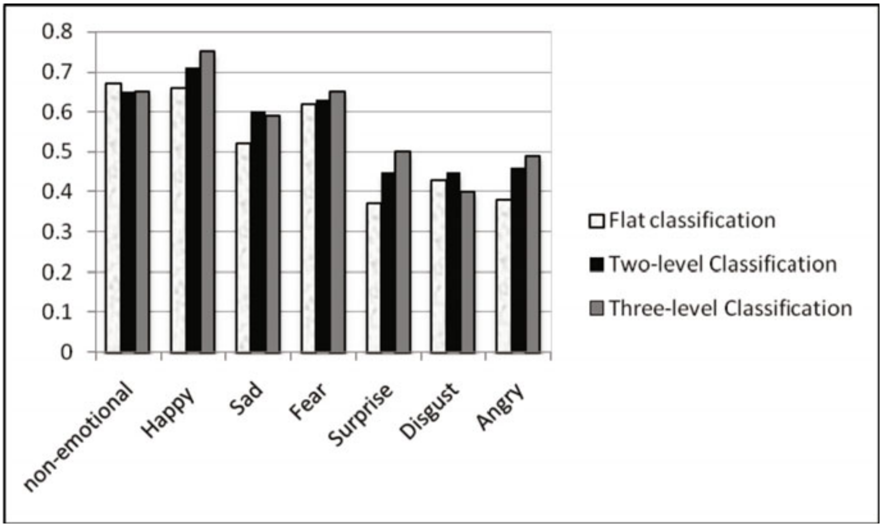


Fig. 1. The comparison of F-measure result of the flat classification with the hierarchical classification approaches for seven classes

In the three-level classifications, the first level is the same as in the previous approach. After determining the non-emotional instances of the first level, in the second level we classify into *positive* and *negative* the instances which level 1 classified as emotional. We only consider *happiness* as positive. Finally, we classify the negative instances into five negative emotion classes. The results of this classification are shown in Table 6.

As we can see in the results of the second level, we increased the precision of the *happiness* class. In the polarity classification level (the second level) the data are almost balanced, with 42% of positive instances. That makes the instances defined as *happiness* more precise.

At the third level, except in the class *disgust*, we see an increase in the F-measure of all classes in comparison to both the two-level and flat classification. It is hard to compare the result of all three applied approaches from the above tables. We have combined all the results in Figure 1, which displays the F-measure of each class for the three approaches.

5 Conclusions and Future Work

The focus of this study was an emotional analysis and classification of emotions in sentences. We first defined two different sets of lexicon-based features to compare with the bag-of-words classification method. As a result, we noticed that unigram features were not much better than polarity lexicon features. Besides, the polarity lexicon features had some benefits over unigrams. In particular, they are supposed to be less corpus-dependent. They also have fewer features than unigram sets; therefore the rest of our experiments were based on the polarity lexicon features.

In the emotional classification we noticed that having non-emotional instances in the dataset degrades the results significantly; therefore we applied a two-level classification which defines the non-emotional instances in one step and considers the rest as emotional. As a result we saw a considerable improvement in the classification results.

The second part of the hierarchical classification experiments considers the polarity of emotions, that is, their positivity or negativity. Here we assume that *happiness* is a positive emotion, while *sadness*, *surprise*, *fear*, *disgust*, and *anger* are negative. We added one more step into our hierarchy to convey the information of the relation between polarity and emotions in our system.

On the other hand, a classifier trained on unbalanced data gives biased results for the classes with more instances. In this case most of the learning algorithms behave just as people do. They mainly learn the dominant class and if they see an instance they do not have information about, they will classify it as an instance of the bigger class, since it is more probable. The hierarchical classification approach was better at dealing with the highly unbalanced data.

In the future, we plan to expand our work by testing on other available emotion-annotated data sets. There are three other available datasets annotated with emotions. The first one is a data set of 700 sentences extracted from blogs, which are annotated with nine emotions and one neutral class [4]. The second available dataset is composed of sentences in the narrative domain of nineteen children's fairy tales which have affect labels from two annotators. The affects are the same as the six emotions we considered, with the difference that the *surprise* class is broken into two subclasses of positive surprise and negative surprise [5]. The last set is the TextAffect dataset available for SemEval 2007, Task 14 [6]. We are interested in seeing how the emotional classification of these data sets will be affected by our proposed classification approach.

Another interesting future task would be to consider different levels of our hierarchy as different tasks which could be handled differently. In the last experiment, the three-level classification, we had three different tasks, namely emotional versus non-emotional, polarity, and emotional classification. Each of these tasks has its own specification; therefore we can definitely benefit from analyzing each task separately and defining different sets of features and classification methods for each task rather than using the same method for every task. One of the first steps for our future work will be using the emotional lexicon and their corresponding features in the third level of our hierarchy – classifying five negative emotions. We believe that using the features derived from the two emotion lexicons will improve the results of the third level.

References

1. Wiebe, J., Wilson, T., Cardie, C.: Annotating Expressions of Opinions and Emotions in Language. Language Resources and Evaluation 39, 165–210 (2005)
2. Ekman, P.: An Argument for Basic Emotions. Cognition and Emotion 6, 169–200 (1992)
3. Aman, S.: Identifying Expressions of Emotion in Text. Master's thesis, University of Ottawa, Ottawa, Canada (2007)
4. Neviarouskaya, A., Prendinger, H., Ishizuka, M.: Compositionality Principle in Recognition of Fine-Grained Emotions from Text. In: Proc. Third International ICWSM Conference, pp. 278–281 (2009)
5. Alm, C.O., Roth, D., Sproat, R.: Emotions from text: machine learning for text- based emotion prediction. In: Proc. Joint Conference on Human Language Technology / Empirical Methods in Natural Language Processing (HLT/EMNLP 2005), Vancouver, Canada, pp. 579–586 (2005)
6. Strapparava, C., Mihalcea, R.: SemEval-2007 Task 14: Affective Text (2007)
7. Strapparava, C., Mihalcea, R.: Learning to Identify Emotions in Text. In: Proc. ACM Symposium on Applied computing, Fortaleza, Brazil, pp. 1556–1560 (2008)
8. Chaumartin, F.: Upar7: A knowledge-based system for headline sentiment tagging. In: Proc. SemEval 2007, Prague, Czech Republic (June 2007)
9. Kozareva, Z., Navarro, B., Vazquez, S., Montoyo, A.: Ua-zbsa: A headline emotion classification through web information. In: Proc. SemEval 2007, Prague, Czech Republic (June 2007)
10. Katz, P., Singleton, M., Wicentowski, R.: Swat-mp: the semeval-2007 systems for task 5 and task 14. In: Proc. SemEval 2007, Prague, Czech Republic (June 2007)
11. Aman, S., Szpakowicz, S.: Using Roget's Thesaurus for Fine-grained Emotion Recognition. In: Proc. Third International Joint Conf. on Natural Language Processing (IJCNLP), Hyderabad, India, pp. 296–302 (2008)
12. Kennedy, A., Inkpen, D.: Sentiment classification of movie reviews using contextual valence shifter. Computational Intelligence 22, 110–125 (2006)

13. Keshtkar, F., Inkpen, D.: Using Sentiment Orientation Features for Mood Classification in Blog Corpus. In: IEEE International Conf. on Natural Language Processing and Knowledge Engineering, Dalian, China, September 24-27 (2009)
14. Koller, D., Sahami, M.: Hierarchically Classifying Documents Using Very Few Words. In: Proc. International Conference on Machine Learning, pp. 170–178 (1997)
15. Kiritchenko, S., Matwin, S., Nock, R., Fazel Famili, A.: Learning and Evaluation in the Presence of Class Hierarchies: Application to Text Categorization. LNCS, pp. 395–406. Springer, Heidelberg (2006)
16. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity: an exploration of features for phrase-level sentiment analysis. Computational Linguistics 35(3), 399–433 (2009)
17. Strapparava, C., Valitutti, A.: WordNet-Affect: an affective extension of WordNet. In: Proc. 4th International Conference on Language Resources and Evaluation (LREC 2004), Lisbon, Portugal, pp. 1083–1086 (2004)
18. Jarmasz, M., Szpakowicz, S.: Roget's Thesaurus and Semantic Similarity. In: Nicolov, N., Bontcheva, K., Angelova, G., Mitkov, R. (eds.) Recent Advances in Natural Language Processing III: Selected Papers from RANLP, John Benjamins, Current Issues in Linguistic Theory, vol. 260, pp. 111–120 (2003)
19. Riloff, E., Wiebe, J.: Learning extraction patterns for subjective expressions. In: Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP 2003), Sapporo, pp. 105–112 (2003)
20. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)

Supervised Machine Learning for Summarizing Legal Documents

Mehdi Yousfi-Monod[1], Atefeh Farzindar[2], and Guy Lapalme[1]

[1] Université de Montréal, Laboratoire RALI
RALI-DIRO Université de Montréal, C.P. 6128, succursale Centre-ville,
Montréal, Québec, Canada H3C 3J7
{yousfim,lapalme}@iro.umontreal.ca
[2] *NLP Technologies Inc.*
1255 University Street, suite 1212, Montréal, Québec, Canada, H3B 3W9
farzindar@nlptechnologies.ca

Abstract. This paper presents a supervised machine learning approach for summarizing legal documents. A commercial system for the analysis and summarization of legal documents provided us with a corpus of almost 4,000 text and extract pairs for our machine learning experiments. That corpus was pre-processed to identify the selected source sentences in extracts from which we generated legal structured data. We finally describe our sentence classification experiments relying on a Naive Bayes classifier using a set of surface, emphasis, and content features.

1 Introduction

Legal information is produced in large quantities and needs to be adequately classified in order to be reliably accessible. In Canada, federal and provincial courts produce around 200,000 decisions each year [1]. Classifying these documents is usually performed by legal experts and requires accuracy and speed. These legal experts often summarize decisions and look for information relevant to specific cases in these summaries. The high quality required for these summaries cannot be achieved by commonly available automatic summarization methods as was shown by Farzindar [2] who compared different summarization methods whose results were evaluated by legal experts. Using these results, *NLP Technologies Inc.* has developed a summarization system, named *DecisionExpress*^{TM}, based on a thematic segmentation of the text, specifically tailored to the legal field. Chieze *et al.* [3] detail the automatic summarization system as well as other legal information services offered by the company. As far as we now, there has been no other work dealing with the large scale and domain specific summarization of documents produced by Canadian federal courts.

DecisionExpress^{TM} relies on a symbolic approach based on a set of linguistic rules developed after a meticulous manual analysis of legal documents. The summaries are produced by extraction of whole sentences, often whole paragraphs, rather than by abstraction (rewriting). The reason is that an abstract may be less

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 51–62, 2010.

accurate and less credible because it is not a direct citation of the decision; reformulation may lead to misinterpretation of the judge's intent. Extracts guarantee that the summary contains only original sentences that can be cited verbatim without having to refer to the original decision. This symbolic summarization approach was developed when no text and extract pairs corpus was available for supervised machine learning. Between June 2008 and June 2009, more than 4,000 decisions have been analyzed and summarized, providing us with a significant and valuable corpus. Unfortunately, the format of the extracts could not be used directly for supervised learning, so documents had to be pre-processed.

In the following section, the summarization process of *DecisionExpress*^{TM} is presented. In Section 3, we report our work on creating the corpus. Our experiments on a model for supervised learning, using the previously generated data, are described and discussed in Section 4. We conclude by introducing new perspectives.

2 Producing the Summary of a Legal Decision

Decisions are available on the Canada courts' websites in HTML[1]. which are analyzed in *DecisionExpress*^{TM} to produce an analytic sheet for each decision containing information extracted from the decision such as the decision's headline and conclusion, the judge's name, the court level and the addressed topics.

Most of the analysis relies on text content rather than the HTML document structure. Since HTML tags define the appearance of the decisions, rather than their structure, and since the presentation as well as its HTML definition is subject to change over time (and it has), we cannot rely on these tags alone to identify the structure of the decisions. Nevertheless, there are cases where text content is not enough and we rely on HTML emphasis to extract some structural elements, as explained below. Linguistic cues, text segments matched by a context-free grammar, are used to identify the decision structure as well as relevant factual information. The output of this analysis is saved using an XML data structure. The division into structural elements relies on a specific knowledge of the legal field [4,5] and defines 4 decision sections or themes: **Introduction**, **Context**, **Reasoning** and **Conclusion**.

Exploiting text content allows the identification of most structural elements of decisions. Unfortunately, subsection titles seldom match regular lexical patterns, so the lexicon cannot be reliably used to identify such subtitles. It is important to locate these structural elements in order to improve the quality of automatic summaries and human reviews. Indeed, if subsections are not identified, their title and content are merged with the previous one, hence losing their legitimate and required salience. Fortunately, in our case there is a quite reliable clue: HTML emphasis. From a manual analysis of a sample HTML document, supported by legal advice, we defined a conditional rule covering most subtitles

[1] For example, `http://decisions.fct-cf.gc.ca/en/2009/2009fc1188/2009fc1188.html` presents a decision of the Federal Court of Canada, in English, on 19 November 2009.

emphasis: subtitles are generally sentences in either bold, underline or italic and not indented. We then identified a list of HTML elements and CSS attributes defining such emphases, and a specific XML attribute was added to matching sentences. The automatic summarization process relies on attributes added by the transducers and stylesheets to the XML tags of the sentences. The process uses a set of rules that match syntactic patterns relying on part-of-speech information and specific lexicon to define salient sentences for each decision's theme. The process consists in keeping a percentage of salient sentences for each theme.

Until May 2009, *DecisionExpress*^{TM} relied on a plain text reviewing system with which lawyers revised automatic extracts by cut-and-paste operations. The extracts were saved as plain text into the database. This is why we have to process such data before being able to use it for supervised learning, as described in the next section. The reviewing task now benefits from an interactive graphical Web interface, named RevSum, in which lawyers insert or remove whole sentences and paragraphs from the summary by simply clicking on them. This new interface saves the summary into the XML document, by adding attributes to selected sentence tags. Hence the XML structure is preserved during the whole process and these texts can be used directly in our learning algorithm experiments.

3 Building the Corpus

DecisionExpress^{TM} relies on a symbolic approach based on linguistic rules according to a meticulous manual analysis of the legal documents, helped by legal experts (lawyers). Between June 2008 and May 2009, more than 4,000 decisions have been created and revised, providing us with a significant and valuable corpus. Unfortunately, the format of the extracts could not be used directly for machine learning. In order to train a categorization model, we need to know which sentences of the source documents were selected for the extracts. So the first step is to identify the source sentence of each extract sentence of the corpus.

Daniel Marcu [6] tackled a related problem in which summaries were abstracts, not extracts. His heuristic consists in removing clauses from the text until the resulting extract is similar enough to the abstract. Our case is simpler and different because sentences are expected to be at least similar so we decided to develop our own method.

Source documents being in XML, each sentence is delimited by an <S> tag. Plain text extract are split into four sections related to the legal themes described above. This difference gave rise to the following issues:

Sentence boundaries detection. The HTML <p> tags around many sentences in source documents that eased the parsing of sentences were not kept in the extracts; therefore, we had to rely on punctuation to deal with abbreviations (e.g. "Mr.") and sentences without end punctuation (e.g. bulleted lists). Fortunately, sometimes there are reliable markers showing the beginning of a sentence, namely paragraph numbers (e.g. "[25]").

Sentence alterations. When generating the summaries, legal experts may have modified sentences, even though, in principle, it was forbidden. The reason is that sometimes it is convenient to remove an unnecessary part of a sentence in a summary[2] or to merge two short sentences. So when matching sentences, we have to look for part of sentences and decide whether it is relevant to identify sentences that have been shortened. We also found cases in which sentences or parts of them have been rewritten. This may cause misspelled words, case changes and even translations. Thus, our similarity function has to be tolerant to slight modifications.

Sentence reordering. Within each of the four themes, we expected that sentences match with the order of the source sentences, however we have found several cases in which the legal experts had reordered the sentences. So we cannot always rely on the order of the source sentences for matching.

Sentence similarities. In the legal field, it is common to see repetition (phrases, clauses or even whole sentences) within a document. Therefore, when identifying sentences, selecting the first text and extract pair that matches may not be the best heuristic. We also have to exploit other clues like ordering.

To deal with those issues, we went through several attempts to identify target sentences while we gradually discovered problematic cases. For each summary, we first determined the sentence boundaries within sections and then matched sentences from the abstract with the ones from the summary using the following three-steps procedure performed iteratively over the four themes:

1. As different sentences may have string similarities and may even include one another, we decided to reduce the risk of wrong matches by trying to first identify the longest sentences. We loop over target sentences, from the longest to the shortest, and for each one, we do a string comparison with each source sentence, also in a decreasing length order. We use the Levenstein edit distance algorithm to allow light modifications (set at 10% of character difference); this level of variation is also allowed in the next steps. We stop when we reach short sentences (less than 50 characters), which are processed in the next step.
2. The shorter the sentences, the greater the chances that they may be similar to others. Some may even be included into longer ones. To reduce risks of wrongful identification, we decided to partially rely on sentence order by trusting the matched sentences in the first step. Sentences are now sorted and processed according to their original order. Identification of the remaining sentences is done only within intervals defined by the previously identified sentences.
3. As some truncated or merged sentences may remain, we try to identify sentence inclusions. We match a summary sentence containing a source sentence if the former's length is within 50% and 150% of the latter. Outside this interval, we consider the sentence would bring noise to the summary, as its extra content was not selected by the experts.

[2] For example, the reference after the colon in "This is a question of mixed fact and law, to be reviewed on a standard of reasonableness: Elezi v. Canada (Minister of Citizenship and Immigration), 2007 FC 240."

Table 1. Corpus distribution over language and fields: immigration (IMM), tax (TAX) and intellectual property (IP)

Domain	IMM	TAX	IP	Total
English	1 765	447	176	2 388
French	1 155	164	8	1 327
Total	2 820	611	184	3 715

Once all sentences of the corpus have been processed, we have a set of XML documents in which each sentence is either tagged as kept in one of the four sections of the summary or not tagged. For some summaries, a significant proportion of the sentences were not identified. This generally happens when sentences are rewritten by lawyers, usually translated. As such documents may bias the training process because unidentified sentences will be considered as negative examples, we decided to remove the documents in which less than 70% of sentences in the summary were matched. From 4,067 documents, we removed 352 and our final corpus is then composed of 3,715 documents, where 94% of the sentences and 93% of the words have been identified. Since we allowed a small editing distance, there are 1.9% of character insertions, deletions or substitutions among identified sentences. Table 1 presents the distribution of English and French decisions in three fields.

4 Machine Learning Experiments

4.1 Categories and Features

Our extract-based summarizer has to classify sentences as being in the summary or not, and our extracts are placed into four sections. We thus have a total of five categories: *not in summary (NIS)*, *Introduction*, *Context*, *Reasoning* and *Conclusion*. Table 2 presents the distribution of the instances of our corpus over all five categories. Depending on the legal field, documents have notable structure dissimilarities as well as differences in the summarization method used by legal experts. We therefore decided to train our models on a single field at a time. As some training features rely on vocabulary, we also decided to deal with one language at a time. In this paper we detail our experiments with a corpus of English decisions from the immigration field as it is the largest sub-corpus we have: 1,765 documents with 65,345 instances. We will describe some results in other fields and languages in Section 4.4.

For the learning process, we split the instances of the corpus into 2/3 for the training set and 1/3 for the test set. The classifier used is the Weka [7] implementation of the popular Naive Bayes, with the supervised discrimination option enabled. We also ran the classification with other bayesian-like and support vector machine classifiers as well as some based on tree decision algorithms but they did not yield better results. We then explored the relevance of several features for our categorization task:

Table 2. Distribution of instances over categories in the corpus

Classes	NIS	Introduction	Context	Reasoning	Conclusion	Total
# Instances	142 277	3 462	18 941	28 693	2 663	196 036
% Instances	72.6%	1.8%	9.7%	14.6%	1.4%	100%

Surface features. Such common features exploit the decision structure of the source document: sentence position in a paragraph, paragraph position in a section, section position, sentence length of words and number of sentences in a paragraph.

Emphasis features. As the analysis of the HTML source decisions preserves part of the emphasis in some sentences and as emphasis is closely related to salience, and thus to relevance, we decided to assess the usefulness of such information. Emphasis features are bold, underline, italic and indent, and take a boolean value.

Content features. We tested 2 features relying on the vocabulary of the decisions. The first uses the sum of each word's *tf* · *idf* score, the result is normalized with the sentence length (in words). The second relies on the legal genre where there are specific words regularly used to express an opinion or declare a fact, which are in sentences generally relevant for the summary. Examples of such words are "apparently", "dismissed", "daughter" or "kill". Over the 1,765 documents of the training corpus, the word "dismissed" appears 1,623 times in all the extracts and 1,582 times in other sentences, while most words usually appear at least 2 to 3 times more in sentences not kept for the summaries. Other instances include the words "paragraphs", "relies" or "procedure". This led us to add such a score, based on the ratio of how many times a term appears in the extract sentences, to how many times it appears in other sentences. This score for a sentence S is the normalized sum of such a ratio of each word:

$$\frac{\Sigma_{w \in S}(\frac{tf_{se}(w)}{tf_{sne}(w)})^2}{length(S)} \tag{1}$$

$tf_{se}(w)$ represents the number of times the word w appears in the corpus in sentences selected for the extracts and $tf_{nse}(w)$ how many times it appears in sentences not selected for the extracts. The power applied to the ratio of frequencies helps discriminating words specific to extracts from others.

4.2 Classification Results and Discussion

We tried different groups of features and the most relevant are shown in Table 3. All features have a positive impact on the classification when considering the F_1-Measure of the whole summary. Our best overall results are obtained by the use of all feature groups, we name this configuration ProdSum (PRobabilistic Decision SUMmarizer).

Table 3. Classification Precision, Recall and F$_1$-Measure based on different feature groups, for each summary sections plus the whole summary of the English immigration corpus. Features are (Sur)face, (Em)phasis and (Con)tent.

	Features	**Intro.**	**Cont.**	**Reas.**	**Concl.**	**Summary**
Precision	Sur	0.644	**0.523**	0.371	0.380	**0.466**
	Sur+Em	0.645	0.490	0.360	0.377	0.438
	Sur+Voc	**0.651**	0.505	**0.390**	**0.389**	0.458
	Sur+Em+Voc	0.649	0.492	0.388	0.387	0.448
Recall	Sur	0.789	0.499	0.190	0.621	0.360
	Sur+Em	0.795	0.595	0.291	0.617	0.447
	Sur+Voc	0.804	0.715	0.356	0.627	0.525
	Sur+Em+Voc	**0.809**	**0.741**	**0.432**	**0.630**	**0.574**
F$_1$-Measure	Sur	0.709	0.511	0.251	0.471	0.406
	Sur+Em	0.712	0.537	0.322	0.468	0.443
	Sur+Voc	0.719	**0.592**	0.372	**0.480**	0.489
	Sur+Em+Voc	**0.720**	**0.592**	**0.409**	**0.480**	**0.503**

The emphasis features have no significant impact on introduction and conclusion categories because sentences in these sections are seldom emphasized. Surface cues greatly help for these two categories and are even enough to achieve our best results, which means that such sections are composed of sentences extracted from constant parts of the decisions. Introduction gets the highest score for both precision (0.649) and recall (0.809) because the most relevant text content of this legal theme is often made of sentences from the first paragraph of the decision. Surprisingly, regarding the context section, adding other features not only increases noise, but also increases recall. Context and conclusion sections reach an acceptable recall, respectively 0.741 and 0.630, mostly through surface features.

The reasoning section, which is usually the longest of the decision, needs other features than the surface ones in order to get an adequate recall; relevant sentences of this section do not solely depend on their position, so we need content and emphasis information to evaluate their relevance.

4.3 Comparison with a Baseline and ASLI

In order to assess the performance of our classification we had to defined two baselines. Our first baseline, adapted from the *start-end* baseline of Farzindar [2], constructs an extract from the first N sentences of each section from the source document. While Farzindar retrieved 15% of the source document (12% from the start and 3% from the end) to create the baseline, we relied on the actual compression ratio of our training corpus, shown in Table 2, which amounts to an average of 27.5%. The extracts of our first baseline were composed of the extraction of the first 1.8% words of the decision's introduction, the first 9.7% of the context, and so on. The last sentence was added in full if it was to be cut by the percentage. It turned out that this compression rate worked better than

Table 4. Comparison of ProdSum to a baseline and ASLI for the English immigration corpus

	System	Intro.	Cont.	Reas.	Concl.	Summary
Precision	Baseline	0.626	0.449	0.182	0.245	0.332
	ASLI	**0.699**	0.390	0.291	0.339	0.362
	ProdSum	0.649	**0.492**	**0.388**	**0.387**	**0.448**
Recall	Baseline	0.544	0.544	0.131	0.470	0.319
	ASLI	**0.878**	0.690	0.330	**0.666**	0.509
	ProdSum	0.809	**0.741**	**0.432**	0.630	**0.574**
F_1-Measure	Baseline	0.582	0.492	0.152	0.322	0.325
	ASLI	**0.778**	0.498	0.309	0.450	0.423
	ProdSum	0.720	**0.592**	**0.409**	**0.480**	**0.503**

the original 15%. Our second baseline is the current automatic summarization system of *DecisionExpress*^{TM}: ASLI. Scores of the latter baseline may be biased as sentences of our corpus have been first selected by ASLI's algorithm. However, the review process, achieved by legal experts, did alter that sentence selection and thus will reduce the bias.

Table 4 provides classification scores for the three systems tested: a baseline, ASLI and ProdSum.

The baseline, while not as efficient as other systems, still managed to get satisfactory scores for the introduction and context sections because most of the relevant information in these sections is found at the beginning. ASLI has a slight advantage over ProdSum when dealing with the introduction, but is generally outperformed for other sections, specifically with respect to the whole summary.

4.4 Results for Other Fields and Language

Table 5 shows the results (F_1-Measure scores) of ProdSum compared to ASLI for two fields, immigration and tax, and two languages, English and French.

We do not provide results for the intellectual property field as there is not enough training data to yield relevant scores as of yet. The experiment on the English tax field resulted in a F_1-Measure score of 0.445, which is a bit lower than the 0.503 score of the immigration field but far greater than the corresponding score obtained by ASLI (0.190). ASLI obtained a low score because it selected too many sentences for the introduction, leading to a 0.058 precision score for that section. It usually works well for the immigration field, but introductions in the tax field cover a large part of the decisions, so most of them should be removed to produce the extract. The underlying reason for that low score is the specialization of the symbolic approach to a specific field. Indeed, ASLI has been developed more specifically for immigration documents, which are more numerous than other fields, while ProdSum, as a statistic approach, adapts better to new fields. The French corpus also works well with ProdSum which yielded, for the immigration field, a F_1-Measure score of 0.483, proving that

Table 5. F_1-Measure scores of ASLI and ProdSum for English and French languages and immigration and tax fields

Language	Domain	# Documents	# Instances	ASLI	ProdSum
English	**IMM**	1765	65345	0.423	**0.503**
	TAX	447	21517	0.190	**0.445**
French	**IMM**	1155	40293	0.433	**0.483**
	TAX	164	8380	0.344	**0.368**

ProdSum is also suitable for French decisions. The small French tax corpus got a score of 0.368, which is comparable to the English version, but may not be relevant because of the low amount of available training data. Finally, ProdSum obtained the best overall results, notwithstanding the field and language.

4.5 Rouge Evaluation

The Rouge metric is typically used to compare automatic extracts with human abstracts. While our reference summaries are extracts, misclassified sentences may contain relevant content which may be captured by Rouge measures. Rouge does not have a default configuration for the French language, and as we used the stemmer and stop word options, we only did runs for the English language, in the immigration and tax fields. Our main goal was to assess performances of our summarizer for each legal theme. Therefore, we evaluated each section separately. We were also curious to know if the full automatic summaries matched well with the references, so we did a run with the full summaries, i.e. extracts with the four sections. All runs used Rouge's configuration[3] of the DUC[4] 2007 conference, which yielded Rouge-{1, 2 and SU4} scores. Table 6 shows F_1-Measure Rouge scores of our experiment. We give F_1-Measure scores instead of the recall ones as we had almost no control on the size of our system and reference summaries, thus noise has to be taken into account. Scores are higher than those which traditional summarizers usually perform because we are comparing our extracts with other extracts made of sentences of the same source documents, not abstracts. Rouge-1 scores are very similar to other ones so we do not display them.

ProdSum gets the overall best results. When dealing with the smallest sections – introduction and conclusion – ASLI gets slightly better scores for the immigration field, due to better precision. The reason is that the symbolic method has rules to detect and exclude citations of the decision, which are not relevant to the summary, whereas our system does not have any feature dealing specifically with such cases. The low classification scores ASLI got for the context and

[3] Rouge version 1.5.5, with the following command line options: `-n 2 -x -m -2 4 -u -c 95 -r 1000 -f A -p 0.5 -t 0 -a` (use Porter stemmer on both models and peers, use 95% confidence interval, bootstrap resample 1000 times to estimate these 95%, compute F-measure with alpha = 0.5).

[4] http://duc.nist.gov/

Table 6. F_1-Measure Rouge scores of all systems for English immigration and tax documents

Domain	Measure	System	Intro.	Cont.	Reas.	Concl.	Full
IMM	Rouge-2	Baseline	0.692	0.496	0.173	0.414	0.435
		ASLI	**0.778**	0.499	0.270	**0.494**	0.577
		ProdSum	0.768	**0.554**	**0.369**	0.479	**0.633**
	Rouge-SU4	Baseline	0.690	0.503	0.192	0.402	0.452
		ASLI	**0.779**	0.503	0.279	**0.479**	0.590
		ProdSum	0.766	**0.556**	**0.376**	0.461	**0.640**
TAX	Rouge-2	Baseline	0.473	0.155	0.090	**0.414**	0.278
		ASLI	0.257	0.147	0.123	0.396	0.598
		ProdSum	**0.507**	**0.403**	**0.445**	0.402	**0.661**
	Rouge-SU4	Baseline	0.473	0.162	0.097	**0.408**	0.289
		ASLI	0.256	0.152	0.125	0.387	0.604
		ProdSum	**0.507**	**0.414**	**0.453**	0.393	**0.667**

reasoning sections of the tax field are confirmed by Rouge scores. The baseline gets the best conclusion scores for the tax field because it selects few sentences, reducing noise thus increasing precision, which is favored by the $F_{alpha=0.5}$-Measure score. Finally, full summaries matched best with ProdSum extracts, notwithstanding the field and measure. It is interesting to note the full summaries scores are globally higher than section scores, regardless of the system, thus indicating that some sentences were wrongly classified in a summary theme, while they actually belonged to another one.

The scores differences are not incidental according to significance tests we did on each legal theme and for full summaries. We used the standard paired t-test on ProdSum and ASLI, for Rouge-SU4 per evaluation score results, on both legal fields. While result differences of the small introduction and conclusion sections proved to be incidental ($0.3 < p - value < 0.9$), ProdSum's results on context and reasoning sections, as well as full summaries, are significantly better ($p - value < 0.0001$) than ASLI's.

5 Related Work

There have been a few other approaches dealing with automatic summarization of legal documents, and the best source for an overview of such works is certainly [8], where the author presents an excellent survey of the area of summarization of court decisions. She describes the context in which court decisions are taken and published and the need for good quality summaries in this area which is comparable to the medical field.

FLEXICON [9] is one of the first summarization system specialized for legal texts, it was a symbolic approach based on the use of keywords found in a legal phrase dictionary. The summaries were not used as such but served for indexing a legal case text collection.

SALOMON [10], developed for summarization of Belgian criminal cases, was the first to explicitly make use of the structure of a case. The system first identifies the discourse structure with text grammars a process similar to the one used in the first phase of *DecisionExpress*^{TM}. The next step produces the summary by selecting relevant paragraphs of each important document section. Paragraphs are represented as vectors of index terms and are grouped by a clustering algorithm. This process aims to removing redundant information and grouping paragraphs into thematically coherent units. SALOMON's approach uses shallow information as ProdSum do but does not rely on machine learning.

Hachey and Grover [11] present an approach closely related to ours. They exploit a corpus of 188 decisions of the House of Lords they have gathered and annotated. Sentences are tagged with rhetorical status, relevance and linguistic information. The authors performed sentence classification experiments with Naive Bayes and maximum entropy models, using shallow information and named entities as features. They only provide prediction scores for individual features, and the best one, F-Score of 31.2, goes to the "thematic words" feature for the Naive Bayes classifier. This feature is a basic $tf \cdot idf$ score, similar to ours. The authors have not performed any manual or automatic content-based evaluation.

6 Conclusion

In this paper we introduced an approach for selecting important sentences from legal documents using supervised machine learning. We first described a system for legal document analysis and summarization which is provided with a valuable and significant corpus of text and extract pairs. That corpus was processed to identify the source sentences contained in the plain text extracts. We also presented our work on generating an XML structured data, dealing with issues specific to the legal field. The machine learning step consisted in running a sentence classification algorithm, Naive Bayes, based on a set of surface, emphasis and content features. Our system, ProdSum, has been compared with a baseline system and with ASLI, the current automatic summarization system of *DecisionExpress*^{TM} (before revision). While ASLI may compete with ProdSum on one or two of the smallest legal themes, our system obtained the best overall results.

While we only used rather standard features, it turned out to be enough to beat the symbolic method. To reach better classification scores, particularly for the context and reasoning legal themes, we plan to explore features based on events and factual information as it is the purpose of such sections to gather temporal and factual evidence in order to support the verdict.

Acknowledgment

We thank the Precarn for partially funding this work. We sincerely thank our lawyers Pia Zambelli and Diane Doray. The authors also thank Fabrizio Gotti and Farnaz Shariat for technical support, Vincenzo Mignacca for his meticulous proofreading of the final version of this paper, and reviewers for their comments and suggestions.

References

1. Plamondon, L., Lapalme, G., Pelletier, F.: Anonymisation de décisions de justice. In: XIe Conférence sur le Traitement Automatique des Langues Naturelles (TALN 2004), May 2004, pp. 367–376 (2004)
2. Farzindar, A.: Résumé automatique de textes juridiques. PhD thesis, Université de Montréal et Université Paris IV-Sorbonne (March 2005)
3. Chieze, E., Farzindar, A., Lapalme, G.: An automatic system for summarization and information extraction of legal information. In: Accepted in Semantic Processing of Legal Texts, pp. 1–20. Springer, Heidelberg (2009)
4. Farzindar, A., Lapalme, G.: Letsum, an automatic legal text summarizing system. In: Gordon, T.F. (ed.) Legal Knowledge and Information Systems, Jurix 2004: the Sevententh Annual Conference, pp. 11–18. IOS Press, Berlin (December 2004)
5. Farzindar, A., Lapalme, G.: Production automatique du résumé de textes juridiques: évaluation de qualité et d'acceptabilité. In: TALN 2005, Dourdan, France, June 2005, vol. 1, pp. 183–192 (2005)
6. Marcu, D.: The automatic construction of large-scale corpora for summarization research, pp. 137–144. University of California, Berkely (1999)
7. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11(1) (2009)
8. Moens, M.F.: Summarizing court decisions. Inf. Process. Manage. 43(6), 1748–1764 (2007)
9. Smith, J., Deedman, C.: The application of expert systems technology to case-based law. In: Proceedings of the First International Conference on Artificial Intelligence and Law, Boston, Mass, pp. 84–93. The Center for Law and Computer Science, Northeastern University (1987)
10. Moens, M.F., Uyttendaele, C., Dumortier, J.: Abstracting of legal cases: the potential of clustering based on the selection of representative objects. J. Am. Soc. Inf. Sci. 50(2), 151–161 (1999)
11. Hachey, B., Grover, C.: Extractive summarisation of legal texts. Artif. Intell. Law 14(4), 305–345 (2006)

Toward a Gold Standard for Extractive Text Summarization

Alistair Kennedy[1] and Stan Szpakowicz[1,2]

[1] School of Information Technology and Engineering
University of Ottawa, Ottawa, Ontario, Canada
{akennedy,szpak}@site.uottawa.ca
[2] Institute of Computer Science
Polish Academy of Sciences, Warsaw, Poland

Abstract. Extractive text summarization is the process of selecting relevant sentences from a collection of documents, perhaps only a single document, and arranging such sentences in a purposeful way to form a summary of this collection. The question arises just how good extractive summarization can ever be. Without generating language to express the gist of a text – its abstract – can we expect to make summaries which are both readable and informative? In search for an answer, we employed a corpus partially labelled with Summary Content Units: snippets which convey the main ideas in the document collection. Starting from this corpus, we created SCU-optimal summaries for extractive summarization. We support the claim of optimality with a series of experiments.

1 Introduction

One of the hardest tasks in Natural Language Processing is text summarization: given a document or a collection of related documents, generate a short – often very short – text which presents only the main points of those documents. There can be a generic summary, when there are no restrictions other than the required compression into the most salient points, or a query-driven summary, when we seek answers to one or more questions, or focus on the broad topic of the query. Language generation is quite a difficult task, for which no easily applicable tools exist in the public domain; in any event, generation would require the creation of a detailed formal model of the summary, itself a formidable task. That is why summarization systems usually rely on *extracting* a set of relevant sentences and then arranging them into a summary. The inherent imperfection of such summaries invites some obvious questions: just how good can an extractive summary ever be? Does the reliance on stitching sentences together rather than generating new text mean that we cannot hope to achieve the quality of summaries generated by hand? We will argue that many of the criteria which underlie summary evaluation can be re-examined by way of building and evaluating upper-bound extractive summaries.

Previous work on finding baseline systems for extractive summarization includes Optimal Position Policy [1] and Sub-Optimal Position Policy [2]. Those

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 63–74, 2010.

are *lower* bounds on how good an automatic text summarization system should be. We present an *upper* bound on how good an extractive summary we can really hope to produce. We evaluate sample summaries using a variety of standard evaluation techniques, including manual evaluation and semi-automatic methods, specifically ROUGE [3]. Unlike algorithms for building baseline summaries, which require no summarizer output for comparison, our gold-standard summaries are built using a corpus labelled with Summary Content Units (SCUs) – Section 1.1 defines them. The SCU-labelled corpus has been compiled every year since 2005, and first noted in [4]. We attempt to maximize the number of SCUs according to one of two criteria. We believe that generating SCU-optimal summaries is an important step toward generating a gold standard for extractive text summarization. There has been earlier work on manually constructing extractive upper bounds [5]. In TAC 2009, a run of that system was evaluated and scored at or near the top in terms of responsiveness, readability and SCU scores. That was a rating with respect to the other peer systems; manually built reference summaries still did better.

The Text Analysis Conference (TAC; formerly Document Understanding Conference, or DUC), organized annually by the National Institute of Standards and Technology (NIST), includes tasks in text summarization. In 2005-2007, the challenge was to generate 250-word summaries of news article collections of 20-50 articles. Summaries were to be built around a query – a few questions on the main topic of the collection and perhaps postulates for how to answer the questions. In 2008-2009 (after a 2007 pilot), the focus has shifted to creating *update summaries*. The document set is split into a few subsets. From each subset, a 100-word summary is generated. The subsets are ordered chronologically, and the goal is to exclude from a summary any information which can be found in a previous document set. For example, given subsets A_1, A_2 and A_3, a summary for A_1, $sum(A_1)$, will be generated normally, while $sum(A_2)$ must not contain any information found in document set A_1. Likewise $sum(A_3)$ should not contain information from document sets A_1 and A_2.

1.1 Manual Evaluation at TAC

Manual summary evaluation[1] at DUC/TAC, financed by NIST, is an expensive but highly useful part of the exercise. It includes *pyramid evaluation*, outlined in [6], which begins with creating several reference summaries and determining what information they contain is most relevant. A relevant element is called a Summary Content Unit (SCU), carried in text by a varying-size fragment, between a few words and a complete sentence. All SCUs, marked in the reference summaries, make up a so-called pyramid, with few frequent SCUs at the top and many rare ones at the bottom. In the actual pyramid evaluation, annotators use a custom-made tool to identify SCU occurrences in peer summaries. More SCUs mean more relevance for a peer summary; there may be redundancy if a SCU appears more than once. If a peer summary contains relevant information absent from reference summaries, the tool allows the creation of a new SCU. Two

[1] See ⟨`www.nist.gov/tac/2009/Summarization/update.summ.09.guidelines.html`⟩

kinds of scores measure the quality of the summary after pyramid evaluation: the pyramid score (precision) and the modified pyramid score (recall) [6]. Only modified pyramid scores are reported in TAC.

Pyramid evaluation supplements the manual summary evaluation for readability and overall responsiveness. Readability evaluates a mixture of grammaticality, non-redundancy, referential clarity, focus, and structure/coherence. For the latter, trained evaluators read peer summaries and score them 1 to 10.

1.2 Semi-automatic Summary Evaluation

Semi-automatic evaluation, notably ROUGE [3], requires hand-made reference summaries to measure lexical overlap between these references and a summary generated by a participating summarization system – a *peer summary* in the TAC terminology. Variations on this method include using synonyms and finding matching sub-sequences. The two used in TAC/DUC are ROUGE-2 and ROUGE-SU4. ROUGE-2 seeks to match bigrams between the peer and reference summaries. ROUGE-SU4 allows for unigrams and for *skip bigrams*, non-adjacent word pairs up to 4 spaces apart. In [7], an upper bound on ROUGE scores was determined by selecting sentences which contained bigrams most frequent in the reference summaries. The results for the 2008 data are presented in Section 5.

Another method used in recent years is Basic Elements (BE) [8]. It works similarly to ROUGE, but requires syntactic parsing to match syntactic structures in reference summaries and peer summaries. Both the BE and ROUGE measures have been found highly correlated with the responsiveness of a summary, 0.975 and 0.972 Pearson correlation respectively on the 2005 DUC data.

2 The SCU-Labelled Corpus

One of the primary advantages of pyramid evaluation is that it provides us with fully annotated peer summaries. Assuming, then, that TAC peers usually build extractive summaries, it becomes feasible to map the sentences from these summaries back to the original corpus [4]. Many sentences in the corpus can be labelled with the list of SCUs they contain, as well as the score for each of these SCUs and their identifiers. [9] reported a mapping back to the original corpus of 83% and 96% of the sentences from the peer summaries in 2005 and 2006 respectively. A dataset has been generated for the DUC/TAC main task data in years 2005-2009, and the update task in 2007. This corpus indicates what useful information is included in a sentence and can be used to give sentences scores. We generate the SCU-optimal summaries by assembling summaries from the highest-scored sentences.

Figure 1 illustrates the format of the data. The example comes from the 2008 data set D0801; the goal was to build a summary around the query “Airbus A380 – Describe developments in the production and launch of the Airbus A380”. The first sentence is tagged with the <annotation> tag indicating that it was used in at least one summary. This sentence appeared in exactly one summary, with ID 0. There are two SCUs. One, with ID 11, is “Airbus A380 flew its maiden

<line>*As opposed to the international media hype that surrounded last week's flight, with hundreds of journalists on site to capture the historic moment, Airbus chose to conduct Wednesday's test more discreetly.* <annotation scu-count=”2” sum-count=”1” sums=”0”><scu uid=”11” label=”Airbus A380 flew its maiden test flight” weight=”4”/><scu uid=”12” label=”taking its maiden flight April 27” weight=”3”/></annotation> </line>

<line>*After its glitzy debut, the new Airbus super-jumbo jet A380 now must prove soon it can fly, and eventually turn a profit.*<annotation scu-count=”0” sum-count=”3” sums=”14,44,57”/> </line>

<line> *“The takeoff went perfectly,” Alain Garcia, an Airbus engineering executive, told the LCI television station in Paris.*</line>

Fig. 1. Positive, negative and unlabelled sentence examples for the query “Airbus A380 – Describe developments in the production and launch of the Airbus A380”

test flight” with a weight of 4. The other, with ID 12, is “taking its maiden flight April 27” with a weight of 2. This is an example of a positive sentence with a weight of 6. The second sentence in Figure 1 is annotated but has a SCU count of 0. This means that the sentence was used – in three summaries numbered 14, 44 and 57 – but no SCU is contained in the sentences. Such sentences are negative examples. The third example in Figure 1 was not used in any summary, so it has no annotations. We call it an *unlabelled* sentence. The complete SCU-labelled corpus contains 19247 labelled sentences from a total set of 91658; Table 1 gives the number of positive, negative and unlabelled sentences.

The labelled part of the corpus contains about 40% positive and 60% negative examples. We cannot assume the same distribution in the unlabelled data, so we cannot really be sure how many positive and negative sentences are in the corpus as a whole. One way of estimating this is to graph the likelihood of a sentence containing a SCU against the number of summaries where this SCU appears. We present this graph in Figure 2; it shows the accuracy and the proportion of the sentences from the data set which appeared in a given number of summaries. When a sentence appears in a large number of summaries, it is more likely to

Table 1. Counts of the positive, negative and unlabelled SCU data

Year	Pos	Neg	Unlabelled	% Labelled
2005	1187	1490	16176	14.2%
2006	988	1368	11642	16.8%
2007	937	975	10670	15.2%
2007-A	201	233	1580	21.5%
2007-B	178	285	955	32.7%
2007-C	164	289	912	33.2%
2008-A	1223	1140	8639	21.5%
2008-B	969	1519	7753	24.3%
2009-A	992	2075	7511	30.0%
2009-B	794	2241	6572	31.6%
Total	7633	11615	72410	21.0%

contain a SCU than when it appears in just one or two summaries. The data for sentences which appear in a large number of peer summaries (5 or more) in Figure 2 are quite erratic. This is largely because there are so few such cases; when we consider sentences which appeared in fewer than 5 summaries, we begin to see a trend. If we perform linear regression on these four points, we will expect an accuracy of about 0.22 on sentences which appeared in zero summaries – the unlabelled data. This would mean that, from the 72410 unlabelled examples, about 15930 would have been positively labelled had they appeared in a summary evaluated using the pyramid method. This suggests that our data set currently identifies about one third of all the positive sentences. That is why our SCU-optimal summaries may not actually contain the highest SCU scores possible, but we will find that they do have very high SCU counts.

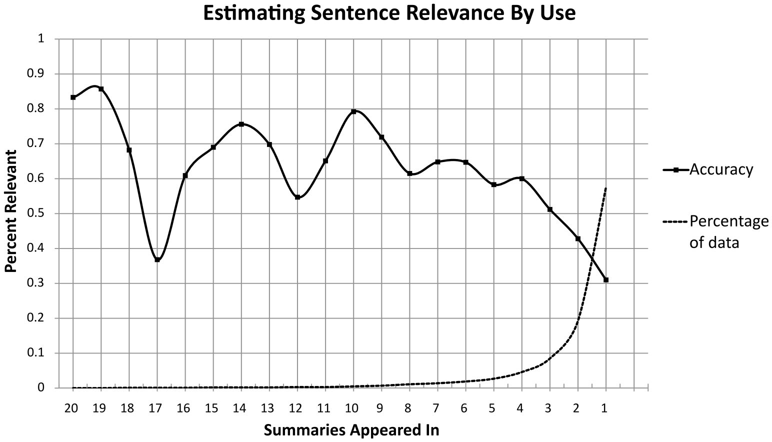


Fig. 2. Frequency of the sentences used by the system summaries

2.1 Previous Users of the SCU-Labelled Corpus

Parts of the SCU-labelled corpus have been used in other research. In [10], the 2005 data are the means for evaluating two sentence-ranking summarization algorithms. In [11], Support Vector Machine is trained on positive and negative sentences from the 2006 DUC data and tested on the 2005 data. The features include sentence position, lexical overlap with the query and others based on text cohesion.

In [2], the SCU-based corpus is used to find a baseline algorithm for update summarization called Sub-Optimal Position Policy (SPP). This is an extension of Optimal Position Policy (OPP) [1] where sentences are selected based on their location in a document. The SCU corpus from 2005-2006 was used for learning SPP, while the 2007 and 2008 data was used for testing.

In [12], the SCU-labelled corpus from 2005 - 2007 is used to identify whether summaries generated automatically tend to be query-focused or query-biased.

A query-focused summary is one built to answer a query, while a query-biased summary is one that selects sentences with as much overlap with the query as possible. It turns out that words found in the query are much more likely to be repeated in machine-generated summaries than in human-made summaries. This is not altogether surprising, because many summarizers determine relevant sentences by measuring lexical overlap with the query.

3 Building a SCU-Optimal Summary

Using the SCU-labelled corpus, we create theoretical upper bounds on how good an extractive summary can possibly be. This is done by selecting sentences in a way to maximize one of two measures:

- combined weight of all SCUs in a summary – maximum SCU weight (MSW);
- combined weight of unique SCUs in a summary – maximum unique SCU weight (MUSW).

Summaries of no more than 100 words are generated, maximizing one of these two criteria. For evaluation we run experiments on the 2008-A and 2009-A TAC data sets, 48 and 44 document sets respectively. Since our aim is to only discover an upper bound for extractive text summarization, not specific to update summaries, we only used the A data sets, not the B sets. We also did not select data from earlier years, because summaries of size 250 became extremely difficult to build for the maximum unique SCU count – see Section 3.2. It should be noted that our SCU-optimal methods will necessarily have higher pyramid scores than any other extractive method tested at TAC. This happens because every sentence they used was labelled in our SCU-labelled corpus and we generate SCU-optimal summaries from this corpus.

3.1 Maximum SCU Weight

Building a summary that maximizes the total SCU score is not difficult. Each sentence has a length and a score where we wish to select the sentences whose combined length is $\leq$ 100 words in a way that maximizes the weight of these sentences. This is in an instance of the well known 0-1 knapsack problem, which can be solved using a dynamic programming algorithm.

We created maximum-weight summaries for all the document sets in the 2008-A and 2009-A data sets. Next we summed all the SCU scores together for each summary. Table 2 shows the results for total SCU weight, total SCU count, unique SCU weight, unique SCU count number of redundant SCUs and total number of sentences. This is the *maximum SCU weight* method (MSW).

3.2 Maximum Unique SCU Weight

For this upper bound we want to maximize the weight of the unique SCUs. Unlike creating summaries which maximize the total SCU weight, we cannot apply

Table 2. Counts for SCU-optimal summaries – MSW & MUSW

	MSW		MUSW	
	2008-A	2009-A	2008-A	2009-A
Total Weight	1430	1932	1178	1464
Total SCUs	518	717	476	593
Unique Weight	932	1104	1132	1298
Unique SCUs	361	454	476	538
Redundant SCUs	157	263	20	55
# of Sentences	213	178	212	167

simple dynamic programming: the score of a sentence depends on every other sentence in the summary. Instead, we built a brute-force algorithm which recursively branches whenever it decides whether to add a sentence to a summary. This algorithm's run time will grow exponentially with the size of the summary generated, but it can still build summaries of up to 100 words in a timely fashion, taking under a minute each on a computer with 2.4 GHz Intel Core 2 Duo processor. (Generating 250-word summaries from the 2005-2007 DUC data becomes prohibitively slow: no summaries built after an hour.) We refer to this method as *maximum unique SCU weight* (MUSW). The results appear in Table 2. This method is the better of the two upper bounds, because our goal should be to maximize unique information. MSW is presented mostly for comparison's sake.

3.3 Sample Summaries

Figure 3 shows sample summaries for the MSW and MUSW methods. The order of the sentences could be changed in a SCU-optimal summary, but this will not change the total SCU score or unique SCU score.

4 Pyramid Evaluation

Naturally, the MWS system will have a higher total SCU weight and the MUSW system will have a higher unique SCU weight as seen in Table 2. We note, however, that the MUSW method still gives redundant SCUs: 4% from 2008-A and 9% from 2009-A. This happens because some SCUs appear in so many sentences that in order to maximize unique SCU weight the summary must repeat some information. It is worth remembering that to maximize unique SCUs does not necessarily mean to eliminate all redundancy. Comparably the MSW summaries have a very high amount of redundancy. On average there are 0.74 (2008-A) and 1.48 (2009-A) redundant SCUs for each sentence. About 30%-37% of SCUs in the MWS summary are redundant.

The modified pyramid score is used as a measurement of recall for how many SCUs were retrieved by a summary. The recall of one of these summaries is the observed SCU weight of the summary, normalized by the average number of SCUs found in the four reference summaries [13].

Maximum SCU weight – MSW
As opposed to the international media hype that surrounded last week's flight, with hundreds of journalists on site to capture the historic moment, Airbus chose to conduct Wednesday's test more discreetly. The A380 will take over from the Boeing 747 as the biggest jet in the skies. So far, Airbus has 154 firm orders for the A380, 27 of them for the freighter version. March 2005: Scheduled first test flight of the plane. June 1994: Airbus begins engineering development of the plane, then known as the A3XX. Assembly of the plane itself is to take place in Toulouse, France.

Maximum unique SCU weight – MUSW
Most A380 traffic will go into just 25 of those airports, Dupont said. March 2005: Scheduled first test flight of the plane. January 23, 2002: Production starts of Airbus A380 components. The A380 will take over from the Boeing 747 as the biggest jet in the skies. Federal Express has ordered 10 of the planes. Assembly of the plane itself is to take place in Toulouse, France. The program, launched in December 2000, banks on a strategy of transporting huge numbers of passengers. International airport standards call for no plane to exceed 80 meters in length and width.

Fig. 3. Summaries generated for the query "Airbus A380 – Describe developments in the production and launch of the Airbus A380" for the 2008 document set D0801-A

$$Modified\_Pyramid\_Score = \frac{\sum_{i=1}^{n} i \times O_i}{\frac{1}{n}\left(\sum_{i=1}^{n} i \times |T_i|\right)}$$

In this formula, O_i is the number of observed SCUs of weight i, T_i represents the set of SCUs of weight i and $|T_i|$ is that set's cardinality. The number of reference summaries, 4 in these data sets, determines n, the maximum weight of a SCU. Table 3 shows the average values of the modified pyramid scores for the MSW and MUSW summaries as well as reference summaries and a random baseline. We have re-implemented the modified pyramid score, so these numbers are not directly comparable to those published by TAC. Jackknifing was used to try and ensure the fairest comparison possible. Note that for this kind of evaluation scores > 1.0 are possible. In fact, since the score is normalized by the combined SCU count from the reference summaries, not the combined SCU scores, the reference summaries will regularly have scores > 1.0. As a lower bound, we also present the results from a system which makes a random selection from the set of positive and negative sentences in the SCU corpus. This baseline is meant to replicate an average extractive summary submitted to TAC.

As can be seen, the scores for SCU-optimal summaries are very high, and scores for MUSW are slightly above those of the reference summaries. Given that we use an estimated one third of the positive sentences from the data set to generate these SCU-optimal summaries, automatic summaries have the potential to contain as much information as human summaries. The random baseline's low score shows how much more room for improvement there is in extractive summarization.

Table 3. Modified pyramid scores for MSW, MUSW, reference summaries and random baseline

	MSW	MUSW	Reference	Random
2008-A	1.06	1.31	1.30	0.39
2009-A	1.23	1.45	1.30	0.29

5 ROUGE

TAC evaluates its systems using two variations on the ROUGE metrics (Section 1.2). ROUGE-2 and ROUGE-SU4 are reported in [14]. Table 4 shows these measures for the MSW and MUSW summaries on the 2008-A and 2009-A data sets. Also in this table are the ranges of scores for the reference and peer summaries for the respective measurements. The recall for these measures for 2008 appears in [14], and the 2009 recall is for now available to the participants. We exclude the scores of some baseline systems also evaluated in TAC 2008 and 2009. For the 2008 data, we also give the upper bounds established in [7]; they are labelled *Max* in the table.

Table 4. ROUGE recall for SCU-optimal and reference/peer summaries

year	measure	MSW	MUSW	Reference Summaries	Peer Summaries	Max
2008-A	ROUGE-2	0.118	0.116	0.108 .. 0.131	0.039 .. 0.111	0.199
	ROUGE-SU4	0.150	0.151	0.140 .. 0.170	0.074 .. 0.143	0.219
2009-A	ROUGE-2	0.105	0.097	0.111 .. 0.149	0.028 .. 0.122	
	ROUGE-SU4	0.140	0.133	0.148 .. 0.184	0.059 .. 0.151	

Both MSW and MUSW fall within the range of the reference summary scores for the 2008 TAC data on ROUGE. For the 2009 data, the ROUGE scores were towards the higher end of the peer summary scores, but they were not quite as good as the reference summaries. ROUGE is really a heuristic method for estimating responsiveness in a summary; it does not directly evaluate content or readability. It certainly does not address redundancy in summaries, considering that the MSW summaries outperformed the MUSW most of the time. That said, these scores show that the SCU-optimal summaries can come quite close to reaching the quality of reference summaries, which would serve to confirm our findings in Section 4.

6 Manual Evaluation

Next we look at the readability and responsiveness of summaries. Unfortunately, the SCU-labelled corpus does not give us any method of determining how readable the summaries we generate are. Readability evaluation takes into account a mixture of grammaticality, non-redundancy, referential clarity, focus, and structure/coherence. Extractive text summarization has both strengths and weaknesses when it comes to readability. Sentences all come from original documents, so they are almost always grammatically correct. The MUSW summaries should do well for non-redundancy, because they contain little repetition, but the MSW summaries have no explicit redundancy checking. Referential clarity can be a problem for any extractive summarization system, if there is no attempt at co-reference resolution. Focus measures how relevant each sentences is to the rest of the summary, while structure and coherence measure whether the sentences are just a heap of information or whether they flow together well; we do not expect the SCU-optimal summaries to perform well on either of these measures, because the flow can even be broken between every two sentences.

Four volunteer annotators helped test the readability and responsiveness of the summaries generated. Each annotator rated 5 kinds of summaries for readability (and its 5 sub-criteria) and responsiveness on a scale 1..10. Two of the summaries were reference summaries generated for TAC, one of the summaries was MSW, one was MUSW and one was a random baseline summary (generated by randomly selecting labelled sentences from the SCU corpus). The probability of selecting a sentence was proportional to the number of peer summaries in which it appeared. This evaluation was done on summaries for 8 different randomly selected document sets (4 each from 2008 and 2009). Table 5 shows the average responsiveness and readability scores for each of these 5 kinds of summaries. We used code from [15] to calculate Krippendorff's α [16] with the interval distance metric to measure inter-annotator agreement. For responsiveness and overall readability we had $\alpha = 0.420$ and $\alpha = 0.459$ respectively.

Regrettably, our sample set is too small to prove conclusively one system's superiority over another, but there are a number of interesting observations which arise from this experiment. Table 5 shows that the human summaries scored better than others on all the measures, with scores between 7.6 and 9.0. The MSUW SCU-optimal summaries and the random baseline had similar performance when comparing overall readability but there was a noticeable difference in responsiveness. The most interesting results come when we look at the sub-criteria of readability. There was little difference between the grammaticality scores. For non-redundancy, the MUSW was not too far below the human summaries, and even the random baseline did not contain much redundancy. In terms of referential clarity, the SCU-optimal summaries and the random baseline differed quite a bit, and generally had scores much lower than the human summaries. The score for this measure may change noticeably by having just one or two additional unclear references. Focus and structure/coherence are measures on which the extractive summaries all performed poorly.

Here is what we can learn from all this: when it comes to readability, non-redundancy is the only sub-measure over which an extractive summarization system can really have influence. A co-reference resolution system might help

Table 5. Average responsiveness and readability scores for each system

Measure	Reference Set 1	Reference Set 2	MSW	MUSW	Random
Responsiveness	7.63	7.88	5.16	6.03	5.38
Readability	8.28	8.22	5.69	6.51	6.46
Grammaticality	*8.75*	*8.53*	*7.81*	*8.31*	*8.34*
Non-Redundancy	*8.65*	*9.00*	*6.69*	*7.91*	*7.53*
Referential Clarity	*8.84*	*8.50*	*6.75*	*6.44*	*7.22*
Focus	*8.09*	*8.28*	*5.34*	*6.00*	*5.75*
Structure/Coherence	*7.91*	*7.81*	*4.38*	*5.31*	*5.06*

improve referential clarity, but in terms of focus and structure/coherence it is difficult to see how these scores can be improved when we restrict ourselves to extractive summarization.

7 Conclusions

We have shown that it is possible to generate summaries which contain content comparable to human summaries, from the perspective of both pyramid and ROUGE evaluation. Despite the high scores on these two measure we found that the responsiveness of the SCU-optimal summaries was not as high as of the reference summaries. When evaluating readability of the summaries, we showed that grammatically of the SCU-optimal summaries is very close to human summaries, and there is the potential to nearly match human summaries in terms of non-redundancy. Other measures based on the coherence of the summaries, however, showed a wide gap between human-written summaries and the SCU-optimal summaries. In future work, we would like to compare our SCU-optimal summaries to other state-of-the-art extractive summaries in terms of readability, responsiveness and modified SCU scores.

Our final conclusion is that it is possible to generate extractive summaries which perform very well on automated measures such as ROUGE, or measures which follow a strict process, as pyramid evaluation does. Ultimately these summaries will not score as well when it comes to manual evaluation, because the readability tends to be low. It may be possible to improve on these SCU-optimal summaries with the addition of co-reference resolution, or perhaps some method of ordering sentences to make them more readable, but from the point of view of content these summaries are as good as can be generated extractively.

Acknowledgments

Our research is supported by the Natural Sciences and Engineering Research Council of Canada and the University of Ottawa. We thank Anna Kazantseva, Richard Keelan and Fazel Keshtkar for help with annotation, and Terry Copeck for maintaining the SCU-labelled corpus. We would also like to thank an anonymous reviewer for the insightful comments.

References

1. Lin, C.Y., Hovy, E.: Identifying topics by position. In: Proc. 5th Conference on Applied Natural Language Processing, Morristown, NJ, USA, pp. 283–290. ACL (1997)
2. Katragadda, R., Pingali, P., Varma, V.: Sentence position revisited: a robust light-weight update summarization 'baseline' algorithm. In: Proc. Third International Workshop on Cross Lingual Information Access, Morristown, NJ, USA, pp. 46–52. ACL (2009)
3. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Proc. ACL workshop on Text Summarization Branches Out, p. 10 (2004)

4. Copeck, T., Szpakowicz, S.: Leveraging pyramids. In: HLT/EMNLP - Document Understanding Workshop, DUC (2005)
5. Genest, P.É., Lapalme, G., Yousfi-Monod, M.: Hextac: the creation of a manual extractive run. In: TAC 2009 Notebook, Gaithersburg, Maryland, USA (November 2009)
6. Nenkova, A., Passonneau, R.J.: Evaluating content selection in summarization: The pyramid method. In: HLT-NAACL, pp. 145–152 (2004)
7. Gillick, D., Favre, B., Hakkani-Tur, D.: The ICSI Summarization System at TAC 2008. In: Proc. of the Text Analysis Conference workshop, Gaithersburg, MD, USA (2008)
8. Hovy, E., Lin, C.Y., Zhou, L., Fukumoto, J.: Automated Summarization Evaluation with Basic Elements. In: Proc. 5th International Conference on Language Resources and Evaluation (LREC), pp. 899–902 (2006)
9. Copeck, T., Inkpen, D., Kazantseva, A., Kennedy, A., Kipp, D., Nastase, V., Szpakowicz, S.: Leveraging duc. In: HLT-NAACL 2006 - Document Understanding Workshop, DUC (2006)
10. Nastase, V., Szpakowicz, S.: A study of two graph algorithms in topic-driven summarization. In: TextGraphs 2006: Proc. TextGraphs: the First Workshop on Graph Based Methods for Natural Language Processing, Morristown, NJ, USA, pp. 29–32. Association for Computational Linguistics (2006)
11. Fuentes, M., Alfonseca, E., Rodríguez, H.: Support vector machines for query-focused summarization trained and evaluated on pyramid data. In: Proc. 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, Morristown, NJ, USA, pp. 57–60. ACL (2007)
12. Katragadda, R., Varma, V.: Query-focused summaries or query-biased summaries? In: Proc. ACL-IJCNLP 2009 Conference Short Papers, Suntec, Singapore, August 2009, pp. 105–108. Association for Computational Linguistics (2009)
13. Passonneau, R.J.: Formal and functional assessment of the pyramid method for summary content evaluation. Natural Language Engineering, 1–25 (2009)
14. Dang, H.T., Owczarzak, K.: Overview of the tac 2008 update summarization task. In: Proc. Text Analysis Conference, pp. 1–16 (2008)
15. Artstein, R., Poesio, M.: Inter-coder agreement for computational linguistics. Comput. Linguist. 34(4), 555–596 (2008)
16. Krippendorff, K.: Content Analysis: An Introduction to Its Methodology. Sage, Thousand Oaks (2004)

MeSH Represented MEDLINE Query Results

Pif Edwards and Vlado Kešelj

Faculty of Computer Science, Dalhousie University
{pedwards,vlado}@cs.dal.ca

Abstract. With a functional Semantic Web on the horizon, practical methods to take advantage of controlled vocabularies should be investigated. Due to the maturity of MEDLINE and UMLS, we have an opportunity, within the medical domain, to begin exploration of ideas which may in the near future be applied to the web in general.

We present an implemented information retrieval system which uses the MeSH terminology as a basis to present MEDLINE results to the user in a hierarchically-browseable way. This demonstrates how to use hierarchical categories to represent a large number of results in manageable way, which is required in many information-gathering tasks, particularly in research of published literature. A pilot evaluation gives evidence of usability of such a system.

Keywords: Natural language processing, information retrieval, user interface design, MeSH, MEDLINE, PubMed.

1 Introduction

1.1 The Research Question

The general question that we explore is: Can an improved user experience be achieved by replacing ranked result lists with hierarchical categorization of search results? Or in more general terms: Is there a better way to browse large result sets?

1.2 Motivation

Information Overload. As of January 22, 2010 MEDLINE contains the citations of 17,843,784 articles and in the first week of January 2010 grew at a rate of 2100/day. [1] This is not abnormal for this collection. In fact, MEDLINE has been growing at a double-exponential rate equaling a compound annual growth of approximately 4.2% over its lifetime. [2] A study from 1985 [3] states that base medical knowledge doubles every 19 years, meaning over a doctor's career medical knowledge increases four times.

With this overall growth in medical knowledge comes an increase in relevant documents for any given medical query. For example, information on HIV is doubling every 22 months. [3] Demner-Fushman *et al.* [4] state that epidemiologists would need to spend over 600 hours a month to read all new articles published

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 75–86, 2010.

in their field. Given these circumstances the question is: When queries match thousands of documents is adequate to simply rank and list these large result sets? That is: How helpful a method of organization is this for the information seeker?

Conventional presentation of large results sets leave lower-ranked results generally ignored and unexplored by users. It is futile to continue to list items with the knowledge that most will never be browsed and may even be detrimental to the user's task. Our browsing models need: to keep pace with this volume of information, to produce better ways to direct users to pertinent articles, to construct ways to help users navigate as they browse.

Exploration of Information Landscape. A list provides few options for exploration. We can skim, periodically sampling results; or plod through every result. Since these results are listed in a linear fashion, when browsing them, we are constrained to visit them more or less linearly. But our understanding is not so constrained. We can think from general to specific. This type of thinking is reflected in a hierarchical browsing model and this model is the basis of our system.

With a hierarchical model, uninteresting results are often categorized together, these subtrees or leaves can be minimized, ignored or deleted. If the user wants to get a feel for the information landscape—for what is 'out there'—a hierarchical structure acts as an informative guide for exploration, each node as a signpost, rather than a long uninformative ranked list where each result is independent of the one before and the one after.

Persistent search. Every time we return to a search engine it is from scratch, that is, our state in the research has changed, enriched by previous queries, but our tool is stateless. This is known in the literature as session-based information retrieval, and this approach is incorporated in the system in order to keep a user's state of research.

2 Related Work

2.1 PubMed

Maintained by the US National Library of Medicine, PubMed (www.pubmed.gov) had 19,591,207 articles as of March 2010. [1] A very active and arguably the most important search engine within the medical domain, PubMed was searched 845 million times in 2007. [1] This tool is the point of experimental comparison for the evaluation of PifMed.

2.2 GOPubMed

As originally conceived GOPubMed (www.gopubmed.com) is very similar to our system. This system uses primarily the GeneOntology (GO) to categorize

PubMed results, but also uses MeSH to accomplish this task. [5] This web-based system began in 2005 and continues to the present. At first results sets were limited to 100, but now stand the result limits tops out at a maximum of 1000 articles. An important difference between GOPubMed and our system is GOPubMed MeSH hierarchy is simplified, truncating all but the top few categories.

2.3 CQA-1.0

This prototype question answering system was developed for clinical evidence-based medicine inquiries, [6] and is the progenitor of our ideas for PifMed. The results of highly structured clinical queries are gathered and presented in hierarchical clusters for users to browse. These clusters do not use the whole MeSH hierarchy and again are simplified in much the same way as GOPubMed. The MeSH terms are used to both generate, label and display clusters.

3 MeSHLINE

The prototype system that we present is called MeSHLINE, taken from a combination of the words MeSH and MEDLINE, which is apt since this project is essentially bringing the MeSH categorization of MEDLINE to the forefront and viewing its use as browsing mechanism with an experimental eye.

3.1 Dependencies

In this section we give a brief description of the services, modules and standards on which MeSHLINE depends.

MeSH. MeSH, which stands for Medical Subject Headings, was developed and is maintained by the National Library of Medicine (NLM) an agency within the National Institute of Health (NIH). First published in 1960 [7], the NLM staff regularly updates this vocabulary, now releasing a new edition of MeSH each year. The 2008 version of MeSH, which is the version used in MeSHLINE, contains 24,767 descriptors.

These descriptors are arranged alphabetically and hierarchically. At the root there are 16 broadly defined main categories, which are further divided into alphabetically-ordered sub-categories. As we follow a path from the root category, down through the 11-level hierarchy, the concepts change from very general, near the root, to very specific, close to the leaves.

To help cope with synonyms, there are an additional 92,000 entry terms to point common terms to descriptor terms. For example, the entry term ‘Vitamin C’ points to the MeSH descriptor ‘Ascorbic Acid’. Only the descriptor terms have been implemented into MeSHLINE.

Unlike many other controlled vocabularies, MeSH was explicitly designed by medical librarians to organize medical document collections, thus ideally suited to this project’s categorization task and why it was our choice among so many others.

MEDLINE. MEDLINE is the largest database searched by PubMed. MeSHLINE depends on one attribute unique to MEDLINE: all articles in MEDLINE have been indexed with MeSH terms by one of the 100 knowledge workers at the NLM. Each MeSH term can optionally be marked as `Major Topic` of a given article by the indexer. A second important feature of MEDLINE is that 79% of all articles in MEDLINE have abstracts. [8] MeSHLINE requires abstracts for tree constructions, but it is important to note MeSHLINE misses out on ∼20% of the articles within MEDLINE due to this dependency. However, often articles without abstracts are generally less relevant to many medical queries. Many are old but some classes of articles such as letters to the editor, corrections and opinion pieces often do not have abstracts regardless of when they were published. These two categories makes up most of the citations excluded from MeSHLINE's search results.

efetch. To query and retrieve results from MEDLINE, MeSHLINE uses a NLM developed tool named *efetch*, which is available in several programming languages from the NLM website [1]. This public domain tool allows MeSHLINE to query MEDLINE (though many other NLM databases accessible with this tool) directly via HTTP and receive results in XML (though a number of other formats are available).

Perl/Tk. MeSHLINE is written completely in Perl and the interface is written in the Perl port of Tk known as Perl/Tk. The version of *efetch* incorporated into MeSHLINE was available in Perl on the NLM website and to the best of our knowledge still is.

3.2 System Information Flow

Figure 1 gives a high-level overview of the information flow of the system during a search session.

To begin a session the user must first, choose a pre-existing index, or enter a query into the entry box and click Query MEDLINE. The existing indices are selectable under the 'Index' tab of the menubar. These large indexes of five and ten thousand articles (e.g. Informatics-10000, ADHD-5000) are presets provided for primarily for testing purposes but also serve as a place holder for future user profile-based functionality.

When the user clicks Query MEDLINE the system ANDs what the user has typed to the user specified preset limits located under the 'MEDLINE Query' tab in the menubar. The minimally required limits are '`AND (hasabstract[text] AND medline[sb])`' and a limit to the number of results returned between 10–10000. Other optional limits include: `Clinical Trial[ptyp]`, `review[ptyp]`, `humans[mesh]`, `free full text[sb]` and `english[lang]`. ANDs and ORs are accepted as valid Boolean search operators in the user specified portion of the query.

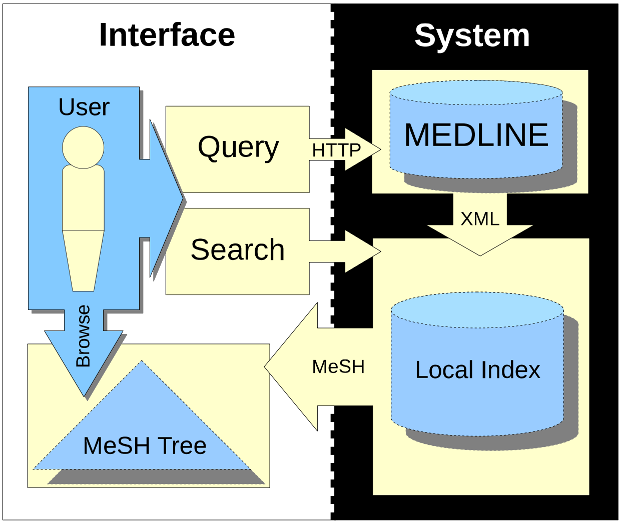


Fig. 1. Shown here is a high-level depiction of the MeSHLINE information flow : 1. user queries MEDLINE; 2. results downloaded in XML via *efetch*; 3. XML indexed locally, for iterative searching; 4. results displayed with a MeSH-based tree structure; 5. MeSH tree is browsed, modified and iteratively searched by user as needed.

The query is sent to MEDLINE via HTTP using the *efetch* utility. These session results are quickly locally indexed so the user may iteratively search this local set with the SEARCH function. Though, as of yet, users have not searched the local set and prefer to re-query MEDLINE should the need arise. However, we still see it as useful as a time saver when dealing with larger result sets which have longer download and processing times.

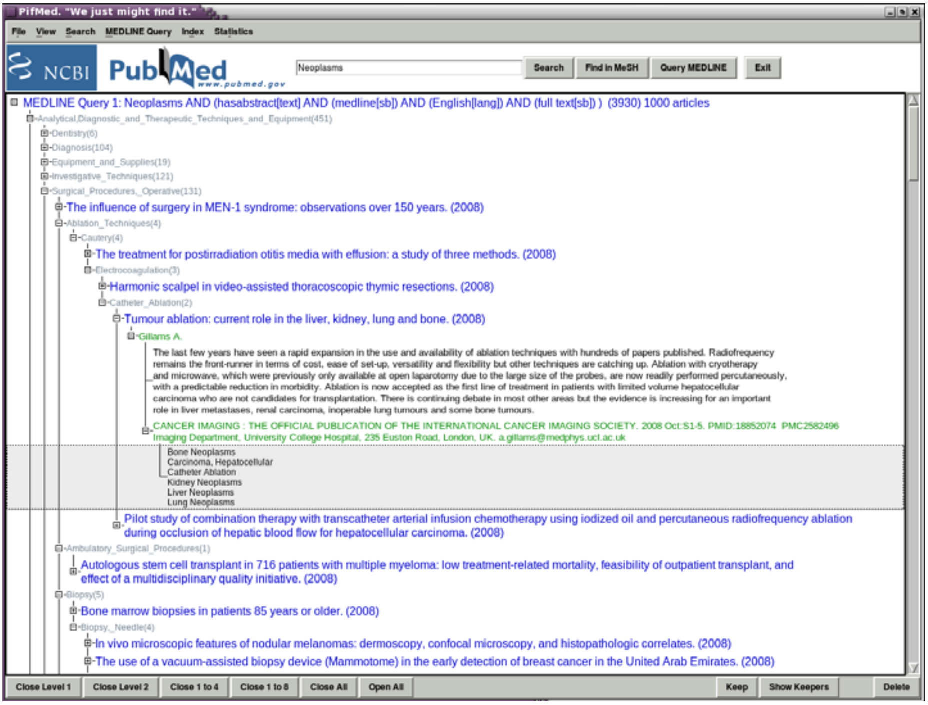
The MEDLINE results, now indexed, are used to build a tree from the MeSH terms attributed to each article. Each MeSH term in an article points to a node on the browsable tree where that article is placed. The tree is custom-built to the query, so only the nodes minimally necessary to reach each article are added to the tree.

Indexing. The local index is based on the Boolean model. The words are stemmed, the stop words are removed, as well as the low-frequency terms. We chose the Boolean model for three major reasons; first, since we index while the users waits, we needed it to be fast; second, the ranking is not of great importance as we are categorizing the results; and finally, most categories return less than five articles and often only one, thus the overhead of using the vector space model for indexing and ranking would not be justified. As a final point on the topic, presently if more than one article is mapped to the same node the articles are presented in the order they are received from MEDLINE, therefore we benefit from the MEDLINE ranking system.

3.3 Interface Design

The interface is designed to be simple and familiar, intentionally focusing the attention of the user on the functionality instead of a logo or colour scheme. In pursuit of familiarity, the choice of colour for the text and graphics is inspired by Google's choice of colors, as well as the PubMed banner. No unnecessary graphics or icons are added to sway users preference or to add confounding factors to influence test results.

Navigation. This method of presentation is familiar to most computer users due to its use in file hierarchies. Presented with the system no users needed any direction on its use. However, early use of the system identified two clear



Neoplasms

```
MEDLINE Query 1: Neoplasms AND (hasabstract[text] AND (medline[sb]) AND (English[lang]) AND (full text[sb]) )  (3930) 1000 articles
  -Analytical,Diagnostic_and_Therapeutic_Techniques_and_Equipment(451)
    -Dentistry(6)
    -Diagnosis(104)
    -Equipment_and_Supplies(19)
    -Investigative_Techniques(121)
    -Surgical_Procedures,_Operative(131)
      -The influence of surgery in MEN-1 syndrome: observations over 150 years. (2008)
      -Ablation_Techniques(4)
        -Cautery(4)
          -The treatment for postirradiation otitis media with effusion: a study of three methods. (2008)
          -Electrocoagulation(3)
            -Harmonic scalpel in video-assisted thoracoscopic thymic resections. (2008)
            -Catheter_Ablation(2)
              -Tumour ablation: current role in the liver, kidney, lung and bone. (2008)
                -Gillams A.
                  The last few years have seen a rapid expansion in the use and availability of ablation techniques with hundreds of papers published. Radiofrequency
                  remains the front-runner in terms of cost, ease of set-up, versatility and flexibility but other techniques are catching up. Ablation with cryotherapy
                  and microwave, which were previously only available at open laparotomy due to the large size of the probes, are now readily performed percutaneously,
                  with a predictable reduction in morbidity. Ablation is now accepted as the first line of treatment in patients with limited volume hepatocellular
                  carcinoma who are not candidates for transplantation. There is continuing debate in most other areas but the evidence is increasing for an important
                  role in liver metastases, renal carcinoma, inoperable lung tumours and some bone tumours.
                  -CANCER IMAGING : THE OFFICIAL PUBLICATION OF THE INTERNATIONAL CANCER IMAGING SOCIETY. 2008 Oct:S1-5. PMID:18852074  PMC2582496
                   Imaging Department, University College Hospital, 235 Euston Road, London, UK. a.gillams@medphys.ucl.ac.uk
                    Bone Neoplasms
                    Carcinoma, Hepatocellular
                    Catheter Ablation
                    Kidney Neoplasms
                    Liver Neoplasms
                    Lung Neoplasms
              -Pilot study of combination therapy with transcatheter arterial infusion chemotherapy using iodized oil and percutaneous radiofrequency ablation
               during occlusion of hepatic blood flow for hepatocellular carcinoma. (2008)
      -Ambulatory_Surgical_Procedures(1)
        -Autologous stem cell transplant in 716 patients with multiple myeloma: low treatment-related mortality, feasibility of outpatient transplant, and
         effect of a multidisciplinary quality initiative. (2008)
      -Biopsy(5)
        -Bone marrow biopsies in patients 85 years or older. (2008)
        -Biopsy,_Needle(4)
          -In vivo microscopic features of nodular melanomas: dermoscopy, confocal microscopy, and histopathologic correlates. (2008)
          -The use of a vacuum-assisted biopsy device (Mammotome) in the early detection of breast cancer in the United Arab Emirates. (2008)
```

Fig. 2. A screenshot of the MeSHLINE user interface. The root of the tree is the MEDLINE query in full. In brackets following the root is the total number of title nodes in the tree, following the total number of title node is the total number of articles. These are not equal because most articles have more than one MeSH term attributed to them. The grey text are MeSH terms, organized into a hierarchy. The blue nodes within the tree are title nodes. Inside a title node is a green author node. Inside the author node is an abstract in black, followed by the bibliographical information. The child of the green bibliographical node is a list of the MeSH keywords attributed to a given article.

problems frustrating browsing. We identified these as: ‘lost in MeSH’ and ‘too many clicks’. We will discuss each of these presently.

Closed and Open Tree. There are three obvious ideas on how to present the starting state of the MeSH tree: all nodes closed, all nodes open, or partly open/closed.

‘All closed’ has two drawbacks. First, the user quickly become frustrated with opening nodes 5—10 levels deep before seeing any search results. Second, this leads the user to forget the query and focus on the MeSH terminology.

‘All open’ is better maybe seen as better, but has two drawbacks of its own. The first problem can be best described in the words of users: ‘Too much clicking’, i.e. a lot of time is spent closing nodes. This is less frustrating since the shortcut of closing nodes up the hierarchy—closer to the root—on long paths saves clicking. The second drawback is that it defeats the purpose of the categorization, since an ‘all open’ tree reads like a long list. When the results are presented thus, the user begins at the top and scrolls a great deal, which was precisely the browse behavior we were trying to avoid, the categorization (i.e. MeSH terminology) falls to the background and is ignored almost completely.

These states maybe selected in the view menu as the default state if they are preferred. Also in the application window, buttons to set the tree into the Open All, Close All states are located along the bottom, below the results (see Figure 2).

3-Click Tree. Our solution was to decide how many clicks were too many clicks; when did the user begin to get frustrated? To us, the answer was four. We designed a tree state which requires only three clicks to get the information you need to decide if an article is interesting or not.

Once a search is completed, the major MeSH headings are shown. These categories are both sufficiently distinct from each other and broad enough in scope to indicate to the user what each likely contain and what they likely do not. When one of these nodes are selected and opened (click 1) all sub-categories with these main headings are also closed, forcing the user to further focus their expectations on what results are possibly contained within. Once one of these nodes is selected and opened (click 2) all paths within that subtree are fully opened down to the article title node. When the user identifies a title of interest, opening this title node (click 3) will reveal the authors, abstract, bibliographical information and the MeSH terms this article has been indexed with, in an open state.

This method seems to have the best of both worlds and since its implementation no user has mentioned “too much clicking” commentary.

Find in MeSH. The Find in MeSH function is put in place to address the problem of the complexity of the MeSH terminology. A user may know of a category but unclear on the path from the root to that known category. If the user enters the category name into the entry box and clicks the Find in MeSH button, MeSHLINE will open all nodes between it and the root, then center the screen over the found category. If the category exists in MeSH but not in the tree generated by the search results, all nodes between the root and where the node should be are opened. If the category does not exist the Find in MeSH function leaves the tree unchanged.

This function is particularly useful when the user finds an article of interest and while browsing the other MeSH keywords used to categorize this article spot a category of interest. The Find in MeSH feature can quickly center the screen on the contents of these newly identified categories for browsing. At the moment no MeSH entry terms are implemented in this feature, only core MeSH terms. The inclusion of these entry terms will be the concern of future work and will likely increase the usefulness of this feature.

Once this feature was implemented the question of the type "Where is this (pointing to a category in a list of terms on-screen) category" disappeared.

Keepers. When the user finds an article they like, they may click the Keep button on the bottom right of the screen (see Figure 2). This collects the author, title, bibliographical information and abstract into a text file for the user to print, email or use however they see fit. Each time they click Keep, the selected article is added to the bottom of the file. This file maybe viewed by clicking the Show Keepers button. Each session opens a new 'keepers' file.

Delete. Articles, branches, sub-trees even entire queries may be deleted by selecting any of the above and clicking the Delete button. To clear clutter and prune away material the user knows to be of no interest (or relevance) helps many users to focus on paths which show more promise. As you can see from Figure 2 the Delete button is in the lower right hand side.

4 Experiments

4.1 Usability Study

First, the user was given a brief 5–6 minute power-point presentation which demonstrated the system. Then the user was asked to use the MeSHLINE system to find an article of interest. The user was made aware before they began that they would later use the same query on PubMed to find the same article they found on MeSHLINE. This process was repeated once. After these searches were completed the user was asked to fill out a short questionnaire,

Qualitative and quantitative data was collected during the user study. For quantitative data collection MeSHLINE reported all user actions to a log. This data was analyzed to measure how long it took to complete queries and where the user (and system) spent most of their time.

The qualitative data was collected primarily through a questionnaire which immediately followed the system and was administered through MeSHLINE itself.

Questionnaire. A three-part questionnaire was issued after the test was completed. The first part was meant to determine their similarity to the intended user. These questions ask them to rate their familiarity with: computers, search, PubMed, MeSH and medicine. An ideal candidate would score 6–7 out of seven in each of these categories.

The second part of the questionnaire was meant to judge the usability of the system. 11 questions in total, each question asks the user to rate MeSHLINE in terms of an aspect of usability (Effectiveness, Efficiency, Easy-to-Learn, Error Tolerance, Engagement) [9] and then directly compare MeSHLINE to PubMed on this aspect. The final question in this section asked for an overall rating of MeSHLINE and an overall comparison of the two systems. All ratings were from 1 to 7.

The third-part of the questionnaire was in the form of short answer and asked the user to state favorite and least favorite feature of MeSHLINE and finally left room for comments, suggestions and to recommend improvements.

System and Hardware. The test system was Desktop PC with a dual 3.2 GHz Intel Pentium 4, 2GB of RAM, 100GB HD, running Linux-based Fedora 8 OS and a 17” LCD screen.

5 Results

5.1 Participants

The intended users of this system are regular users of PubMed. Therefore we primarily focused on the needs of medical researchers, medical students and medical librarians. Our ideal user would score a 6–7 out of 7 in each category in Table 1. According to the results of Table 1 you can see that knowledge of MeSH is the low point, but overall, our pilot group is satisfactorily close to our target user. We expect familiarity with MeSH to be the low point for the majority of users, these results reflect this assumption.

5.2 Quantitative Results

The quantitative data in Table 3 presented some unexpected results. The fact that MeSHLINE was ∼80% faster than PubMed was a surprise. To explain the difference my first instinct is to find fault with the study design. It may the case that hidden in the order of use is a confounding factor that is affecting the outcome. Perhaps the fact that the user found the paper with one tool means that tool lead them to that result, a result that would not so easily be repeated using the other tool. This may be the case, however, each of these users had a very specific paper they were looking for in at least one of their two queries, which throws doubt on this as an explanation.

To remove possible influence of confounding factors in this regard, the order of the systems will be randomized for any future studies.

5.3 Qualitative Results

On the whole, the users in our pilot study found MeSHLINE preferable to PubMed, though not resoundingly so. Users gave MeSHLINE the edge in terms of Easy-to-Learn (5.1), Effectiveness (4.8) and Efficiency (4.8) and lower grades for Error Tolerance (4.4) and Engagement (4.1). The overall usability averages to 4.7 out of 7. See Table 4 for the full results.

Table 1. In part one of our questionnaire each user is asked to rate their familiarity in each of these areas. Our ideal user would score a 6–7 in each of these categories.

User			**Statistics**				
1	2	3	Min	Median	Max	Average	Familiarity with...
7	7	7	7	7	7	7	computers
7	7	6	6	7	7	6.7	Computer search
5	4	2	2	4	5	3.7	MeSH
5	6	6	5	6	6	5.7	PubMed
4	7	7	4	7	7	6	Medicine
			4.8	6.2	6.4	5.8	Average Participant

Table 2. Shows MEDLINE search results for ‘neoplasms’ returning limits of 100, 1000, and 10000 of 716517 possible results. 100 is quite fast, 1000 is still acceptable, but ∼12.5 minutes to manage 10000 results is much to slow for all but the most dedicated user. The default limit for results is 500 (average 16–22s total wait time) for this reason.

Result Limit	MEDLINE Download	Index XML	Build Tree	Display Tree	Total
100	1s	1s	1s	1s	4s
500	6s	3s	6s	3s	18s
1000	10s	6s	12s	11s	39s
10000	138s	63s	118s	425s	744s

6 Limitations

Very large query result sets (10000+) are too slow for general use as shown in Table 2.

To what degree previous knowledge of MeSH maybe necessary to best navigate this system is still unknown. We believe use promotes understanding of the MeSH terminology, but willingness to learn MeSH, what rate users learn MeSH and the what depth of knowledge of MeSH is needed to best make use of this categorization are all unknowns. At the moment users did not seem uncomfortable with the system, but comfort levels have not been measured and we anticipate this may be an impediment to adoption as the system of choice.

Table 3. This table shows the results of our quantitative data collection. The quantitative data from user one was unfortunately corrupted and not usable for analysis.

User2	User3	Query
156	103	Q1: MeSHLINE (secs)
245	212	Q1: PubMed (secs)
89	**109**	**Q1: Difference (secs)**
57.05%	**105.83%**	**MeSHLINE Faster by X%**
213	96	Q2: MeSHLINE (secs)
341	220	Q2: PubMed (secs)
128	**124**	**Q2: Difference (secs)**
60.09%	**129.17%**	**MeSHLINE Faster by X%**
108.5	116.5	Average Difference in seconds
58.57%	117.50%	Average Result
	142	Total Average search time: MeSHLINE
	254.5	Total Average search time: PubMed
	112.2	Total Average difference
	79.23%	Total Average Result

Table 4. Shown here is the final analysis of the qualitative results. Averages of each of the measures of usability along the right are used to calculate the final score, which is on the bottom line.

min	median	max	average	measure
2	5	7	4.8	Effective
3	5	7	4.8	Efficient
3	4	6	4.1	Engaging
2	4	7	4.4	Error Tolerant
4	5	7	5.1	Easy to Learn
2	**5**	**7**	**4.7**	**Total Average Usability**

7 Future Work

One benefit of this model lies in the persistence of state of the users browsing. We would like to extend state persistence past one session by implementing user profiles. These user profiles would log past queries and maintain previous results trees in the state they were in when they were last modified. This would be particularly useful for an information monitoring task, where search trees could be re-queried, new entries added automatically marked as ‘unseen’ or ‘new’ and this marking reflected on parent nodes up the hierarchy.

User profiles and persistence make tree pruning a more worthwhile effort. Efforts deleting branches and articles would not go wasted. These ‘pruned’ search results may be valuable enough to share with other users, or be used as the basis for automatic query generation for new searches.

Finally, we would like to organize future user studies with specific information needs in mind, to measure the systems usability for specific tasks where high recall is important and others where browsing is central to the task.

8 Conclusion

In conclusion, we have shown knowledge-based methods can be used to speed browsing of large result sets. Our pilot study indicates users may prefer this browsing model to the existing conventional model. The encouraging results of our pilot study, mixed with the small sample size, signal a larger user-study is needed to reinforce these findings.

References

1. US National Library of Medicine: Data news and update information, http://www.nlm.nih.gov/bsd/revup/revup_pub.html (last access 2010)
2. Hunter, L., Cohen, K.B.: Biomedical language processing: What's beyond pubmed? Molecular Cell 21(5), 589–594 (2006)
3. Smith, R.: What clinical information do doctors need? British Medical Journal 313(7064), 1062–1068 (1996)
4. Demner-Fushman, D., Hauser, S., Thoma, G.: The role of title, metadata and abstract in identifying clinically relevant journal articles. In: AMIA Annual Symposium Proceedings, vol. 2005, pp. 191–195. American Medical Informatics Association (2005)
5. Doms, A., Schroeder, M.: GoPubMed: exploring PubMed with the gene ontology. Nucleic Acids Research 33(Web Server Issue), W783–W786 (2005)
6. Lin, J., Demner-Fushman, D.: Semantic Clustering of Answers to Clinical Questions. In: AMIA Annual Symposium Proceedings, vol. 2007, pp. 458–462. American Medical Informatics Association (2007)
7. US National Library of Medicine: Fact sheet medical subject headings MeSH, http://www.nlm.nih.gov/pubs/factsheets/mesh.html (last access 2010)
8. US National Library of Medicine: Fact sheet MEDLINE, http://www.nlm.nih.gov/pubs/factsheets/medline.html (last access 2010)
9. Stone, D., Jarrett, C.: User interface design and evaluation. Morgan Kaufmann, San Francisco (2005)

Argumentation-Based Reasoning with Inconsistent Knowledge Bases*

Xiaowang Zhang[1,2], Zhihu Zhang[1], Dai Xu[1], and Zuoquan Lin[1]

[1] School of Mathematical Science, Peking University, Bejing 100871, China
[2] School of Mathematical Science, Anhui University, Hefei 230039, China
{zxw,zhzhang,xudai,lzq}@is.pku.edu.cn

Abstract. In this paper, we present an argumentation-based approach to dealing with inconsistency occurring in knowledge bases. We investigate several important logical properties of such an argumentation-based entailment relation and show its promising advantages in paraconsistent reasoning for inconsistent knowledge bases. Moreover, two basic inference problems, namely, satisfiability of concepts and query entailment, are discussed under our semantics. We provide a workable example in order to show the justifiability of the argumentation-based semantics.

1 Introduction

Inconsistency handling is an important problem in Artificial Intelligence (AI). In a real world, many reasons potentially bring inconsistent knowledge, such as modeling errors, migration from other formalisms, merging ontologies and ontology evolution. It is unrealistic to expect that real knowledge bases (KBs) and databases are always prefect and logically consistent [2]. Unfortunately, if a knowledge base is inconsistent then inference can be described as exploding, or trivialized. That is, anything, no matter whether it is meaningful, can follow from an inconsistent set of assumptions. With KBs being widely applied in many fields such as Semantic Web, coping with inconsistent KBs becomes more essential in AI.

There are many existing approaches to handling inconsistency occurring in KBs. The main idea of those approaches is to avoid the explosive entailment, i.e., making the inference rule *ex falso quodlibet* invalid to obtain some meaningful information from an inconsistent KB. They are based on two fundamentally different standpoints. One is based on the assumption that inconsistencies indicate erroneous data which are to be repaired in order to obtain consistent KBs (e.g., by selecting consistent subsets for the reasoning process) [3,4,5]. Unfortunately, based on this view, we may lose useful knowledge so that we might not obtain more reasonable conclusions from inconsistent KBs. The other, so-called *paraconsistent approach*, is based on the assumption that inconsistencies are treated as a natural phenomenon in realistic data which are to be tolerated by applying a non-standard reasoning method to obtain meaningful answers (e.g., by borrowing multi-valued semantics) [6,7,8,9,10,11]. However, the inference power of

* The primary version of this paper firstly presented in [1].

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 87–99, 2010.

them is rather weak because some inference rules are not available. For instance, four-valued DLs [6,7,8] and three-valued DLs [11] do not hold three basic inference rules (or resolution rules) such as *modus ponens*, *disjunctive syllogism* and *modus tollens*. Quasi-classical DLs [10] hold resolution rules while tautologies cannot be driven by quasi-classical entailment from any KB. Though two views are incompatible, they are limited because inconsistencies are inadequately treated and are not further analyzed in detail but either isolated even being discarded or ignored by evaluating a value "B"(i.e., both *true* and *false*).

Dung [12] introduced an argumentation framework to provide a proof-theoretic semantics for non-monotonic logic. Based on Dung's argumentation framework, Besnard and Hunter [13] presented a new argumentation framework to handle inconsistency in propositional logic. The advantage of Besnard and Hunter's argumentation framework is that the arguing process of a finite set of arguments always terminates. In recent years, there are some argumentation-based approaches to resolving some other important tasks in AI, such as general ontologies engineering[14], ontologies reasoning based on defeasible logic programs [15], and querying over multiple ontologies [16].

In this paper, based on Besnard and Hunter's argumentation framework, we provide an argumentation-based approach to handle inconsistency occurring in DL $\mathcal{ALC}$ KBs (KBs for short).We select DL $\mathcal{ALC}$ because $\mathcal{ALC}$ is a basic member of DL family. The main innovations and contributions of this paper can be summarized as follows:

- We define arguments of $\mathcal{ALC}$ KBs. An argument is a pair of a minimal consistent set Φ of axioms and an axiom ϕ where the set Φ entails the axiom ϕ. We present the relationship between two arguments called undercut to characterize the conflict between them. A conservative relation over undercuts, as a preferential order, is introduced to characterize the capability of arguing an axiom in a KB. We introduce canonical undercuts by collecting the maximal conservative undercuts in order to represent all undercuts of an argument.
- We develop an argumentation framework for $\mathcal{ALC}$. Firstly, we introduce argument trees to capture the procedure of arguing an axiom (or a query) among a given KB via the dialogue mechanism. Then, based on argument trees, we present an argumentation framework to demonstrate the relationship between a KB and an axiom. Finally, we use categorisers of an axiom to mark whether other axioms from a given KB are for/against the axiom.
- We propose an argumentative entailment relationship from an $\mathcal{ALC}$ KB to an $\mathcal{ALC}$ axiom based on binary argumentation for $\mathcal{ALC}$. Two basic inference problems, namely, satisfiability of concepts and query entailment, are discussed under the argumentative semantics.

This paper is structured as follows. Section 2 reviews briefly the syntax and semantics of $\mathcal{ALC}$. Section 3 defines arguments in $\mathcal{ALC}$. Section 4 proposes an argumentative semantics for $\mathcal{ALC}$. Section 5 discusses reasoning with inconsistent KBs and repairing KB based on argumentation. In the last section, we summarize our work and discuss the future work. Due to space limitation, all proofs are presented in an online technical report at http://www.is.pku.edu.cn/~zxw/publication/TR-ADL.pdf.

2 Description Logic $\mathcal{ALC}$

Description logics (DLs) is a well-known family of knowledge representation formalisms. For more comprehensive background reasoning, we refer the reader to Chapter 2 of the DL Handbook [17].

In this paper, we consider $\mathcal{ALC}$ which is a simple yet relatively expressive DL. $\mathcal{AL}$ is the abbreviation of attributive language and $\mathcal{C}$ denotes "*complement*". Let N_C and N_R be pairwise disjoint and countably infinite sets of *concept names* and *role names* respectively. Let N_I be an infinite set of *individual names*. In this paper, we use the letters A and B for concept names, the letters $R, S, \ldots$ for role names, the letters $C, D, \ldots$ for concepts, the letters $a, b, \ldots$ for individuals. $\top$ and $\bot$ denote the *universal concept* and the *bottom concept* respectively.

With the symbols $\top$ and $\bot$, we furthermore denote the top concept and the bottom concept, respectively. Complex concepts in $\mathcal{ALC}$ can be formed from these inductively as follows:

- $\top$, $\bot$, each concept name is a concept;
- if C and D are concepts, then $C \sqcap D$, $C \sqcup D$, and $\neg C$ are concepts;
- if C is a concept and R is a role, $\forall R.C$ and $\exists R.C$ are concepts.

An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a non-empty domain $\Delta^{\mathcal{I}}$ and a mapping $\cdot^{\mathcal{I}}$ which maps every concept to a subset of $\Delta^{\mathcal{I}}$ and every role to a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. For all concepts C, D and a role$R$, satisfies conditions as follows: $\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$; $\bot^{\mathcal{I}} = \emptyset$; $(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$; $(C_1 \sqcap C_2)^{\mathcal{I}} = C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$; $(C_1 \sqcup C_2)^{\mathcal{I}} = C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}}$; $(\exists R.C)^{\mathcal{I}} = \{x \mid \exists y, (x, y) \in R^{\mathcal{I}}$ and $y \in C^{\mathcal{I}}\}$; and $(\forall R.C)^{\mathcal{I}} = \{x \mid \forall y, (x, y) \in R^{\mathcal{I}}$ implies $y \in C^{\mathcal{I}}\}$.

A *general concept inclusion axiom (GCI)* or *a terminology* is an inclusion statement of the form $C \sqsubseteq D$. It is the statement about how concepts are related to each other. A finite set of GCIs is called a *TBox*. We can also formulate statements about individuals. A *concept* (*role*) *assertion axiom* has the form $C(a)$ ($R(a, b)$). An *ABox* contains a finite set of concept axioms and role axioms. In the ABox, one describes a specific state of affairs of an application domain in terms of concept and roles.

To give a semantics to ABoxes, we need to extend interpretations to individual names. For each individual name a, $\cdot^{\mathcal{I}}$ maps it to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$.

An interpretation $\mathcal{I}$ satisfies a concept axiom $C(a)$ if and only if $a^{\mathcal{I}} \in C^{\mathcal{I}}$; $\mathcal{I}$ satisfies a role axiom $R(a, b)$ if and only if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$. A KB $\mathcal{K}$ consists of a TBox and an ABox, i.e., it is a set of GCIs and assertion axioms. An interpretation $\mathcal{I}$ is a *model* of a DL (TBox or ABox) axiom if and only if it satisfies this axiom; and it is a model of an KB $\mathcal{K}$ if it satisfies every axiom in $\mathcal{K}$. An ABox $\mathcal{A}$ is consistent w.r.t. a TBox $\mathcal{T}$ if and only if there exists a common model of $\mathcal{T}$ and $\mathcal{A}$.

Given a KB $\mathcal{K}$ and a DL axiom ϕ, we say $\mathcal{K}$ *entails* ϕ, denoted $\mathcal{K} \models \phi$, if and only if every model of $\mathcal{K}$ is a model of ϕ. A concept C is *satisfiable* if and only if there exists an individual a such that the ABox $\{C(a)\}$ is consistent; and *unsatisfiable* otherwise. A concept C is *satisfiable* w.r.t. a TBox $\mathcal{T}$ if and only if there exists a model $\mathcal{I}$ of $\mathcal{T}$ such that $C^{\mathcal{I}} \neq \emptyset$ is consistent; and *unsatisfiable* otherwise.

3 Arguments in Description Logic $\mathcal{ALC}$

This section starts to build up a formal theory of argumentation-based reasoning in $\mathcal{ALC}$. To do this, we will follow the structure of the argumentation process which can be divided into the following three steps:

1. Defining the structure of the arguments and the interactions between arguments.
2. Valuating the arguments defined in the previous step based on the interactions between arguments.
3. Selecting the justified conclusions.

In this section, we mainly focus on the problems in the first step and restate main definitions and properties presented in [1]. We first define the underlying structure of arguments in $\mathcal{ALC}$. Next, we give the definition of different interactions between both arguments. Finally, we show some basic properties of these interactions.

Similar to logical arguments [13] built based on propositional logic or first-order logic, we define the underlying structure of arguments in $\mathcal{ALC}$ as follows:

Definition 1. *Given a KB $\mathcal{K}$, let Φ be a set of axioms and ϕ an axiom in $\mathcal{ALC}$. An* argument *is a pair $\langle \Phi, \phi \rangle$ such that*

(1) *Φ is classically consistent;*
(2) *$\Phi \models \phi$;*
(3) *Φ is a minimal subset of $\mathcal{K}$ satisfying* (2).

If $\mathbf{A} = \langle \Phi, \phi \rangle$ is an argument*, we say that $\mathbf{A}$ is an argument for ϕ and we also say that Φ is a support for ϕ. In addition, we call Φ the* support *of $\mathbf{A}$ and ϕ the* consequent *of $\mathbf{A}$. Sup$(\mathbf{A}) = \Phi$ and Con$(\mathbf{A}) = \phi$.*

Example 1. *Let $\mathcal{K} = (\{Penguin \sqsubseteq Bird, Bird \sqsubseteq Fly, Penguin \sqsubseteq \neg Fly, Swallow \sqsubseteq Bird, Swallow \sqsubseteq \forall HasFood.\neg Fish, Penguin \sqsubseteq \exists HasFood.Fish, Swallow \sqsubseteq \neg Penguin\}, \{Penguin(Tweety), \neg Fly(Tweety)\})$ be an $\mathcal{ALC}$ KB. From $\mathcal{K}$, we know that penguins are birds; birds can fly; penguins cannot fly; swallows are birds; swallows do not eat fishes; penguins eat fish; swallows are not penguins and tweety is a penguin and tweety cannot fly.*
Some arguments are as follows:

$$
\begin{aligned}
\mathbf{A}_1 &= \langle\{Penguin(Tweety), Penguin \sqsubseteq Bird, Bird \sqsubseteq Fly\}, Fly(Tweety)\rangle\\
\mathbf{A}_2 &= \langle\{\neg Fly(Tweety), Penguin \sqsubseteq Bird, Bird \sqsubseteq Fly\}, \neg Penguin(Tweety)\rangle\\
\mathbf{A}_3 &= \langle\{\neg Fly(Tweety), Penguin(Tweety), Penguin \sqsubseteq Bird\}, Bird \sqcap \neg Fly(Tweety)\rangle\\
\mathbf{A}_4 &= \langle\{\neg Fly(Tweety), Penguin(Tweety), Bird \sqsubseteq Fly\}, Penguin \sqcap \neg Bird(Tweety)\rangle\\
\mathbf{A}_5 &= \langle\{\neg Fly(Tweety)\}, \neg Fly(Tweety)\rangle\\
\mathbf{A}_6 &= \langle\{Penguin(Tweety), Penguin \sqsubseteq \neg Fly\}, \neg Fly(Tweety)\rangle\\
\mathbf{A}_7 &= \langle\{Penguin(Tweety), Penguin \sqsubseteq Bird, Bird \sqsubseteq Fly\}, Penguin \sqcap Fly(Tweety)\rangle\\
\mathbf{A}_8 &= \langle\{Penguin(Tweety), Penguin \sqsubseteq \exists HasFood.Fish\}, \exists HasFood.Fish(Tweety)\rangle\\
\mathbf{A}_9 &= \langle\{Penguin(Tweety), Swallow \sqsubseteq \neg Penguin\}, \neg Swallow(Tweety)\rangle\\
\mathbf{A}_{10} &= \langle\{\neg Fly(Tweety), Bird \sqsubseteq Fly\}, \neg Bird(Tweety)\rangle
\end{aligned}
$$

Arguments are actually not independent. In contrast, they have relations with each other. Now, we are ready to give two kinds of relations "*encompassment*" relation and "*defeat*" relation on arguments in $\mathcal{ALC}$. We first define how some arguments encompass others.

Definition 2. *An argument $\langle \Phi, \phi \rangle$ is* more conservative *than an argument $\langle \Psi, \psi \rangle$ iff $\Phi \subseteq \Psi$ and $\psi \models \phi$, written by $\langle \Psi, \psi \rangle \preceq_c \langle \Phi, \phi \rangle$. If $\phi \not\models \psi$ then $\langle \Phi, \phi \rangle$ is called* strictly more conservative *than an argument $\langle \Psi, \psi \rangle$, written by $\langle \Psi, \psi \rangle \prec_c \langle \Phi, \phi \rangle$.*

In short, we define a pre-order relation on arguments using conservative relation. For instance, the argument $\mathbf{A}_4 \prec_c \mathbf{A}_{10}$ in Example 1.

It is clear to conclude that $\mathbf{A} \preceq_c \langle \emptyset, \top \rangle$ for any argument $\mathbf{A}$ because the empty KB is consistent. In this paper, we mainly consider non-empty finite KBs.

Next, we are going to define the interactions between arguments that are used to express the defeat relation on arguments. However, it is not easy to build such interactions directly based on the syntax of $\mathcal{ALC}$ like logical arguments [13] built based on propositional logic or first-order logic because the constructors $\sqcap, \sqcup$ in the syntax of $\mathcal{ALC}$ are not logical connectives $\wedge, \vee$. Thus, we need to introduce two logical connectives "*conjunction*" ($\wedge$) and "*disjunction*" ($\vee$) defined as follows. Let $\mathcal{K}$ be a KB and ϕ, ψ axioms in $\mathcal{ALC}$. Then

- $\mathcal{K} \models \phi \wedge \psi$ iff $\mathcal{K} \models \phi$ and $\mathcal{K} \models \psi$;
- $\mathcal{K} \models \phi \vee \psi$ iff $\mathcal{K} \models \phi$ or $\mathcal{K} \models \psi$.

We call $\phi \wedge \psi$ (or $\phi \vee \psi$) *conjunction* of axioms (or *disjunction* of axioms).

Compared with concept conjunction $\sqcap$ and concept disjunction $\sqcup$ in the syntax of $\mathcal{ALC}$ defining on concepts, the above new connectives $\wedge$ and $\vee$ are two logical connectives effecting on axioms.

Besides these two logical connectives, we further need to define a negation connective to represent the logically negative information, which will be used to express the defeat relation among arguments.

Definition 3. *Let $\mathcal{K}$ be a KB, C, D concepts, a an individual and ϕ, ψ two axioms. A new negative constructor called* quasi-negation *(denoted by $\sim$) on axioms or conjunction (disjunction) of axioms is defined as follows:*

- *$\mathcal{K} \models \sim C(a)$ iff $\mathcal{K} \models \neg C(a)$;*
- *$\mathcal{K} \models \sim C \sqsubseteq D$ iff $\mathcal{K} \models C \sqcap \neg D(\iota)$ for some individual ι in $\mathcal{K}$;*
- *$\mathcal{K} \models \sim (\phi \wedge \psi)$ iff $\mathcal{K} \models \sim \phi \vee \sim \psi$;*
- *$\mathcal{K} \models \sim (\phi \vee \psi)$ iff $\mathcal{K} \models \sim \phi \wedge \sim \psi$.*

Axioms, conjunction of axioms, disjunction of axioms and their quasi-negations are called *extended axioms*. Note that extended axioms are not in the language of KBs but used to build interactions between arguments in $\mathcal{ALC}$. We define a *defeat* relation between arguments in $\mathcal{ALC}$ in the following.

Definition 4. *An argument $\langle \Psi, \psi \rangle$ is a* defeater *of an argument $\langle \Phi, \phi \rangle$ such that $\psi \models \sim (\phi_1 \wedge \ldots \wedge \phi_n)$ for some $\{\phi_1, \ldots, \phi_n\} \subseteq \Phi$. An* undercut *for an argument $\langle \Phi, \phi \rangle$ is an argument $\langle \Psi, \sim (\phi_1 \wedge \ldots \wedge \phi_n) \rangle$ where $\{\phi_1, \ldots, \phi_n\} \subseteq \Phi$.*

In Example 1, the argument $\mathbf{A}_5$ is a defeater for the argument $\mathbf{A}_1$ and $\mathbf{A}_7$; the argument $\mathbf{A}_2, \mathbf{A}_3, \mathbf{A}_4$ are undercuts for the argument $\mathbf{A}_1$.

By Definition 4, it is easy to see that undercuts are defeaters. And it is also simple to get the next result.

Proposition 1. *If $\mathbf{A}'$ is a defeater for $\mathbf{A}$ then there exists an undercut $\mathbf{A}''$ for $\mathbf{A}$ where $\mathbf{A}' \preceq_c \mathbf{A}''$.*

Proposition 1 shows that for any defeater for an argument, there is an undercut for the argument that is more conservative than the defeater.

Since arguments can be ordered by pre-order relation $\prec_c$ as defined in Definition 2, it is nature to introduce the notion of *maximally conservative defeater* and *maximally conservative undercut*.

Definition 5. *Let $\mathbf{A}$ and $\mathbf{A}'$ be two arguments in $\mathcal{ALC}$. We say that $\mathbf{A}'$ is a* maximally conservative defeater *for $\mathbf{A}$ iff $\mathbf{A}'$ is a defeater for $\mathbf{A}$, and there is no defeater $\mathbf{A}''$ for $\mathbf{A}$, $\mathbf{A}' \prec_c \mathbf{A}''$. A* maximally conservative undercut *of $\mathbf{A}$ is analogously defined. We further define a* canonical undercut *for an argument $\mathbf{A}$ as a maximally conservative undercut for $\mathbf{A}$.*

In Example 1, the arguments $\mathbf{A}_2$, $\mathbf{A}_3$ and $\mathbf{A}_4$ are canonical undercuts of $\mathbf{A}_1$. The argument $\mathbf{A}_2$ is a canonical undercut of $\mathbf{A}_8$.

Proposition 2. *Given an argument $\mathbf{A}$, if $\mathbf{A}'$, $\mathbf{A}''$ are different canonical undercuts for $\mathbf{A}$ then $\mathbf{A}' \not\preceq_c \mathbf{A}''$ and $\mathbf{A}'' \not\preceq_c \mathbf{A}'$.*

Example 2. *Given a KB $\mathcal{K} = (\emptyset, \{C(a), D(a), \neg C(a), \neg D(a)\})$, both the following $\mathbf{A}_{11} = \langle\{\neg C(a)\}, \sim (C \sqcap D)(a)\rangle$ and $\mathbf{A}_{12} = \langle\{\neg D(a)\}, \sim (C \sqcap D)(a)\rangle$ are canonical undercuts for $\mathbf{A}_{13} = \langle\{C(a), D(a)\}, C \sqcap D(a)\rangle$, but neither is more conservative than the other.*

In Example 2, $\mathbf{A}_{11}$ and $\mathbf{A}_{12}$ are different canonical undercuts for $\langle\{C(a), D(a)\}, C \sqcap D(a)\rangle$. Sup$(\mathbf{A}_{11})$ is different from Sup$(\mathbf{A}_{12})$ while Con$(\mathbf{A}_{11})$ is the same as Con$(\mathbf{A}_{12})$.

Proposition 3. *For each defeater $\mathbf{A}_1$ for an argument $\mathbf{A}$, there exists a canonical undercut $\mathbf{A}_2$ for $\mathbf{A}$ such that $\mathbf{A}_1 \preceq_c \mathbf{A}_2$.*

4 Argumentative Semantics for $\mathcal{ALC}$

Basic concepts of argumentation theory in DLs are defined in the previous section (Step 1). Next, based on them, we valuate the argumentation framework (Step 2) and introduce an argumentation-based semantics for $\mathcal{ALC}$ (Step 3).

4.1 Argumentation Framework for Knowledge Bases

Firstly, we introduce an argumentation framework for $\mathcal{ALC}$ KBs. Given an $\mathcal{ALC}$ KB $\mathcal{K}$ and an axiom ϕ, an argumentation framework of ϕ w.r.t. $\mathcal{K}$ is composed of argument trees for/against ϕ defined as follows:

Definition 6. *Given an axiom ϕ, an* argument tree *for ϕ is a tree where the nodes are arguments such that*

(1) *the root is an argument for ϕ;*
(2) *for no node $\langle \Phi, \psi \rangle$ with ancestor nodes, $\langle \Phi_1, \psi_1 \rangle \dots \langle \Phi_n, \psi_n \rangle$ is Φ a subset of $\Phi_1 \cup \dots \cup \Phi_n$;*
(3) *the children nodes of a node* $\mathbf{A}$ *consist of all canonical undercuts for* $\mathbf{A}$*, which obeys* (2).

In Example 2, the argument tree T for $C \sqcap D(a)$ is shown in Fig.1.

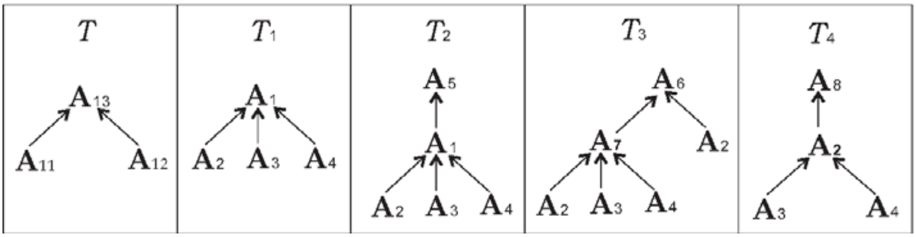


Fig. 1. Argument Trees

The following proposition ensures the argument tree in Definition 6 is well defined.

Proposition 4. *Any argument tree of an axiom ϕ in a finite KB $\mathcal{K}$ is finite*[1].

In particular, the argument trees for classically consistent KBs have the following characteristic.

Proposition 5. *If an $\mathcal{ALC}$ KB $\mathcal{K}$ is classically consistent, then all argument trees have exactly one node.*

Intuitively, an argument tree containing one node, i.e., only one root node, is always *successful* (defined later).

Definition 7. *An* argumentation framework *of an axiom ϕ is a pair of sets $\langle \mathcal{P}, \mathcal{C} \rangle$ where $\mathcal{P}$ is the set of argument trees of ϕ and $\mathcal{C}$ is the set of argument trees for $\sim \phi$.*

In Example 1, we obtain the argumentation framework $\langle \mathcal{P}, \mathcal{C} \rangle$ of $Fly(Tweety)$ is T_1, T_2, T_3 shown in Fig.1.

Here T_1 is an argument tree of $Fly(Tweety)$ and T_2, T_3 are argument trees of $\neg Fly$ $(Tweety)$. So the argumentation framework of $Fly(Tweety)$ is $\langle \langle \{T_1\}, \{T_2, T_3\} \rangle \rangle$.

In Example 1, the argumentation framework of $\exists HasFood.Fish(Tweety)$ is $\langle \mathcal{P}, \mathcal{C} \rangle$ where $\mathcal{C} = \emptyset$ and $\mathcal{P}$ containing an argument tree T_4 shown in Fig.1.

Proposition 6. *Let $\mathcal{K}$ be an $\mathcal{ALC}$ KB. Given an argumentation framework $\langle \mathcal{P}, \mathcal{C} \rangle$, if $\mathcal{K}$ is classical consistent, then each argument tree in $\mathcal{P}$ has exactly one node and $\mathcal{C}$ is the empty set.*

[1] A tree is finite iff it has a finite number of branches and a finite depth.

4.2 Argumentative Semantics

In the subsection, we define an argumentative semantics based on both binary argumentation and our argumentation framework.

At first, we define a concept so-called *successful argument tree* in argumentation framework. If $\mathbf{A}_1, \mathbf{A}_2$ and $\mathbf{A}_3$ are three arguments such that $\mathbf{A}_1$ is undercut by $\mathbf{A}_2$ and $\mathbf{A}_2$ is undercut by $\mathbf{A}_3$ then $\mathbf{A}_3$ is called a *defence* for $\mathbf{A}_1$. We define the "*defend*" relation as the transitive closure of "*being a defence*". An argument tree is said to be *successful* iff every leaf defends the root node. The *categorizer* is a function, denoted by $\mathbf{c}$, from the set of argument trees to $\{0, 1\}$ such that $\mathbf{c}(T) = 1$ iff T is successful.

The *categorization* of a set of trees is the collection of their categorizer values. The *accumulator* of a query ϕ is a function, denoted by $\mathbf{a}$, from categorizations to the set $\{(1,1), (1,0), (0,1), (0,0)\}$. Let $\langle X, Y\rangle$ be a categorization of argumentation framework of an axiom ϕ, then $\mathbf{a}(\langle X, Y\rangle) = (w(X), w(Y))$ where $w(Z) = 1$ iff $1 \in Z$.

Definition 8. *The* valuation *is a function, denoted by* $\mathbf{v}$*, from a set of axioms to a set* $\{$*both(*B*), true(*t*), false(*f*), unknown(*U*)*$\}$*, defined as follows:*

(1) $\mathbf{v}(\phi) = \mathrm{B}$ *iff* $\mathbf{a}(\langle X, Y\rangle) = (1, 1)$*;*
(2) $\mathbf{v}(\phi) = \mathrm{t}$ *iff* $\mathbf{a}(\langle X, Y\rangle) = (1, 0)$*;*
(3) $\mathbf{v}(\phi) = \mathrm{f}$ *iff* $\mathbf{a}(\langle X, Y\rangle) = (0, 1)$*;*
(4) $\mathbf{v}(\phi) = \mathrm{U}$ *iff* $\mathbf{a}(\langle X, Y\rangle) = (0, 0)$*;*

where $\langle X, Y\rangle$ *is a categorization of argumentation framework of* ϕ *in* $\mathcal{K}$*.*

Note that the accumulator of query Q can be given an intuitive explanation as follows:

- (1,1) means the answer to Q is "*both*", i.e., both true and false;
- (1,0) means the answer to Q is "*true*", i.e., not false but true;
- (0,1) means the answer to Q is "*false*", i.e., not true but false;
- (0,0) means the answer to Q is "*unknown*", i.e., neither true nor false.

Definition 9 (Argumentative Entailment). *Let* $\mathcal{K}$ *be a KB and* ϕ *an axiom in* $\mathcal{ALC}$*. We say* $\mathcal{K}$ argumentatively entails *(a-entails, for short)* ϕ*, denoted by* $\mathcal{K} \models_a \phi$*, iff there exists a successful argument tree of* ϕ*. In this case, we call* $\models_a$ argumentative entailment (relationship) *(a-entailment, for short).*

As a result, we state that there exists a good relationship among a-entailment, accumulator and categorization.

Theorem 1. *Let* $\mathcal{K}$ *be a classically consistent KB and* ϕ *an axiom in* $\mathcal{ALC}$*. Then the following propositions are equivalent each other:*

(1) $\mathcal{K} \models_a \phi$*;*
(2) $\mathbf{v}(\phi) \in \{\mathrm{B}, \mathrm{t}\}$*;*
(3) *there exists an argument tree* T *of* ϕ *such that* $\mathbf{c}(T) = 1$.

The following property ensures that the a-entailment over classically consistent KBs preserves the classical entailment.

Theorem 2. *Let* $\mathcal{K}$ *be a classically consistent KB and* ϕ *an axiom in* $\mathcal{ALC}$*.* $\mathcal{K} \models_a \phi$ *iff* $\mathcal{K} \models \phi$*.*

However, if a KB $\mathcal{K}$ is classically inconsistent, then $\models_a$ is weaker than $\models$. Clearly, $\mathcal{K} \models_a \sim \phi$ does not necessarily hold if $\mathcal{K} \not\models_a \phi$ for any axiom ϕ. $\mathcal{K} \models_a \psi$ is not inferred from $\mathcal{K} \models_a \phi$ and $\mathcal{K} \models_a \sim \phi \vee \psi$. Since $\{\phi, \sim \phi\} \not\models_a \psi$ for any axiom ψ, we conclude the following result.

Theorem 3. *The argumentative semantics is paraconsistent.*

5 Argumentation-Based Reasoning and Repair

5.1 Argumentation-Based Reasoning

Argumentative semantics is presented in the previous section. In this section, we mainly discuss two basic inference problems under the argumentative semantics, namely satisfiability and query entailment, which are two important tasks in DLs.

Firstly, we define a satisfiability called *argumentative satisfiability*. Let $\mathcal{K}$ be an $\mathcal{ALC}$ KB. A concept C is *argumentatively satisfiable* (a-satisfiable, for short) w.r.t. $\mathcal{C}$ iff for some individual a, there exists at least a successful argument tree w.r.t. $\mathcal{K}$ for $C(a)$; and *argumentatively unsatisfiable* (a-unsatisfiable, for short) otherwise.

As a result, we show that the argumentative satisfiability preserves the classical satisfiability in DLs.

Theorem 4. *Let $\mathcal{K}$ be an $\mathcal{ALC}$ KB. If a concept C is classical satisfiable w.r.t. $\mathcal{K}$ then C is a-satisfiable w.r.t. $\mathcal{K}$.*

Intuitively, a-satisfiable concepts are more general than satisfiable concepts. Based on the property, the classical incoherent TBox, i.e., there exists a classical unsatisfiable concept in the TBox, can be taken as a newly coherent TBox called *argumentatively coherent* (a-coherent, for short) under the argumentative semantics. In Example 1, the concept $\neg Fly$ is a-satisfiable and others concept are classical satisfiable.

There are two basic inference problems so-called *instance checking* and *subsumption checking* in a KB $\mathcal{K}$ under the argumentative semantics defined as follows:

- *instance checking*: given a concept C and an individual a, a is an *argumentative instance* (a-instance, for short) of concept iff $\mathcal{K} \models_a C(a)$.
- *subsumption checking*: a concept C is *argumentatively subsumed* (a-subsumed, for short) by a concept D iff $\mathcal{K} \models_a C \sqsubseteq D$.

In fact, two basic problems can be taken as a-entailment problems called *query a-entailment* problems.

By Theorem 2, query a-entailment problems in a classical consistent KB are equivalent to the query entailment problems. However, if a KB $\mathcal{K}$ is classically inconsistent, then some meaningful consequents (not all meaningless information) could be answered by judging whether there exists a successful argument tree w.r.t. $\mathcal{K}$ for/against the query. Intuitively, these consequents a-entailed by a KB are *justifiable*, i.e., they can protect themselves in arguing process with others axioms in the KB. In this case, it shows that the argumentative semantics embodies an important property so-called *justifiability* [18].

In Example 1, since the accumulator of $Fly(Tweety)$ is $(0,1)$, $\mathbf{v}(\neg Fly(Tweey))$ is t. Therefore, $\mathcal{K} \models_a \neg Fly(Tweey)$ and $\mathcal{K} \not\models_a Fly(Tweey)$. Though $\mathcal{K} \models Fly(Tweety)$ and $\mathcal{K} \models \neg Fly(Tweety)$, it shows that $\neg Fly(Tweety)$ is "*justifiable*" in $\mathcal{K}$ because the fact $\neg Fly(Tweety)$ has other arguments which support it, while $Fly(Tweety)$ is "*unjustifiable*". In short, our argumentation-based reasoning can justifiably answer a query by analyzing other arguments for/against the query in a given KB, instead of roughly answering the query by "*true*" or "*false*" in the classical semantics.

5.2 Argumentation-Based Repair

As a practical application of AI, *knowledge base repair* [19] is obviously an important task in the Semantic Web. I.e., when a KB $\mathcal{K}$ is inconsistent, there exists at least one contradiction in $\mathcal{K}$. Thus it is usually required to "repair" $\mathcal{K}$, by discarding its contradiction. In this sense, contradiction is treated as *problematical* knowledge. A repaired KB should not contain problematical knowledge. In fact, our argumentative semantics is exactly a tool to analyze and evaluate problematical knowledge in given KBs.

Based on binary argumentation [13], the value of each axiom of an inconsistent KB might be one of {B, t, f, U}. Next, we discuss which axioms should be removed from an inconsistent KB $\mathcal{K}$. Intuitively, we firstly wipe off those axioms whose values are f because there exists an argument supporting their negations. For instance, if the value of an axiom $C(a)$ is f then there exists an argument $\langle \Phi, \neg C(a) \rangle$ and $\Phi \subseteq \mathcal{K}$. Thus $C(a)$ is not naturally contained in $\mathcal{K}$ from classical KBs repair. Analogously, we should not remove those axioms whose values are t from $\mathcal{K}$. It is difficult to treat axioms whose values are B or U. Let's consider an example. Let $\mathcal{K} = \{\neg C(a), C \sqcup D(a), \neg D(a)\}$ be an inconsistent KB. It easily shows that the value of $\neg C(a)$ and $\neg D(a)$ are U. Intuitively, inconsistencies occur only when $\neg C(a)$ and $\neg D(a)$ are together. That is, neither $\neg C(a)$ nor $\neg D(a)$ is reliable. At last, we obviously find that axioms whose value are B should be discarded since their negation are valued to t. For instance, the value of $\bot(a)$ w.r.t. $\{\bot(a)\}$ is B while $\bot(a)$ is obviously a contradiction.

From the above discussion, we will develop a repair operator by only considering those axioms whose values are t as candidate members of a new KB after repairing. Before we introduce an argumentation-based repair operator, note that two fact: (1) TBoxes store inner knowledge of KBs and ABoxes store outer knowledge of KBs; (2) classical consistent ABoxes together with incoherent TBoxes may bring inconsistencies; (3) an $\mathcal{ALC}$ TBox is inconsistent only when it contains a GCI $\top \sqsubseteq \bot$. For instance, let $\mathcal{A} = \{C(a)\}$ be an ABox and $\mathcal{T} = \{C \sqsubseteq D, C \sqsubseteq \neg D\}$ an incoherent TBox. It is easy to show that $(\mathcal{A}, \mathcal{T})$ is inconsistent. For simplified consideration, we mainly discuss KBs whose TBoxes are classically coherent and consistent. In this case, we only repair the ABox w.r.t. the TBox instead of repairing the whole KB.

We argue that the argumentative entailment relation ($\models_a$) defined in the previous section is enough for inconsistency-tolerant reasoning and query answering in KBs. However, in KB repair scenario, readability is often more desirable than logical equivalence. In what follows, we introduce a specific repair operator which fully preserve the axiom structure of the original KB, and at the same time, is logically close to $\mathcal{K}$ with holding justifiability.

Next, we present a practicable repair operator to implement $\mathcal{ALC}$ ABox repair.

Definition 10 (Repair Operator). *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a KB in $\mathcal{ALC}$ where $\mathcal{T}$ is classically coherent and consistent. We define a repaired KB $\triangle_a(\mathcal{K}) = (\mathcal{T}, \triangle_a(\mathcal{A}))$ s.t.*

$$\triangle_a(\mathcal{A}) = \{C(t) \in \mathcal{A} \mid \mathcal{K} \models_a C(t) \textit{ and } \mathcal{K} \not\models_a \neg C(t)\}.$$

Note that $\triangle_a(\mathcal{K})$ is always finite if $\mathcal{K}$ is finite[2]. That is, our argumentation-based repair operator is well defined.

Theorem 5. *Let $\mathcal{K}$ be a KB in $\mathcal{ALC}$. We have*

(1) $\triangle_a(\mathcal{K})$ *is classically consistent;*
(2) *if $\mathcal{K}$ is classically consistent, then $\triangle_a(\mathcal{K}) \equiv \mathcal{K}$;*
(3) *if $\triangle_a(\mathcal{K}) \models \phi$ then $\mathcal{K} \models_a \phi$ where ϕ is an $\mathcal{ALC}$ axiom.*

In Theorem 5, (1) ensures that the repair result of a KB is classically consistent; (2) states that there is no change between a classical consistent KB and its repair result; and (3) shows that those consequences which are entailed by the repair result could also be a-entailed by its original KB.

For instance, let $\mathcal{K} = (\{Bird \sqsubseteq Fly\}, \{Bird(b_1), \neg Fly(b_1), Bird(b_2)\})$ be a KB. $\mathcal{K}$ is classically inconsistent. Then $\triangle_a(\mathcal{K}) = (\{Bird \sqsubseteq Fly\}, \{Bird(b_2)\})$. Because $\{Bird(b_1), \neg Fly(b_1)\}$ conflicts with $Bird \sqsubseteq Fly$, two axioms ($Bird(b_1)$, $\neg Fly(b_1)$) are not included in $\triangle_a(\mathcal{K})$. However, the axiom $Bird(b_2)$ does not contradict any axiom. Then, $Bird(b_2)$ is included in $\triangle_a(\mathcal{K})$.

Intuitively, given an inconsistent KB $\mathcal{K}$, the KB $\triangle_a(\mathcal{K})$ after argumentation-based repairing $\mathcal{K}$ is a reasonable and suitable candidate because it excludes those knowledge which might contradict or conflict with others, and restores those knowledge which could be justified.

6 Conclusion and the Future Work

In this paper, we have presented an argumentation-based approach to reason with inconsistent DL-based KBs. Firstly, we have defined arguments for axioms w.r.t. a KB and the relationship between arguments such as defeat and undercut. Then we have developed an argumentation framework of KBs to capture the structural inter-relationships both an axiom and other axioms in KBs and we have proposed a binary-argumentation-based called argumentative semantics. We have shown the argumentative semantics could be justifiably employed to cope with inconsistent KBs. Furthermore, an argumentation-based operator is introduced to repair inconsistent KBs with maintaining consistency and justifiability. Searching arguments is a key task of implementing argumentation-based reasoning with inconsistent KBS since the argumentation framework are built based on arguments. Though the work of searching arguments in DL-Lite KBs is discussed in [20], it is still an open problem in expressive DLs such as $\mathcal{ALC}$. Finding an efficient approach to search arguments in expressive DL-based KBs will be considered as our future work.

[2] $\mathcal{K}$ is finite iff the number of axioms in $\mathcal{K}$ is finite.

Acknowledgements

The authors would like to thank Yue Ma, Kedian Mu, Guilin Qi, Shuang Ren and Guohui Xiao for helpful comments and discussions. This work is supported by the program of the National Natural Science Foundation of China (NSFC) under grant number 60973003.

References

1. Zhang, X., Zhang, Z., Lin, Z.: An argumentative semantics for paraconsistent reasoning in description logic ALC. In: Proc. of DL 2009, UK, CEUR-WP 477, CEUR-WS.org (2009)
2. Bertossi, L.E., Hunter, A., Schaub, T. (eds.): Inconsistency Tolerance. LNCS, vol. 3300. Springer, Heidelberg (2005)
3. Schlobach, S., Cornet, R.: Non-standard reasoning services for the debugging of description logic terminologies. In: Proc. of IJCAI 2003, Mexico, pp. 355–362. Morgan Kaufmann, San Francisco (2003)
4. Huang, Z., van Harmelen, F., ten Teije, A.: Reasoning with inconsistent ontologies. In: Proc. of IJCAI 2005, UK, Professional Book Center, pp. 454–459 (2005)
5. Qi, G., Du, J.: Model-based revision operators for terminologies in description logics. In: Proc. of IJCAI 2009, USA, pp. 891–897 (2009)
6. Patel-Schneider, P.F.: A four-valued semantics for terminological logics. Artif. Intell. 38(3), 319–351 (1989)
7. Ma, Y., Hitzler, P., Lin, Z.: Algorithms for paraconsistent reasoning with OWL. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 399–413. Springer, Heidelberg (2007)
8. Odintsov, S.P., Wansing, H.: Inconsistency-tolerant description logic. part II: A tableau algorithm for CACL$^{\mathrm{C}}$. J. Applied Logic 6(3), 343–360 (2008)
9. Zhang, X., Lin, Z.: Paraconsistent reasoning with quasi-classical semantic in ALC. In: Calvanese, D., Lausen, G. (eds.) RR 2008. LNCS, vol. 5341, pp. 222–229. Springer, Heidelberg (2008)
10. Zhang, X., Xiao, G., Lin, Z.: A tableau algorithm for handling inconsistency in OWL. In: Aroyo, L., Traverso, P., Ciravegna, F., Cimiano, P., Heath, T., Hyvönen, E., Mizoguchi, R., Oren, E., Sabou, M., Simperl, E. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 399–413. Springer, Heidelberg (2009)
11. Zhang, X., Lin, Z., Wang, K.: Towards a paradoxical description logic for the semantic web. In: Link, S. (ed.) FoIKS 2010. LNCS, vol. 5956, pp. 306–325. Springer, Heidelberg (2010)
12. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell. 77(2), 321–358 (1995)
13. Besnard, P., Hunter, A.: A logic-based theory of deductive arguments. Artif. Intell. 128(1-2), 203–235 (2001)
14. Tempich, C., Simperl, E.P.B., Luczak, M., Studer, R., Pinto, H.S.: Argumentation-based ontology engineering. IEEE Intelligent Systems 22(6), 52–59 (2007)
15. Gomez, S.A., Chesnevar, C.I., Simari, G.R.: An argumentative approach to reasoning with inconsistent ontologies. In: Proc. of KROW 2008, Australia. CRPIT 90, pp. 11–20. ACS (2008)
16. Black, E., Hunter, A., Pan, J.Z.: An argument-based approach to using multiple ontologies. In: Godo, L., Pugliese, A. (eds.) SUM 2009. LNCS (LNAI), vol. 5785, pp. 68–79. Springer, Heidelberg (2009)

17. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, Cambridge (2003)
18. Dung, P.M.: An argumentation-theoretic foundations for logic programming. J. Log. Program. 22(2), 151–171 (1995)
19. Kalyanpur, A.: Debugging and repair of OWL ontologies. PhD thesis, College Park, MD, USA, Adviser-Hendler, James (2006)
20. Zhang, X., Lin, Z.: An argumentation-based approach to handling inconsistencies in DL-Lite. In: Mertsching, B., Hund, M., Aziz, Z. (eds.) KI 2009. LNCS (LNAI), vol. 5803, pp. 615–622. Springer, Heidelberg (2009)

Finding a Single, All, or the Most Probable Solution to a Finite or Non-finite Interval Algebra Network

André Trudel

Jodrey School of Computer Science, Acadia University,
Wolfville, Nova Scotia, B4P 2R6, Canada
Andre.Trudel@acadiau.ca

Abstract. We present a unified approach for finding a single, all, or the most probable solution to an Interval Algebra network. The network may contain finite, non-finite, or a mixture of both types of temporal intervals.

Keywords: Allen's interval algebra, interval algebra networks, temporal reasoning.

1 Introduction

Allen [1] defines a temporal reasoning approach based on intervals and the 13 possible binary relations between them. The relations are before (b), meets (m), overlaps (o), during (d), starts (s), finishes (f), and equals (=) (see Table 1). Each relation has an inverse. The inverse symbol for b is bi and similarly for the others: mi, oi, di, si, and fi. The inverse of equals is equals.

A relation between two intervals is restricted to a disjunction of the basic relations, which is represented as a set. For example, (A m B) V (A o B) is written as A {m,o} B. The relation between two intervals is allowed to be any subset of I = {b, bi, m, mi, o, oi, d, di, s, si, f, fi, =} including I itself.

An IA (interval algebra) network is a graph where each node represents an interval. Directed edges in the network are labelled with subsets of I. By convention, edges labelled with I are not shown. An IA network is consistent (or satisfiable) if each interval in the network can be mapped to a real interval such that all the constraints on the edges hold (i.e., one disjunct on each edge is true).

P. van Beek wrote efficient C code to solve IA networks. His complete package includes approximately 7500 lines of code. The algorithms are described in [13]. To use the code, one must be an expert C programmer, and have a deep understanding of the algorithms. The implementation's emphasis is on efficiency, not on user friendliness.

P. van Beek's algorithms have also been implemented in the constraint logic programming language ECLiPSe [2] by Fruhwirth [3]. Another Eclipse implementation [5] uses a meta constraint solving approach. The edges in the IA network are the variables. The domain of each variable is a set of subsets of I. The domains, although finite, can grow large if they include all the subsets of I. The meta qualitative constraint solver requires code for the operations of intersection and composition, and meta-heuristic rules. Note that both [3, 5] also handle quantitative constraints which are not dealt with in this paper.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 100–110, 2010.

Table 1. Allen's interval relations

Relation	**Symbol**	**Example**
X before Y	b	XXX YYY
X meets Y	m	XXXYYY
X overlaps Y	o	XXXX YYYY
X during Y	d	XXX YYYYYYY
X starts Y	s	XXX YYYYYYY
X finishes Y	f	XXX YYYYYYY
X equals Y	=	XXX YYY

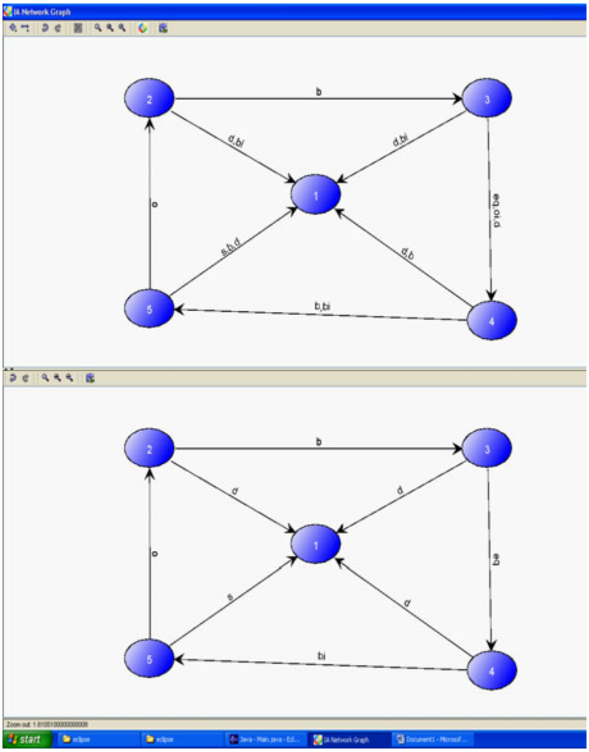


Fig. 1. A solution

An implementation based on converting an IA network to a finite domain constraint satisfaction problem and then using local search is presented in [9]. IA networks have also been solved using SAT methods [7].

In this paper, we present an IA network solution method which in addition to finding a single solution can find all the solutions to the network. All the solutions can then be used to solve a probabilistic version of the network. The network's intervals are not assumed to be finite which is the case with previous implementations. The network may contain finite, non-finite, or a mixture of both types of temporal intervals.

All previous IA network implementations are non-trivial. The user must be an expert in the implementation language and software. The average user is usually not capable of making even simple extensions or modifications. Just getting the software to function is usually a challenge! The objective of our implementation is user friendliness and ease of use. The user is presented with a graphical user interface (GUI). No knowledge of the underlying implementation language and algorithms are required. Simply by pointing and clicking with the mouse, the user draws an IA network in the GUI and then clicks a button to request a solution. If a solution exists, it is drawn below the input network. The target audience is researchers that need to quickly verify or generate a few solutions, and students entering the temporal area. The implementation could also be used as a teaching tool.

In the following sections, we present the GUI, followed by the implementation details. An initial description of the implementation appears in [12].

2 Graphical User Interface

The GUI has 2 windows which are initially both empty. The user enters an IA network in the top window. There are buttons for drawing nodes and edges, and most of the graph is drawn with the mouse. Using the keyboard, the user enters the interval relationships on the edges separated by commas. There are no restrictions on the size or shape of the graph.

After entering the IA network, the user clicks a button to request a solution. If the network is inconsistent, a warning message is displayed. Otherwise, a solution is displayed in the bottom half of the GUI. The edges and nodes are re-drawn as entered by the user. Each edge label will be one of the interval relationships originally entered by the user and the entire network is consistent.

Fig. 1 is a screenshot of the GUI. The top window contains the IA network entered by the user. The solution is shown in the bottom window. Note that edges not entered by the user are assumed to exist with I labels, but are not displayed. For example, in the IA network in the top half of Fig. 1 there is a hidden edge from node 2 to node 4 with a label of I. The edge is also not shown in the solution in the bottom half of Fig. 1.

In addition to finding a solution, the GUI also has a button for finding all the solutions to an IA network. As with finding a single solution, the user enters the network in the top half of the GUI. After clicking the all solutions button, solutions are shown one at a time in the bottom window. The user clicks to scroll through the solutions. For example, after entering the IA network in the top half of Fig. 2 and clicking the all solutions button, the first solution is shown in the bottom half of Fig. 2. Note that each edge label in the network in the top half of Fig. 2 can appear in a solution for a total of 8 possible solutions. Each of the 8 solutions are presented to the user one at a time. Note that other solutions involving changing labels on hidden edges are also possible, but not shown to the user.

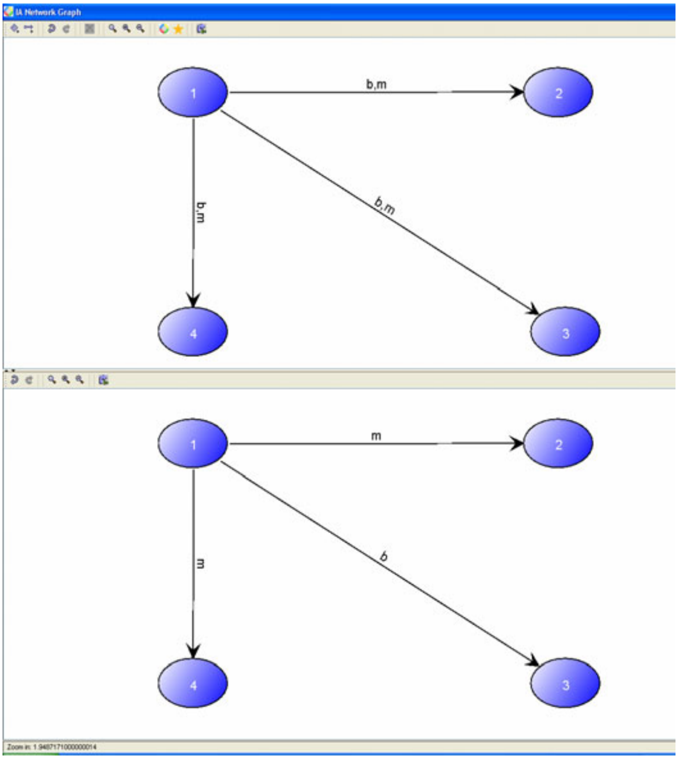


Fig. 2. All solutions

A probabilistic version of the network in the top of Fig. 2 is shown in the top of Fig. 3. Each relation is assigned a probability, and the probabilities on an edge sum to 1. One interpretation for the numbers is preference. For example, on the edge (1,4) there is a strong preference for “m” over “b”. In the diagonal edge, there is no preference. If there is no edge between two nodes, it is assumed to be present with the label I and each relation has equal probability 1/13.

A solution to a probabilistic IA (PIA) network is a consistent labelling which maximizes the product of the probabilities associated with each label in the solution [8]. For example, the solution to the PIA network in the top half of Fig. 3 is shown in the bottom half. Although not visible in the figure, the GUI also prints out the value of the product of the labels in the solution which is 0.315 in this case.

Allen’s Interval Algebra [1] is defined strictly in terms of finite intervals. Let us now extend the algebra to include non-finite intervals. An IA network can now contain finite, and intervals which are infinite in one or both directions. For example in the network at the top of Fig. 4, interval 1 is finite, interval 2 is infinite on the left and finite on the right, interval 3 is infinite in both directions, and interval 4 is finite on the left and infinite on the right. As proven in [15], if all the edges are properly labelled, finite interval IA network software can be used to solve non-finite interval IA networks. A solution to the non-finite interval network is shown at the bottom of Fig. 4.

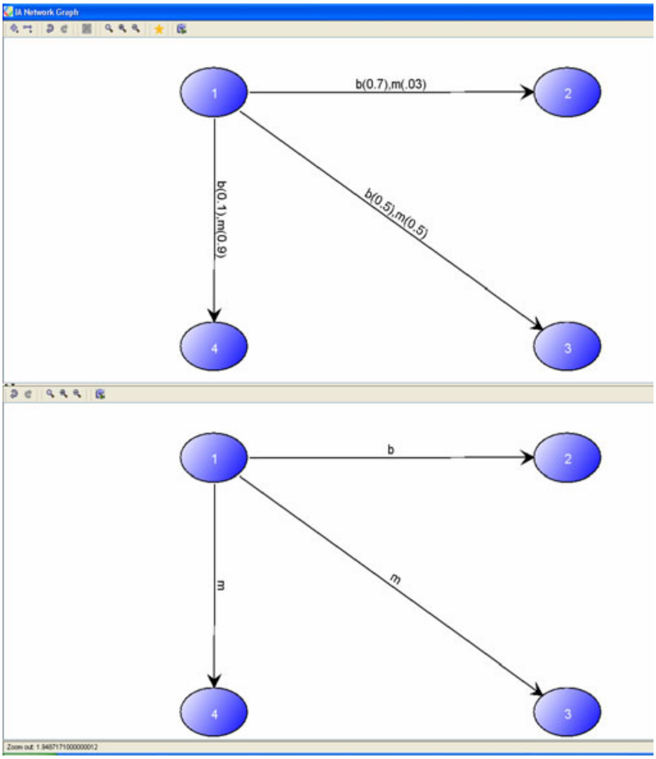


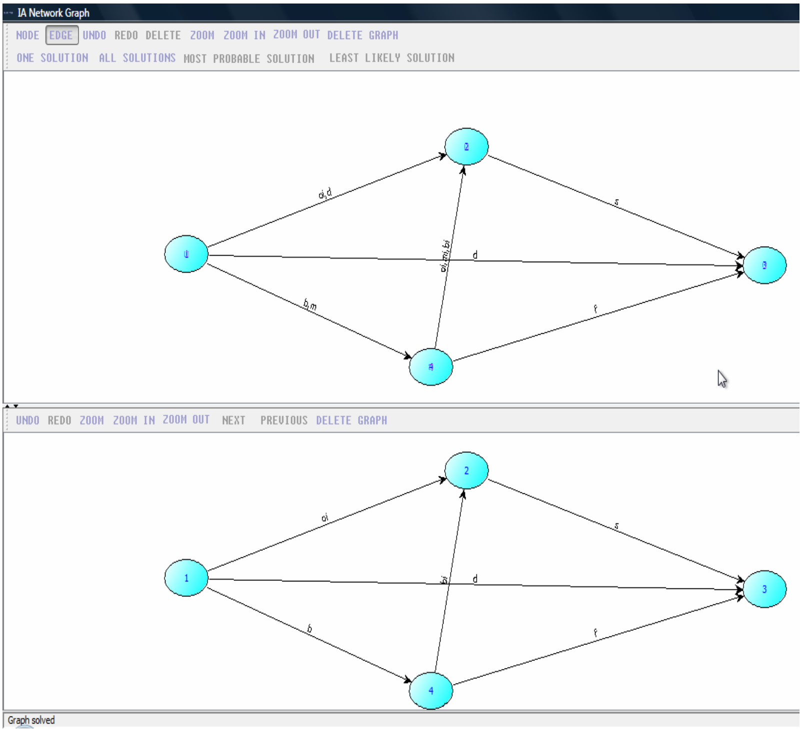
Fig. 3. Most probable solution

3 Implementation

The implementation components are shown in Fig. 5. The user interacts with the implementation via the GUI which is built using JGraph [4]. After the user has entered a graph and requested a solution, control is given to the Java program. The program first reads in a constraint logic programming template. The template is updated with information from the user's network. The completed logic program is then passed on to Eclipse [2]. Eclipse will solve the graph and then return the solution to the Java program. The java program will then display the solution in the GUI. Note that the user only interacts with the GUI. The other components are hidden from the user.

An IA network is traditionally defined to be a binary CSP with infinite domains. The intervals are the variables. The domain of each variable is the set of pairs of reals of the form (x,y) where x<y. The constraint between two variables i and j is the label on the edge (i,j) in the IA network.

During the past three decades, research on IA networks and finite domain CSPs has progressed relatively independently. The reason is that algorithms specifically designed for finite domains are usually not applicable to infinite domains. Although part of the area's folklore, it was not widely known that IA networks are indeed finite domain CSPs. For example, van Beek and Manchak [13] write that "two of their



Graph solved

Fig. 4. Non-finite interval network

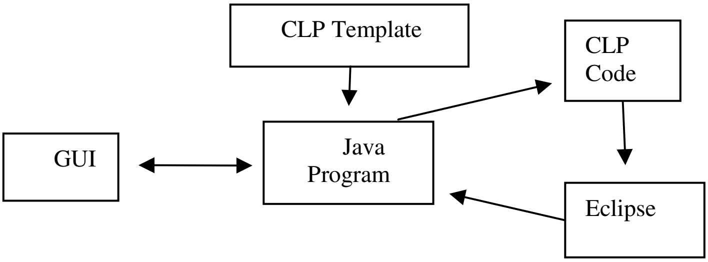


Fig. 5. Implementation

heuristics cannot be applied in our context as the heuristics assume a constraint satisfaction problem with finite domains, whereas IA networks are examples of constraint satisfaction problems with infinite domains".

Thornton et al. [9] show how to convert an IA network into an equivalent non-binary CSP with finite integer domains. They observe that the relative positions of the interval endpoints in an IA network can be used to determine consistency:

Theorem 1: Each interval in an IA network with *n* intervals can be mapped to a real interval such that all the constraints on the edges hold if and only if each interval in the IA network can be mapped to an interval with integer end-points in the range 1,…,2n such that all the constraints on the edges hold.

Based on theorem 1, Thornton et al. [9] convert an IA network to a non-binary CSP with finite domains. Each endpoint becomes a variable with domain {1,…,2n}. A label on an edge from X=(X-,X+) to Y=(Y-,Y+) in the IA network imposes a constraint on some or all of the variables X-, X+, Y-, and Y+. For example, X {d} Y generates the constraint (Y- < X-) & (X+ < Y+) which is non-binary since the constraint involves 4 variables (it is not two separate and independent binary constraints). They then apply local search techniques on the non-binary CSP.

A result equivalent to theorem 1 was derived earlier by Ligozat [6]. In [6], a purely algebraic proof in terms of p-intervals is provided for the equivalence between general IA networks and networks with integer end-points in the range of 1 to 2n.

Using theorem 1, it is also possible to convert an IA network to a binary CSP with finite integer domains [10 and 11].

In our implementation, we convert the IA network to a non-binary CSP with finite integer domains and use Eclipse to solve it. Details are given in the following section.

4 Eclipse Components

In this section, we return to the implementation diagram in Fig. 5 and explain how the IA network entered by the user is solved as a finite domain non binary CSP. The Java program in Fig. 5 first reads in the CLP template file, which contains constraint code for Allen's relations. The relations are implemented by placing restrictions on the endpoints. For example, interval X = (XL,XR) is before Y = (YL,YR) if and only if XR < YL. In Eclipse, we write: b(XL,XR,YL,YR,1) :- XR < YL. The "1" in the last parameter of b is a numeric representation of b and is used to keep track of the relationships on the edges. The relations are numbered from 1 to 13. The after relationship bi is implemented in terms of before: bi(XL,XR,YL,YR,2) :- b(YL,YR,XL,XR,1). The other relations are similarly implemented. The CLP template file also contains clauses to enforce that the left endpoint of each interval precedes its right endpoint.

The Java program copies the contents of the CLP template file to the CLP code file. Graph specific code is then added to the file. Assume we are given an IA network with n intervals (nodes) numbered from 1 to n. The left and right endpoints of the i'th interval are labelled Li and Ri respectively. The set of endpoints is represented in Eclipse as: EndPoints = [L1,R1,L2,R2,…,Ln,Rn]. For example, if n=4 we have: EndPoints = [L1,R1,L2,R2,L3,R3,L4,R4]. The range of each interval endpoint must be explicitly specified and is between 1 and 2n. In Eclipse, this is written in the following format: EndPoints :: 1..2n. For example, if n=4 we write: EndPoints :: 1..8. The edges are also labelled. For example, if there are 5 edges: Edges = [E1, E2, E3, E4, E5].

Every edge constraint is a disjunction of relationships. We represent the disjunction directly. For example, let the constraint on edge E1 between intervals 1 and 2 be meets or overlaps (i.e., {m,o}). This constraint is represented in Eclipse as:

(m(L1,R1, L2,Y2,E1); o(L1,R1, L2,Y2,E1)).

Singleton labels are represented directly. For example if instead we have {m} we simply write: m(L1,R1, L2,Y2,E1).

Next, we request Eclipse to generate a solution with the query: finda_solution (Edges). If a solution is found, Edges will be bound to a list of integers. Each integer represents the Allen relation for that edge.

The CLP code file generated for the network in Fig. 1 is shown in Fig. 6. Note that it is typical that only 1 page of Eclipse code is generated to solve the IA network.

We extend the implementation to find all the solutions to an IA network by backtracking over solutions. We must be careful what we backtrack over:

- **Labels:** Traditional IA network software assumes that each pair of nodes has an edge between them. If an edge is not explicitly shown, it is assumed to have a label of I. For example, the network in the top half of Fig. 2 becomes the one in Fig 6. The number of solutions involving the solid edges of Fig 6. has not changed, it remains at 8. Also, note that path consistency will not reduce the initial labeling of Fig 6. If we backtrack over the labels in the network, we must consider 17,576 possible solutions. Notice the combinatorial explosion with a network of only 4 nodes!

```
:-lib(ic).
:-lib(ic_global).
:-lib(propia).
b(_,XR,YL,_,1):- XR < YL.
bi(XL,XR,YL,YR,2):- b(YL,YR,XL,XR,_).
m(_,XR,YL,_,3):- XR = YL.
mi(XL,XR,YL,YR,4):- m(YL,YR,XL,XR,_).
o(XL,XR,YL,YR,5):- XL < YL, XR > YL, XR < YR.
oi(XL,XR,YL,YR,6):- o(YL,YR,XL,XR,_).
d(XL,XR,YL,YR,7):- XL > YL, XR < YR.
di(XL,XR,YL,YR,8):- d(YL,YR,XL,XR,_).
s(XL,XR,YL,YR,9):- XL = YL, XR <YR.
si(XL,XR,YL,YR,10):- s(YL,YR,XL,XR,_).
f(XL,XR,YL,YR,11):- XL > YL, XR = YR.
fi(XL,XR,YL,YR,12):- f(YL,YR,XL,XR,_).
eq(XL,XR,YL,YR,13):- XL = YL, XR = YR.
lrConstraint([]).
lrConstraint([L,R|T]):- L < R,lrConstraint(T).
finda_solution(Edges) :-
     Edges =[E1,E2,E3,E4,E5,E6,E7,E8],
     EndPoints= [L1,R1,L2,R2,L3,R3,L5,R5,L4,R4],
     EndPoints:: 1..10,
     lrConstraint(EndPoints), % insert 1 constraint for each edge
     (s(L5,R5,L1,R1,E1);b(L5, R5, L1, R1,E1);d(L5, R5, L1, R1,E1)),
     (d(L2, R2, L1, R1,E2);bi(L2, R2, L1, R1,E2)),
     (d(L3, R3, L1, R1,E3);bi(L3, R3, L1, R1,E3)),
     (d(L4, R4, L1, R1,E4);b(L4, R4, L1, R1,E4)),
     b(L2, R2, L3, R3,E5),
     (eq(L3,R3,L4,R4,E6);oi(L3,R3,L4,R4,E6);d(L3, R3, L4, R4,E6)),
     (b(L4, R4, L5, R5,E7);bi(L4, R4, L5, R5,E7)),
     o(L5, R5, L2, R2,E8),
     labeling(EndPoints).
```

Fig. 6. CLP code file

- **Endpoints:** If instead, we use software based on Thornton et al.'s [9] approach, we have a network of 4 nodes and must find an assignment of each interval's endpoints to an integer in the range from 1 to 8. For example, in the bottom half of Fig. 2, there are 70 different ways that the endpoints of nodes 1 and 3 can be

> assigned to integers in the range from 1 to 8 so that b holds. For example, one assignment is (1,2) and (5,8). For meets, there are 56 different assignments for the endpoints. Therefore, to find all possible solutions by backtracking over the possible endpoint assignments, we must consider over 60 billion possibilities.

Note that in the above, the worst case number of possibilities to backtrack over is given. Clever algorithms and heuristics can reduce the number of possibilities.

The approach we adopted is to backtrack over labels, but ignore all I labelled edges. We only backtrack over labels specifically entered by the user. We first generate a candidate solution which has one label on each edge. We check if this is a solution. We then generate the next candidate solution and so on. For example, for the network in the top half of Fig. 2 we generate and test 8 candidate solutions.

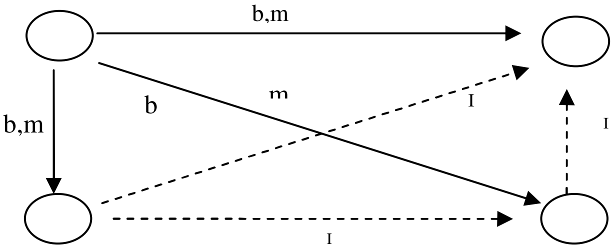


Fig. 7. Simple IA network with missing edges included

We added code to the "CLP Code" file in Fig. 4 which first generates all the possible solutions. We then apply the original code for finding a single solution to each possible solution to verify if indeed it is a solution. The set of valid solutions are passed back to the "Java Program". The program stores the solutions in a two dimensional array and displays the solutions in the GUI one at a time. For example, the 8 possible solutions are shown one at a time in the window's bottom half in Fig. 2. The Eclipse code generated for this example is shown in Fig. 8.

We use the all solutions feature to solve a probabilistic IA network. We first strip off the probabilities, and then generate and store all the solutions. For each solution, we compute a value by re-assigning a probability to each label in the solution and taking the product. The solution with the highest value is the solution to the probabilistic network. This extra processing is added to the "Java Program" in Fig. 4. The Eclipse code was not modified. It is trivial to add code to find the least likely, or median solution.

Since finding a single solution to an IA network is an NP complete problem, finding all the solutions is not feasible for large problem instances. In the worst case, an *n*-node network has $13^{n(n-1)/2}$ candidate solutions. It is easy to construct 10-node networks that take overnight to find all the solutions. Our implementation is targeted at small problem instances when finding all the solutions. Details of an efficient implementation that solves large PIA networks appear in [14].

No changes to the system were required to implement the non-finite interval networks. As proven in [15], IA networks containing non-finite intervals can be solved using finite interval software. Before the software is used, the network must be pre-processed. Missing edges are not assumed to be labelled with I. Instead, the label assigned to missing edges depends on the types of the intervals incident with the edge.

5 Future Work

The implementation can be extended to present an alternative representation of the solution to the user. When solving a network, the Eclipse code assigns each interval in the *n* node network to a finite interval in the range 1 to 2*n*. These finite interval assignments can be used to draw the intervals. For example, another representation of the solution in the bottom of Fig. 2 is shown in Fig. 9.

The Java program in Fig. 4 calls an Eclipse program to solve the network. The Java program can be modified to use other solution software (e.g., van Beek's code, or Zhang's et al [14] code for PIA networks).

If the network is inconsistent, a message is displayed. Instead, if a sub-network is causing the inconsistency, it could be highlighted. The user then has the option to modify or correct the network. Algorithms for finding minimal inconsistent sub-networks appear in [14].

```
%import required libraries - same as in Fig. 6
% Allen's 13 interval constraint relations - same as in Fig. 6
% left enpoint is strictly less than right endpoint for each interval
% same as in Fig. 6
% only keep valid solutions; prune out the bad ones
prune_Temp([],S,S).
prune_Temp([E|R],X,Y) :- finda_solution(E),!,prune_Temp(R,[E|X],Y).
prune_Temp([_|R],X,Y) :- prune_Temp(R,X,Y).
% Everything above is problem independent.
% Code below has to be customized for each problem.
finda_solution(Edges) :-
     Edges =[E1,E2,E3],
     EndPoints= [L1,R1,L4,R4,L2,R2,L3,R3],
     EndPoints:: 1..8,
     lrConstraint(EndPoints), % insert 1 constraint for each edge
     (b(L1, R1, L2, R2,E1);m(L1, R1, L2, R2,E1)),
     (b(L1, R1, L3, R3,E2);m(L1, R1, L3, R3,E2)),
     (b(L1, R1, L4, R4,E3);m(L1, R1, L4, R4,E3)),
     labeling(EndPoints).
findall_solutions(Solutions) :-
        bag_create(Bag),
        (  % insert 1 member for each edge
            member(E1,[1,3]), member(E2,[1,3]), member(E3,[1,3]),
             E = [E1,E2,E3],
            bag_enter(Bag, E), fail
        ; true), bag_dissolve(Bag, Temp)
        %Temp contains all the possible single edge graph labellings
        %Need to prune out invalid labellings
        prune_Temp(Temp, [], Solutions).
```

Fig. 8. Eclipse code for the all solutions example

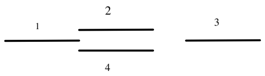


Fig. 9. Alternate solution representation

6 Conclusion

The work described in this paper has two unique features. The first is that it is the only implementation which can find a single, all, or probabilistic solution to an IA network. The intervals are not restricted to being finite.

The second unique feature of our approach is that due to its GUI, the implementation can be used by non-technical users. The user does not need to learn specialized software and algorithms. The implementation allows the user to draw any IA network and solve it. Also, the implementation has pedagogical value.

Acknowledgments. Research is supported by an NSERC Discovery Grant.

References

1. Allen, J.F.: Towards a general model of action and time. Artificial Intelligence 23(2), 123–154 (1984)
2. ECLiPSe, `http://www.eclipse-clp.org/`
3. Fruhwirth, T.: Temporal reasoning with constraint handling rules. Technical report ECRC-94-05, European computer-industry research centre, Germany (1994)
4. JGraph, `http://www.jgraph.com/`
5. Lamma, E., Milano, M., Mello, P.: Temporal reasoning in a meta constraint logic programming architecture. In: Third international workshop on temporal representation and reasoning (TIME 1996), Florida, pp. 128–135 (1996)
6. Ligozat, G.: Weak representations of interval algebras. In: 8th National Conference on Artificial Intelligence (AAAI 1990), Boston, USA, pp. 715–720 (1990)
7. Pham, D.N., Thornton, J., Sattar, A.: Modelling and solving temporal reasoning as propositional satisfiability. Artificial Intelligence 172(15), 1752–1782 (2008)
8. Ryabov, V., Trudel, A.: Probabilistic Temporal Interval Networks. In: 11th International Symposium on Temporal Representation and Reasoning (TIME 2004), Tatihou Island, France, pp. 64–67 (2004)
9. Thornton, J., Beaumont, M., Sattar, A., Maher, M.: A local search approach to modeling and solving interval algebra problems. Journal of logic and computation 4(1), 93–112 (2004)
10. Trudel, A.: How to convert a qualitative temporal CSP into a finite domain binary CSP. In: Spatial and Temporal Reasoning workshop held during the 18th International Joint Conference on Artificial Intelligence (IJCAI 2003), Acapulco, Mexico, pp. 121–124 (2003)
11. Trudel, A., Zhang, H.: Exploiting the relationship between IA networks and finite domain CSPs. In: 12th International Symposium on Temporal Representation and Reasoning (TIME 2005), Vermont, USA, pp. 177–179 (2005)
12. Trudel, A.: Finding all the solutions to an IA network. In: Workshop on Spatial and Temporal Reasoning held during IJCAI 2005, Edinburgh, Scotland, pp. 71–76 (2005)
13. van Beek, P., Manchak, D.W.: The design and experimental analysis of algorithms for temporal reasoning. Journal of Artificial Intelligence Research 4, 1–18 (1996)
14. Zhang, K., Trudel, A.: Efficient heuristics for solving probabilistic interval algebra networks. In: The 13th International Symposium on Temporal Representation and Reasoning (TIME 2006), Budapest, Hungary, pp. 111–118 (2006)
15. Trudel, A.: Interval Algebra networks with infinite intervals. In: 16th International Symposium on Temporal Representation and Reasoning (TIME 2009), Bressanone-Brixen, Italy, pp. 141–146 (2009)

A Dynamic Pricing Approach in E-Commerce Based on Multiple Purchase Attributes

Tapu Kumar Ghose and Thomas T. Tran

School of Information Technology and Engineering
University of Ottawa, Ottawa, ON K1N 6N5, Canada
{tghos009,ttran}@site.uottawa.ca
http://www.site.uottawa.ca

Abstract. In this paper, we propose an approach of dynamic pricing where buyers purchase decision is dependent on multiple preferred purchase attributes such as product price, product quality, after sales service, delivery time, sellers' reputation. The approach requires the sellers, by considering the five attributes, to set an initial price of the product with the help of their prior knowledge about prices of the product offered by other competing sellers. Our approach adjusts the selling price of products automatically with the help of neural network in order to maximize seller revenue. The experimental results portray the effect of considering the five attributes in earning revenue by the sellers. Before concluding with directions for future works, we discuss the value of our approach in contrast with related work.

Keywords: Dynamic Pricing, Multiple Purchase Attributes, Electronic Commerce.

1 Introduction

In dynamic pricing products prices always respond to the fluctuation of the market and hence the prices keep on changing with the tick of a clock. Every seller wants to set the selling price of their products so that their revenue is maximized. Determining selling prices of products is a challenging task for the sellers to sustain in the market. The purpose of the dynamic pricing problem is to determine the selling prices such that sellers receive maximum revenue. Usually, a customer before buying a product selects a store/seller for the purchase. The selection may be done under multiple attributes (preferences), such as best price offered, after-sale services, product quality, delivery time, sellers' reputation etc. Therefore, the sellers have to provide a competitive price for a product in response to variation in the market parameters such as competitors' prices and consumers purchase preferences. There exist intelligent agents, called pricebots, which enable online sellers to dynamically calculate a competitive price for a product. According to Dasgupta et al. [8], "these intelligent agents provide a convenient mechanism for implementing automated dynamic pricing algorithms for the sellers in an online economy". However, some intelligent agents

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 111–122, 2010.

use a number of assumptions for the dynamic pricing in online markets. Some intelligent agents assume that sellers are provided with complete knowledge of market parameters, while some other agents consider product price as the only attribute that determines consumers' purchase decision [8]. In recent decades extensive research has been done in dynamic pricing. Some of the research made an assumption that there is only one seller in the market [16]. On the contrary, in real life sellers have limited or no prior knowledge about the market parameters (e.g., buyer's reservation price, competitive sellers' price and profit etc). In addition, in reality there exist several competitive sellers in online market.

The goal of this work is to address the problem of dynamic pricing in a competitive online economy, where a buyer's purchase decision is determined by multiple attributes. From the knowledge of our literature review [8,6,7], the most common attributes that can play vital role in determining customers' purchase decision would include product price, product quality, delivery time, after-sale service, and sellers' reputation. In our model we consider these mentioned five attributes in determining a competitive price for a product P. We use feed-forward neural network to determine a competitive price for the products in order to maximize sellers' revenue. In our simulation we showed that once the sellers set an initial price of the product, our model adjusts the price of the product automatically with the help of neural network in order to maximize profits. In setting the initial price of a product, we assume that sellers use their prior knowledge about the prices of the product offered by other competing sellers. The remaining of the paper is organized as follows: Section 2 provides background information on feed-forward neural network. Sections 3 discusses related work. Section 4 presents our proposed approach for dynamic pricing. Section 5 represents results and analysis from our simulation. Section 6 provides a brief discussion on our approach. Section 7 concludes the paper with future research directions.

2 Feed-Forward Neural Network

In feed-forward Neural Networks the nodes in input layer accept information from outside the network, while the nodes in output layer send information outside the network. Each node, also known as unit, is connected to one or more other nodes by directed links. Each link contains a numerical weight, for instance $W_{i,j}$ indicates the strength of the connection between unit i and unit j [9]. Each unit u_i has an activation value a_i which acts as output of the unit. The activation value is calculated as follows:

$$a_i = f\left(\sum_{j=0}^{i-1} W_{j,i} a_j\right) . \tag{1}$$

where $\sum_{j=0}^{i-1} W_{j,i} a_j$ is the weighted sum of the inputs to unit u_i and f is the activation function applied to the weighted sum. We have chosen logistic sigmoid function as the activation function.

$$f(x) = \frac{1}{1 + \exp^{-x}}.$$

3 Related Works

Over the past few years there can be observed a noticeable rise in interest of dynamic pricing in commercial and research communities. In spite of rich literatures in the field, majority of the research works do not consider the competition markets [13]. Kephart et al. [1], for their work, considered a picture where a monopolist seller willing to maximize his/her revenue, provided buyers demand curve is random and unpredictable. Li et al. [17] studied the enterprises' dynamic decision problem on price strategies (dynamic pricing decision) in duopolistic retailing market under uncertain market state. Chinthalapati et al. [10] used machine learning based approach to study price dynamics in an electronic retail market. In the study they have taken price attributes into consideration that would determine a customer's buying decision. Dasgupta et al. [11] studied dynamic pricing in a multi-agent economy which consisted of buyers and competing sellers. They had taken price as the only attribute which take part in buyers purchase decision. In contrast, our model considers four more attributes (product quality, delivery time, after sale service and sellers' reputation) other than price. In fact our model is general enough to work for any number of attributes. Moreover, our model is not limited to two competitive sellers. Our model can work for both monopolist market and a competitive market with multiple sellers.

Dimicco et. al [5], by using Learning Curve Simulator, analyzed performance of two adaptive pricing algorithms: Goal-Directed (GD) and Derivative-Following (DF). They considered both monopoly and competitive economy of finite markets where goods like airlines ticket, sport events ticket, perishable goods have to be sold by finite time horizon. Alexandre et. al [14] discussed on the problems of dynamic pricing in finite time horizon. They considered a retailer who has to set the price of a good to optimize the total expected revenues over a period of time T. Their model is dependent on demand curve of the products. Kong [12], in his paper, examined seller strategies for dynamic pricing in a market for which a seller has finite time horizon to sell its inventory. Dasgupta et. al [15], in their paper, employed push strategies mechanism for dynamic pricing where they make use of demand curve. The authors considered time-limited goods in a supplier driven marketplace where goods are sold by maintaining strict deadline. On the contrary, our model of dynamic pricing does not require the sellers to figure out the demand curve of the products. Moreover, the model is not limited to goods with finite time horizon.

Greenwald and Kephart [2] explored no-regret learning for probabilistic pricing algorithm. In their model they considered an economy for single homogeneous goods. On the other hand, our model is not restricted to homogeneous goods. Our model is not concerned about how many sellers are there in the market, whereas, Tesauro and Kephart [3], in their experiment they assumed that there are only two competing sellers participating in the economy who alternatively take turns in adjusting their prices at each time step.

4 Design of the Proposed Model

In our model, for determining the price, we used a feed-forward neural network which contains three layers: input layer, hidden layer and output layer. The network we designed consists of five units in the input layer, one for each attribute mentioned in Section 1. The input layer also consists of one extra unit u_0 as the bias unit. We set the value of a_0 to the production cost of the product. Usually, sellers are not willing to sell their products below the production cost of the corresponding products. Hence, we considered the production cost of the product as the output of the bias unit. All the units accept numerical values as input. Initially, all the values a_1, a_2, a_3, a_4 and a_5 of the input units are set by the sellers. In setting the initial price of a product, we assume that sellers use their prior knowledge about the prices of the product offered by other competing sellers in the market.

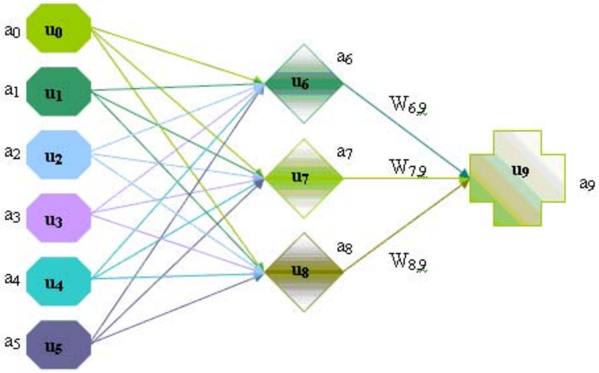


Fig. 1. A three-layered Feed-forward Neural Network for Price Determination

Our model can accept more attributes. One additional unit in the input layer needs to be added for each new attribute. On the other hand, in order to remove an attribute from the network the corresponding unit from the input layer, along with all the links that are connected to the unit, has to be eliminated. This implies that our model will work for any number of attributes.

4.1 Dynamic Pricing Algorithm

The price of the product determined by the network (Fig. 1) can be found by using final output a_9. The value of a_9 can be calculated with the aid of equation (1) as follows:

$$Finaloutput, a_9 = f\left(W_{6,9}a_6 + W_{7,9}a_7 + W_{8,9}a_8\right). \tag{2}$$

$$\begin{aligned} where, a_6 &= f\left(W_{0,6}a_0 + W_{1,6}a_1 + W_{2,6}a_2 + W_{3,6}a_3 + W_{4,6}a_4 + W_{5,6}a_5\right) \\ a_7 &= f\left(W_{0,7}a_0 + W_{1,7}a_1 + W_{2,7}a_2 + W_{3,7}a_3 + W_{4,7}a_4 + W_{5,7}a_5\right) \\ a_8 &= f\left(W_{0,8}a_0 + W_{1,8}a_1 + W_{2,8}a_2 + W_{3,8}a_3 + W_{4,8}a_4 + W_{5,8}a_5\right) \end{aligned}$$

We sub-divide the process of dynamic pricing by our model of neural network into two phases: training phase and price determination phase. In the training phase we train our network with a set of training pattern. A training pattern consists of a set of inputs with desired output. A typical set of training pattern for our model would look as Table 1. Each row of Table 1 represents a training pattern which contains a set of inputs with corresponding desired output. Initially we assume that the buyers have equal preference on all the five attributes that we are considering. Therefore, we associate each link between input units and hidden units with equal weights. The purpose of the training process is to adjust the weights between the links such that the errors are minimized. To obtain this goal we feed units of the input layer of our network with the corresponding input values ($Input_{ij}$) from each training pattern. We then determine the output from our network and compare it with the corresponding desired output ($Output_i$) of the training pattern to calculate error. Finally, we update weights between the links depending on the calculated errors. In our model, during the process of training, the errors between the links are minimized by using back-propagation technique. The training process can be portrayed by the following steps:

i Input values from a training pattern to units of the input layer of the network.
ii If the current training pattern is the first training pattern of the training set, then associate the links between input units and hidden units with equal weight, i.e., 0.2. Also, associate the links between hidden units and output unit equally, i.e., 0.33.
iii Determine the value from the output layer.
iv Compute the error, i.e., the difference between desired output of the training pattern and the value obtained in step *iii*.
v If the error is more than zero then go to step *viii*.
vi If the error is approximately zero and there is more training pattern left, then take the next training pattern and go to step *i*.
vii If the error is approximately zero and there is no more training pattern left, then terminate the training process.

Table 1. General set of training pattern of our model

Product Price	Product Quality	Delivery Time	Sellers' Reputation	After Sales Service	Desired Output
$Input_{11}$	$Input_{12}$	$Input_{13}$	$Input_{14}$	$Input_{15}$	$Output_1$
$Input_{21}$	$Input_{22}$	$Input_{23}$	$Input_{24}$	$Input_{25}$	$Output_2$
$Input_{31}$	$Input_{32}$	$Input_{33}$	$Input_{34}$	$Input_{35}$	$Output_3$

viii Update the weights of the links using back-propagation technique to minimize the error.
ix Go to step *iii*.

The training phase updates weights between the links of the network as needed so that it can provide better output. Once the training process is complete, our model of network is ready to determine a competitive price for a product, P, from the price determination phase as follows:

i Set the production cost of the product, P as the input to bias unit of the input layer and set the weights of the links associated with bias unit to 1.
ii Set the values (a_i) of the input units for the corresponding purchase attributes of product (as mentioned in Section 3.1) by using prior knowledge about the prices of product offered by other competing sellers.
iii Run the network and derive the price from the output layer.
iv Set the price from the output layer as the product price.

4.2 Error Minimization and Price Determination

The error is minimized at each iteration from step *iii* through step *ix* of training phase. Once the price of a specific product is determined from the output layer from step *iv* of price determination phase, the weights of the links remain unchanged. In step *ii* of price determination phase we assume that sellers use their prior knowledge of price offered by other sellers in the market. This indicates that our model keep an eye on the competitive price set by other sellers in the competing market.

In dynamic online economy, the price of products keeps on changing with the tick of clock. In order to sustain in the competitive online economy a seller needs to update his/her price in response to price fluctuation by other competing sellers in the market. While updating the price by using our model, we go through training and price determination phase of our model to recalculate the price. Before recalculating the price we analyze the revenue earned by using the selling price, P_r, for the product, P, that was generated from our model. If the revenue earned is greater than zero, then in step *ii* of training phase instead of taking 0.2 as the weight, $W_{i,j}$, between the input units and the hidden units, we use the weights, $W_{i,j}$, that were determined during the last iteration of the training phase at the time of determining P_r and go through the process again. For instance, assume that the value of $W_{1,6}$ was 0.38 when the product price was determined from step *iv* of price determination phase. In such scenario, we would like to set the value of $W_{1,6}$ to 0.38 instead of 0.2 in step *ii* of training phase and run the process again. Moreover, at the time of determining P_r we store the values of input units from step *i* and values of output unit from step *iv* of price determination phase as the historical data. We use this historical data as an additional training set during the training phase. On the other hand, if there was no revenue earned then the entire process is run by providing a new set of inputs in step *ii* of price determination phase.

5 Results and Analysis

We simulated our model in an e-commerce market place to examine if the model performs better than the simple pricing algorithm outlined in the following subsection. We also analyzed if a seller earns more revenue by employing our model instead of the Derivative-Following (DF) strategy proposed in [10]. In derivative following (DF) strategy, initially, product prices are set randomly and profitability is observed. The product prices are increased in the same direction unless the observed profitability falls. If the observed profitability falls then product prices are decreased as long as profit is encountered. It requires keeping track of past average profit of each state, and increases the prices till the profitability level falls [10].

5.1 Simple Pricing Algorithm

A seller, by taking five attributes of our model into consideration, can employ a simple pricing algorithm to determine a competitive price of products. A simple pricing algorithm may take at least production cost of a product as initial selling price of the product. If a buyer prefers to enjoy any additional attributes such as after sale service of the product, then the algorithm may wish to add some additional price for each supplementary attributes. Finally, the algorithm would provide a selling price of the products. Since in online economy prices of products do not remain static, a seller has to update his/her offered price of the products. While updating the prices, there can arrive two different scenarios for a seller who employs the simple pricing algorithm. First, the algorithm, while updating the price by adding extra price for additional attributes, does not use any information of how other competing sellers in the market set their selling price. Since the algorithm has no knowledge about market parameters, it uses some random extra prices for additional attributes. Hence, the price can be too low or too high which may lead to earn inadequate revenue for a seller. In the second scenario let us assume that a seller who employed the simple algorithm, do some manual search on the prices offered by other vendors. The algorithm uses information obtained from the seller's search to determine a selling price for products. However, the manual search could be time consuming. It might take hours to days or even longer to gather information by manual search. Since prices change within very short span of time in online market, the information acquired by manual search during relatively large span of time might become outdated. Consequently, the algorithm would be using obsolete information which may lead to generate inappropriate output. In contrary, our model outputs a competitive selling price of products by providing importance to five attributes based on historical data, which implies it does not rely on any manual search. In addition, our technique, while determining competitive selling price, considers the sellers make use of their prior knowledge of the prices set by other competing sellers in providing initial input to the model. This indicates that our model keep an eye on the competitive price set by other sellers in competing market and utilizes fresh information of price.

5.2 Train Network

We assume that sellers use their historical data as the training patterns to the network. A training pattern consists of a set of inputs with desired output. We began our simulation by training the network of our model with 10 sets of training patterns so that errors can be minimized as much as possible by using back propagation algorithm. We trained our network for nine different numbers of epochs or iterations: 10, 50, 100, 500, 1000, 5000, 10000, 50000 and 100000. As the training continues, after each iteration or epoch, the network calculates amount of error. The calculated error is then used to update the weights of the links by using back propagation algorithm so that error is minimized in the next iteration. Practically the value of error never becomes zero, but approaches to zero. We let our network to tolerate an error of amount 0.01 and 0.001. We run our network with five different learning rates: 0.01, 0.005, 0.001, 0.0005 and 0.0001. Analysis of the training process in the following sub-section indicates that the model performs better if we use 50000 epochs with 0.005 learning rate during training the network.

5.3 Determine Training Parameters

The purpose of the model is to generate a competitive price for a product with respect to the price offered by other competing sellers in the market. The more number of training patterns are used to train the network in training phase, the better knowledge the model will contain. This would lead to generate a more competitive price. Hence, the performance of the model depends on the training phase. Besides number of training patterns used, the training process is dependent on three parameters: (i) number of epochs used, (ii) Learning rate and (iii) Error tolerance. Use of proper values for these parameters plays a vital role in the model's performance. We trained our model by using different values (as mentioned previous subsection) for these parameters with 10 sets of training patterns. Once training process is complete, our model is ready to use to determine a selling price for a product. We used our trained network to determine the price of a product (lets call it P). We then derive suitable values for the parameters by analyzing performance of the model through investigating the experimental results shown later in this section. Since our model requires initial selling price of the product to be set by the seller, we used the prices of Table 2 as initial selling price based on five different attributes of the product. We assumed the production cost of P is 645.00. According to our model, the output produced from the output layer of our model is considered as the selling price, P_r, at which a sellers, S, would be selling P. If there is M out of N buyers in the market willing to buy P at a cost of P_r, then the revenue earned by S can be calculated from the product of M and P_r.

While training our model, we found that the amount of revenue earned per product after selling it to a single buyer is closely identical to each other for different learning rates when lower number of epochs is used. Amount of revenue earned is increased gradually with higher number of epochs used. Another finding

Table 2. Initial Selling Price of Product P

Product Price	645.00
Product Quality	648.99
Delivery Time	745.99
After Sale Service	805.99
Sellers Reputation	718.99

was that that the model performs better if 0.01 is chosen as learning rate. The elapsed time for training our model increases gradually with increasing number of epochs. However, there is a rapid increase in elapsed time after 100,000 epochs. Therefore, we would not like to use more than 100,000 epochs during simulating an e-commerce market place in the following subsection. While training our model, we let our model to accept an error tolerance of 0.01 and 0.001. From 5000 number of epochs onwards the model delivers similar output for error tolerance of 0.01 and 0.001.

Under the above circumstances of analysis, for training our model during simulating an e-commerce market place in the following subsection we would like to use 100,000 epochs, 0.01 learning rate and 0.01 error tolerance.

5.4 Results

We simulated our model in an e-commerce market place to evaluate performance of our model. We consider a market place where three sellers (namely, $Seller_{simple}$, $Seller_{DF}$ and $Seller_{om}$) wish to sell a product P to 200 different buyers with five distinct preferable purchase attributes of products. $Seller_{simple}$ employs a **simple** pricing algorithm described in Section 5.1. $Seller_{DF}$ uses **d**erivative-**f**ollowing (**DF**) strategies and $Seller_{om}$ follows **o**ur **m**odel.

We run the market for ten rounds, with twenty buyers in each round. After each round we calculate the revenue earned by each seller. We then compare it with the revenue earned by the corresponding seller in previous round to determine the direction (positive or negative) of revenue earned. At the end of each round, we allow the sellers to update their selling prices. In DF strategies, the price of product is updated, by some amount, in the direction of revenue earned. We consider that $Seller_{DF}$ updates his/her selling price by a random amount between 0.00% and 5.00% of the current selling price. On the other hand, $Seller_{simple}$ updates the selling price either by using direction of revenue earned or by using information of prices set by other two sellers in the market during one of the previous rounds. We made an assumption that simple pricing algorithm performs manual search, which is time consuming, to gather information regarding other sellers' selling price. Therefore, the information may not be available to $Seller_{simple}$ at the end of each round. In addition, we assume that if the information is available, then due to manual slow searching process the information of the immediate previous round is not available to $Seller_{simple}$. For simplicity, we consider that if the information is available, then $Seller_{simple}$ updates the selling price of P by using average value of the prices set by other two sellers during the (r-2)th round, where r is the current round. In contrary, if the information is not available, $Seller_{simple}$ uses direction of revenue to update the price. We used a randomly-generated probability to determine if the information is available to $Seller_{simple}$. $Seller_{om}$ always uses our model to update the price. For simplicity we assume that in all ten rounds of the market there are equal numbers of buyers (four out of twenty) preferred each given five attributes.

We begin our simulation of the market by assuming that at the beginning of the first round $Seller_{simple}$ and $Seller_{DF}$ use data from Table 2 to set their selling price of the product P whose production cost is 645.00. $Seller_{om}$ uses information from Table 2 with 100,000 epochs, 0.01 learning rate and 0.01 error tolerance to generate a selling price (650.487) for P. Figure 2 summarizes the total revenue earned at the end of each round by the three different sellers who employed three distinct pricing algorithms.

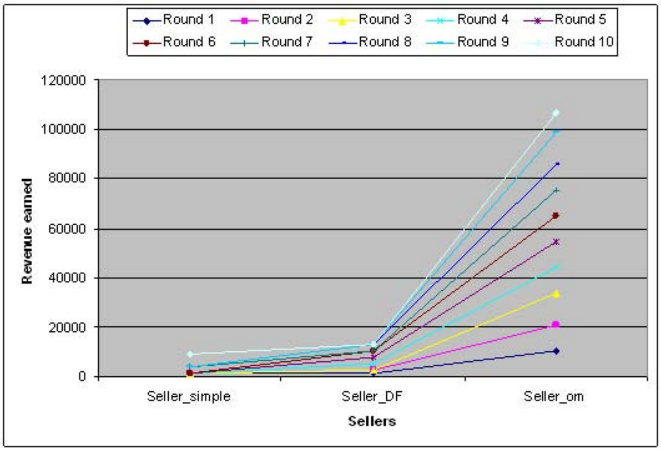


Fig. 2. Total revenue earned by three sellers

Initially, all the sellers managed to earn some revenue. Among the three sellers, the growth of revenue earned by $Seller_{simple}$ was the slowest. $Seller_{simple}$ failed to earn any revenue at the end of most of the rounds. Apart from first round, $Seller_{simple}$ earned some revenue after the end of seventh and tenth round. The performace of $Seller_{DF}$ in terms of earning revenue was better than $Seller_{simple}$, however, he/she could not beat $Seller_{om}$ in any of the rounds. On the contrary, $Seller_{om}$, who employed our model, earned revenue at each round. Moreover, after each round, $Seller_{om}$ earned higher revenue than that of revenue earned by other two sellers. At the end of tenth round, $Seller_{DF}$ earned nearly 43% more revenue than $Seller_{simple}$. On the other hand, $Seller_{om}$ earned nearly eight times more revenue than $Seller_{DF}$.

6 Discussion

The experimental results show that our model can attract more buyers compared to other two buyers, because we have considered multiple attributes in determining a selling price for the product P. Attracting more buyers from wider range of preferred attributes implies that more revenue can be earned. Various pricing algorithms are followed in present online economy. Among them game-theoric pricing (GT), myoptimal pricing (MY), derivative following (DF), and Q-learning (Q) are practiced widely. Game-theoretic (GT) strategy makes an assumption that all other competing sellers use game-theoretic [4]. However, in present world different sellers employ distinct pricing strategies. GT uses complete information regarding buyer population. Moreover, it does not use any historical data. In contrast, historical data plays an important role in understanding changing behavior of the market. We used little historical data of the price offered by other competitive sellers. In addition, our model is not concerned about what strategies are being used by other competing sellers. Similarly, My-optimal (MY) strategy does not dependant on whether other sellers employing different pricing strategies or not, but it is concerned about buyers demand curve and also the prices set by other sellers in the economy. MY also assumes that prices set by other competing sellers will remain unchanged [4]. On the contrary, sellers are always willing to change their offered price for the sake of sustaining in competing market. Hence, our model always keeps an eye on the random prices set by other sellers. Q-learning strategy is based on reinforcement learning and makes use of both buyers' demand curve and knowledge about competitors pricing strategies. On the other hand, our model does not rely on buyer demand curve. In short, we attempt to address the problem of dynamic pricing in a competitive online economy, where a buyer's purchase decision is determined by multiple attributes. By taking multiple purchase attributes into account we can attract more number of buyers which lead to earning more revenue.

7 Conclusion and Future Work

The proposed approach described here considered multiple purchase attributes to determing product price dynamically by using feed-forward neural network. We simulated an e-commerce market place with 200 buyers, three sellers where all the sellers trying to sell a product P. The experimental results showed that the seller employing our model earned higher revenue than that of earned by other two sellers who followed simple pricing algorithm and derivative-following (DF) strategies. We would like to compare our approach with other existing well known approach of dynamic pricing, like game-theoretic (GT), my-optimal (MY) etc. Our model made an assumption that sellers have limited prior knowledge about market parameters in setting the initial price of the products. We would like to eliminate the assumption from our model by employing a web crawler tool in our application in order to learn the information on prices set by other competitive sellers in the market. Using this information we may set the initial price of the products such that the sellers no need to initialize the product prices while using our model.

References

1. Kephart, J., Brooks, C., Das, R.: Pricing information bundles in a dynamic environment. In: ACM Conference on Electronic Commerce 2001, pp. 180–190 (2001)
2. Greenwald, A., Kephart, J.: Probabilistic pricebots. Agents, 560–567 (2001)
3. Tesauro, G., Kephart, J.: Foresight-based pricing algorithms in agent economies. Decision Support Systems 28(1-2), 49–60 (2000)
4. Greenwald, A., Kephart, J., Tesauro, G.: Strategic pricebot dynamics. In: ACM Conference on Electronic Commerce 1999, pp. 58–67 (1999)
5. DiMicco, J., Greenwald, A., Maes, P.: Dynamic pricing strategies under a finite time horizon. In: ACM Conference on Electronic Commerce, pp. 95–104 (2001)
6. Bar-Isaac, H., Tadelis, S.: Seller Reputation. Foundations and Trends in Microeconomics 4(4), 273–351 (2008)
7. Chen, Y., Tsao, C., Lin, C., Hsu, I.: A Conjoint Study of the Relationship between Website Attributes and Consumer Purchase Intentions. In: Pacific Asia Conference on Information Systems, PACIS (2008)
8. Dasgupta, P., Hashimoto, Y.: Multi-attribute dynamic pricing for online markets using intelligent agents. In: AAMAS (2004)
9. Russell, S.J., Norvig, P.: Artificial Intelligence: A modern Approach, 2nd edn. Prentice-Hall, Englewood Cliffs (2005)
10. Chinthalapati, V., Yadati, N., Karumanchi, R.: Learning Dynamic Prices in MultiSeller Electronic Retail Markets With Price Sensitive Customers, Stochastic Demands, and Inventory Replenishments. IEEE, Los Alamitos (2006)
11. Dasgupta, P., Das, R.: Dynamic Service Pricing for Brokers in a Multi-Agent Economy. IEEE, Los Alamitos (2000)
12. Kong, D.: One Dynamic Pricing Strategy in Agent Economy Using Neural Network Based on Online Learning. In: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (2004)
13. Luo, L., Xiao, B., Deng, J.: Dynamic pricing decision analysis for parallel flights in competitive markets, pp. 323–327. IEEE, Los Alamitos (2005)
14. Carvalho, A., Puterman, M.: Dynamic pricing and reinforcement learning. IEEE, Los Alamitos (2003)
15. Dasgupta, P., Moser, L., Melliar-Smith, P.: Dynamic Pricing for Time-Limited Goods in a Supplier-Driven Electronic Marketplace. Electronic commerce research (2005)
16. Gallego, G., Ryzin, G.: Optimal dynamic pricing of inventories with stochastic demand over finite horizons. Manage. Sci. 40(8), 999–1020 (1994)
17. Li, C., Wang, H., Zhang, Y.: Dynamic pricing decision in a duopolistic retailing market. In: Proceedings of the 6th World Congress on Intelligent Control and Automation, Dalian, China (June 2006)

The IMAP Hybrid Method for Learning Gaussian Bayes Nets

Oliver Schulte[1], Gustavo Frigo[1], Russell Greiner[2], and Hassan Khosravi[1]

[1] School of Computing Science, Simon Fraser University,
Burnaby, B.C., Canada V5A 1S6
{oschulte,gafrigo,hkhosrav}@cs.sfu.ca
[2] Department of Computing Science, University of Alberta,
Edmonton, Alberta Canada T6G 2E1
greiner@cs.ualberta.ca

Abstract. This paper presents the I-map hybrid algorithm for selecting, given a data sample, a linear Gaussian model whose structure is a directed graph. The algorithm performs a local search for a model that meets the following criteria: (1) The Markov blankets in the model should be consistent with dependency information from statistical tests. (2) Minimize the number of edges subject to the first constraint. (3) Maximize a given score function subject to the first two constraints. Our local search is based on Graph Equivalence Search (GES); we also apply the recently developed SIN statistical testing strategy to help avoid local minima. Simulation studies with GES search and the BIC score provide evidence that for nets with 10 or more variables, the hybrid method selects simpler graphs whose structure is closer to the target graph.

1 Introduction

Bayes nets [18] are a widely used formalism for representing and reasoning with uncertain knowledge. A Bayes net (BN) model is a directed acyclic graph (DAG) $G = \langle \mathbf{V}, \mathbf{E} \rangle$ whose nodes $\mathbf{V}$ represent random variables and whose edges $\mathbf{E}$ represent statistical dependencies, together with conditional probability tables that specify the distribution of a child variable given each instantiation of its parents. In this paper we consider Gaussian Bayes networks with the following properties: (1) all variables are continuous, (2) a child variable is a linear function of its parent variables plus a Gaussian error term, (3) all error terms are independent.

There are two well established general approaches to learning a BN structure. Constraint-based (CB) methods employ a statistical test to detect conditional (in)dependencies given a sample d, and then compute a BN G that fits the (in)dependencies [23]. Score-based methods search for models that maximize a model selection score [13]. Hybrid methods aim to combine the strengths of both approaches [24, 8, 12]. Evaluations have shown that for DAGs with *discrete* variables, the best hybrid methods outperform both purely score-based and purely constraint-based methods [24]. We introduce a new hybrid model selection criterion and develop a novel search strategy for the criterion that integrates

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 123–134, 2010.

statistical tests and score functions in the context of continuous variables. Our new criterion combines constraints and score functions as follows: (1) A DAG G should satisfy the *Markov boundary condition*, meaning that for any two nodes X and Y, no statistically significant correlation is found between X and Y given the neighbors and spouses (co-parents) of X. (2) The model G should have the minimum number of edges among the graphs that satisfy the boundary condition. (3) Among the minimum-edge graphs satisfying the boundary condition, our criterion selects the ones that maximize a given scoring function.

There are theoretical, statistical and computational motivations for this composite selection criterion. It is well-known in Bayes net theory that a BN model that represents the target or operating distribution generating the data must satisfy the Markov boundary condition. It is widely accepted that a graphical model G of the target distribution should be edge-minimal, meaning that no subgraph of G represents the target distribution [18, Ch.3.3], [17, Ch.2.4]. Minimizing the number of edges implies edge-minimality. Schulte et al. provide a learning-theoretic justification for minimizing the number of edges as a small-sample selection criterion [22]. *Statistical motivation* is provided by the observation that standard model selection criteria like the Bayes Information Criterion (BIC; [17, Ch.8.3.2]) tend to favor overly complex models when applied to linear models [19]. Our simulations provide further empirical evidence to support this finding. Our composite criterion addresses overfitting by assigning higher priority to minimizing the number of edges rather than to maximizing the score. Thus the criterion favors adding an edge only if this is necessary for representing a statistically significant correlation found in the data, even if adding the edge improves the model selection score. A *computational motivation* for adding the model selection score is that the problem of finding minimum-edge graphs consistent with a set of given dependencies is NP-hard [4, Lm. 4.5]; the score serves as a heuristic for exploring the search space.

For experimental evaluation, we adapted the state-of-the-art Graph Equivalence Search (GES) procedure [16, 5]. We report a number of measurements comparing GES and our constrained GES, based on the well-established BIC score function. Simulation results for both randomly generated and real-world target BN structures compare the graphs learned with and without (in)dependency constraints to the target graph. For graphs with 10 nodes and greater, we observe that BIC significantly overfits the data in the sense that it produces graphs with too many adjacencies. Our simulations illustrate how adding (in)dependency constraints corrects some of this overfitting tendency of the BIC score function. The constrained search produces simpler models (i.e., with fewer adjacencies) whose structure is closer to the target graph, as measured by the number of correctly/incorrectly placed edges. Our source code is available for anonymous ftp access at ftp://ftp.fas.sfu.ca/pub/cs/oschulte/imap/.

Paper Organization. The next section reviews basic notions from Bayes net theory. Section 3 discusses the major design choices in our system, including our adaptation of GES search. It provides a proof of consistency (asymptotic correctness) for our hybrid search procedure. Section 4 presents simulation studies that compare constrained GES search with the BIC score to regular GES search with the same score.

Related Work. *Score-based Methods.* A number of score functions are widely used in structural equation modelling, such as AIC and model chi-square [14]. We focused our study on the BIC information criterion, for several reasons. (1) BIC is one of the best established in the SEM literature. (2) BIC is widely used for evaluating Bayes nets in computer science studies [8, 25]; it is the default score for Gaussian models in CMU's Tetrad system [6]. (3) Other standard criteria like AIC penalize complex structures less than BIC so the tendency of BIC towards complex models corrected by our algorithm is even stronger with these criteria.

Hybrid Methods. Tsamardinos et al. [24] recently presented a hybrid method (max-min hill climbing) for discrete variables that treats the tests of statistical outcomes as constraints. While this work indicates that independence constraints from a statistical test can improve a score-based search, Hay et al. [12] show that because it accepts independence null hypotheses, max-min hill climbing is sensitive to type II errors. This paper extends our earlier work [21] as it treats only dependencies (rejections of the null hypothesis) as "hard" constraints. However, the previous algorithm addressed the problem of *underfitting* in score-based BN learning with discrete variables, whereas the problem in BN learning in Gaussian models is overfitting. Therefore the previous method adds more adjacencies than regular score-based search, whereas the method of this paper adds fewer adjacencies. Other previous hybrid BN learning algorithms (e.g., [8, 11]) consider statistical measures (*e.g.*, mutual information), but do not incorporate the outcome of a statistical test as a constraint that the learned model must satisfy. To our knowledge, the hybrid methods whose description and evaluation have been published to date, deal with discrete variables rather than continuous ones.

2 Basic Definitions

The definition and theorems cited in this section are standard; for further details see [17, 18, 23]. We consider Bayes nets for a set of random variables $\mathbf{V} = \{X_1, \ldots, X_n\}$ where each X_i is real-valued. A **Bayes net structure** $G = \langle \mathbf{V}, \mathbf{E} \rangle$ for a set of variables $\mathbf{V}$ is a directed acyclic graph (DAG) over node set $\mathbf{V}$. A Bayes net (BN) is a pair $\langle G, \theta_G \rangle$ where θ_G is a set of parameter values that specify the probability distributions of each variable conditioned on instantiations of its parents. A BN $\langle G, \theta_G \rangle$ defines a p.d.f over $\mathbf{V}$. In a linear Gaussian BN, each child Y is a linear function of its parents $X_1, ..., X_k$ so $Y = \sum_{i=1}^{k} a_i X_i + \varepsilon_Y$, where the error term ε_Y has a normal distribution with mean 0 and variance σ_Y^2. The variance of ε_Y and the coefficients a_i are parameters of the model. The mean and variance of each root node are further parameters of the model. We make the standard assumption that the error terms for different variables are uncorrelated. The BIC score is defined as $BIC(G, \mathsf{d}) = L(\hat{G}, \mathsf{d}) - par(G) \cdot ln(m)/2$

where $\hat{G} = \hat{G}(d)$ is the BN G with its parameters instantiated to be the maximum likelihood estimates given the sample d, the quantity $L(\hat{G}, \mathsf{d})$ is the log-likelihood of $\hat{G}$ on the sample d, the sample size is denoted by m, and $par(G)$ is the number of free parameters in the structure G.

Two nodes X, Y are **adjacent** in a BN if G contains an edge $X \rightarrow Y$ or $Y \rightarrow X$; an adjacency is a pair of adjacent nodes. An **unshielded collider** in G is a triple of nodes connected as $X \rightarrow Y \leftarrow Z$, where X and Z are not adjacent. The **pattern** $\pi(G)$ of DAG G is the partially directed graph over $\mathbf{V}$ that has the same adjacencies as G, and contains an arrowhead $X \rightarrow Y$ if and only if G contains an unshielded collider $X \rightarrow Y \leftarrow Z$. We assume familiarity with the notion of d-separation [18]. We write $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{S})_G$ to denote that two disjoint sets $\mathbf{X}$ and $\mathbf{Y}$ of vertices are d-separated by a third set $\mathbf{S}$ in G. If two sets $\mathbf{X}$ and $\mathbf{Y}$ are not d-separated by $\mathbf{S}$ in graph G, then $\mathbf{X}$ and $\mathbf{Y}$ are **d-connected** by $\mathbf{S}$ in G, written $(\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y}|\mathbf{S})_G$. We write $\mathcal{D}(G)$ for the set of all d-connections $(\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y}|\mathbf{S})_G$ or conditional dependencies that hold in a graph G. Two DAGs G and G' satisfy exactly the same dependencies iff they have the same patterns (*i.e.*, $\mathcal{D}(G) = \mathcal{D}(G')$ iff $\pi(G) = \pi(G')$ [17, Th.2.4]). We take the set of dependencies associated with a pattern π to be the set of dependencies in any DAG G whose pattern is π. For a node X, we refer to the set of its parents, children and co-parents (*i.e.*, other parents of its children) as the **Markov blanket** of X in G, written $MB_G(X)$. Given its Markov blanket $MB(X)$, each node X is d-separated from all other nodes outside of the Markov blanket.

Let ρ be a joint probability density function (p.d.f.) for variables $\mathbf{V}$. If $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$ are three disjoint sets of variables, then $\mathbf{X}$ and $\mathbf{Y}$ are **stochastically independent given S**, denoted by $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{S})_\rho$, if $\rho(\mathbf{X}, \mathbf{Y}|\mathbf{S}) = \rho(\mathbf{X}|\mathbf{S})\, \rho(\mathbf{Y}|\mathbf{S})$ whenever $\rho(\mathbf{S}) > 0$. A BN structure G is an **I-map** of p.d.f. ρ if for any three disjoint sets of variables $\mathbf{X}, \mathbf{Y}$ and $\mathbf{Z}$ we have $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{S})_G$ implies $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{S})_\rho$. For a given BN structure G and joint density function ρ, there is a parametrization θ_G such that ρ is the joint density for $\mathbf{V}$ defined by $\langle G, \theta \rangle$ only if G is an I-map of P. As the characteristic feature of our approach is searching for a graph that satisfies this condition, we refer to it as "I-map learning". The next section describes an implementation of I-map learning.

3 Algorithm Design for I-Map Learning

We first discuss employing statistical tests for detecting conditional (in) dependencies, then integrating statistical testing with a score-based local search.

Use of Statistical Tests. I-map learning requires a statistical significance test for testing conditional independence hypotheses of the form $X \perp\!\!\!\perp Y|\mathbf{S}$. Our system architecture is modular, so the test can be chosen to suit the type of available data and application domain. We followed other CB methods and used Fisher's z-statistic for testing whether a given partial correlation is 0 [23, Ch.5.5]. For a given pattern graph G, say that node Y is a *proper spouse* of node X if X and Y have a common child but are not adjacent. The set of *nonchildren* of X and Y are the nodes that are adjacent to X or Y but not children of either; denote this set by $NC_G(X, Y)$. (In a completely directed graph, these are just the parents of X and Y; our definition applies to partially directed patterns as well.) Our basic test selection strategy applies the chosen significance test to the following independence hypotheses, for each ordered pair of nodes (X, Y).

1. The **Markov blanket independencies** $\{X \perp\!\!\!\perp Y | MB_G(X) : Y \notin MB_G(X)\}$.
2. The **spousal independencies** $\{X \perp\!\!\!\perp Y | NC_G(X) : Y \text{ is a proper spouse of } X\}$.

These independence tests are well-suited for pattern-based search since the Markov blanket, adjacencies, and common children are determined by the pattern alone. The spousal independencies distinguish nodes on the Markov blanket that are both neighbors and spouses from nodes that are spouses only. If a graph entails a Markov blanket hypothesis (resp, spousal independency hypothesis), but a suitable test rejects the independency hypothesis, this is evidence that the graph is not correct. I-map learning implements the Markov blanket testing strategy through a procedure `find-new-dependencies(`G`)` that takes as input a new graph G adopted during the local search, tests the new Markov blanket and spousal hypotheses for the graph G, and returns the set of rejected independence hypotheses. Every time the local search moves to a new graph structure G, the procedure `find-new-dependencies` is applied to G to augment the cache of observed dependency constraints (cf. [21]). The procedure `find-new-dependencies` tests a set of independence hypotheses, so issues of multiple hypothesis testing arise. Any multiple hypothesis testing method can be employed to implement the functionality of `find-new-dependencies` [2, 9]. Like many other constraint-based and hybrid systems, we simply carry out multiple hypotheses at the same fixed significance level [23, 8, 15]. At an intermediate stage, our method also integrates one of the most recent CB algorithms, the "condition on nothing and everything else" strategy of SIN graphical model selection [9]: For any two variables X and Y, test (1) the unconditional correlation betwen X and Y and (2) the correlation conditional on all other variables.

Heuristic Search Algorithm for I-map Learning. For our simulations we adapt the state-of-the art GES (Greedy Equivalence Search) local search algorithm. We describe GES only in sufficient detail to indicate how we adapt it. During its growth phase, GES moves from a current candidate pattern π to the highest-scoring pattern π' in the upper neighborhood $\mathsf{nbdh}^+(\pi)$. A pattern π' in $\mathsf{nbdh}^+(\pi)$ contains exactly one more adjacency than π, and may have arrows reversed, subject to several conditions that ensure that $\mathcal{D}(\pi) \subset \mathcal{D}(\pi')$, i.e., π' entails a strict superset of the dependencies entailed by π. The growth phase terminates with a pattern π when no graph in $\mathsf{nbdh}^+(\pi)$ has higher score than π. During the subsequent shrink phase, GES moves from a current candidate pattern π to the highest-scoring pattern π' in the lower neighborhood $\mathsf{nbdh}^-(\pi)$. A pattern π' in $\mathsf{nbdh}^-(\pi)$ contains exactly one less adjacency than π, and may have arrows reversed, subject to several conditions that ensure that $\mathcal{D}(\pi') \subset \mathcal{D}(\pi)$, i.e., π' entails is a strict subset of the dependencies entailed by π. GES terminates with a pattern π when no graph in $\mathsf{nbdh}^-(\pi)$ has higher score than π. The constrained version IGES (for I-map + GES) constrains the GES neighborhoods so they satisfy a given set of observed dependencies. Formally, the *growth*

Algorithm 1. The IGES procedure adapts GES based on the neighborhood structures nbdh^{+} and nbdh^{-} to perform constrained score optimization with a statistical testing method

```
   Input: data sample d for random variables $\mathbf{V}$.
   Calls: score evaluation function score($\pi$, d), statistical testing procedure
   find-new-dependencies($\pi$, d).
   Output: BN pattern constrained by (in)dependencies detected in the data.
 1: initialize with the disconnected pattern $\pi$ over $\mathbf{V}$, and the empty dependency set
    $\mathcal{D}$.
 2: for all Variables $X, Y$ do
 3:    test the hypothesis $X \perp\!\!\!\perp Y$ on sample d
 4:    if $X \perp\!\!\!\perp Y$ is rejected by statistical test, add to detected dependencies stored in
       $\mathcal{D}$
 5: end for
 6: {begin growth phase}
 7: while there is a pattern $\pi'$ in $\mathsf{nbdh}^{+}_{\mathcal{D}}(\pi, \mathcal{D})$ do
 8:    choose $\pi'$ in $\mathsf{nbdh}^{+}_{\mathcal{D}}(\pi, \mathcal{D})$ with maximum score
 9:    $\mathcal{D} := \mathcal{D} \cup$ find-new-dependencies($\pi'$, d)
10: end while
11: {begin shrink phase}
12: while there is a pattern $\pi'$ in $\mathsf{nbdh}^{-}_{\mathcal{D}}(\pi, \mathcal{D})$ with greater score than current pattern
    $\pi$ do
13:    choose $\pi'$ in $\mathsf{nbdh}^{-}_{\mathcal{D}}(\pi, \mathcal{D})$ with maximum score
14: end while
15: {prune pattern $\pi$ further with "nothing and everything else" SIN tests}
16: for any two variables $X$ and $Y$ that are adjacent in $\pi$,
    if $X \perp\!\!\!\perp Y$ or $X \perp\!\!\!\perp Y | \mathbf{V} - \{X, Y\}$ are not rejected by the statistical test,
    remove the link between $X$ and $Y$.
17: repeat growth phase and shrink phase once (lines 6-14).
18: Return the current pattern $\pi$.
```

neighborhood constrained by dependencies $\mathcal{D}$ is defined as follows:

$$\pi' \in \mathsf{nbdh}^{+}_{\mathcal{D}}(\pi) \quad \text{if and only if} \quad \pi' \in \mathsf{nbdh}^{+}(\pi) \text{ and } (\mathcal{D}(\pi') \cap \mathcal{D}) \supset (\mathcal{D}(\pi) \cap \mathcal{D}).$$

The growth phase keeps expanding a candidate structure to entail more of the observed dependencies $\mathcal{D}$, and terminates when all observed dependencies are covered. To check if a graph expansion covers strictly more dependencies, we keep a cache of dependencies that have not yet been covered during the growth phase, and go through these dependencies in order to see if any of them are covered by a candidate graph. The *shrink neighborhood constrained by dependencies* $\mathcal{D}$ is defined as follows:

$$\pi' \in \mathsf{nbdh}^{-}_{\mathcal{D}}(\pi) \text{ if and only if } \pi' \in \mathsf{nbdh}^{-}(\pi) \text{ and } (\mathcal{D}(\pi') \cap \mathcal{D}) \supseteq (\mathcal{D}(\pi) \cap \mathcal{D}).$$

The shrink phase moves to higher-scoring patterns in the GES lower neighborhood, subject to the constraint of fitting the observed dependencies, until a local score maximum is reached. Algorithm 1 gives pseudocode for IGES search.

Analysis of Search Procedure. A score function is *consistent* if, as the sample size increases indefinitely, with probability 1 all graphs that maximize the score are I-maps of the target distribution. The score function is *decomposable* if the score of a graph can be computed from scores for each node given its parents. The standard analysis of CB methods assumes the correctness of the statistical tests, which holds in the sample size limit [7, 23]. Under these assumptions, our local search method is consistent. The proof is available at [20].

Proposition 1. *Suppose that the statistical test returns only valid dependencies in target graph G during an execution of Algorithm 1 (with or without SIN testing), and that the score function is consistent and decomposable. Then as the sample size increases indefinitely, with probability 1, the algorithm terminates with an I-map π of the target distribution defined by G.*

Number of Statistical Calls. The *computational overhead* compared to regular local score optimization is the number of statistical calls. For a graph G with n nodes, the number of Markov blanket independence hypotheses is on the order of $O(\binom{n}{2})$—two tests for each pair of nodes X, Y that are not in each other's Markov blanket. By taking advantage of the structure of the local search procedure, we can often reduce the set of hypotheses to be tested to an equivalent but smaller set. For example, if the local search adds a single edge $X \rightarrow Y$ to a graph G, the only nodes whose Markov blanket has been affected are X, Y and the parents of Y. Assuming that the target graph has constant degree (cf. [23, Ch.5.4.2.1]), only a linear number of new independence tests is required at each stage of the search. Thus we expect that in practice, the order of independence tests required will be $O(n \times ca)$ where ca is the total number of candidate structures examined during the local search. Our simulations provide evidence for this hypothesis (Section 4).

4 Empirical Evaluation of Hybrid Criterion with Standard Search+Score Method

We performed a large number of simulations, and summarize the main findings. More details are available in an extended version [20]. Our code is written in Java and uses many of the tools in the Tetrad package [6]. The following learning methods were applied with the BIC score function.

1. Score-based search: GES starting with the empty graph.
2. Constraint-based search: PC algorithm [23] with z test and significance level $\alpha = 5\%$.
3. Backward Selection [10]: start with the complete DAG with all edges, apply the shrink phase of GES search.
4. Hybrid search method: IGES + SIN search with z test and significance level $\alpha = 5\%$. We also refer to this as the I-map pruned method.

Experiments with Synthetic Data. The target models considered were randomly generated networks with 5-20 variables. We used Tetrad's random DAG generating functions to build the networks [6] as follows. (1) A parent and a child are chosen at random and the corresponding edge is added to the random graph unless it causes a cycle in the resulting graph. The number of edges is also determined randomly, with the constraint that there are at most twice as many edges as nodes. (2) Linear coefficients are drawn uniformly from the union of the intervals $(-0.5, -1.5)$ and $(0.5, 1.5)$. Variance parameters are drawn uniformly from the interval $(1.0, 3.0)$. Means are drawn from a standard normal distribution with mean 0 and variance 1. For each graph, we drew samples of various sizes (ranging from 100 to 20,000). We repeated the simulation 30 times, resulting in 30 random graphs for each combination of sample size and node count. Our graphs and tables display the average of the 30 networks for all measurements.

Model Complexity and Structure. Our key findings are graphed in Figures 1 and 2. Figure 1 shows that the hybrid criterion together with the SIN tests effectively reduces the overfitting tendency of the regular score-based criterion, as measured by the number of edges in the learned model versus the number in the true graph. Without the SIN tests, the improvement is not as great. We measured the quality of the graph structure by combining adjacencies in the target structure (true positive) vs. adding adjacencies not present in the target structure (false positive) using the F-measure from information retrieval [26, p.146], which is defined as

$$\frac{2(\text{True Positive})}{2(\text{True Positive}) + (\text{False Positive}) + (\text{False Negative})}.$$

Higher F-measures are better. In general, the GES search produces more false positives than IGES search and fewer false negatives, as our edge-ratio measurements confirm. Figure 2 shows that the adjacency F-measure for the hybrid criterion is slightly worse for graphs with less than 10 nodes. This is because the overfitting tendency of the BIC score is small for small graphs, as our edge-ratio measurements confirm, so the overall balance of false positives and false negative is slightly favorable for unconstrained GES search. As the graph size increases, so does the number of false positives relative to graph size in GES, which means that the F-measure balance becomes favorable for the hybrid criterion.

Performance of Statistical Testing Strategy. A number of measurements concern the behavior of the testing strategy. A standard measure for the performance of a multiple hypothesis testing method is the *false discovery rate* (FDR) [2], which is defined as #rejected true independence hypotheses/#tested independence hypotheses. For the SIN independence hypotheses we also measured the *false acceptance rate* (FAR), defined as #false accepted independence hypotheses/#tested independence hypotheses. In our simulations, with the significance level fixed at $\alpha = 5\%$, the FDR in random graphs was on average no greater than α, which is a good result in light of the Bonferroni inequality. In fact, for most experimental constellations the FDR was below 1.5%; it peaks at 3.5%

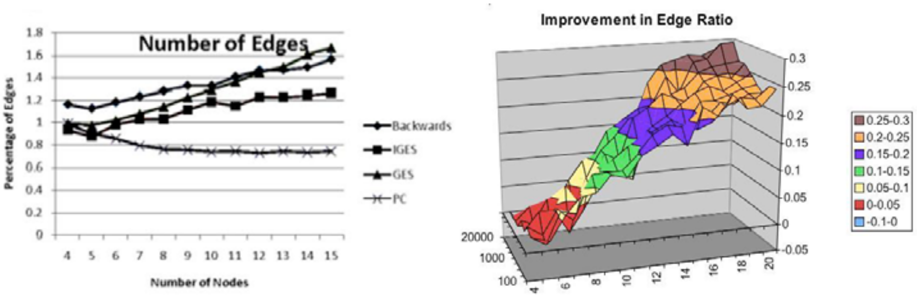


Fig. 1. Left: The figure shows the distribution of the edge ratio for the comparison methods, defined as #edges in target graph/#edges in learned graph. A ratio of 1 is ideal. The x-axis indicates the number of nodes, the y-axis the average edge ratio over all sample sizes for the given graph size (30 graphs per sample size and number of nodes). The average edge ratio for IGES+SIN is closer to 1 than for GES, which has a clear tendency towards more complex models. The improvement increases with sample size and network size. Right: The improvement of the edge ratio attained by IGES+SIN; the y-axis shows edge-ratio(IGES+SIN)-edge-ratio(GES). The improvement increases with sample size and network size.

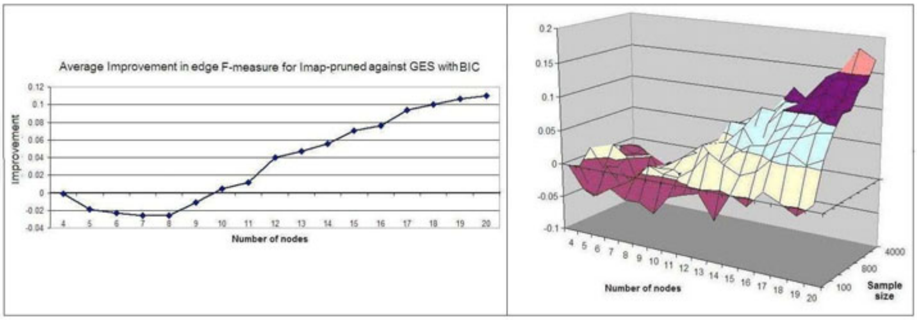


Fig. 2. Left: Average improvement in adjacency F-measure of IGES+SIN over the GES algorithm (both using BIC score) plotted against number of nodes. The x-axis indicates the number of nodes, the y-axis the difference IGES-GES for the average edge ratio over all sample sizes for the given graph size (30 graphs per sample size and number of nodes). Starting around 10 nodes, the average F-measure for IGES + SIN is better than for GES, which has a tendency towards overly complex models. Right: The improvement in adjacency F-measure increases with sample size and network size.

with sample size = 100, number of nodes = 4. For sample size 1,000 the average FAR is about 20%, and decreases linearly to about 5% for sample size 10,000. The results support our strategy of treating rejections of the null hypothesis as much more reliable than acceptances. Both FAR and FDR decrease with sample size. The FDR also depends on the size of the graph, as it increases somewhat with larger graphs.

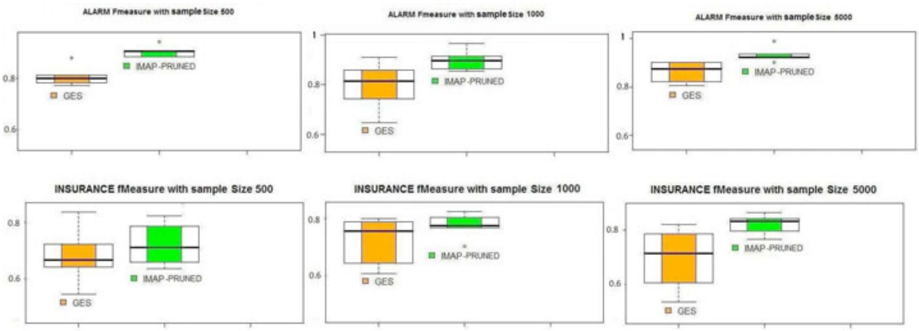


Fig. 3. Boxplots comparing the F-measure in the Alarm and Iinsurance networks for 3 different sample sizes, for GES search vs. IGES+SIN search (= Imap-pruned). Higher F-Measure values indicate a closer fit to the target structure. This plot shows the average F-measures over 5 random samples drawn for the given sample size. To better display the differences for each setting, the top figures uses a different scale from the bottom figure.

We also examined the *computational overhead* incurred by carrying out statistical testing in addition to score-based search. Our results show that the number of independence tests is roughly linear in the length of the search. The exact slope of the line depends on the sample and graph sizes; averaging over these and plotting the number of independence tests as a function of number of candidate graphs examined during the search, we find that the number of tests performed is about 6 for each graph generated.

Simulations with Real World Networks. Our simulations with real-word BNs with more nodes—Alarm [1] (37 nodes) and Insurance [3] (25 nodes)—confirm that with larger graphs, the difference in model quality increases.[1] We observed an improvement in adjacency F-measure for the constrained method, both on average and in the variance of the results, as illustrated in Figure 3.

Conclusion and Future Work

This paper presented a hybrid method for learning linear Gaussian BN structures. Compared to traditional score-based approaches, the statistical testing performed by a hybrid method detects regularities in the data that constrain the search and can guide it towards a better model. Compared to traditional constraint-based methods, the model selection score serves as a heuristic to search for a structure that satisfies the observed (in)dependency constraints. Also, a hybrid method can adopt a strategy for selecting statistical hypotheses that focuses on a relatively small set of tests that can be performed reliably. Our testing strategy was based on the Markov blanket. We treated only rejections

[1] These networks models were originally constructed with discrete variables. We followed the approach of Schmidt et al. [19] of using the same graph structure with continuous domains for the nodes.

of independence hypotheses as hard constraints on the score-based search. This makes our hybrid method less sensitive to the failures of independence tests, which are known to be the main problem for constraint-based methods.

We showed how to adapt a generic local search+score procedure for the constrained optimization required by the hybrid criterion. Evidence from simulation studies with the well-established BIC criterion indicates that, when the number of variables exceeds about 10, the additional constraints from statistical tests help select a model that is appropriately complex in that it fits the target graph structure better than the model selected by unconstrained learning. Our hybrid method appears to be a principled and effective way to address overfitting in learning Gaussian Bayes networks that combines ideas from both score-based and constraint-based learning to address the weakness of each.

Acknowledgements. This research was supported by NSERC discovery grants to RG and OS. RG gratefully acknowledges financial support from the Alberta Ingenuity Centre for Machine Learning. We are grateful to Joseph Ramsey for providing us with the source code of the Tetrad project.

References

[1] Beinlich, I., Suermondt, H., Chavez, R., Cooper, G.: The ALARM monitoring system. In: AIME 1989, pp. 247–256 (1989)

[2] Benjamini, Y., Hochberg, Y.: Controllling the false discovery rate—a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society 57(1), 289–300 (1995)

[3] Binder, J., Koller, D., Russell, S., Kanazawa, K.: Adaptive probabilistic networks with hidden variables. Machine Learning 29 (1997)

[4] Bouckaert, R.R.: Bayesian belief networks: from construction to inference. PhD thesis, Universiteit Utrecht (1995)

[5] Chickering, D.: Optimal structure identification with greedy search. JMLR 3, 507–554 (2003)

[6] The Tetrad project: Causal models and statistical data (2008), `http://www.phil.cmu.edu/projects/tetrad/`

[7] Cooper, G.: An overview of the representation and discovery of causal relationships using Bayesian networks. In: Glymour, C., Cooper, G. (eds.) Computation, Causation, and Discovery, pp. 4–62. MIT, Cambridge (1999)

[8] de Campos, L.: A scoring function for learning Bayesian networks based on mutual information and conditional independence tests. JMLR, 2149–2187 (2006)

[9] Drton, Perlman: A SINful approach to Bayesian graphical model selection. Journal of Statistical Planning and Inference 138, 1179–1200 (2008)

[10] Edwards, D.: Introduction to Graphical Modelling. Springer, New York (2000)

[11] Friedman, N., Pe'er, D., Nachman, I.: Learning Bayesian network structure from massive datasets. In: UAI, pp. 206–215 (1999)

[12] Hay, M., Fast, A., Jensen, D.: Understanding the effects of search constraints on structure learning. Technical Report 07-21, U Mass. Amherst CS (April)

[13] Heckerman, D.: A tutorial on learning with Bayesian networks. In: NATO ASI on Learning in graphical models, pp. 301–354 (1998)

[14] Klein, R.: Principles and practice of structural equation modeling. Guilford, New York (1998)

[15] Margaritis, D., Thrun, S.: Bayes. net. induction via local neighbor. In: NIPS, pp. 505–511 (2000)

[16] Meek, C.: Graphical Models: Selecting causal and statistical models. PhD thesis, CMU (1997)

[17] Neapolitan, R.E.: Learning Bayesian Networks. Pearson Education, London (2004)

[18] Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, San Francisco (1988)

[19] Schmidt, M., Niculescu-Mizil, A., Murphy, K.: Learning graphical model structure using L1-regularization path. In: AAAI (2007)

[20] Schulte, O., Frigo, G., Greiner, R., Khosravi, H.: The IMAP hybrid method for learning Gaussian Bayes nets: Full version, `ftp://ftp.fas.sfu.ca/pub/cs/oschulte/imap/imap-linear.pdf`

[21] Schulte, O., Frigo, G., Greiner, R., Khosravi, H.: A new hybrid method for Bayesian network learning with dependency constraints. In: Proceedings IEEE CIDM Symposium, pp. 53–60 (2009)

[22] Schulte, O., Luo, W., Greiner, R.: Mind change optimal learning of bayes net structure. In: Bshouty, N.H., Gentile, C. (eds.) COLT. LNCS (LNAI), vol. 4539, pp. 187–202. Springer, Heidelberg (2007)

[23] Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search. MIT Press, Cambridge (2000)

[24] Tsamardinos, I., Brown, L.E., Aliferis, C.F.: The max-min hill-climbing bayesian network structure learning algorithm. Machine Learning 65(1), 31–78 (2006)

[25] van Allen, T., Greiner, R.: Model selection criteria for learning belief nets: An empirical comparison. In: ICML, pp. 1047–1054 (2000)

[26] Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)

Feature Subset-Wise Mixture Model-Based Clustering via Local Search Algorithm

Younghwan Namkoong[1], Yongsung Joo[2], and Douglas D. Dankel II[1]

[1] Computer and Information Science and Engineering,
University of Florida, FL 32611, USA
{ynamkoon,ddd}@cise.ufl.edu
[2] Department of Statistics, Dongguk University, Seoul, South Korea
yongsungjoo@dongguk.edu

Abstract. In clustering, most feature selection approaches account for all the features of the data to identify a single common feature subset contributing to the discovery of the interesting clusters. However, many data can comprise multiple feature subsets, where each feature subset corresponds to the meaningful clusters differently. In this paper, we attempt to reveal a feature partition consisting of multiple non-overlapped feature blocks that each one fits a finite mixture model. To find the desired feature partition, we used a local search algorithm based on a Simulated Annealing technique. During the process of searching for the optimal feature partition, reutilization of the previous estimation results has been adopted to reduce computational cost.

Keywords: Clustering, Feature Selection, Finite mixture model.

1 Introduction

A natural way to analyze data having little or no available prior information (e.g. class labels) is to classify these data into a number of groups based on a similarity measure. The resulting groups are structured so that each group is heterogeneous to the other groups while data objects in the same group are homogeneous [19]. This unsupervised classification technique, called clustering, is one of the most popular approaches for exploratory data analysis in the application areas of text data mining and gene expression data analysis [5].

To discover interesting patterns in these applications, most clustering algorithms take into account all of the features represented in the data. However, in many cases, most of these features may be meaningless and disturb the analysis process. There is a need to select the proper subset of features needed to represent the clusters, but the absence of prior knowledge makes this task a difficult problem [10]. Nevertheless, feature selection has been studied as an effective technique to improve the quality of clustering results.

Although many efficient approaches have been developed for feature selection in clustering, most of them focused on either extracting relevant features or removing irrelevant and/or redundant features from all the features in the

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 135–146, 2010.

dataset[13]. However, it is noted that there can be a number of disjoint feature subsets in the original feature vector and they cannot be simultaneously revealed through these previous approaches.

In this paper, we present a novel approach to reveal a feature partition containing various clusters fitted on the finite mixture models. For each feature subsets, we applied AIC (Akaike Information Criterion) which minimizes the Kullback-Leibler (KL) divergence between the estimated and the true model to determine the goodness-of-fit of these mixture models achieved via the deterministic annealing expectation maximization (DAEM) algorithm [1,18]. The desired feature partition can be formed by aggregating the obtained feature subsets. Finding the desired feature partition can be a challenging problem because the number of possible feature partitions grows hyper-exponentially with the number of features, therefore a local search algorithm based on a Simulated Annealing (SA) technique was used [8]. Our approach shows insensitivity to the various initial feature partitions with the output providing useful information about how each feature subset contributes to the discovery of finite mixture model-based clusters.

This paper is organized as follows. After discussing feature selection approaches, we describe the structure of the feature partition model and relevant procedures to estimate the finite mixture model for clustering. Then, we present how to find the optimal feature partition determining the fitted model representing the clusters for each feature subset. The simulation results demonstrate that our approach successfully identifies the expected feature subsets.

2 Related Work

Feature selection has been extensively discussed in classification, but is relatively challenging in clustering due to the absence of prior knowledge [6]. Feature selection in clustering can be divided into the three categories; filter approaches, wrapper approaches, and the simultaneous process of feature selection and clustering [3]. In the filter approach, selecting features is determined by assessing the relevance of features in the dataset. In contrast, the wrapper approach regards the feature selection algorithm as a “wrapper” to the clustering algorithm for evaluating and selecting a relevant subset of features [10]. Sahami (1998) used mutual information for measuring feature similarity to remove redundant features [17]. Liu et al. proposed a feature selection method tailored for text clustering that used some efficient feature selection methods such as Information Gain and Entropy-based Ranking [11]. Most previous approaches tried to find relevant features based on the clustering result, so they were restricted to determining the number of clusters or choosing the closest model to represent clusters, called model selection. Law et al. (2004) incorporated mixture-based clustering by estimating feature saliency via the EM algorithm, with the use of the minimum message length criterion for the Gaussian mixture model to select the best number of clusters [10]. Constantinopoulos et al. (2006) integrated a mixture model formulation and Bayesian method to deal with both the feature

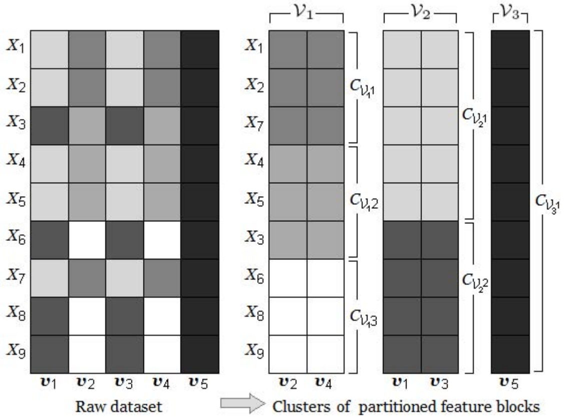


Fig. 1. Illustration of feature partitioning for mixture model-based clustering (a simple raw dataset, and its expected feature subsets and the clusters for each feature subset). In this example, $\mathcal{V} = \{\mathcal{V}_1, \mathcal{V}_2, \mathcal{V}_3\}$, where $\mathcal{V}_1 = \{v_2, v_4\}$, $\mathcal{V}_2 = \{v_1, v_3\}$, and $\mathcal{V}_3 = \{v_5\}$.

selection and the model selection problems at the same time. This approach becomes more robust for sparse datasets. Recently, sparse principal component analysis was utilized for clustering and feature selection in [13].

3 Feature Partition Model Formulation

Let $\mathbf{v} = (v_1, \ldots, v_D)$ be a D-dimensional feature vector. Data $\mathbf{X} = (\mathbf{X}_1^T, \ldots, \mathbf{X}_N^T)^T$ consist of N random objects of $\mathbf{v}$. Given $\mathbf{v}$, a partition of features $\mathcal{V}$ consists of K mutually exclusive nonempty subsets of features, denoted by $\mathcal{V} = \{\mathcal{V}_k;\ k \in \{1, \ldots, K\}$ and $K \in \{1, \ldots, D\}\}$, where $\mathcal{V}_k$ is the k^{th} subset of features. For $\mathcal{V}_k$, $\bigcup_{k=1}^{K} \mathcal{V}_k = \mathbf{v}$ and $\mathcal{V}_k \cap \mathcal{V}_{k'} = \emptyset$, where $k' \in \{1, \ldots, K\}$ and $k \neq k'$.

Let $U_{\mathcal{V}_k}$ denote a submatrix of $\mathbf{X}$ corresponding to $\mathcal{V}_k$. Based on the concept of mixture model-based clustering, $U_{\mathcal{V}_k}$ can be expressed by $G_k \in \{1 \ldots N\}$ clusters, each of which follows a probability distribution. Based on the general form of the mixture model, $f(U_{\mathcal{V}_k}; \theta_k)$ for G_k can be defined as

$$f(U_{\mathcal{V}_k}; \theta_k) = \sum_{g=1}^{G_k} p_{kg} \phi(U_{\mathcal{V}_k}; \theta_{kg}), \tag{1}$$

where, θ_{kg} is the vector of the unknown parameters, $\phi(U_{\mathcal{V}_k}; \theta_{kg})$ is the probability density of $U_{\mathcal{V}_k}$, and p_{kg} is a mixture proportion with the following constraint: $0 \leq p_{kg} \leq 1$ and $\sum_g p_{kg} = 1$. For simplicity, we assume that $\mathbf{X}$ follows a Gaussian distribution.

Since our approach assumes that all $\mathcal{V}_k$s are mutually independent, each $U_{\mathcal{V}_k}$ can lie in different Gaussian mixture models with different G_k mixture components. Each mixture component, denoted by $C_{\mathcal{V}_k g}$, represents the object index set of the g^{th} cluster in the $U_{\mathcal{V}_k}$, where $\bigcup_{g=1}^{G_k} C_{\mathcal{V}_k g} = U_{\mathcal{V}_k}$, $C_{\mathcal{V}_k g} \neq \emptyset$, and

$C_{\mathcal{V}_k g} \cap C_{\mathcal{V}_k g'} = \emptyset$ for any $g' \in \{1, \ldots, G_k\}$ and $g \neq g'$. Fig. 1 illustrates a simple example of the feature partition model.

4 Estimation of the Finite Mixture Model via the Deterministic Annealing EM Algorithm

Based on equation (1) and the above assumption, the log-likelihood function of $\mathbf{X}$, $\mathcal{L}(\theta|\mathbf{X})$, can be written as $\mathcal{L}(\theta|\mathbf{X}) = \sum_{k=1}^{K} \mathcal{L}_k(\theta_k|U_{\mathcal{V}_k})$, where

$$\sum_{k=1}^{K} \mathcal{L}_k(\theta_k|U_{\mathcal{V}_k}) = \sum_{k=1}^{K} \left(\sum_{n=1}^{N} \log \sum_{g=1}^{G_k} p_{kg} \phi(U_{\mathcal{V}_k n}|\theta_{kg}) \right) \tag{2}$$

and $\theta_k = (\theta_{k1}, \ldots, \theta_{kg}, \ldots, \theta_{kG_k})$. θ_{kg} and $\phi(U_{\mathcal{V}_k n}|\theta_{kg})$ are the parameter values and the Gaussian probability density function of the g^{th} cluster, respectively.

The estimates maximizing $\mathcal{L}_k(\theta_k|U_{\mathcal{V}_k})$ for fixed $\mathcal{V}_k$ and G_k can be usually achieved via the EM algorithm [4]. For $\mathcal{V}_k$, let Z be missing variables, where $Z_{kgn} = 1$ or 0 if the n^{th} object is assigned to $C_{\mathcal{V}_k g}$ or not. Then, the complete data log-likelihood function is

$$\mathcal{L}_k(\theta_k|U_{\mathcal{V}_k}, Z_k) = \sum_{n=1}^{N} \log \sum_{g=1}^{G_k} Z_{kgn} p_{kg} \phi(U_{\mathcal{V}_k n}; \theta_{kg}). \tag{3}$$

Starting with an initial parameter values $\theta_k^{(0)}$, the EM algorithm alternates the E-step and the M-step to update θ_k. In the i^{th} E-step, the conditional expectation of the complete data log-likelihood, called the Q function, is computed:

$$Q(\theta_k, \theta_k^{(i)}) = E[\mathcal{L}_k(\theta_k|U_{\mathcal{V}_k}, Z_k)|U_{\mathcal{V}_k}; \theta_k^{(i)}]. \tag{4}$$

In the M-step, new parameter estimates, $\theta_k^{(i+1)}$, maximizing $Q(\theta_k, \theta_k^{(i)})$ are calculated. This process stops when a convergence condition, $\theta_k^{(i+1)} = \theta_k^{(i)}$, is satisfied.

Because the EM algorithm is susceptible to local maxima due to the monotonic convergence property, one can consider utilizing the DAEM algorithm that uses a modified log-likelihood including the "thermodynamic free energy" parameter β $(0 < \beta < 1)$ [18]. Specifically, the DAEM algorithm starts with a small initial β, which is close to 0. Then, until β becomes 1, the DAEM algorithm alternates the E and M steps by gradually increasing β to obtain a better local (and possibly global) maximum.

For a specific instance of the above process, the Q function at the $i + 1^{th}$ iteration is computed as follows:

$$Q(\theta_k, \theta_k^{(i)}) = \sum_{n=1}^{N} \sum_{g=1}^{G_k} \left[\zeta_g(U_{\mathcal{V}_k n}; \theta_k^{(i)}) \log p_{kg} \phi(U_{\mathcal{V}_k n}; \theta_k^{(i)}) \right], \tag{5}$$

where $\zeta_g(U_{\mathcal{V}_k};\theta_k^{(i)})$ is the posterior probability that the n^{th} data object in the $U_{\mathcal{V}_k}$ belongs to the g^{th} mixture component,

$$\zeta_g(U_{\mathcal{V}_k n};\theta_k^{(i)}) = \frac{\left(p_{kg}^{(i)}\phi_g(U_{\mathcal{V}_k n};\mu_{kg}^{(i)},\Sigma_{kg}^{(i)})\right)^\beta}{\sum_{\ell=1}^{G_k}\left(p_{k\ell}^{(i)}\phi_\ell(U_{\mathcal{V}_k n};\mu_{k\ell}^{(i)},\Sigma_{k\ell}^{(i)})\right)^\beta} \qquad (6)$$

where $n = 1,\ldots,N$ and $g = 1,\ldots,G_k$. We slightly modified the range of β so that β becomes stable by gradually decreasing from a positive integer as the initial value of β.

5 Model Selection Based on Information Criteria

Because the estimated model can vary depending on the values of the parameters, an appropriate model amongst many candidate models should be selected. One possible approach for this problem is to choose a model that is the most similar to the "true" model. To measure the similarity between the true model and the estimated model, the Kullback-Leibler (KL) divergence is a good choice [9]. For $\mathbf{X}$, θ, and $\hat{\theta}$, the KL-divergence between the probability density function of the true model $\phi(\mathbf{X};\theta)$ and an estimated model $g(\mathbf{X};\hat{\theta})$ can be defined as

$$\begin{aligned} &KL(\phi(\mathbf{X};\theta)||\varphi(\mathbf{X};\hat{\theta})) \\ &\quad= \int \phi(\mathbf{X};\theta)\log\frac{\phi(\mathbf{X};\theta)}{\varphi(\mathbf{X};\hat{\theta})}d\mathbf{X}. \end{aligned} \qquad (7)$$

KL-divergence is equal to zero when the fitted model is equivalent to the true model. Otherwise, KL-divergence is always positive and grows as the dissimilarity between the two models increases [15]. This property implies that minimizing KL-divergence is equivalent to maximize the following term derived from the equation (7):

$$\int \phi(\mathbf{X};\theta)\log\varphi(\mathbf{X};\hat{\theta})d\mathbf{X}. \qquad (8)$$

Based on this property, the Akaike Information Criterion (AIC) was proposed to estimate the KL-divergence between the true model and the fitted model [1]. In the model selection process, an estimated model is regarded as the best fitted model when the score of AIC is minimized. For the given $\mathcal{V}_k$, $AIC(\mathcal{V}_k)$ is denoted by

$$AIC(\mathcal{V}_k) = -2\times\mathcal{L}_k(\hat{\theta}_k|U_k) + 2\lambda_k, \qquad (9)$$

where $\mathcal{L}_k(\hat{\theta}_k|U_k)$ is the maximum log-likelihood, and λ_k is the number of parameters $\hat{\theta}_k$. Using (9), the model selection process starts with $G_k = 1$. Until satisfying the stopping criterion that the score of $AIC(\mathcal{V}_k)$ is no longer decreasing, this process repeats by increasing $G_k = G_k + 1$. In particular, when the best fitted model for a U_k contains only one mixture component (cluster), $\mathcal{V}_k$ can be regarded as a feature subset having less-contribution in clustering. Since feature subsets are mutually independent, the AIC for $\mathcal{V}$ can be computed as the sum of $AIC(\mathcal{V}_k)$s, expressed by:

$$J(X, \mathcal{V}, \theta) = \sum_{k=1}^{K} AIC(\mathcal{V}_k). \tag{10}$$

Note that minimizing equation (10) implies the discovery of feature partition consisting of multiple feature subsets where each can be expressed by the best-fit mixture model for clustering. Accordingly, equation (10) is utilized as an objective function in our approach.

6 Simulated Annealing with RJMCMC Technique for Finding the Feature Partition

Searching for the feature partition minimizing the objective function (10) is not feasible, because the number of all possible partitions for a fixed number of features is known as the Bell number [16],

$$B(D) = \sum_{K=1}^{D} \frac{1}{K!} \sum_{k=1}^{K} (-1)^{K-k} \binom{K}{k} k^D, \tag{11}$$

which grows hyper-exponentially in D and prevents finding a solution through exhaustive search. Moreover, for each feature subset $\mathcal{V}_k$ in a given feature partition $\mathcal{V}$, the model selection process as well as the mixture model-based clustering via the DAEM algorithm aggravate the prohibitive computational complexity. For such a combinatorial optimization problem, Monte Carlo optimization can be an acceptable technique [7]. Let $\pi(\mathcal{V})$ be a probability mass function related with $\mathcal{V}$, denoted by

$$\pi(\mathcal{V}) = \frac{e^{-J(X,\mathcal{V},\theta)}}{\sum_{V \in \Omega(V)} \left\{ e^{-J(X,V,\theta)} \right\}}, \tag{12}$$

where $\Omega(V)$ represents the state space of all possible feature partitions. The optimal feature partition $\mathcal{V}^*$ is achieved by evaluating the objective function (10) for the $\mathcal{V}$s randomly generated from $\pi(\mathcal{V})$. If the objective function $J(X, \mathcal{V}, \theta)$ decreases, candidate feature partitions that are close to the modal score of $\pi(\mathcal{V})$ can be more frequently generated [7].

Using the above Monte Carlo optimization, a special local search algorithm based on the Metropolis-Hastings (MH) algorithm, called the biased random walk algorithm [2], can be a suitable technique to search for the $\mathcal{V}^*$. Before discussing the biased random walk algorithm, we briefly explain the basic concepts of the MH algorithm. Suppose that the t^{th} state is the current state. Let $\mathcal{V}^{(t)}$ be a feature partition at the current state. A candidate feature partition $\mathcal{V}'^{(t)}$

can be proposed by an MH algorithm and the probability for transition from $\mathcal{V}^{(t)}$ to $\mathcal{V}'^{(t)}$ can be denoted by $p(\mathcal{V}'^{(t)}|\mathcal{V}^{(t)})$. Then, as a feature partition at the next state $\mathcal{V}^{(t+1)}$, either $\mathcal{V}^{(t)}$ or $\mathcal{V}'^{(t)}$ is selected with an acceptance probability γ, defined as

$$\gamma = \min\left[1, \alpha = \frac{\pi(\mathcal{V}'^{(t)})p(\mathcal{V}^{(t)}|\mathcal{V}'^{(t)})}{\pi(\mathcal{V}^{(t)})p(\mathcal{V}'^{(t)}|\mathcal{V}^{(t)})}\right]. \tag{13}$$

To be specific, the feature partition at the $t + 1^{th}$ state $\mathcal{V}^{(t+1)} = \mathcal{V}'^{(t)}$ when $\alpha \geq 1$ or $p_{\mathrm{r}} < \alpha < 1$ for p_{r} drawn uniformly at random from $\in (0, 1)$. Otherwise, $\mathcal{V}^{(t+1)} = \mathcal{V}^{(t)}$.

In the biased random walk algorithm, it is assumed that a candidate feature partition at the t^{th} state $\mathcal{V}'^{(t)}$ is generated by updating the index of one feature in $\mathcal{V}^{(t)}$. This implies $p(\mathcal{V}'^{(t)}|\mathcal{V}^{(t)}) = p(\mathcal{V}^{(t)}|\mathcal{V}'^{(t)})$, so α in equation (13) can be reduced to $\pi(\mathcal{V}'^{(t)})/\pi(\mathcal{V}^{(t)})$. Even though the biased random walk algorithm can find $\mathcal{V}^*$ theoretically over the $\Omega(\mathcal{V})$, the relative narrow search region at each state causes a slow convergence. Furthermore, when all $\mathcal{V}'^{(t)}$s that can be generated from $\mathcal{V}^{(t)}$ have a much lower objective function score, it is difficult to escape from the $\mathcal{V}^{(t)}$. This situation can be regarded as a local maxima problem. To alleviate this problem, it is worthwhile to utilize the Simulated Annealing (SA) technique [8], an annealing process to search for the global optimum.

In SA, the temperature parameter T ($T > 0$) represents the degree of random transition between states, meaning that a candidate state tends to be accepted at a high temperature. Our search algorithm, which combines the SA technique with the biased random walk algorithm, starts with the initial feature partition $\mathcal{V}^{(0)}$ and the initial temperature parameter $T^{(0)}$ set to a high value. Until the final state becomes stable, when $T \to 0$, this search algorithm explores $\mathcal{V}^*$ by gradually decreasing T. At the t^{th} state, a candidate feature partition $\mathcal{V}'^{(t)}$ is accepted by the following probability γ':

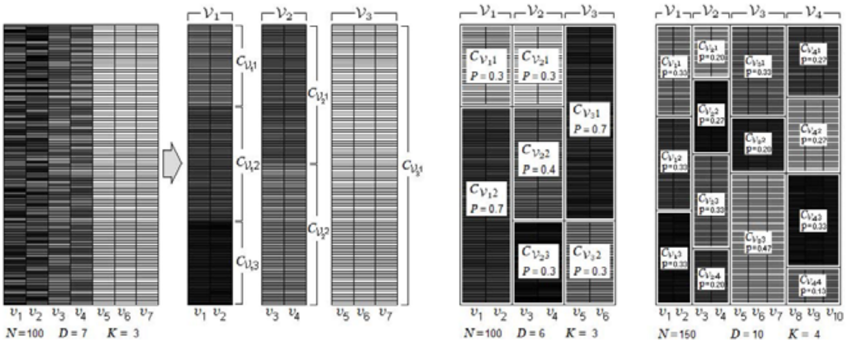
$$\gamma' = \min\left[1, \alpha = \frac{\pi(\mathcal{V}'^{(t)})}{\pi(\mathcal{V}^{(t)})}\exp\{T^{(t)}\}\right], \tag{14}$$

where $T^{(t)} = \rho \times T^{(t-1)}$, $t \geq 1$, and ρ is a cooling rate, $\rho \in (0, 1)$.

During the search for $\mathcal{V}^*$, evaluating the fitted mixture model via the DAEM algorithm at each state requires a burdensome computational time. This can be mitigated by an incremental estimation technique that reutilizes the value of $\pi(\mathcal{V}^{(t)})$ that was previously computed [14]. By reducing the number of unnecessarily repeated evaluation of $\pi(\mathcal{V}^{(t)})$, the total computation time can be drastically reduced. In addition, the initialization-insensitive property of the DAEM algorithm supports the use of the incremental estimation.

7 Experimental Results and Discussion

Our approach was evaluated with various types of datasets. From the simulation results, we discuss the convergence and insensitiveness of the proposed approach.



(a) Dataset-I (b) Dataset-II (c) Dataset-III

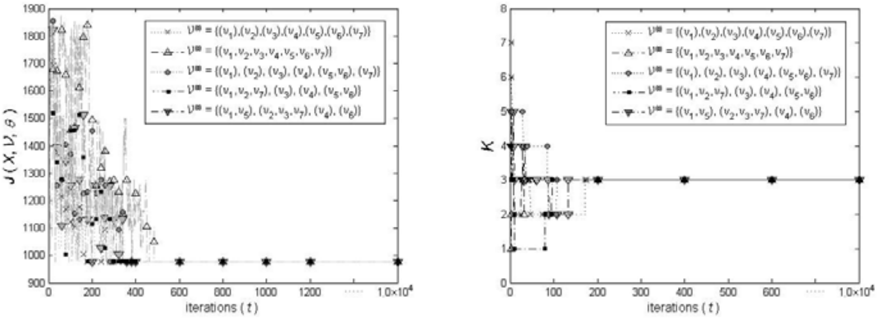
Fig. 2. Synthetic datasets

Table 1. The parameter estimates of the dataset-I

	θ : parameter values used to generate the 1st dataset			$\hat{\theta}$: parameter estimates obtained from the dataset-I		
$\mathcal{V}$	$[\ (v_1, v_2); (v_3, v_4); (v_5, v_6, v_7)\ ]$			$[\ (v_1, v_2); (v_3, v_4); (v_5, v_6, v_7)\ ]$		
G_k	$G_1 = 3$	$G_2 = 2$	$G_3 = 1$	$\hat{G}_1 = 3$	$\hat{G}_2 = 2$	$\hat{G}_3 = 1$
p_{kg}	$p_{11} = 0.30$ $p_{21} = 0.50$	$p_{12} = 0.40$ $p_{22} = 0.50$	$p_{13} = 0.30$ $p_{31} = 1.00$	$\hat{p}_{11} = 0.30$ $\hat{p}_{21} = 0.50$	$\hat{p}_{12} = 0.40$ $\hat{p}_{22} = 0.50$	$\hat{p}_{13} = 0.30$ $\hat{p}_{31} = 1.00$
μ_{kg}	$\mu_{11} = [7.15,\ 7.00]$ $\mu_{21} = [4.92,\ 4.93]$	$\mu_{12} = [4.06,\ 4.15]$ $\mu_{22} = [8.99,\ 8.97]$	$\mu_{13} = [1.13,\ 0.99]$ $\mu_{31} = [14.0,\ 14.0,\ 14.0]$	$\hat{\mu}_{11} = [7.15,\ 7.00]$ $\hat{\mu}_{21} = [4.92,\ 4.93]$	$\hat{\mu}_{12} = [4.06,\ 4.15]$ $\hat{\mu}_{22} = [8.99,\ 8.97]$	$\hat{\mu}_{13} = [1.13,\ 0.99]$ $\hat{\mu}_{31} = [13.96,\ 13.96,\ 13.97]$
Σ_{kg}	$\Sigma_{11} = \begin{bmatrix} 0.25 & 0.20 \\ 0.20 & 0.25 \end{bmatrix}$ $\Sigma_{21} = \begin{bmatrix} 0.25 & 0.18 \\ 0.18 & 0.25 \end{bmatrix}$	$\Sigma_{12} = \begin{bmatrix} 0.25 & 0.19 \\ 0.19 & 0.25 \end{bmatrix}$ $\Sigma_{22} = \begin{bmatrix} 0.25 & 0.19 \\ 0.19 & 0.25 \end{bmatrix}$	$\Sigma_{13} = \begin{bmatrix} 0.25 & 0.21 \\ 0.21 & 0.25 \end{bmatrix}$ $\Sigma_{31} = \begin{bmatrix} 0.25 & 0.19 & 0.20 \\ 0.19 & 0.25 & 0.19 \\ 0.20 & 0.19 & 0.25 \end{bmatrix}$	$\hat{\Sigma}_{11} = \begin{bmatrix} 0.27 & 0.22 \\ 0.22 & 0.28 \end{bmatrix}$ $\hat{\Sigma}_{21} = \begin{bmatrix} 0.28 & 0.21 \\ 0.21 & 0.23 \end{bmatrix}$	$\hat{\Sigma}_{12} = \begin{bmatrix} 0.28 & 0.26 \\ 0.26 & 0.34 \end{bmatrix}$ $\hat{\Sigma}_{22} = \begin{bmatrix} 0.28 & 0.18 \\ 0.18 & 0.21 \end{bmatrix}$	$\hat{\Sigma}_{13} = \begin{bmatrix} 0.24 & 0.21 \\ 0.21 & 0.25 \end{bmatrix}$ $\hat{\Sigma}_{31} = \begin{bmatrix} 0.26 & 0.22 & 0.19 \\ 0.22 & 0.27 & 0.20 \\ 0.19 & 0.20 & 0.21 \end{bmatrix}$

Three synthetic datasets were generated on the basis of the following parameters, K, G_ks, $\mathcal{V}$, p_{kg}s, μ_{gd}s, and Σ_{gd}s where $g \in \{1, \ldots, G_k\}$, $d \in \{1, \ldots, D\}$, and $k \in \{1, \ldots, K\}$. For $\mathcal{V}$, $\mathcal{V}_k$ is composed of the different number of $\mathcal{C}_{\mathcal{V}_k g}$ and its corresponding p_{kg}. Each X_n lies in a Gaussian distribution corresponding to μ_{kg} and Σ_{kg}. For example, the dataset-I contains three feature subsets: $\mathcal{V}_1$, $\mathcal{V}_2$, and $\mathcal{V}_3$. Each $\mathcal{V}_k$ has 3, 2, and 1 mixture components, respectively. Table 1 provides the values of parameters for creating the dataset-I. Fig. 2(a) illustrates the expected results as well as the overall structures of the dataset-I. The dataset-II and the dataset-III, see Fig. 2, have different shapes and more complex structures than the dataset-I. In particular, the structure of the dataset-II is sometimes called a checkerboard structure.

Our feature partition search algorithm was executed with the datasets for 1.0×10^4 iterations. To cover the overall cases for each dataset, we used 5 different initial partition $\mathcal{V}^{(0)}$s. For instance, the $\mathcal{V}^{(0)}$s used for the dataset-I are: (1) $\mathcal{V}^{(0)}$=$\{(v_1), (v_2), (v_3), (v_4), (v_5), (v_6), (v_7)\}$, (2) $\mathcal{V}^{(0)} = \{(v_1, v_2, v_3, v_4, v_5, v_6, v_7)\}$, (3) $\mathcal{V}^{(0)}$=$\{(v_1), (v_2), (v_3), (v_4), (v_5, v_6), (v_7)\}$, (4) $\mathcal{V}^{(0)}$=$\{(v_1, v_2, v_7), (v_3), (v_4), (v_5, v_6)\}$, and (5) $\mathcal{V}^{(0)}$=$\{(v_1, v_5), (v_2, v_3, v_7), (v_4), (v_6)\}$. Specifically, each $\mathcal{V}_k$ in (1) is a singleton feature subset, $\mathcal{V}^{(0)}$ of (2) is a feature subset with all



(a) Trajectories of chains

(b) Trajectories of K

Fig. 3. Experimental results with the various $\mathcal{V}^{(0)}$s for the dataset-I

features, and the remaining three $\mathcal{V}^{(0)}$s were randomly generated. To support enough randomness, $T^{(0)} = 400.0$ and $\rho = 0.997$. For the DAEM algorithm, $\beta^{(0)} = 4.0$ and $\beta^{(i)} = \beta^{(i-1)} \times 0.998$.

Through the results shown in Fig. 3(a), our search method demonstrated insensitivity to the various initial feature partitions, $\mathcal{V}$s, by showing successful convergence to the minimum score of the objective function $J(X, \mathcal{V}, \theta)$. Fig. 3(b) shows the trajectory of the change to the total number of feature subsets in $\mathcal{V}^{(t)}$, corresponding to the score of the objective function.

As shown in Table 1, our method found both feature subsets and the parameter estimates near the true parameter values. On the other hand, for the dataset-I, when the conventional clustering was performed with all features, the best-fit mixture model-based clusters consisting of three mixture components were discovered and two features (i.e., v_1 and v_2) were identified as the informative features for clustering. This result shows that the conventional clustering algorithms are inadequate to mine a number of feature subsets being useful to explain diverse clustering outputs at the same time. For the dataset-II and the dataset-III, our approach revealed the expected feature partition as well as the mixture model to represent the clusters for each feature subset. We omit these results due to the space limitation of this paper.

Our method seeks the desired feature partition using the incremental estimation and has been tested on the three datasets described above. Fig. 4 shows the results of the total evaluation time. In all cases, the growth rate of the execution time for each iteration becomes stable after a few thousands of iterations. In contrast to the approach that estimates at all iterations, the incremental estimation method showed better performance with drastically reduced computational time.

We also applied our approach to two widely used real datasets available from the UCI Machine Learning Repository ('diabetes' and 'waveform') [20]. The diabetes dataset consist of 768 instances (500 and 268 instances tested positive and negative for diabetes, respectively) described by 8 numeric-valued attributes. The waveform dataset consists of 5000 21-dimensional instances that can be grouped by 3 classes of waves. All features except one associated with class

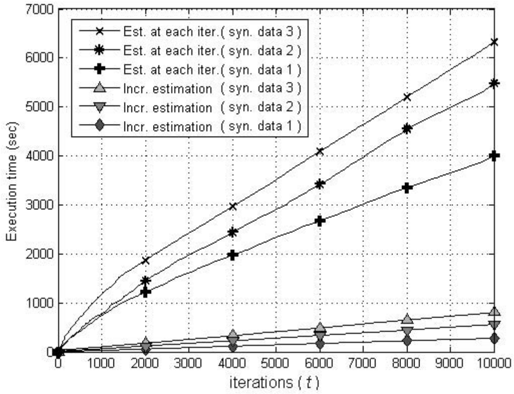


Fig. 4. Performance evaluation of the incremental estimation

labels were generated by combining two of three shifted triangular wave forms and then adding Gaussian noise [12]. Because the clustering results of our approach can be much different from those of the other methods, it can be difficult in comparing between them. Moreover, this implies the difficulty in directly evaluating the clustering accuracy. Therefore, in our experiments, we assessed clustering accuracy using the feature subset that shows the most similar clustering results to the original class labels. Table 2 summarizes the experimental results with datasets including the properties of these datasets.

In the experiment with diabetes dataset, the 2nd feature block contained two mixture components and its clustering accuracy to the "true" class labels was 0.6602. For the 1st feature block, v_1 had similar degrees of negative relationships with v_7 in all three clusters. The 3rd feature block containing a single feature was composed of three mixture components. Both the 1st and the 3rd feature blocks showed relatively meaningless analytical information than the 2nd feature block. The experimental results of the waveform dataset seem to cover

Table 2. The summary of experimental results (N: the number of data samples, D: the number of dimensions, K: the actual number of feature blocks, $\hat{K}$: the estimates of the number of feature blocks, G: the actual number of clusters in each feature block, and $\hat{G}$: the estimates of the number of clusters in each feature block)

Dataset	N	D	$\hat{K}(K)$	$\hat{G}(G)$	$\mathcal{V}$
Dataset-I	100	7	3(4)	[3,2,1] ([3,2,1])	$[\ (v_1, v_2); (v_3, v_4); (v_5, v_6, v_7)\ ]$
Dataset-II	100	7	3(4)	[2,2,2] ([2,2,2])	$[\ (v_1, v_2); (v_3, v_4); (v_5, v_6)\ ]$
Dataset-III	150	10	4(4)	[3,4,3,4] ([3,4,3,4])	$[\ (v_1, v_2); (v_3, v_4); (v_5, v_6, v_7); (v_8, v_9, v_{10})\ ]$
Diabetes	768	8	2(N/A)	[3,2,3] (N/A)	$[\ (v_1, v_7); (v_2, v_3, v_4, v_5, v_6); (v_8)\ ]$
Waveform	5000	21	3(N/A)	[1,1,3] (N/A)	$[\ (v_1); (v_{21}); (v_2, v_3, \ldots, v_{19}, v_{20})\ ]$

a result of the feature selection method. Each of two features, v_1 and v_{21}, was represented by a singleton cluster based on univariate Gaussian distribution. This implies that they can be regarded as less informative features. For the 3rd feature block containing all features except v_1 and v_{21}, the mixture-model based clustering results showed 0.7690 clustering accuracy to the assigned class labels.

8 Conclusion

In many applications for cluster analysis, data can be composed of a number of feature subsets where each is represented by a number of diverse mixture model-based clusters. However, in most feature selection algorithms, this kind of cluster structure has rarely been interesting because they accounted for discovery of a single informative feature subset for clustering. In this paper, we proposed a novel approach to find a set of the feature subsets based on the Gaussian mixture model at the same time. In our approach, each feature subset is represented by a number of best-fit mixture model-based clusters by utilizing the AIC criterion and DAEM algorithm. To avoid an unreasonable processing time when searching for the optimal feature partition, the Simulated Annealing-based reversible jump Markov Chain Monte Carlo technique has been utilized. Also, the total computational cost to search for the optimal feature partition can be reduced through the use of an incremental estimation. Experimental results demonstrated that our approach is insensitive to the various initial feature partitions.

Our method can be used in various application areas such as text data mining and gene expression data analysis. Specifically, in text data mining, there are many features to represent text documents. Through our approach, they can be partitioned over the various feature subsets that identify genres, authors, styles, and other categories, and each document can be assigned different clusters across the above diverse feature subsets. However, approaches should be considered to address the problem of how to scale up feature selection in text clustering where the number of features is in the order of thousands. To alleviate this problem, as a preprocessing step, one can select related features with clustering through the likelihood ratio test. This issue is under investigation.

References

1. Akaike, H.: A new look at the statistical model identification. IEEE Transactions on Automatic Control 19(6), 716–723 (1974)
2. Booth, J.G., Casella, G., Hobert, J.P.: Clustering using objective functions and stochastic search. Journal of the Royal Statistical Society B 70(1), 119–139 (2008)
3. Constantinopoulos, C., Titsias, M.K., Likas, A.: Bayesian feature and model selection for Gaussian mixture models. IEEE Trans. Pattern Anal. Mach. Intell. 28(6), 1013–1018 (2006)
4. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum-likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society B 39, 1–38 (1977)

5. Han, J., Kamber, M.: Data mining: concepts and techniques. Morgan Kaufmann Publishers, San Francisco (2001)
6. Jain, A.K., Zongker, D.E.: Feature selection: Evaluation, application, and small sample performance. IEEE Trans. Pattern Anal. Mach. Intell. 19(2), 153–158 (1997)
7. Green, P.J.: Reversible jump Markov Chain Monte Carlo computation and Bayesian model determination. Biometrika 82, 711–732 (1995)
8. Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P.: Optimization by simulated annealing. Science 220, 671–680 (1983)
9. Kullback, S., Leibler, R.A.: On information and sufficiency. The Annals of Mathematical Statistics 22(1), 79–86 (1951)
10. Law, M.H.C., Figueiredo, M.A.T., Jain, A.K.: Simultaneous feature selection and clustering using mixture models. IEEE Trans. Pattern Anal. Mach. Intell. 26(9), 1154–1166 (2004)
11. Liu, T., Liu, S., Chen, Z., Ma, W.-Y.: An evaluation on feature selection for text clustering. In: Fawcett, T., Mishra, N. (eds.) ICML. AAAI Press, Menlo Park (2003)
12. Liu, H., Motoda, H.: Computational Methods for Feature Selection. Chapman & Hall/CRC, Boca Raton (2007)
13. Luss, R., dAspremont, A.: Clustering and feature selection using sparse principal component analysis. CoRR abs/0707.0701 (2007)
14. Neal, R.: Markov chain sampling methods for Dirichlet process mixture models. Technical Report 9815, Department of statistics, University of Toronto (1998)
15. Roberts, S.J., Holmes, C., Denison, D.: Minimum-entropy data partitioning using reversible jump markov chain monte carlo. IEEE Trans. Pattern Anal. Mach. Intell. 23(8), 909–914 (2001)
16. Rota, G.-C.: The Number of Partitions of a Set. American Mathematical Monthly 71(5), 498–504 (1964)
17. Sahami, M.: Using Machine Learning to Improve Information Access. Ph.D. thesis, Stanford University, CA (1998)
18. Ueda, N., Nakano, R.: Deterministic annealing EM algorithm. Neural Networks 11(2), 271–282 (1998)
19. Xu, R., Wunsch II, D.: Clustering (IEEE Press Series on Computational Intelligence). Wiley-IEEE Press (2009)
20. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository (2007), http://archive.ics.uci.edu/ml/index.html

Cascading Customized Naïve Bayes Couple

Guichong Li[1], Nathalie Japkowicz[1], Trevor J. Stocki[2], and R. Kurt Ungar[2]

[1] Computer Science, University of Ottawa,
800 King Edwards, Ottawa, Canada
{jli136,nat}@site.uottawa.ca
[2] Radiation Protection Bureau, Health Canada,
Ottawa, ON, Canada
{trevor_stocki,kurt_ungar}@hc-sc.gc.ca

Abstract. Naïve Bayes (NB) is an efficient and effective classifier in many cases. However, NB might suffer from poor performance when its conditional independence assumption is violated. While most recent research focuses on improving NB by alleviating the conditional independence assumption, we propose a new Meta learning technique to scale up NB by assuming an altered strategy to the traditional Cascade Learning (CL). The new Meta learning technique is more effective than the traditional CL and other Meta learning techniques such as Bagging and Boosting techniques while maintaining the efficiency of Naïve Bayes learning.

Keywords: Naïve Bayes, Meta Learning, Cascade Learning.

1 Introduction

Naïve Bayes (NB) is a simple Bayesian classifier which assumes conditional independence of the domain's attributes. Research has shown that NB exhibits high accuracy over other classifiers in many cases [5]. Moreover, because it is efficient, due to its linear time complexity for training, NB has been widely applied to many practical applications such as text classification [20].

The main problem with NB is that it suffers from poor performance if the conditional independence assumption is violated [20]. Previous research emphasizes the improvement of NB by alleviating the assumption [5][19][26]. For example, NB can be improved by selecting an effective set of attributes which satisfy the conditional independence assumption [17] or by combining decision tree with Naïve Bayes [16]. More recent research focuses on One-Dependent-Estimation (ODE) [14][26][29][30] by building a simplified Bayesian network structure in that each node has a single parent node.

However, these improved approaches sometimes still suffer from either ineffectiveness or a high time and space complexity as compared to the original NB while they achieve quite a success in some circumstances. More simply, one can apply a Meta learning technique such as Adaptive Boosting (AdaBoost) [9] to improve individual classifiers. This seems more attractive because a Meta learner promises a high efficiency by using a linear combination of individual classifiers. Unfortunately, the

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 147–160, 2010.

previous Meta learning techniques attempted [2][25] failed to scale up NB due to the stability of NB [1][7][23][25].

In this paper, we propose a new Meta learning technique to scale up NB by altering the strategy of the traditional Cascade Learning (CL) [11][24][28]. When the traditional CL achieves its goal by constructing a new feature space for the Meta level, it is observed that it often failed to construct a proper feature space for discrimination in the Meta level. Instead, the proposed new Meta learning technique, called Cascading Customized Couple (CCC), is a domain-based CL built by the construction of subdomains from the original domain.

The main advantages of this method consist of the following two aspects:

Firstly, it is more efficient than previously proposed approaches for scaling up NB because it is a linear combination of two NB classifiers without a significant increase of the time and space complexity of NB; Secondly, it is more effective than either the traditional CL in most cases by assuming an altered strategy to the CL, or other Meta learning techniques such as Bagging and Boosting techniques for scaling up NB.

2 Previous Research

2.1 Meta Learning and Cascade Learning

Previous research has proposed Meta learning techniques to scale up individual classifiers. Bootstrap aggregating (Bagging) [2] builds a Meta learner by building diverse models on subsamples (or bags) obtained by sampling uniformly with replacement on the original training set. Adaptive Boosting (AdaBoost) is a Meta Learning algorithm [9], which improves any learning algorithm by repeatedly calling a weak classifier. In each round, the weights of each incorrectly classified example are increased and the weights of each correctly classified example are decreased, so that the new classifier focuses on the incorrectly classified examples. MultiBoostAB [25] assumes the Adaboost technique and wagging (sampling with different weights) technique to enhance the original AdaBoost by wagging a set of sub-committee of classifiers so that each of them is formed by AdaBoost.

Cascade Learning (CL) [11][24][28] is quite different from the Bagging and Boosting techniques in that it emphasizes a straightforward relationship among individual components. For example, previously proposed cascade learning techniques such as cascade generalization [11] and stacked generalization [24][28] build a set of classifiers on the original domain. They also output class probability distributions for each instance in the original domain. A new domain can be constructed from the original domain by using these class probability distributions as its feature values in the Meta level.

Although Bagging, AdaBoost, and MultiBoostAB demonstrates success in many practical applications, they often suffer from failures to scale up NB due to the stability of NB [1][7][23][25]. CL, on the other hand, has difficulty creating a proper feature space for purpose of discrimination in the Meta level so that it is improper to be used for scaling up NB in many cases.

2.2 Naïve Bayes and Enhancement

Given a training set with a probability distribution P, in supervised learning, Bayesian learning defines a classifier with a minimized error, i.e.,

$$y_\mathrm{i} = c_\mathrm{i} = \arg\max_{c_i \in C} P(c_i \mid x) \equiv \arg\max_{c_i \in C} P(x|c_i)P(c_i)$$
$$= \arg\max_{c_i \in C} P(a_1, a_2, ..., a_\mathrm{n} \mid c_i)P(c_i) \qquad (2.1)$$

Naïve bayes (NB) [6][18] assumes the probabilities of attributes $a_1, a_2, \ldots, a_\mathrm{n}$ to be conditionally independent given the class c_i. Therefore, the right side of (2.1) becomes

$$P(x \mid c_i) = P(a_1, a_2, ..., a_n \mid c_i) = \prod_{j=1}^{n} P(a_j \mid c_i)$$

NB can be trained in the linear time complexity O(*mn*), where *m* is the number of attributes and *n* is the number of examples in the training set. Moreover, NB is a stable classifier, and has exhibited a high performance over other classifiers in many applications.

On the other hand, a number of studies promise the improvement of NB by overcoming the restrictions of the conditional independence assumption. We summarize these methods into the following three categories:

(I) Select a subset of attributes such that they satisfy the conditional independence assumption. For example, Selective Bayesian Classifiers (SBC) [19] uses forward selection in a greedy search to find an effective subset of attributes, with which a better Naïve Bayes is built.

(II) Combine a decision tree with Naïve Bayes. For example, NBTree [16] builds a local NB on each leaf of a decision tree.

(III) Build a simplified Bayesian network structure by allowing a simple relationship between attributes given a class [14][26][29][30]. Tree Augmented Naïve Bayes (TAN) [10] extends tree-like Naïve Bayes, in which the class node directly points to all attribute nodes, and an attribute node has only one parent attribute. TAN has a time complexity of $O(m^2 logm)$ for structure learning, and then a time complexity $O(nm^2+km^2v^2+m^2logm)$ for training, where *m* and *n* are defined as above; *k* is the number of classes, and *v* is the average number of values for each attribute.

TAN is also regarded as a One-Dependent Estimation (ODE) technique. Other ODE techniques [14][26][30][29] improve TAN without searching for the simplified Bayesain network structure. For example, Aggregating One-Dependence Estimators (AODE) [26] achieves higher accuracy than NB by averaging over a constrained group of 1-dependence NB models built on a small space. AODE has the time and space complexities of $O(nm^2)$ and $\mathrm{O}(k(nm)^2)$, respectively. Although other ODE techniques including AODE is more efficient than TAN, they are still more complicated than NB with respect to the time and space complexities.

In this paper, we consider a novel Meta learning technique to scale up NB.

3 Cascading Customized Naïve Bayes Couple

3.1 Cascade Learning

As we know from Section 2.1, traditional CL defines the base level and constructs a new feature space for the Meta level [28]. Given a dataset D = {(y_i, x_i), i = 0,…,

$n-1\}$, where y_i is a class value, x_i is a vector of attribute values, and n is the number of examples in D, for the purpose of scaling up NB, we build K NB models on D in the base level by using k-cross validation for diversity. The kth model outputs a probability prediction p_{ik} for an input (y_i, x_i), where $k = 0, \ldots, K-1$. Therefore, we obtain a new dataset $\{(y_i, p_{ik}), k = 0,\ldots, K-1$, and $i = 0,\ldots, n-1\}$ for the Meta level where another NB model is built on the new feature space.

However, this CL is not easily realized in practice because it is unclear how to construct a proper feature space for discrimination in the Meta level [24], and it is also difficult to make NB diverse because NB is a stable learner. To overcome this obstacle, we adopt an altered strategy for CL and propose a new definition of a domain-based CL as follows.

Definition 3.1. Domain-based Cascade Learning (DCL) is an ensemble learning technique that learns individual classifiers from the original domain by building each component on its own sub-domain which can be reconstructed in terms of the outputs of previously built component learners. All components together make a decision.

The above definition of DCL, or simply CL without any confusion, finds its roots in previous research related to cascade generalization [11] and stacked generalization [24][28]. Some ideas similar to CL in Definition 3.1 have been raised in previous research. For example, the Cascade-Correlation method learns the architecture of artificial neural networks using a strategy similar to CL [8]. A mixture of local experts [13] can be built on a partitioned domain by assuming the Expectation Maximization (EM) algorithm for the mixture [15]. Recursive Bayesian Classifiers (RBC) [17] is a suggestive schema that uses a hierarchy of probabilistic summaries instead of a single schema, i.e., RBC partitions instances and recursively builds simple NB on each partition. Our method proposed in this paper can be regarded as an implementation of this hierarchy.

3.2 Customized Classifier

The main idea behind this altered CL is related to a new classifier, called *Customized Classifier* (CC). For example, NB is a linear classifier [6], and it can become a CC. Suppose that a training set is divided into many small subsets. Therefore, an individual NB classifier can be built on the partitioned training set for classifying a target subset. We describe several related concepts as follows.

Definition 3.2. A labeled training set can be regarded as a *domain*, which describes some domain knowledge. A subset of the training set can be regarded as a *sub-domain.*

A sub-domain does not have to contain examples belonging to the same category although examples in each original class naturally constitute a sub-domain. Not any sub-domain but those sub-domains that cross the class boundary are more likely to be informative.

The original domain can be divided into many sub-domains if necessary, and sub-domains can be labeled by *artificial class labels* as additional classes. Therefore, a classifier can be customized on the partitioned domain in terms of the related sub-domain.

Definition 3.3. A *Customized Classifier* (CC) is a classifier, which can classify an input in the related sub-domain, and can reject the classification on an input outside the sub-domain.

Definition 3.3 can be further explained as follows. A CC can output the class distribution of an input with respect to both of the original classes and the additional classes. For an input outside the related sub-domain, the CC intends to classify it by outputting a class membership probability of 0 or an equal class membership probability for the original classes. This leads to the *rejection* of classification on the input by eliminating its effect on the final classification if an averaging combination rule is used.

Although the number of CC is not limited, we emphasize the use of a couple of individual CCs in this altered CL. In particular, we suggest Cascading Customized Couple (CCC) for scaling up NB. The principles of CC and CCC can be simply described in Example 3.1.

Example 3.1. Given a domain D with two original classes, c_1, c_2, without any loss of generality, two CC classifiers, denoted as H_1 and H_2, are built from D, where H_1 has its sub-domain S_1 and the outside of the sub-domain is labeled by c_3; H_2 has its sub-domain S_2 and the outside of the sub-domain is labeled by c_4.

Given an input $x \in S_1$ with a true label c_1, because x is in the sub-domain S_1 of H_1, H_1 can correctly classify x. Suppose we have $P_1(c_1|x) = 0.6$, $P_1(c_2|x) = 0.3$, and $P_1(c_3|x) = 0.1$.

Because x is not in the sub-domain S_2 of H_2, H_2 cannot classify x into either c_1 or c_2. Instead, H_2 classifies x into its additional class c_4, Suppose we have

$P_2(c_1|x) = P_2(c_2|x) = 0.2$, and $P_2(c_4|x) = 0.6$, i.e., H_2 rejects classifying x into either c_1 or c_2 because $P_2(c_1|x) = P_2(c_2|x)$. Therefore,

$p_1 = (P_1(c_1|x) + P_2(c_1|x) / 2 = 0.4$

$p_2 = (P_1(c_2|x) + P_2(c_2|x) / 2 = 0.25$

As a result, $c_1 = \arg\max_i (P_1(c_i \mid x) + P_2(c_i \mid x)/2)$, where $i = 1, 2$. □

3.3 Learning Algorithm

The main issue is that it is difficult to exactly build CCs. Instead, we intend to build approximate CCs for diversity as follows.

Given a training set D, the initial domain is the whole training set with original class labels. The first CC_0 is built on D using a base learner, i.e., NB. CC_0 actually is a traditional classifier built on the original training set. The learning algorithm is completed if CC_0 totally fits the training set D without misclassifications.

Otherwise, the misclassifications need to be further classified. To this end, we can add additional classes to label the corresponding correct classifications. As a result, all misclassifications become a sub-domain, and the outside of the sub-domain is labeled by artificial labels as the additional classes. The original training set D becomes a new training set D_1 with the sub-domain containing the misclassifications and the additional classes corresponding to the classified examples. The second CC_1 is built on D_1 and the learning algorithm ends up with a couple of CC classifiers.

Because CC_1 is built in terms of the outputs of CC_0, we say that they are cascaded with each other on D, and they are combined with each other to become a CCC classifier. However, we intend to only build approximate CCs, i.e., CC_0 and CC_1, for diverse models.

```
CCC algorithm
input   $D$: original domain;
        L: a specified base learner, e.g., NB
output  $H$:$CCC$, the resulting CCC classifier
1  saveLabels($D$)
2  $B = \varnothing$
   // first CC
3  $h_1$ = L($D$), $B = B \cup \{h_1\}$
4  $E = h_1(D)$, CT = $D - E$
   // second CC
5  if(|CT| < |D|)
6     addClasses(CT, $D$, 0)
7     $h_2$ = L($D$), $B = B \cup \{h_2\}$
8  $H(x) = \tilde{c} = \arg\max_{c \in C}\left(P''(H(x) = c)\right)$,
   8.1 $P''(H(x) = c) = \frac{P'(H(x) = c)}{\sum_{c' \in C} P'(H(x) = c')}$
   8.2 $P'(H(x) = c) = \frac{1}{|B|}\sum_{h \in B} P(h(x) = c)$, where $c \in C$
   8.3 $P(h(x) = c) = \frac{P(h(x) = c)}{\sum_{c' \in C'} P(h(x) = c')}$
9  restoreLabels($D$)
10 return $H$:CCC(B)
```

Fig. 1. Cascading Customized Couple induction algorithm

According to the above discussion, we propose the Cascading Customized Couple (CCC) induction algorithm to learn a CCC classifier, as shown in Figure 1.

The CCC algorithm builds a CCC classifier with its two inputs: the original training set *D* and a base learner L(), i.e., NB. Because the algorithm performs labeling on *D*, the algorithm initially saves all original labels of examples in the training set *D* by saveLabels() at Step 1 while it restores all original labels of examples at Step 9 by restoreLabels() after it builds the couple of approximate CC classifiers or simply CC classifiers without any confusion.

At Step 2, *B* is initialized as an empty set, which is used for collecting the resulting CC classifiers. The first CC learner h_1 is built on *D* at Step 3 using the base learner L() = NB() to build a traditional NB classifier. At Step 4, the misclassifications *E* of h_1 on *D* are computed, and the correct classifications CT on D are obtained by removing the misclassifications *E* from *D*.

At Step 5, if |CT| = |*D*|, then the first CC fits in *D*, CCC does not build the second CC for classifying training errors. Otherwise, the algorithm classifies those misclassifications contained in *E* from Steps 6 to 7.

At Step 6, addClasses() is used for adding artificial class labels to the original domain *D*, and re-label those correct classifications obtained at Step 4 to the corresponding additional classes in terms of their original class labels. In the last phase, at Step 8, the learning algorithm defines a CCC classifier, which is an ensemble learner containing the resulting CC classifiers in *B*, with a modified averaged combination schema for decisions, where *C* is a set of original class labels while *C′* is a set of original class labels and artificial class labels.

```
addClasses algorithm
input   S: subdomain;
        i: beginning index
        D: original domain
output  D': a new domain with additional classes
begin
1  L = $\emptyset$, L'= $\emptyset$, j = 0, k = i
2  foreach p $\in$ S
3      c = p.classLabel
4      if(c $\notin$ L)
5          L[j] = c      // current class label
6          L'[j] = c.k   // c.k is a new class label
7          p.classLabel = L'[j]
8          k++; j++
9      else
10         p.classLabel = L'[j'], where L[j'] = c
11 addClassLabels(L', D)
end.
Proc addClassLabels(L', D)
begin
12   i = 0; k = |D.classes|
13   foreach c = L'[i]
14      D.classes[k + i] = c
15      i++
end
```

Fig. 2. addClasses Algorithm in CCC Algorithm

The CCC induction algorithm defines additional classes on a training set for building cascaded CC classifiers. These additional classes help define the hyperplanes of a CC classifier to classify its sub-domain. Additional classes are defined by invoking the procedure addClasses(), as shown in Figure 2, where input *i* is used as an indicator variable for defining new additional classes.

The addClasses procedure from Step 2 to Step 10 re-labels each instance *p* in *S* with a new additional class label *c.k*. But *p* will be re-labelled with the same label if the current label of *p* is the same as the current label of another instance. L records all the current labels and L′ records all the new additional class labels. Finally, at Step 11, the procedure extends the list of original class labels of D with new additional class labels.

saveLabels() at Step 1 and restoreLabels() at Step 9, as shown in Figure 1, can be simply implemented in linear time complexity. Also, addClasses() adds additional class labels into D for those correctly classified examples in CT in a linear time complexity. A base learner NB is a traditional induction algorithm, which has a linear training time O(*mn*)[6]. CCC has the time complexity O(*kmn*) for misclassifications, where k is the number of the original classes. Therefore, the CCC has a linear time complexity O(kmn) for training.

Because CCC changes the class labels of examples for training cascaded CC classifiers, it only requires an extra space O(n) for saving the original class labels of examples. Therefore, a customized NB classifier requires the space complexity of O((k+k')mv+n) for its parameters and the original labels of examples during the training time, where k' is the number of additional classes. The space complexity of the resulting CCC with a base learner NB for a nominal case is O((k+k')mv) while the space complexity for a continuous case is O((k+k')m).

3.4 An Example

Given a dataset V with two classes, i.e., the square class (minority), denoted as '□', and the diamond class (majority), denoted as '◊', as shown in Figure 3(a), we show how CCC works on this synthetic dataset.

CCC with a base NB builds its first classifier. CCC runs the learned NB to correctly classify most examples, which are assigned to a minus class, denoted as a '–', and a solid dot class, denoted as '•', for the original diamond class and the original square class, respectively, as shown in Figure 3(b). The learner also misclassifies a few diamond examples and a few square examples. As we can see, the NB classifier as a linear classifier divides the original domain with a straight line.

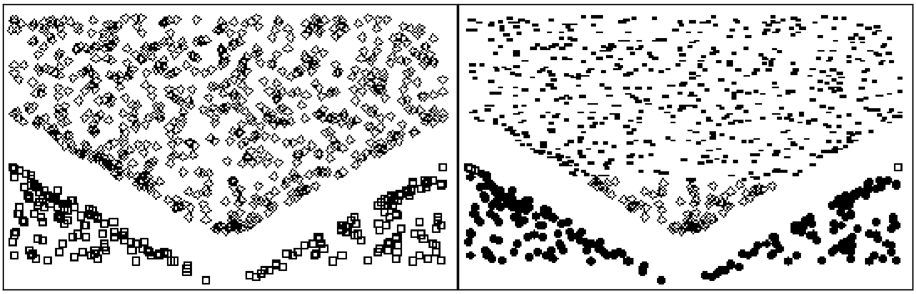

(a) (b)

Fig. 3. An Example of CCC with a base NB

CCC continues by building the second classifier and classifying the remaining misclassifications on the new sub-domain after all correct classifications are re-labeled with additional class labels, i.e., '–' and '•' for those classified data points belonging to the diamond class and the square class, respectively.

Finally, the resulting CCC is composed of two CC classifiers: the first one is can be regarded as the level-0 model outputting correct classifications and misclassifications instead of probability vectors in the traditional CL while the second one can be regarded as the level-1 model built on the customized domain. The combination rule defined at Steps 8, 8.1, 8.2, and 8.3 in CCC consists of two normalizations. Clearly, this new strategy is quite different from the traditional CL, Bagging, and AdaBoost techniques.

4 Experiments

We conducted experiments to evaluate the proposed new Meta learner CCC on 33 datasets. 32 benchmark datasets are chosen from the UCIKDD repository [12] and another one is a synthesized dataset, which is obtained from a scientific application [21]. The scientific application is described as follows: a possible method of explosion detection for the Comprehensive nuclear-Test-Ban-Treaty [21][22] consists of monitoring the amount of radioxenon in the atmosphere by measuring and sampling the activity concentration of Xe-131m, Xe-133, Xe-133m, and Xe-135 by radionuclide monitoring. Several samples are synthesized under different circumstances of nuclear explosions, and combined with various measured levels of normal concentration backgrounds to synthesize a training dataset, which is called Explosion, for use with machine learning methods. The characteristics of these datasets are described in Table 1, where the columns are the names of the datasets, the number of attributes (#attr), the number of instances (#ins), the number of classes (#c).

To justify the proposed Meta classifier CCC to scale up NB, we chose four Meta learning algorithms for scaling up Naïve Bayes (NB) [18], i.e., Stacking [28], AdaBoost [9], MultiBoostAB [25], and Bagging [2]. These induction algorithms are chosen from the Waikato Environment for Knowledge Analysis (Weka) tools [27].

NB is set with Gaussian Estimator for continuous values and Maximum Likelihood Estimator for nominal values; Stacking is set with 10 folds and NB as the base learner and Meta learner, thus denoted as SNB; AdaBoost runs with the base learner NB and 10 models, thus denoted as BNB; MutilBoostAB is set with 10 iterations and the base learner NB, thus denoted as MNB; Bagging is set with 10 iterations and the base learner NB, thus denoted as BgNB. Other parameters of these learning algorithms are set with their default settings.

Table 1. The characteristics of datasets

Datasets	#attr	#ins	#c	Datasets	#attr	#ins	#c
Anneal	39	898	5	Lymph	19	148	4
Audiology	70	226	24	Mushroom	23	8124	2
Autos	26	205	6	P-tumor	18	339	21
Balance-s	5	625	3	Segment	20	2310	7
Breast-w	10	699	2	Sick	30	3772	2
Colic	23	368	2	Sonar	61	208	2
Credit-a	16	690	2	Soybean	36	683	18
Diabetes	9	768	2	Splice	62	3190	3
Glass	10	214	6	Vehicle	19	846	4
Heart-s	14	270	2	Vote	17	435	2
Hepatitis	20	155	2	Vowel	14	990	11
Hypothyroid	30	3772	4	Waveform	41	5000	3
Ionosphere	35	351	2	Zoo	18	101	7
Iris	5	150	3	Adult	15	48842	2
kr-vs-kp	37	3196	2	Shuttle	10	58000	7
Labor	17	57	2	Explosion	5	92630	2
Letter	17	20000	26				

Table 2. Performance (Accuracy) of CCC, NB and three Meta classifiers on 30 benchmark datasets. Those strings such as 'ww' represent the results of the corrected paired t-test (first) and Wilcoxon rank test (second), respectively. '-' is the case for tie.

Datasets	CCC	NB		SNB		BNB		MNB		BgNB	
Anneal	95.43	86.41	ww	35.42	ww	93.65	-w	87.92	ww	86.75	ww
Audiology	72.13	72.12		60.86	ww	79.24	ll	72.12		72.12	
Autos	63.87	54.94	ww	38.79	ww	55.40	-w	59.55		56.64	-w
Balance-s	90.71	90.71		91.67		90.63		90.87		90.31	
Breast-w	96.07	96.14		96.14		95.35	-w	96.07		96.21	
Colic	83.00	78.61	-w	79.02	-w	77.04	ww	79.72	-w	78.48	-w
Credit-a	81.74	78.04	ww	78.41	ww	81.74		79.06	ww	78.12	ww
Diabetes	75.39	75.46		75.39		76.04		76.24		75.72	
Glass	48.85	47.42		28.94	ww	47.42		47.42		48.15	
Heart-s	84.81	84.26		84.44		83.52		84.63		84.07	
Hepatitis	84.60	83.00		83.31		83.90		84.63		83.65	
Hypothyroid	97.26	95.27	ww	94.07	ww	95.27	ww	95.39	ww	95.37	ww
Ionosphere	92.32	83.05	ww	83.34	ww	91.18		91.03		82.48	ww
Iris	94.67	95.00		95.00		95.67		95.67		95.67	
kr-vs-kp	94.79	87.81	ww	88.08	ww	94.82		89.57	ww	87.75	ww
Labor	93.67	94.67		93.83		93.00		95.50		95.50	
Letter	67.44	64.04	ww	56.16	ww	64.04	ww	64.55	ww	64.11	ww
Lymph	80.02	82.36		72.98	-w	79.40		82.36		82.71	
Mushroom	99.62	95.78	ww	96.98	ww	100.00	ll	99.59		95.78	ww
P-tumor	46.91	49.41	-l	26.69	ww	49.41	-l	49.41	-l	48.37	
Segment	85.19	80.30	ww	82.88	ww	80.30	ww	80.39	ww	80.41	ww
Sick	96.55	92.92	ww	92.70	ww	93.53	ww	93.23	ww	92.66	ww
Sonar	79.61	69.04	ww	69.27	ww	80.58		75.99	-w	69.74	ww
Soybean	92.38	92.90		92.46		91.88		93.04	-l	92.68	
Splice	95.91	95.49	-w	95.38	ww	93.97	ww	95.27	ww	95.44	-w
Vehicle	50.94	45.68	ww	45.74	ww	45.68	ww	45.68	ww	45.27	ww
Vote	93.56	90.22	ww	90.22	ww	95.97	ll	91.38	-w	90.11	ww
Vowel	73.23	63.43	ww	65.00	ww	79.75	ll	69.29	-w	63.84	ww
Waveform	80.64	79.96	ww	82.37	ll	79.96	ww	80.32	-w	79.99	ww
Zoo	97.05	95.09		95.55		97.05		97.55		95.09	
Average	**82.95**	79.98		75.70		82.18		81.45		80.11	

To compare the CCC with previous approaches to scale up NB, we also chose five algorithms to build the corresponding classifiers, i.e., SBC, TAN, NBTree, AODE, and HNB. Because the ODE classifiers such as AODE and HNB only can work on nominal cases, numeric attributes are discretized and missing values are replaced by using the unsupervised methods Discretize and ReplaceMissingValues in Weka, respectively, before experiments can be done.

Experiments were conducted by 2 runs of the 10-fold cross validation. For each round, classifiers were built on 9 folds, and were tested on the holdout fold. The process was repeated 10 times, and the results were averaged. We used the paired t-test and the Wilcoxon signed rank test for significance testing on the results (accuracies) from two classifiers. The Wilcoxon signed-rank test or Wilcoxon test is a non-parametric statistical hypothesis test for two repeated measurements under the assumption of independence of the differences. It is an alternative to the paired t-test when these measurements cannot be assumed to be normally distributed. Wilcoxon test is used for single training set rather than multiple datasets [4].

We first conducted experiments to compare CCC with the selected Meta learners to scale up NB. The results on first 30 benchmark datasets are shown in Table 2, where the columns of the corresponding approaches are followed by additional columns, which depict statistical test results ‘w ‘or ‘l’; ‘w’ represents a win of CCC against the corresponding approach while ‘l’ represents a loss of CCC against the corresponding approach; otherwise, both approaches are tied; the averaged accuracies are also reported in the bottom of the table.

As we can see, CCC can improve NB in many cases with respect to the paired t-test. CCC only degrades NB on P-tumor with respect to the Wilcoxon test. CCC also outperforms other Meta learners in most cases. In particular, CCC does not lose to MNB and Bagging in any case with respect to the paired t-test. We summarize all the results of the paired t-test, as shown in Table 3, where ##/##/## represents the numbers of win, tie, and loss of CCC over other Meta learning approaches, respectively. As we can see from the row ‘CCC’, it is clearly shown that CCC outperforms other Meta approaches to scale up NB.

Experimental results on three large datasets are shown in Table 4. The results show that CCC degrades NB and is inferior to other Meta learners only on the Adult case. In Explosion, CCC is tied with the other Meta learners due to a large variance in their results with respect to the paired t-test although the accuracy of CCC is higher than that of other Meta learners. We further analyze the Adult case, where the class distribution is 11687:37155. Our experiment is also to calculate the True Positive Rate (TPR) of the minority class. As a result, those TPR from CCC, NB, SNB, BNB, MNB, BgNB are 0.8117, 0.5116, 0.534, 0.5116, 0.5138, and 0.511, respectively. This shows that CCC can more precisely classify the minority class than other Meta learners on this case. We emphasize that this is rational in many practical applications [3].

Table 3. Summary of paired t-test

	NB	SNB	BNB	MNB	BgNB
SNB	6/21/3				
BNB	1/21/8	3/13/14			
MNB	0/23/7	2/17/11	6/22/2		
BgNB	0/30/0	3/21/6	8/21/1	6/24/0	
CCC	15/15/0	18/11/1	8/18/4	9/21/0	14/16/0

Table 4. Performance of CCC, NB, and other Meta classifiers on three Large datasets

Dataset	CNB	NB		SNB		BNB		MNB		BgNB	
Adult	81.80	83.25	ll	83.34	ll	83.25	ll	83.29	ll	83.23	ll
Shuttle	96.03	93.01	ww	33.01	ww	92.82	ww	93.20	ww	92.93	ww
Explosion	99.15	91.02		99.68		91.02		91.02		91.12	
Average	89.98	86.82		72.93		87.32		87.24		86.85	

We compared CCC with previously proposed approaches such as SBC, TAN, AODE, and HNB, which scale up NB by alleviating the conditional independence assumption. As mentioned before, experiments were done after numeric attributes were discretized with the Discretize method and missing values were replaced by using the ReplaceMissingValues method in Weka. Some experimental results are reported in Table 5.

Table 5. Performance (Accuracy) of CCC, NB, and previous approaches for improving NB

Datasets	CNB	NB		SBC		TAN		NBTree		AODE		HNB	
Balances	91.12	91.20		91.20		87.28	ww	91.20		89.84	-w	90.01	
Breastw	97.21	97.28		96.57	ww	94.64	ww	97.21		96.93		96.21	
Colic	81.91	79.17	-w	83.82		80.02	-w	80.02		80.95		81.08	
Credita	85.29	84.71		84.42		85.00		85.07		86.01		84.93	
Diabetes	75.66	76.04		76.76		75.07		75.33		76.95		76.69	
Hearts	84.81	84.63		80.00	ww	78.89	-w	82.59	ww	83.89	-w	82.41	
Hepatitis	84.25	83.60		82.27		84.54		80.73	ww	84.56		82.29	
Iris	95.33	94.67		96.67		91.00	ww	95.00		95.33		93.00	-w
krvskp	94.79	87.81	ww	94.34		92.10	ww	98.26	ll	91.27	ww	92.27	ww
Labor	98.17	98.17		82.67	ww	90.17	-w	97.33		94.67		92.83	-w
Lymph	84.38	84.02		77.62	ww	85.48		83.07		86.38		82.69	
Splice	95.91	95.49	-w	52.57	ww	52.57	ww	95.49		96.00		59.11	ww
Vote	93.78	90.22	ww	95.63	-l	94.71		94.60		94.48		94.25	
Zoo	95.05	94.05		91.59		94.09		94.59		94.55		98.05	
Average	89.83	88.65		84.72		84.68		89.32		89.42		86.13	

As we can see, CCC is very competitive with these approaches to scale up NB in these cases although CCC might lose to these approaches (other approaches, e.g., AODEsr [30] and WAODE [14], are omitted without loss of generality) in other cases (omitted due to space limitation). In the Splice case, SBC, TAN, and HNB unexpectedly failed to scale up NB. CCC is more successful than SBC, TAN, and HNB in this case if the uncorrelated attribute 'Instance Name' remains in the training set [29]. The results in Table 5 are quite attractive in practical applications because CCC only builds two individual NB models. CCC is much more efficient and has much less space demand than previously proposed approaches including recent ODE classifiers such as AODE and HNB.

5 Conclusion and Future Work

It has been observed that previously proposed Meta learning techniques, Bagging, AdaBoost, and MultiBoostAB, failed to scale up Naïve Bayes (NB) due to the stability of NB. In this paper, we propose a new Meta learning technique to improve NB. This technique is different from recent research which focuses on One-Dependent Estimation (ODE) techniques such as AODE and HNB. The new Meta learner adopts the Domain-based Cascade Learning (DCL), which is regarded as an altered strategy for traditional Cascade Learning for building diverse NB models. We propose the Cascading Customized Couple (CCC) algorithm, which only builds a couple of NB. Our analysis and experimental results show that CCC is more successful than the previously proposed Meta learning techniques used to scale up NB, and more efficient than those ODE techniques. CCC is also very competitive with the ODE techniques in some cases. This is very attractive in practical applications such as text classification in that one is confronted with large datasets.

Because it is observed in Table 2 that traditional Meta learning techniques such as AdaBoost can be more successful than CCC in some cases, it is suggested that CCC might be further enhanced by incorporating with the Boosting techniques for Bayesian learning.

References

[1] Baur, E., Kohavi, R.: An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine Learning 36, 105–139 (1999)

[2] Breiman, L.: Bagging Predictors. Machine Learning 24(3), 123–140 (1996)

[3] Chawla, N.V., Japkowicz, N., Kolcz, A.: Editorial to the special issue on learning from imbalanced data sets. ACM SIGKDD Explorations 6(1), 1–6 (2004)

[4] Demsar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)

[5] Domingos, P., Pazzani, M.: Beyond independence: Conditions for the optimality of the sample Bayesian classifier. Machine Learning 29, 103–130 (1997)

[6] Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis. A Wiley Intersience Publication, Hoboken (2000)

[7] Elkan, C.: Boosting and Naïve Bayesian Learning. Technical Report CS97-557, University of California, Davis (1997)

[8] Fahlman, S., Lebiere, C.: The cascade-correlation learning architecture. In: Touretzky, D. (ed.) Advances in Neural Information Processing Systems, vol. 2, pp. 524–532. Morgan Kaufman, San Mateo (1990)

[9] Freund, Y., Schapire, R.E.: A short Introduction to Boosting. Journal of Japanese Society for Artificial Intelligence 14(5), 771–780 (1999)

[10] Friedman, N., Geiger, D., Goldszmith, M.: Bayesian network classifiers. Machine Learning 29, 131–163 (1997)

[11] Gama, J., Brazdil, P.: Cascade generalization. Machine Learning 41, 315–343 (2000)

[12] Hettich, S., Bay, S.D.: The UCI KDD Archive. University of California, Department of Information and Computer Science, Irvine, CA (1999), `http://kdd.ics.uci.edu`

[13] Jacobs, R., Jordan, M., Nowlan, S., Hinton, G.: Adaptive Mixtures of Local Experts. Neural Computation 3, 79–97 (1988)

[14] Jiang, L., Zhang, H.: Weightily Averaged One-Dependence Estimators. In: Proceedings of the 9th Biennial Pacific Rim International Conference on Artificial Intelligence, pp. 970–974 (2006)

[15] Jordan, M., Jacobs, R.: Hierarchical Mixtures of Experts and the EM Algorithm. Neural Computation 6, 181–214 (1994)

[16] Kohavi, R.: Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid. In: Proceedings of the Second International conference on Knowledge Discovery and Data Mining, pp. 202–207 (1996)

[17] Langley, P.: Induction of recursive Bayesian classifiers. In: Brazdil, P.B. (ed.) ECML 1993. LNCS, vol. 667, pp. 153–164. Springer, Heidelberg (1993)

[18] Langley, P., Iba, W., Thompson, K.: An analysis of Bayesian classifiers. In: Proceedings of the 10th National Conference on Artificial Intelligence, pp. 223–228. AAAI Press and MIT Press (1992)

[19] Langley, P., Sage, S.: Induction of selective Bayesian classifiers. In: Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, pp. 399–406. Morgan Kaufmann, San Francisco (1994)

[20] Rennie, J., Shih, L., Teevan, J., Karger, D.: Tackling the Poor Assumptions of Naive Bayes Text Classifiers. In: Proceedings of International Conference on Machine Learning, pp. 616–623 (2003)

[21] Stocki, T.J., Blanchard, X., D'Amours, R., Ungar, R.K., Fontaine, J.P., Sohier, M., Bean, M., Taffary, T., Racine, J., Tracy, B.L., Brachet, G., Jean, M., Meyerhof, D.: Automated radioxenon monitoring for the comprehensive nuclear-test-ban treaty in two distinctive locations: Ottawa and Tahiti. J. Environ.Radioactivity 80, 305–326 (2005)

[22] Sullivan, J.D.: The comprehensive test ban treaty. Physics Today 151 (1998)

[23] Ting, K., Zheng, Z.: A study of Adaboost with naive Bayesian classifiers: weakness and improvement. Computational Intelligence 19(2), 186–200 (2003)

[24] Ting, K., Witten, I.: Issues in Stacked Generalization. Journal of Artificial Intelligence Research 10, 271–289 (1999)

[25] Webb, G.I.: MultiBoosting: A technique for combining boosting and wagging. Machine Learning 40(2), 159–196 (2000)

[26] Webb, G.I., Boughton, J., Wang, Z.: Not So Naive Bayes: Aggregating One-Dependence Estimators. Machine Learning 58(1), 5–24 (2005)

[27] Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)

[28] Wolpert, D.: Stacked generalization. Neural Networks 5, 241–260 (1992)

[29] Zhang, H., Jiang, L., Su, J.: Hidden Naive Bayes. In: Twentieth National Conference on Artificial Intelligence, pp. 919–924 (2005)

[30] Zheng, F., Webb, G.I.: Efficient lazy elimination for averaged-one dependence estimators. In: Proceedings of the 23th International Conference on Machine Learning, pp. 1113–1120 (2006)

Semi-supervised Probability Propagation on Instance-Attribute Graphs

Bin Wang and Harry Zhang

Faculty of Computer Science, University of New Brunswick
P.O. Box 4400, Fredericton, NB, Canada E3B 5A3
bin.wang@unb.ca

Abstract. Graph-based methods have become one of the most active research areas of semi-supervised learning (SSL). Typical SSL graphs use instances as nodes and assign weights that reflect the similarity of instances. In this paper, we propose a novel type of graph, which we call *instance-attribute graph*. On the instance-attribute graph, we introduce another type of node to represent attributes, and we use edges to represent certain attribute values. The instance-attribute graph thus moreexplicitly expresses the relationship between instances and attributes. Typical SSL graph-based methods are nonparametric, discriminative, and transductive in nature. Using the instance-attribute graph, we propose a nonparametric and generative method, called *probability propagation*, where two kinds of messages are defined in terms of corresponding probabilities. The messages are sent and transformed on the graph until the whole graph become smooth. Since a labeling function can be returned, the probability propagation method not only is able to handle the cases of transductive learning, but also can be used to deal with the cases of inductive learning. From the experimental results, the probability propagation method based on the instance-attribute graph outperforms the other two popular SSL graph-based methods, *Label Propagation* (LP) and *Learning with Local and Global Consistency* (LLGC).

Keywords: Semi-supervised Learning, Graph-based Methods, Factor Graph, Instance-attribute Graph, Probability Propagation.

1 Introduction

Over the past decade, semi-supervised learning (SSL) has emerged as an exciting direction in machine learning research. SSL algorithms solve classification problems when only a relatively small percentage of training data is labeled. Hence, SSL has drawn the attention of the machine learning community, particularly for practical applications, such as natural language processing and bioinformatics, where hand-labeling the data is expensive but unlabeled data are abundant and easily obtained. One of the most active research areas of SSL is graph-based SSL. However, even though proper graph construction is the important issue for any graph-based application, there are relatively few literatures available that deal

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 161–172, 2010.

with graph construction. Graph design is more of an art than a science [5]. Typically, graphs for graph-based SSL methods are defined so that nodes represent labeled and unlabeled instances in the dataset and edges are assigned weights reflecting the similarity of instances.

In this paper, we propose a novel type of graph, which we call *instance-attribute graph*, to express the relationship between instances and attributes instead of among instances themselves. Inspired by *factor graph* [1], we treat the attributes as variables and the instances as factor functions over attributes. Therefore, we use two kinds of nodes on the instance-attribute graph: instance nodes represent instances; and attribute nodes represent attributes. The edges between instance nodes and attribute nodes demonstrate the certain values between instances and attributes.

In typical SSL graph-based methods, where graphs only have instance nodes, we usually assume label smoothness over the graph. On the graph, any nodes connected by an edge with a large weight tend to have the same label so that the labels can be propagated throughout the graph. Therefore, graph methods are nonparametric and discriminative in nature. On the other hand, SSL graph-based methods are intrinsically transductive. That is, their methods only return the values of labeling functions instead of returning the functions themselves. It is not easy to extend graph-based methods to be inductive because adding the unseen instance nodes would destroy the graph that has already been built and learned.

The instance-attribute graph, alternatively, offers a nonparametric and generative method called *probability propagation*, which we explore in this paper for application in SSL classification. The probability propagation method attempts to estimate a labeling function that could potentially be defined on an instance-attribute graph. There are two kinds of messages, instance messages and attribute messages, which are defined according to corresponding probabilities. On the graph, attribute nodes and instance nodes have their own states. When the states of nodes are not consistent, messages will be sent and transformed between instance nodes and attribute nodes until the whole graph is smooth. Since the labeling function can finally be returned from the algorithm, the probability propagation method not only is able to handle the cases of transductive learning, but also can be used to deal with the cases of inductive learning. We conduct the experiments based on UCI datasets to compare the probability propagation method with two popular SSL graph-based methods, *Label Propagation* [2] and *Learning with Local and Global Consistency* [3]. The experimental results show that the probability propagation method outperforms these two graph-based methods in SSL scenario.

We have organized the rest of this paper as follows. In Section 2, we briefly introduce some related works on SSL graph-based methods and some inference methods based on the factor graph. In Section 3, we propose the instance-attribute graph. In Section 4, we describe the probability propagation method. In Section 5, we present the experiments of the probability propagation method against other two graph-based methods. Finally, we give the conclusion and project future work.

2 Related Works

2.1 SSL Graph-Based Methods

There are several approaches to graph construction for SSL. Some construct graphs using domain knowledge. For example, Balcan et al. [4] build graphs for video surveillance using strong domain knowledge, where the graph of webcam images consists of time edges, color edges and face edges. Alternatively, some construct graphs using kNN design [5], where each node is connected to its k-nearest-neighbor under some distance measure. One of the most popular kinds of graphs used for SSL is *exp*-weighted graphs [2] [3]. In an *exp*-weighted graph, nodes i and j are connected by a weighted edge, where the weight w_{ij} is calculated by a Gaussian function $w_{ij} = exp(-d(i,j)^w/\alpha^2)$.

Many graph-based methods propagate labels from labeled nodes to unlabeled nodes on the graph. *Label Propagation* (LP) [2] is a typical example. The algorithm of LP is an iterative procedure, where all the nodes continue to propagate their labels to their neighbors until the whole graph is smooth. In this procedure, the class labels of labeled nodes are clamped so that labeled nodes are able to constantly "push" the class boundary through the high density region to the low density region.

Many graph-based methods can be expressed in a regularization framework that consists of a loss function and a regularizer. The method of *Learning with Local and Global Consistency* (LLGC) [3] is a typical one, where it uses the loss function $\sum_{i=1}^{n}(f_i - y_i)^2$ and the normalized Laplacian in the regularizer. The underlying idea of LLGC is to let each node iteratively spread its label information to its neighbors while retaining its initial information at the same time. The algorithm of LLGC is also an iterative procedure, which will not end until a globally stable state is achieved on the graph.

Some of other works on SSL graph-based methods include Zhu et al. [6], who propose the Gaussian random fields and harmonic function method that uses a quadratic loss function with infinity weight and a regularizer based on the graph combinatorial Laplacian; and Joachims [7], who proposes the spectral graph transducer.

2.2 Factor Graph and Related Methods

Typical graphs used by graph-based methods only have one kind of nodes to represent variables. Such a graph defines a potential global function for variables. Usually, a function can be expressed as a product of factors over subsets of variables. For example, we can factorize the joint distribution $P(\mathbf{X})$ over a set of variables in the form of product of factors: $P(\mathbf{X}) = \prod_s f_s(\mathbf{X}_s)$, where $\mathbf{X}$ denotes the entire set of variables, $\mathbf{X}_s$ denotes a subset of the variables, and f_s denotes a factor which is a function of a corresponding set of variables $\mathbf{X}_s$. Factor graphs make factorization more explicit by introducing a different kind of nodes for the factors, called factor nodes, in addition to the nodes that represent the variables. Undirected edges connect each factor node to all the variable nodes on which that factor depends.

We can then interpret the factor graph from the statistic physics point of view: on a factor graph, the nodes each have local states, and those local states define the energy of the factor graph; we can thus infer from the graph the minimized energy which is represented by the stable states of nodes. On the other hand, inference methods for factor graphs often involve message propagation. Once a message is sent from a node, that message always denotes a change for the local state of that node. The change often ruins the global stability of the graph and increases its energy. Inference methods utilize a set of message propagations and transformations in order to make graphs stable again and to minimize the energy on the fixed points of local states. Well-known inference algorithms include the sum-product algorithm [1] and the belief propagation [8].

Braunstein et al. [9] adopt the factor graph representation to tackle the satisfiability problem (SAT). The Boolean variables are linked to variable nodes on a factor graph, and the clauses are linked to factor nodes. A factor node a is connected to a variable node i by an edge whenever the variable x_i (or its negation) appears in the clause a. There are also two kinds of messages defined in the factor graph, "cavity biases" and "cavity fields". For evaluating the messages, a warning propagation algorithm is proposed, which is similar to the sum-product algorithm. This algorithm converges at a set of fixed points of cavity biases when the factor graph uses a tree-structure. Finally, the set of assignments for all the variables can be found by that set of cavity biases.

3 Instance-Attribute Graph

In this paper, we propose *instance-attribute graph* for graph-based SSL. As we have discussed, factor graphs introduce factor nodes to explicitly represent factor functions over variables. In the datasets used for machine learning applications, each instance typically consists of a set of attributes. If we regard the attributes as variables, the instances can be described as factor functions over attributes. Therefore, we build the instance-attribute graph to express the relationship between instances and attributes. An instance-attribute graph thus consists of two kinds of nodes, where the instance nodes represent instances and attribute nodes represent attributes. The edges between instance nodes and attribute nodes represent the certain values of such attributes.

Let us consider a simple example of an instance-attribute graph. Suppose we have a dataset, in which there are 4 instances: $\{E_1, E_2, E_3, E_4\}$, and each instance consists of 2 binary attributes: $\{a, b\}$. The attribute values of instances are shown as: $E_1 = \{0, 1\}$, $E_2 = \{1, 0\}$, $E_3 = \{1, 1\}$, $E_4 = \{0, 0\}$. Following the above definition, the instance-attribute graph for this dataset is built as in Figure 1.

Notice that the instance-attribute graph is different from the factor graph used either by the sum-product algorithm [1] or by the SAT problem [9]. On the factor graph used by the sum-product algorithm, factor nodes only connect variable nodes, for which representing variables appear in the factors. Similarly, on the factor graph used for the SAT problem, variable nodes only connect to function

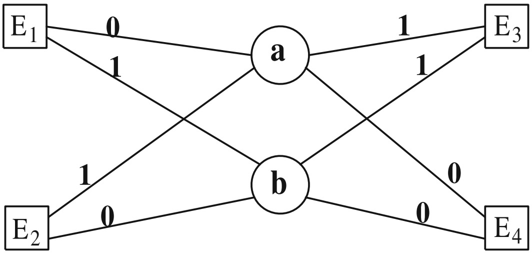


Fig. 1. An example of instance-attribute graph

nodes that contain those variables in the corresponding clauses. On the instance-attribute graph, however, one instance node connects all of the attribute nodes[1]. Moreover, the edges on the factor graph used by the sum-product algorithm do not include any information about attribute values; and the edges on the factor graph built for the SAT problem are represented in only two ways (i.e. solid edges denote the variables themselves and dash edges denote their negations). However, the edges on the instance-attribute graph are assigned corresponding attribute values.

4 Probability Propagation Based on Instance-Attribute Graph

Before introducing the method, we first define a general instance-attribute graph in the SSL scenario. Given a dataset $\mathbf{D}$, there are l labeled instances: $\{(E_1, c_1),\ldots (E_l, c_l)\}$, and $n - l$ unlabeled instances: $\{E_{l+1}, E_{l+2} \ldots, E_n\}$, where $c_i \in \mathbf{C}$ denotes the class label of E_i, and $\mathbf{C}$ is the set of possible class labels. The dataset $\mathbf{D}$ has an attribute set $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m\}$.[2] The value of attribute a_j for instance E_i is represented by $\mathbf{a}_j(E_i)$. The instance-attribute graph for dataset $\mathbf{D}$ is shown in Figure 2.

Based on this general instance-attribute graph, we propose the *probability propagation* method for the problem of SSL classification. The probability propagation method aims to estimate a labeling function $L(E_i) = \arg\max_{c \in \mathbf{C}} P(c|E_i)$ that is potentially defined on an instance-attribute graph, where $P(c|E_i)$ is the class membership probability of E_i. By Bayes theorem, the class membership probability $P(c|E_i)$ is calculated as $P(c|E_i) = P(E_i|c)P(c)/P(E_i)$. Assume that the attribute values are conditionally independent given the class label. So we have

$$P(c|E_i) = \frac{P(c)\prod_{j=1}^{m} P(\mathbf{a}_j(E_i)|c)}{P(E_i)}, \tag{1}$$

where $P(c)$ is the prior probability of class label c, m is the total number of attributes, $P(\mathbf{a}_j(E_i)|c)$ is the probability that attribute $\mathbf{a}_j$ has the value of $\mathbf{a}_j(E_i)$

[1] Here we assume that there is no missing value in the dataset.

[2] Here we assume that all the instances consist of the same attributes.

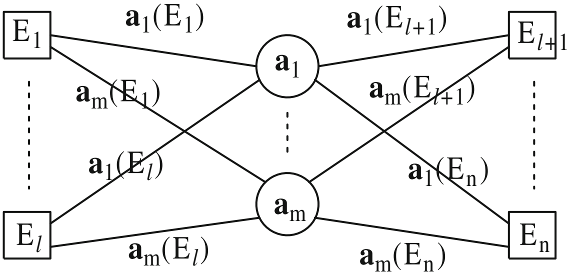


Fig. 2. The general instance-attribute graph

given class label c. In practice we are only interested in the numerator of that fraction, since the denominator does not depend on class labels and can be treated as the constant. So we can rewrite the labeling function as

$$L(E_i) = \arg\max_{c \in \mathbf{C}} P(c) \prod_{j=1}^{m} P(\mathbf{a}_j(E_i)|c). \tag{2}$$

From the Equation 2, we can see that class labels are determined by the product of prior class probability and probabilities $P(\mathbf{a}_j(E_i)|c)$. Coming back to the instance-attribute graph, probabilities $P(\mathbf{a}_j(E_i)|c)$ just reflect the relationship between attribute node $\mathbf{a}_j$ and instance node E_i given class label c through their edge, on which $\mathbf{a}_j(E_i)$ is assigned. Hence, we define the attribute node message $\mu^c_{\mathbf{a}_j \to E_i}$, which is propagated from attribute node $\mathbf{a}_j$ to instance node E_i given class label c, as the probability that $\mathbf{a}_j$ has the value of $\mathbf{a}_j(E_i)$ given c (i.e. $P(\mathbf{a}_j(E_i)|c)$).

After confirming the attribute node message, we next consider how to define the messages sent from instance nodes to attribute nodes. Typically, an instance node message reacts to a coming attribute node message based on its internal property with other attribute node messages from all the neighbors of this instance node except the given attribute node. So we define the instance node message $\mu^c_{E_i \to \mathbf{a}_j}$, which is sent from instance node E_i to attribute node $\mathbf{a}_j$ given class label c, as the probability of c given the value of E_i for $\mathbf{a}_j$ (i.e. $P(c|\mathbf{a}_j(E_i))$.

We now begin to estimate the messages. First, we show how to estimate the instance node message $\mu^c_{E_i \to \mathbf{a}_j}$. Based on the Bayes Theorem, $P(c|\mathbf{a}_j(E_i))$ can be estimated as

$$P(c|\mathbf{a}_j(E_i)) = \frac{P(\mathbf{a}_j(E_i)|c)P(c)}{P(\mathbf{a}_j(E_i))}. \tag{3}$$

Now we focus on probability $P(a_j(E_i)|c)$ in Equation 1. If we multiply $P(c|E_i)$ and Equation 1's denominator $P(E_i)$, and divide the product of all the items of Equation 1's numerator except $P(a_j(E_i)|c)$, then $P(a_j(E_i)|c)$ can be calculated as

$$P(\mathbf{a}_j(E_i)|c) = \frac{P(c|E_i)P(E_i)}{P(c)\prod_{1\le k\le m\wedge k\ne j} P(\mathbf{a}_k(E_i)|c)}. \tag{4}$$

In the instance-attribute graph, the internal properties of instance nodes are their class membership probabilities, i.e. $P(c|E_i)$. The product of a set of probabilities: $\{P(\mathbf{a}_k(E_i)|c)|1 \le k \le m \wedge k \ne j\}$, just combines all the attribute node messages from the neighbors of E_i except the message from the target attribute node $\mathbf{a}_j$. Substitute Equation 4 into Equation 3, we have

$$P(c|\mathbf{a}_j(E_i)) = \frac{P(c|E_i)P(E_i)}{P(\mathbf{a}_j(E_i))\prod_{1\le k\le m\wedge k\ne j} P(\mathbf{a}_k(E_i)|c)}. \tag{5}$$

Note that $P(E_i)/P(\mathbf{a}_j(E_i))$ is a constant when we focus on the estimation of $P(c|\mathbf{a}_j(E_i))$. Thus we ignore this term and estimate instance node message $\mu^c_{E_i\to\mathbf{a}_j}$ as

$$\mu^c_{E_i\to\mathbf{a}_j} = \frac{P(c|E_i)}{\prod_{1\le k\le m\wedge k\ne j} P(\mathbf{a}_k(E_i)|c)}. \tag{6}$$

Next, we show how to estimate the attribute node message $\mu^c_{\mathbf{a}_j\to E_i}$. From the definition, the attribute node message is equal to the estimate of $P(\mathbf{a}_j(E_i)|c)$. In practice, the frequencies of attribute values and class labels have been used in the estimation. The estimate of $\mu^c_{\mathbf{a}_j\to E_i}$ is shown as follows[3]

$$\mu^c_{\mathbf{a}_j\to E_i} = \frac{1+\sum_{i=1}^n \theta^c(a_j^k, E_i)P(c|a_j^k)}{|\mathbf{a}_j| + \sum_{s=1}^{|\mathbf{a}_j|}\sum_{i=1}^n \theta^c(a_j^s, E_i)P(c|a_j^s)}, \tag{7}$$

where n is the number of instances, $|\mathbf{a}_j|$ is the number of possible values of attribute $\mathbf{a}_j$, and θ^c is a function: $\theta^c(a_j^k, E_i) = 1$ iff the value of attribute $\mathbf{a}_j$ is a_j^k in instance E_i and E_i's class label is c; otherwise $\theta^c(a_j^k, E_i) = 0$.

For calculating the labeling function L, we also need to estimate the class prior probability $P(c)$, which is shown as

$$P(c) = \frac{1+\sum_{i=1}^n P(c|E_i)}{|\mathbf{C}| + n}, \tag{8}$$

where $|\mathbf{C}|$ is the number of possible class labels. To avoid the numerators becoming too small or arriving at zero, we use the method of m-estimate [10] in Equation 7 and Equation 8.

By now we are ready to present the procedure for the probability propagation algorithm. After building up the instance-attribute graph, we initialize the attribute node messages and class prior probabilities by labeled instances. For labeled instances, the class membership probabilities are assigned according to the uniform distribution. For labeled instances, their class membership probabilities are assigned as: $P(c|E_i) = 1$ iff E_i's class label is equal to c; otherwise

[3] For convenience, we use $a_j^k = \mathbf{a}_j(E_i)$ to denote the attribute value, where k denote the index of values of attribute $\mathbf{a}_j$.

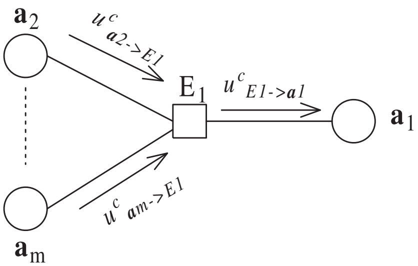


Fig. 3. An example of instance node message propagation

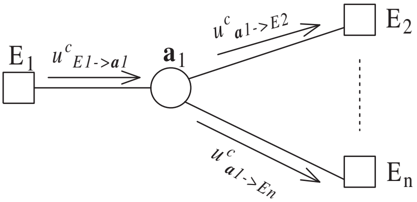


Fig. 4. An example of attribute node message propagation

$P(c|E_i) = 0$. At the beginning, attribute node messages are propagated along edges to the instance nodes. Node E_i, for example, reacts to attribute node message $\mu^c_{\mathbf{a}_j \to E_i}$ by sending instance node message $\mu^c_{E_i \to \mathbf{a}_j}$ estimated by Equation 6. Once the attribute nodes receive the messages from instance nodes, they will update their emitting attribute node messages. The attribute node message, i.e. $\mu^c_{\mathbf{a}_j \to E_i}$, is updated by Equation 7. The procedures of propagations are depicted in Figure 3 and Figure 4. The propagations continue until all the transformations of messages are finished on each edge. Subsequently, the class membership probabilities of each instance is updated based on current attribute probabilities. The class prior probabilities are estimated again based on the updated class membership probabilities as well. Then the attribute nodes send messages again. This iterative procedure does not end until the change of probabilities is smaller than a threshold. The formal procedure of the algorithm is described as follows.

```
Algorithm: Probability Propagation
  Input: An instance-attribute graph
  Output: Labels of unlabeled instances and the labeling function
  begin
    Initialize;
    repeat
        Send the attribute node messages along each edge for each
            class label;
        Calculate instance node messages and update the attribute
            probability estimates using subroutine UPDATE(E);
        Update the class membership probability estimates of
            unlabeled instances;
        Update the class prior probability estimations;
    until: The change of probabilities is smaller than the threshold
    Label the unlabeled instances by Equation 2;
  end
```

The subroutine **UPDATE** is shown as follows.

```
Subroutine: UPDATE(E)
  Input: The set of all the edges E on the graph
  Output: Updated attribute probability estimates
  begin
    repeat
        Randomly select an edge from E;
        for each class label
            Calculate the instance node message sent along
                the given edge for given class label;
            Send the instance node message to the connected
                attribute node;
            Update the attribute node message for given class label;
            Send the updated attribute node messages along the edges
                which have the same attribute value;
        end
        Remove the selected edge from E;
    until: E is empty
    Estimate the attribute probabilities;
  end
```

The procedure for the probability propagation algorithm acts like a conversation between the instance nodes and the attribute nodes through exchanging messages. On one hand, the attribute node messages represent the global states because they are estimated based on the whole class distribution of the graph. Sending an attribute node message is akin to sending a survey that asks the connected instance node whether or not it agrees with the state of the attribute node. On the other hand, the local states of the instance nodes are represented by their class membership probabilities. The estimations of instance node messages are not only based on the local states of instance nodes, but are also influenced by the messages from other attribute nodes. Therefore, an instance node message received by an attribute node can be regarded as the degree to which the connected instance node and other attribute nodes agree. Once an attribute node receives the result of the survey, it will update its state in terms of the opinions from the connected instance node and other attribute nodes. After the entire conversation is finished, the instance nodes update themselves in order to make the instance-attribute graph smoother. The attribute nodes then send their survey again to challenge whether or not the graph has been smooth. If it has been smooth, the algorithm ends; if not, the next round of conversation begins.

As outlined above, the message propagation methods are guaranteed to converge when the factor graph is tree-structured. But the convergence is not guaranteed when there are loops or cycles in the factor graph. If our task is to estimate the exact probabilities from the factor graphs, the graphs containing cycles will make the computations of the message propagation methods infeasible. However, the message propagation methods on the loopy factor graph can still work well for other tasks, for example, decoding the compound codes [11] and approximating the free energy [12]. It has been shown in [8] that, if we ignore the existence of loops, the propagation schemes still have a chance to bring all the messages to some stable equilibrium as time goes on. It is obvious that there are loops in the instance-attribute graph. The probability propagation based on instance-attribute graph also runs the risk of un-convergence. But our task is to label the unlabeled instances instead of estimating the exact probabilities. Thus the probability propagation algorithm can still have good performance on the real-world applications, which will be shown in the following experiments.

5 Experiments

We conduct the experiments to compare the performance of the probability propagation method based on the instance-attribute graph to the performances of two SSL graph-based methods, such as LP and LLGC. The experiments are based on 15 datasets from Weka [13]. Since both LP and LLGC are for transductive learning, we follow the format of transductive learning[4]. We utilize the implementation of LLGC in Weka, and implement the LP algorithm and the probability propagation algorithm in Weka framework. For each dataset, 10% of the dataset is used as the set of labeled instances, and the other instances are used as unlabeled instances. In the experiments, the accuracy scores of each algorithm are obtained via 10 runs of ten-fold cross validation. Runs with each algorithm are carried on the same training sets and evaluated on the same test sets. Finally, we conduct two-tailed t-test with a 95% confidence level to compare each pair of algorithms. The results are shown in Table 1.

Table 1. Experimental results on accuracy for semi-supervised graph-based methods

Dataset	PP	LP	LLGC
zoo	76.45	38.48	41.63
labor	87.4	58.87	56.8
iris	88.67	34.13	41.33
vote	88.97	62.42	66.02
breast-cancer	62.31	70.4	70.29
lymph	59.34	51.05	51.87
hepatitis	73.31	79.38	79.56
balance-scale	58.13	51.68	49.3
glass	42.22	32.42	34.93
heart-h	61.96	62.84	63.34
heart-c	63.9	53.27	51.81
colic.ORIG	59.89	65.66	65.66
heart-statlog	81.67	53.04	56.89
credit-a	83.88	54.78	54.51
colic	71.55	60.49	61.74
t-test	w/t/l	3/2/10	4/2/9

[4] In transductive learning, the entire dataset is divided into two sets: labeled instances set and unlabeled instances set. All of the instances can be seen during the training procedure.

In Table 1, the two-tailed t-test results are shown in the bottom row, where each entry has the format of $w/t/l$. It means that, compared with the probability propagation algorithm, the algorithm in the corresponding column wins w times, ties t times, and loses l times. From the experiments, we observe that the probability propagation algorithm outperforms both LP and LLGC, where LP algorithm wins 3 times and loses 10 times, and LLGC algorithm wins 4 times and loses 9.

From the experimental results, we can see that probability propagation method based on instance-attribute graph outperforms other two methods. As mentioned before, LP and LLGC propagate labels through the weighted edges in the graphs, and the weights are calculated based on the Gaussian kernel. The computation of weights happens only at the point where instances interact with each other at the attribute level. The relationships between instances and attributes are not explicitly utilized. In the probability propagation algorithm, however, an instance node can communicate with other instances through their shared attribute nodes by propagating messages. An instance node has better opportunity to make use of the domain prior knowledge and the structure of dataset.

6 Conclusion and Future Works

In this paper, we propose a novel type of graph, the instance-attribute graph, to express the relationship between instances and attributes in the scenario of SSL. An instance-attribute graph consists of two types of nodes, such as instance nodes and attributes nodes. The edges on the instance-attribute graph connect instance nodes and attribute nodes, and represent the certain values of attributes. Based on the instance-attribute graph, we design the probability propagation method for the problem of SSL classification. In the probability propagation method, two kinds of messages, such as attribute node messages and instance node messages, are defined according to corresponding probabilities. The messages are sent and transformed on the instance-attribute graph until the whole graph is smooth. Since a labeling function can be returned from the algorithm, the probability propagation method not only is able to handle the cases of transductive learning, but also can be used to deal with the cases of inductive learning. From the experimental results, the probability propagation method based on the instance-attribute graph outperforms other two popular SSL graph-based methods, such as LP and LLGC.

In the future, we will continue to study some issues of the probability propagation method based on the instance-attribute graph in SSL scenario. The first issue is the scalability of this method. Since the instance-attribute graph is built by instances and attributes, the increase of the amount of either instances or attributes will cause the computation to become incredibly expensive. Also the convergence of the probability propagation method is an issue because there is no guarantee of convergence for looping graphs. Furthermore, we will compare with other SSL graph-based methods under the circumstance of inductive learning.

References

1. Kschischang, F.R., Member, S., Frey, B.J., Loeliger, H.: Factor graphs and the sum-product algorithm. IEEE Transactions on Information Theory 47, 498–519 (2001)
2. Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation, Technical Report CMU-CALD-02-107, Carnegie Mellon University (2002)
3. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: Advances in Neural Information Processing Systems 16, pp. 321–328. MIT Press, Cambridge (2004)
4. Balcan, M.-F., Blum, A., Choi, P.P., Lafferty, J., Pantano, B., Rwebangira, M.R., Zhu, X.: Person identification in webcam images: An application of semi-supervised learning. In: ICML Workshop on Learning with Partially Classified Training Data (2005)
5. Zhu, X.: Semi-supervised learning with graphs, PhD dissertation, Carnegie Mellon University (2005)
6. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using Gaussian fields and harmonic functions. In: ICML, pp. 912–919 (2003)
7. Joachims, T.: Transductive learning via spectral graph partitioning. In: ICML, pp. 290–297 (2003)
8. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann Publishers Inc., San Francisco (1988)
9. Braunstein, A., Mézard, M., Zecchina, R.: Survey propagation: An algorithm for satisfiability. In: Random Struct & Algorithms, vol. 27, pp. 201–226. John Wiley & Sons, Inc., New York (2005)
10. Mitchell, T.M.: Mahchine learning. McGraw-Hill International Edtions (1997)
11. Kschischang, F.R., Frey, B.J.: Iterative decoding of compound codes by probability propagation in graphical models. IEEE Journal on Selected Areas in Communications 16, 219–230 (1998)
12. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Bethe free energies, Kikuchi approximations, and belief propagation algorithms. Technical report TR-2001-10 (2001), http://www.merl.com/reports/TR2001-16/index.html
13. Witten, I.H., Frank, E.: Data mining - practical machine learning tools and techniques with Java implementation. Morgan Kaufmann, San Francisco (2000)

Offline Recognition of Handwritten Numeral Characters with Polynomial Neural Networks Using Topological Features

El-Sayed M. El-Alfy

College of Computer Sciences and Engineering,
King Fahd University of Petroleum and Minerals,
Dhahran 31261, Saudi Arabia
alfy@kfupm.edu.sa

Abstract. Group-Method of Data Handling (GMDH) has been recognized as a powerful tool in machine learning. It has the potential to build predictive neural network models of polynomial functions using only a reduced set of features which minimizes the prediction error. This paper explores the offline recognition of isolated handwritten numeral characters described with non-Gaussian topological features using GMDH-based polynomial networks. In order to study the effectiveness of the proposed approach, we apply it on a publicly available dataset of isolated handwritten numerals and compare the results with five other state-of-the-art classifiers: multilayer Perceptron, support-vector machine, radial-basis function, naïve Bayes and rule-based classifiers. In addition to improving the classification accuracy and the per-class performance measures, using GMDH-based polynomial neural networks has led to significant feature dimensionality reduction.

Keywords: Machine learning, GMDH, Polynomial networks, Pattern recognition, Handwritten numeral character recognition, Non-Gaussian topological features.

1 Introduction

Automatic recognition of handwritten numerals, as an important part of optical character recognition (OCR), has been a very active research topic over the past twenty years [1], [2]. Many practical applications have been developed such as reading numbers on bank checks, reading zip codes in postal addresses for mail sorting systems, processing of hand-written forms, and recognition of handwriting on computer tablets. Both offline and online recognition have been intensively studied for different languages such as English, Japanese, Chinese, French, Farsi, Arabic, Bangla, and Hindi [2]-[7]. The online systems, where recognition takes place while capturing writing, are less complex than offline systems, where recognition takes place using static pre-stored images, as they can benefit from the temporal information on the order of strokes made by the writer [8].

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 173–183, 2010.

The problem has several challenges resulting from the variations in the same numeral pattern that depends on the writing styles of individuals. It gets more complex for cursive based languages. Several techniques have been proposed using a variety of features and classifiers attempting at improving the classification accuracy. Features used include statistical features such as templates, measurements of density of points, characteristic loci, moments, and mathematical transforms [9]-[11]. It also includes shape features from their skeletons such as loops, endpoints, junctions, arcs, concavities and convexities, and strokes [12].

Among the proposed classifiers are hidden Markov model (HMM) [7], support vector machines [13], fuzzy logic [14], rough sets [15], neural networks [12], [17] and radial-basis function [16]. Although traditional neural networks with back-propagation learning have been proposed to solve the problem among other classifiers, their prohibitive learning time required to converge limits their application to practical problems.

GMDH-based polynomial networks [18], [19] have several advantages as compared to other forms of feed-forward neural networks. It builds simple network models of polynomial functions by automatically selecting relevant inputs, network structure, number and type of neurons in each layer, and neurons' connections and polynomial descriptions. It is recognized for its fast learning and strength in feature dimensionality reduction while building the optimal prediction model.

The aim of this paper is to explore the application of GMDH-based polynomial networks for the recognition of isolated handwritten numeral characters. The recognition performance of the proposed approach will be evaluated using a publicly available dataset and will be compared with five other machine learning approaches: naïve Bayes (NB), rule-based (RB), support vector machine (SVM), multilayer Perceptron (MLP), and radial-basis function (RBF) algorithms. Also it pinpoints the relevant topological features for the recognition of each handwritten numeral.

The rest of the paper is organized as follows. Section 2 describes the isolated handwritten numeral recognition problem and the topological non-Gaussian features. Section 3 describes the polynomial neural network approach. Section 4 describes the database used and the experimental results. It also compares the effectiveness of the proposed approach with other state-of-the-art classifiers. Section 5 concludes the paper and pinpoints some future work.

2 Handwritten Numeral Recognition and Features

The process of offline handwritten numerals recognition starts with a set of pre-stored scanned images. As shown in Fig. 1, handwriting is not perfect; even for the same writer numeral characters can have different deformations and shapes. With machine learning algorithms, there are typically three phases. The first phase is preprocessing of the set of raw images for each class. During this phase if numerals are not isolated, they are segmented and stored separately then normalized and scaled to fit into a preset frame. After that, features are extracted and each image is represented by a feature vector. The resulting processed dataset is split into two portions one for training and the other one for testing. In the second phase, a classification model is built using the training dataset. Finally the model is evaluated on the testing dataset.

In our experiments, numerals are represented by binary images with 1 means black pixel and 0 means white pixel. Each image is 32 × 24 pixel array; see Fig. 2. Images are pre-processed and all numerals are oriented normally and scaled equally in the horizontal and vertical direction so that the top and bottom or left and right of each numeral touch the edges of the array. For recognition purpose, images are processed and each numeral is described by a feature vector consisting of a 16 non-Gaussian topological features.

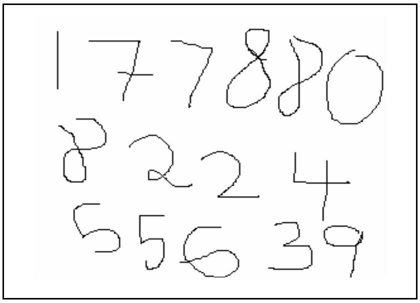


Fig. 1. Illustration of handwritten numeral characters; the same digit can have various forms and deformations

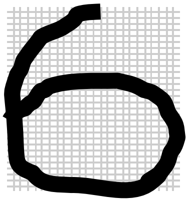


Fig. 2. Illustration of handwritten numeral characters; the same digit can have various forms and deformations

These features, as indicated in Table 1, belong to five types: convexity features, width features, energy distribution features, loop-related features, and slope features. The detailed description of the features can be found in [12]. A brief description is provided next for the sake of completeness and clarity. There are four convexity features x_1, x_2, x_3, and x_6 which correspond to upper-right, lower-right, upper-left, and lower-left directions respectively. A convexity feature is set to 1 if the numeral is convex in the corresponding direction; otherwise it is set to 0. There are three binary features x_4, x_5, and x_7; each indicates whether the maximum width is less than seven pixels in the bottom 6 rows, top 6 rows, and all rows respectively. Loop-related features, x_8, x_9, x_{10}, x_{11}, and x_{12}, describe respectively the number of loops, the location of the top loop, the location of the second loop, the width of the horizontal bar below the top loop, and the number of black pixels above the topmost row having two regions. The energy distribution features (x_{13}, x_{14}, and x_{16}) measure the relative distribution of black pixels in various parts of the numeral image. x_{13} is the ratio between the number of black pixels in the left half to the number of black pixels in the right half of the image. x_{14} is the ratio between the number of black pixels in the top half to the number of black pixels in the bottom half of the image. x_{16} is the ratio between the number of black pixels in the top 10 rows to the number of black pixels in the bottom 10 rows of the image. The slope feature x_{15} denotes the sum of the squared errors for a straight line fit to the rightmost pixels in the image.

Table 1. Non-Gaussian topological features

Category	Designation	Type
Convexity	x_1, x_2, x_3, x_6	Binary
Width-related	x_4, x_5, x_7	Binary
Loop-related	x_8, x_9, x_{10}, x_{11}, x_{12}	Numeric
Engery distribution	x_{13}, x_{14}, x_{16}	Numeric
Slope	x_{15}	Numeric

3 GMD-Based Polynomial Neural Network Model

Group-Method of Data Handling (GMDH) is a supervised machine learning methodology for automatically constructing a neural network of polynomial functions that fits a training dataset of solved examples [19]. GMDH-based networks are self organizing. Thus the number of layers, the number of neurons in each layer, the polynomial for each neuron, and the connections among various neurons are all selected automatically during training to optimize the network performance. The automation of model synthesis not only lessens the burden on the analyst but also safeguards the model generated from being influenced by human biases and misjudgments. During training, the learning algorithm starts with simple regression relationships between the target (*a.k.a.* dependent) variable and the predictor (*a.k.a.* independent) variables. It then proceeds to derive more accurate representations in the next iteration. To prevent exponential growth and limit model complexity, the algorithm selects only relationships having good predicting powers within each phase. Iteration is stopped when the new generation regression equations start to have poorer prediction performance than those of the previous generation, at which point the model starts to become overspecialized and therefore unlikely to perform well with new data. The original idea of GMDH-based algorithm was introduced by an Ukrainian scientist, A. G. Ivakhnenko, in 1968.

To illustrate these steps for the classical GMDH approach, consider an estimation database of n_e observations (rows) and $m+1$ columns for m independent variables (x_1, x_2, ..., x_m) and one dependent variable y. In the first iteration we assume that our predictors are the actual input variables. The initial rough prediction equations are derived by taking each pair of input variables ($<x_i, x_j>: i = 1, 2, ..., m; j = 1, 2, ..., m$) together with the output y and computing the quadratic regression polynomial [19]:

$$y = A + B\,x_i + C\,x_j + D\,x_i^2 + E\,x_j^2 + F\,x_i\,x_j. \tag{1}$$

Each of the resulting *m(m-1)/2* polynomials is evaluated using data for the pair of x variables used to generate it, thus producing new estimation variables (z_1, z_2, ..., $z_{m(m-1)/2}$) which would be expected to describe y better than the original variables. The resulting z variables are screened according to some selection criterion and only those having good predicting power are kept. The original GMDH algorithm employs an additional and independent selection set of n_s observations for this purpose and uses the regularity selection criterion based on the root mean squared error r_k over that data set, where,

$$r_k^{\,2} = \sum_{\ell=1}^{n_s} (y_\ell - z_{k\ell})^2 \Bigg/ \sum_{\ell=1}^{n_s} y_\ell^{\,2} \quad ;k = 1,2,...,m(m-1)/2 \cdot \tag{2}$$

Only those polynomials (and associated z variables) that have r_k below a prescribed limit are kept and the minimum value, r_{min}, obtained for r_k is also saved. The selected z variables represent a new database for repeating the estimation and selection steps in the next iteration to derive a set of higher-level variables. At each iteration, r_{min} is compared with its previous value and the process is continued as long as r_{min} decreases or until a given complexity is reached. An increasing r_{min} is an indication of the model becoming overly complex, thus over-fitting the estimation data and performing poorly in predicting the new selection data. Keeping model complexity checked is an important aspect of GMDH-based algorithms, which keep an eye on the final objective of constructing the model, *i.e.*, using it with new data previously unseen during training. The best model for this purpose is that providing the shortest description for the data available. Computationally, the resulting GMDH model can be seen as a layered network of partial quadratic descriptor polynomials, each layer representing the results of an iteration.

A number of GMDH methods have been proposed which operate on the whole training data set thus avoiding the use of a dedicated selection set. The GMDH learning approach uses the predicted squared error (PSE) criterion for selection and stopping to avoid model overfitting, thus eliminating the problem of determining when to stop training in neural networks. The criterion minimizes the expected squared error that would be obtained when the network is used for predicting new data. The PSE error is calculated as:

$$PSE = FSE + CPM(2K/n)\sigma_p^{\,2}, \tag{3}$$

where *FSE* is the fitting squared error on the training data, *CPM* is a complexity penalty multiplier selected by the user, *K* is the number of model coefficients, *n* is the number of samples in the training set, and σ_p^2 is a prior estimate for the variance of the error obtained with the unknown model. This estimate does not depend on the model being evaluated and is usually taken as half the variance of the dependent variable y [2]. As the model becomes more complex relative to the size of the training set, the second term increases linearly while the first term decreases. *PSE* goes through a minimum at the optimum model size that strikes a balance between accuracy and simplicity (exactness and generality). The user may optionally control this trade-off using the *CPM* parameter. Larger values than the default value of 1 lead to simpler models that are less accurate but may generalize well with previously unseen data, while lower values produce more complex networks that may overfit the training data and degrade actual prediction performance.

GMDH builds networks consisting of various types of polynomial functional elements. The network size, element types, connectivity, and coefficients for the optimum model are automatically determined using well-proven optimization criteria, thus reducing the need for user intervention compared to neural networks. This simplifies model development and reduces the learning/development time and effort. The models take the form of layered feed-forward abductive networks of functional elements (nodes); see Fig. 3. Elements in the first layer operate on various combinations of the independent input variables (x's) and the element in the final layer produces the predicted output for the dependent variable y. In addition to the main layers of the network, an input layer of normalizers convert the input variables into an internal representation as Z scores with zero mean and unity variance, and an output unitizer unit restores the results to the original problem space. GMDH supports the following main functional elements:

(i) A white element which consists of a constant plus the linear weighted sum of all outputs of the previous layer, i.e., the white output, O_{white}, is given by,

$$O_{white} = w_0 + w_1x_1 + w_2x_2 + ... + w_nx_n, \tag{4}$$

where $x_1, x_2,..., x_n$ are the inputs to the element and $w_0, w_1, ..., w_n$ are the element weights.

(ii) Single, double, and triple elements which implement a third-degree polynomial expression with all possible cross-terms for one, two, and three inputs respectively; for example, the double output, O_{Double}, is given by,

$$O_{Double} = w_0 + w_1x_1 + w_2x_2 + w_3x_1^2 + w_4x_2^2 + w_5x_1x_2 + w_6x_1^3 + w_7x_2^3 \tag{5}$$

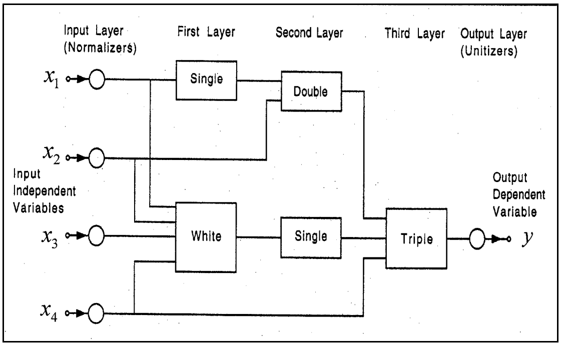


Fig. 3. A typical polynomial neural network model showing various types of functional units

4 Empirical Evaluation

To study the effectiveness of the proposed approach, we performed several experiments using a database of raw images of Arabic numerals that has been collected from 3000 persons by the IRS (Internal Revenue Service) [12]. In our experiments, 300 samples for each numeral are selected randomly. Each sample is described using 16 input features, as explained in Section 2, and labeled with the corresponding numeral value. The dataset was randomly divided into a training set and a testing set. The training set has 2400 samples whereas the testing set has 600 samples. The number of samples for each class in the training and testing datasets is as shown in Table 2.

Table 2. Description of the training and testing datasets

	Class										Overall
	0	1	2	3	4	5	6	7	8	9	
N_{train}	237	254	233	233	241	235	248	236	237	246	2400
N_{test}	63	46	67	67	59	65	52	64	63	54	600

The multiclass classification problem is converted to ten "one-versus-all" binary classification problems. However, the final decision is made by combining the outputs of all networks rather than using a threshold value for each separate model. Using all the training samples, ten networks have been built using abductive reasoning. The structures of these networks are as shown in Fig. 4. Each network model indicates the more relevant features for each class of the numerals. In addition each network model computes a real value as an output. The real-valued outputs of all models are combined to predict the class of the test sample. The network that has the maximum value wins and the corresponding class is taken as the final decision. This helps in breaking ties between various networks decisions.

Table 3 shows the relevant features that have been selected for each numeral and the percentage reduction in the feature vector dimensionality. Table 4 shows the accuracy of each classifier where GMDH refers to the polynomial neural network, SVM refers to support vector machine, MLP refers to multilayer Perceptron, RBF refers to radial-basis function, NB refers to naïve Bayes, and RB refers to rule-based learning. Table 5 shows the per-class performance measures, where *Pr*, *TP*, *FP*, and *FM* refer to precision, true positive rate, false positive rate, and F-measure. The performance measures are expressed as percentage. The detailed definitions and equations for these measures are available in [20]. However, for the sake of completeness brief definitions are as follows:

- *Pr* is the number of correctly recognized samples of a specific class divided by the total number of samples recognized as being of that class. *Pr* is a measure of fidelity of recognition.
- *TP* is the ratio between the number of correctly recognized samples of a specific class and the total number of existing samples that are actually of that class. *TP* is also known as recall.
- *FP* is the ratio between the number of incorrectly recognized samples as being of a specific class and the total number of existing samples that are not actually of that class.
- *FM* is the harmonic mean of precision and recall.

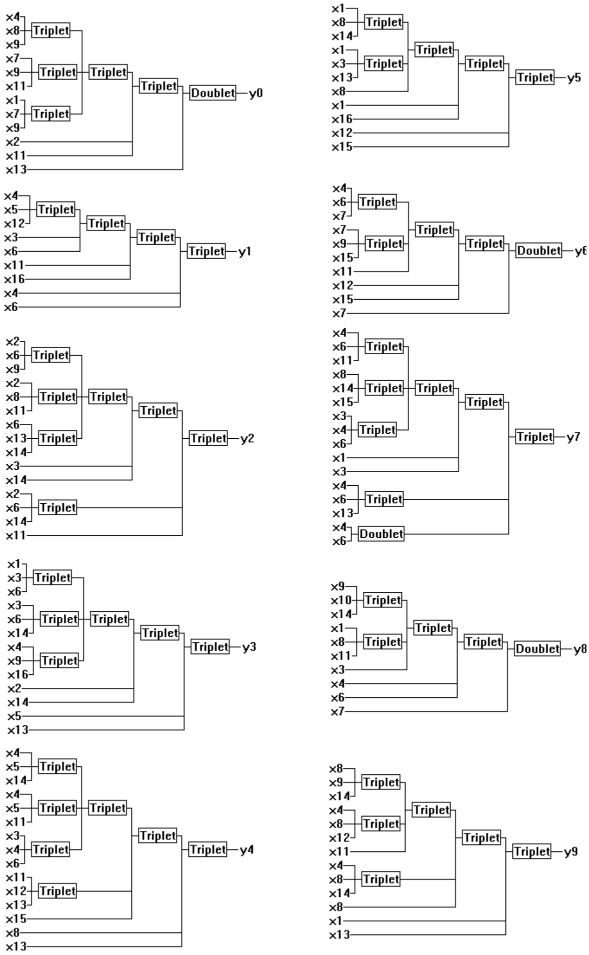


Fig. 4. Polynomial neural network models for identification of various numerals (at CPM =1); the output of each model is a real-value which is feed into a decision unit that generates the index of the maximum value as an output

Table 3. Most relevant features selected by the polynomial neural network for various numerals

Class	Relevant Features	Reduction (%)
0	x_1, x_2, x_4, x_7, x_8, x_9, x_{11}, x_{13}	50.0
1	x_3, x_4, x_5, x_6, x_{11}, x_{12}, x_{16}	56.25
2	x_2, x_3, x_6, x_9, x_8, x_{11}, x_{13}, x_{14}	50.0
3	x_1, x_3, x_4, x_5, x_9, x_{13}, x_{14}	56.25
4	x_3, x_4, x_5, x_6, x_8, x_9, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}	31.25
5	x_1, x_3, x_8, x_9, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}	43.75
6	x_4, x_6, x_7, x_9, x_{11}, x_{12}, x_{15}	56.25
7	x_1, x_3, x_4, x_6, x_8, x_{11}, x_{13}, x_{14}, x_{15}	43.75
8	x_1, x_3, x_4, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{14}	37.5
9	x_1, x_4, x_8, x_9, x_{11}, x_{12}, x_{13}, x_{14}	50.0

Table 4. Comparing the overall accuracy for various classification methods

	GMDH	SVM	MLP	RBF	NB	RB
Accuracy (%)	91.83	88.83	89	89	87.67	87.67

Table 5. Comparing the per-class performance measures for the polynomial neural network and other classifiers; percentage values are approximated to one-decimal digit

Method	Measure	Numeral Characters									
		0	1	2	3	4	5	6	7	8	9
GMDH	*Pr*	93.8	100	92.2	85.3	92.9	93.4	89.1	92.5	98.3	83.6
	TP	95.2	100	88.1	86.6	88.1	87.7	94.2	96.9	90.5	94.4
	FP	0.7	0	0.9	1.9	0.7	0.8	1.1	0.9	0.2	1.8
	FM	94.5	100	90.1	85.9	90.4	90.5	91.6	94.7	94.2	88.7
SVM	*Pr*	85.9	100	84.5	86.9	84.0	92.3	94.1	87.5	98.3	79.4
	TP	96.8	100	89.6	79.1	71.2	92.3	92.3	87.5	90.5	92.6
	FP	1.9	0	2.1	1.5	1.5	0.9	0.6	1.5	0.2	2.4
	FM	91.0	100	87.0	82.8	77.1	92.3	93.2	87.5	94.2	85.5
MLP	*Pr*	88.4	100	93.2	84.9	90.2	92.1	90.9	83.3	93.3	78.5
	TP	96.8	100	82.1	83.6	78	89.2	96.2	85.9	88.9	94.4
	FP	1.5	0	0.8	1.9	0.9	0.9	0.9	2.1	0.7	2.6
	FM	92.4	100	87.3	84.2	83.6	90.6	93.5	84.6	91.1	85.7
RBF	*Pr*	85.9	95.8	93.0	88.7	91.5	90.6	84.5	86.3	93.3	83.3
	TP	96.8	100	79.1	82.1	72.9	89.2	94.2	98.4	88.9	92.6
	FP	1.9	0.4	0.8	1.3	0.7	1.1	1.6	1.9	0.7	1.8
	FM	91	97.9	85.5	85.3	81.1	89.9	89.1	92.0	91.1	87.7
NB	*Pr*	87	92	92.0	74.0	97.8	92.1	86.0	85.1	98.2	80.7
	TP	95.2	100	68.7	80.6	76.3	89.2	94.2	98.4	87.3	92.6
	FP	1.7	0.7	0.8	3.6	0.2	0.9	1.5	2.1	0.2	2.2
	FM	90.9	95.8	78.6	77.1	85.7	90.6	89.9	91.3	92.4	86.2
RB	*Pr*	89.4	75	83.9	81.4	92.2	93.2	90.4	89.4	96.6	87.5
	TP	93.7	97.8	77.6	85.1	79.7	84.6	90.4	92.2	88.9	90.7
	FP	1.3	2.7	1.9	2.4	0.7	0.8	0.9	1.3	0.4	1.3
	FM	91.5	84.9	80.6	83.2	85.5	88.7	90.4	90.8	92.6	89.1

5 Conclusion

A new approach is investigated in this paper for the recognition of isolated handwritten numerals. The approach is based on the self-organizing abductive learning for building a neural network of polynomial neurons using the group method data handling (GMDH). Experimental work has been conducted to study the effectiveness of the proposed approach on one of the publicly available datasets. Encouraging results have been attained as compared to five other conventional classifiers.

Acknowledgments. The author acknowledges the support of King Fahd University of Petroleum and Minerals (KFUPM) during the development of this work.

References

1. Liu, C., Nakashima, K., Sako, H., Fujisawa, H.: Handwritten Digit Recognition: Benchmarking of State-of-the-Art Techniques. Pattern Recognition 36, 2271–2285 (2003)
2. Cheriet, M., El Yacoubi, M., Fujisawa, H., Lopresti, D., Lorette, G.: Handwriting Recognition Research: Twenty Years of Achievement and Beyond. Pattern Recognition 42, 3131–3135 (2009)
3. Weideman, W.E., Manry, M.T., Yau, H.-C., Gong, W.: Comparisons of a Neural Network and a Nearest-Neighbor Classifier via the Numeric Handprint Recognition Problem. IEEE Trans. on Neural Networks 6, 1524–1530 (1995)
4. Alaei, A., Nagabhushan, P., Pal, U.: Fine Classification of Unconstrained Handwritten Persian/Arabic Numerals by Removing Confusion amongst Similar Classes. In: International Conference on Document Analysis and Recognition, pp. 601–605 (2009)
5. Wen, Y., Lu, Y., Shi, P.: Handwritten Bangla Numeral Recognition System and Its Application to Postal Automation. Pattern Recognition 40, 99–107 (2007)
6. Mizukami, Y.: A Handwritten Chinese Character Recognition System Using Hierarchical Displacement Extraction Based on Directional Features. Pattern Recognition Letters 19, 595–604 (1998)
7. Awaidah, S., Mahmoud, S.: A Multiple Feature/Resolution Scheme to Arabic (Indian) Numerals Recognition Using Hidden Markov Models. Signal Processing 89, 1176–1184 (2009)
8. Plamondon, R., Srihari, S.: Online and Off-line Handwriting Recognition: A Comprehensive Survey. IEEE Trans. on Pattern Analysis and Machine Intelligence 22, 63–84 (2000)
9. Liu, C., Nakashima, K., Sako, H., Fujisawa, H.: Handwritten Digit Recognition: Investigation of Normalization and Feature Extraction Techniques. Pattern Recognition 37, 265–279 (2004)
10. Ping, Z., Lihui, C.: A Novel Feature Extraction Method and Hybrid Tree Classification for Handwritten Numeral Recognition. Pattern Recognition Letters 23, 45–56 (2002)
11. Mahmoud, S.: Arabic (Indian) Handwritten Digits Recognition Using Gabor-Based Features. In: International Conference on Innovations in Information Technology, pp. 683–687 (2008)
12. Gong, W., Yau, H.-C., Manry, M.T.: Non-Gaussian Feature Analyses Using a Neural Network. Progress in Neural Networks 2, 253–269 (1994)
13. Sadri, J., Suen, C.Y., Bui, T.D.: Application of Support Vector Machines for Recognition of Handwritten Arabic/Persian Digits. In: Proceeding of the Second Conference on Machine Vision and Image Processing & Applications (MVIP 2003), Tehran, Iran (2003)
14. Sadok, M., Alouani, A.: A Fuzzy Logic Based Handwritten Numeral Recognition Expert System. In: Proceedings of the Twenty-Ninth Southeastern Symposium on System Theory, pp. 34–38 (1997)
15. Kim, D., Bang, S.-Y.: A Handwritten Numeral Character Classification Using Tolerant Rough Set. IEEE Tran. on Pattern Analysis and Machine Intelligence 22, 923–937 (2000)
16. Hwang, Y., Bang, S.: Recognition of Unconstrained Handwritten Numerals by a Radial Basis Function Neural Network Classifier. Pattern Recognition Letters 18, 657–664 (1997)
17. Cao, J., Ahmadi, M., Shridhar, M.: A Hierarchical Neural Network Architecture for Handwritten Numeral Recognition. Pattern Recognition 30, 289–294 (1997)

18. Abdel-Aal, R.E., El-Alfy, E.-S.M.: Constructing Optimal Educational Tests Using GMDH-Based Item Ranking and Selection. Neurocomputing 72, 1184–1197 (2009)
19. Barron, A.R.: Predicted Squared Error: A Criterion for Automatic Model Selection. In: Farlow, S.J. (ed.) Self-organizing Methods in Modeling: GMDH Type Algorithms, pp. 87–103. Marcel Dekker, New York (1984)
20. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)

Automatic Parameter Settings for the PROAFTN Classifier Using Hybrid Particle Swarm Optimization

Feras Al-Obeidat, Nabil Belacel, Juan A. Carretero, and Prabhat Mahanti

University of New Brunswick, Canada
National Research Council Canada

Abstract. In this paper, a new hybrid metaheuristic learning algorithm is introduced to choose the best parameters for the classification method PROAFTN. PROAFTN is a multi-criteria decision analysis (MCDA) method which requires values of several parameters to be determined prior to classification. These parameters include boundaries of intervals and relative weights for each attribute. The proposed learning algorithm, identified as PSOPRO-RVNS as it integrates particle swarm optimization (PSO) and Reduced Variable Neighborhood Search (RVNS), is used to automatically determine all PROAFTN parameters. The combination of PSO with RVNS allows to improve the exploration capabilities of PSO by setting some search points to be iteratively re-explored using RVNS. Based on the generated results, experimental evaluations show that PSOPRO-RVNS outperforms six well-known machine learning classifiers in a variety of problems.

Keywords: Knowledge Discovery, Particle swarm optimization, Reduced Variable Neighborhood Search, Multiple criteria classification, PROAFTN, Supervised Learning.

1 Introduction

Data classification in machine learning algorithms is a widely used supervised learning approach. It requires the development of a classification model that identifies behaviors and characteristics of the available objects to recommend the assignment of unknown objects to predefined classes [3]. The goal of the classification is to accurately assign the target class for each instance in the dataset. For instance, in medical diagnosis, patients are assigned to disease classes (positive or negative) according to a set of symptoms. In this context, the classification model is built on historical data and then the generated model is used to classify unseen instances.

Multi-criteria decision analysis (MCDA) [22] is another field that addresses the study of decision making [14] and classification problems. In recent years, the field of MCDA has been attracting researchers and decision-makers from many areas, including health, data mining and business [26]. The classification problem in MCDA consists of the formulation of the decision problem in the form of class prototypes that are used for assigning objects to classes. Each prototype is described by a set of attributes and is considered to be a good representative of its class [17].

PROAFTN is a relatively new MCDA classification method introduced in [6] and belongs to the class of supervised learning algorithms. PROAFTN has successfully been

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 184–195, 2010.

applied to many real-world practical problems such as medical diagnosis, asthma treatment, and e-Health [7,9,10]. However, to apply the PROAFTN classifier, the values of several parameters need to be determined prior to classification. These parameters include the boundaries of intervals and the relative weight for each attribute. This consists of the formulation of the decision problem in the form of prototypes – representing each class – to be used for assigning each object to the target class.

Recently, some related work introduced in [2] was done to improve PROAFTN's performance. There, the unsupervised discretization algorithm *k*-means and a Genetic Algorithm (GA) were used to obtain the best number of clusters and optimal prototypes obtained after completion of a clustering process. Here, a new method is proposed based on particle swarm optimization (PSO) and Reduced Variable Neighborhood Search (RVNS) to obtain all PROAFTN training parameters. This study proposes a different approach than the one proposed in [2] in that the formulation of the optimization problem is entirely different. The integrating or hybridization of PSO and RVNS significantly improves the exploration and search strategies.

The rest of the paper is organized as follows: in Section 2, the PROAFTN method as well as the PSO and RVNS algorithms are briefly presented. In Section 3, the proposed approach PSOPRO-RVNS to learn PROAFTN is introduced. The description of the datasets, experimental results, and comparative numerical studies are presented in Section 4. Finally, conclusions are summarized in Section 5.

2 Overview of PROAFTN, PSO and RVNS

2.1 PROAFTN Method

PROAFTN belongs to the class of supervised learning algorithms [6]. Its procedure can be described as follows. From a set of n objects known as a training set, consider a is an object which requires classification; assume this object a is described by a set of m attributes $\{g_1, g_2, \dots, g_m\}$ and let $\{C^1, C^2, \dots, C^k\}$ be the set of k classes. The different steps of the classification procedure follow.

Initialization. For each class C^h, $h = 1, 2, \dots, k$, a set of L_h prototypes $B^h = \{b_1^h, b_2^h, \dots, b_{L_h}^h\}$ are determined. For each prototype b_i^h and each attribute g_j, an interval $[S_j^1(b_i^h), S_j^2(b_i^h)]$ and the preference thresholds $d_j^1(b_i^h) \geq 0$ and $d_j^2(b_i^h) \geq 0$ are defined where $S_j^2(b_i^h) \geq S_j^1(b_i^h)$, with $j = 1, 2, \dots, m$ and $i = 1, 2, \dots, L_h$.

Figure 1 depicts the representation of PROAFTN's intervals. To apply PROAFTN, the pessimistic interval $[S_{jh}^1, S_{jh}^2]$ and the optimistic interval $[q_{jh}^1, q_{jh}^2]$ for each attribute in each class need to be determined [8], where:

$$q_{jh}^1 = S_{jh}^1 - d_{jh}^1 \qquad\qquad q_{jh}^2 = S_{jh}^2 + d_{jh}^2 \tag{1}$$

applied to:

$$q_{jh}^1 \leq S_{jh}^1 \qquad\qquad q_{jh}^2 \geq S_{jh}^2 \tag{2}$$

Hence, $S_{jh}^1 = S_j^1(b_i^h)$, $S_{jh}^2 = S_j^2(b_i^h)$, $q_{jh}^1 = q_j^1(b_i^h)$, $q_{jh}^2 = q_j^2(b_i^h)$, $d_{jh}^1 = d_j^1(b_i^h)$, and $d_{jh}^2 = d_j^2(b_i^h)$. The following subsections explain the different stages required to classify object a to class C^h using PROAFTN.

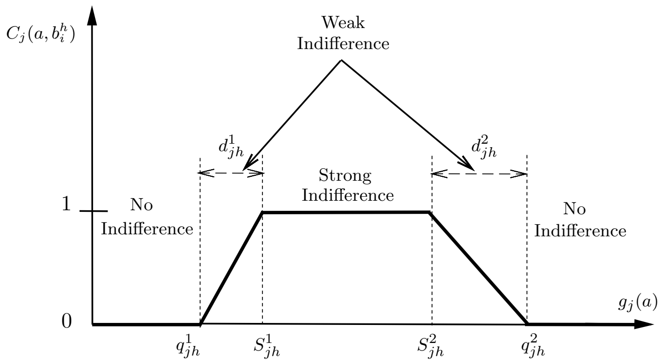


Fig. 1. Graphical representation of the partial indifference concordance index between the object a and the prototype b_i^h represented by intervals

Computing the fuzzy indifference relation $I(a,b_i^h)$. The initial stage of the classification procedure is performed by calculating the fuzzy indifference relation $I(a,b_i^h)$. The fuzzy indifference relation is based on the concordance and non-discordance principle which represents the relationship (*i.e.*, the membership degree) between the object to be assigned and the prototype [5,8]. It is formulated as:

$$I(a,b_i^h) = \left(\sum_{j=1}^{m} w_{jh} C_j(a,b_i^h)\right) \prod_{j=1}^{m} \left(1 - D_j(a,b_i^h)^{w_{jh}}\right) \tag{3}$$

where w_{jh} is a weighting factor that measures the relative importance of a relevant attribute g_j of a specific class C^h: $w_{jh} \in [0,1]$ and $\sum_{j=1}^{m} w_{jh} = 1$ Also, $C_j(a,b_i^h)$, $j = 1,2,\ldots,m$, is the degree that measures the closeness of the object a to the prototype b_i^h according to the attribute g_j. The calculation of $C_j(a,b_i^h)$ is given by:

$$C_j(a,b_i^h) = \min\{C_j^1(a,b_i^h), C_j^2(a,b_i^h)\}, \tag{4}$$

where

$$C_j^1(a,b_i^h) = \frac{d_j^1(b_i^h) - \min\{S_j^1(b_i^h) - g_j(a), d_j^1(b_i^h)\}}{d_j^1(b_i^h) - \min\{S_j^1(b_i^h) - g_j(a), 0\}}$$

and

$$C_j^2(a,b_i^h) = \frac{d_j^2(b_i^h) - \min\{g_j(a) - S_j^2(b_i^h), d_j^2(b_i^h)\}}{d_j^2(b_i^h) - \min\{g_j(a) - S_j^2(b_i^h), 0\}}$$

Finally, $D_j(a,b_i^h)$ is the discordance index that measures how far object a is from prototype b_i^h according to attribute g_j. Two veto thresholds $\varepsilon_j^1(b_i^h)$ and $\varepsilon_j^2(b_i^h)$ [6], are used to define this value, where the object a is considered perfectly different from the prototype b_i^h based on the value of attribute g_j. Generally, the determination of veto thresholds through inductive learning is risky. These values need to be obtained by an expert familiar with the problem. However, this study is focused on an automatic approach.

Therefore, the effect of the veto thresholds is eliminated by setting them to infinity. As a result, only the concordance principle is used. Thus, Eq (3) is summarized as:

$$I(a,b_i^h) = \sum_{j=1}^{m} w_{jh} C_j(a,b_i^h) \tag{5}$$

To sum up, three comparative procedures between object a and prototype b_i^h according to attribute value g_j are performed (Fig. 1):

- case 1 (strong indifference): $C_j(a,b_i^h) = 1 \Leftrightarrow S_{jh}^1 \leq g_j(a) \leq S_{jh}^2$
- case 2 (no indifference): $C_j(a,b_i^h) = 0 \Leftrightarrow g_j(a) \leq q_{jh}^1$, or $g_j(a) \geq q_{jh}^2$
- case 3 (weak indifference): The value of $C_j(a,b_i^h) \in (0,1)$ is calculated based on Eq. (4). (*i.e.,* $g_j(a) \in [q_{jh}^1, S_{jh}^1]$ or $g_j(a) \in [S_{jh}^2, q_{jh}^2]$)

Evaluation of the membership degree $\delta(a,C^h)$. The membership degree between object a and class C^h is calculated based on the indifference degree between a and its nearest neighbor in B^h. The following formula identifies the nearest neighbor:

$$\delta(a,C^h) = \max\{I(a,b_1^h), I(a,b_2^h), \ldots, I(a,b_{L_h}^h)\} \tag{6}$$

Assignment of an object to the class. The last step is to assign object a to the right class C^h. To find the right class the following decision rule is used:

$$a \in C^h \Leftrightarrow \delta(a,C^h) = \max\{\delta(a,C^i)/i \in \{1,\ldots,k\}\} \tag{7}$$

2.2 Particle Swarm Optimization Algorithm

Particle Swarm Optimization (PSO) is a population-based and adaptive optimization method introduced by Eberhart and Kennedy in (1995) [13]. The fundamental concepts of PSO are intuitively inspired by social swarming behavior of birds flocking or fish schooling. Thus, compared with Genetic Algorithms (GAs), the evolution strategy in PSO is inspired from social behavior of living organisms, whereas the evolutionary strategy of GAs is inspired from biological mechanisms (*i.e.*, mating and mutation). More specifically, in the PSO evolutionary process, potential solutions, called particles, move about the multi-dimensional search space by following and tracking the current best particles in the population.

From an implementation perspective, PSO is easy to implement and computationally efficient compared to GAs [18,20]. The general procedure of PSO is outlined in Algorithm 1.

Functionality of PSO. Each particle in the swarm has mainly two variables associated with it. These variables are: the *position vector* $\mathbf{x}_i(t)$ and the *velocity vector* $\mathbf{v}_i(t)$. Thus, each particle $\mathbf{x}_i(t)$ is represented by a vector $[x_{i1}(t), x_{i2}(t), \ldots, x_{iD}(t)]$ where i is the index number of each particle in the swarm, D represents the dimension of the search space and t is the iteration number.

Algorithm 1. PSO Evolution Steps

```
Step 1: **Initialization phase**, Initialize the *swarm*
**Evolution phase**
**repeat**
    Step 2: Evaluate *fitness* of each particle
    Step 3: Update personal best position for each particle
    Step 4: Update global best position for entire population
    Step 5: Update each particle's velocity
    Step 6: Update each particle's position
**until** (termination criteria are met or stopping condition is satisfied)
```

During the evolutionary phase, each of the N_{pop} particles in the swarm is drawn toward an optimal solution based on the updated value of $\mathbf{v}_i$ and the particle's current position $\mathbf{x}_i$. Thus, each particle's new position $\mathbf{x}_i(t+1)$ is updated using:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \tag{8}$$

The new position of the particle is influenced by its current position and velocity, the best position it has visited (*i.e.*, its own experience), called personal best and denoted here as $\mathbf{P}_i^{Best}(t)$, and the position of the best particle in its neighborhood, called the global best and represented by $\mathbf{G}^{Best}(t)$. At each iteration t, the velocity $\mathbf{v}_i(t)$ is updated based on the two best values $\mathbf{P}_i^{Best}(t)$ and $\mathbf{G}^{Best}(t)$ using the formula,

$$\mathbf{v}_i(t+1) = \underbrace{\varpi(t)\mathbf{v}_i(t)}_{\text{inertial parameters}} + \underbrace{\tau_1\rho_1(\mathbf{P}_i^{Best}(t) - \mathbf{x}_i(t))}_{\text{personal best velocity components}} + \underbrace{\tau_2\rho_2(\mathbf{G}^{Best}(t) - \mathbf{x}_i(t))}_{\text{global best velocity components}} \tag{9}$$

where $\varpi(t)$ is the inertia weight factor that controls the exploration of the search space. Factors τ_1 and τ_2 are the individual and social components/weights, respectively, also called the acceleration constants, which change the velocity of a particle towards the $\mathbf{P}_i^{Best}(t)$ and $\mathbf{G}^{Best}(t)$, respectively. Finally, ρ_1 and ρ_2 are random numbers between 0 and 1. During the optimization process, particle velocities in each dimension d are evolved to a maximum velocity $v_{d_{max}}$ (where $d = 1, 2 \ldots, D$).

2.3 Reduced Variable Neighborhood Search (RVNS)

RVNS is a variation of the metaheuristic Variable Neighborhood Search (VNS) [15,16]. The basic idea of the VNS algorithm is to find a solution in the search space with a systematic change of neighborhood. In RVNS, two procedures are used: shake and move. Starting from the initial solution (the position of prematurely converged individuals) $\mathbf{x}$, the algorithm selects a random solution $\mathbf{x}'$ from first neighborhood. If the generated $\mathbf{x}'$ is better than $\mathbf{x}$, it replaces $\mathbf{x}$ and the algorithm starts all over again with the same neighborhood. Otherwise, the algorithm continues with the next neighborhood structure. The pseudo-code of RVNS is presented in Algorithm 2.

Algorithm 2. RVNS Procedure

```
Require:
  Define neighborhood structures $N_k$ for $k = 1, 2, \ldots, k_{max}$, that will be used in the search
  Get the initial solution $\mathbf{x}$ and choose stopping condition
  repeat
     $k \leftarrow 1$
     while $k < k_{kmax}$ do
        Shaking:
          Generate a point $\mathbf{x}'$ at random from the $k$th neighborhood of $\mathbf{x}$ $(\mathbf{x}' \in N_k(\mathbf{x}))$
        Move or not:
        if $\mathbf{x}'$ is better than the incumbent $\mathbf{x}$ then
           $\mathbf{x} \leftarrow \mathbf{x}'$
           $k \leftarrow 1$
        else
           set $k \leftarrow k+1$
        end if
     end while
  until stopping condition is met
```

3 Problem Formulation

In most cases, and depending on the population size, PSO may not be able to explore the entire search space thoroughly. To improve the search and exploration process, RVNS forces individuals to 'jump' to another location in the solution space while PSO allows these points to continue the search using past experience. As discussed earlier, to apply PROAFTN, the intervals $[S^1_{jh},\, S^2_{jh}]$ and $[q^1_{jh}, q^2_{jh}]$ that satisfy the constraints in Eq. (2) and the weights w_{jh} in Eq. (5) are to be obtained for each attribute g_j belonging to each class C^h. In this study, the induction of weights is based on the calculation of entropy and information gain [25]. An entropy measure of a set of objects is calculated as $Entropy = -\sum_{i=1}^{C}(p_i)\ log_2\ (p_i)$, where p_i is the proportion of instances in the dataset that take the i_{th} value of the target attribute and C represents the number of classes.

To simplify the constraints in Eq. (2), a variable substitution based on Eq. (1) is used. As a result, parameters d^1_{jh} and d^2_{jh} are used instead of q^1_{jh} and q^2_{jh}, respectively. Therefore, the optimization problem, which is based on maximizing classification accuracy to provide the optimal parameters, is defined here as

$$\begin{aligned} P:\text{Maximize} \quad & f(S^1_{jh}, S^2_{jh}, d^1_{jh}, d^2_{jh}, n) \\ \text{Subject to:} \quad & S^1_{jh} \leq S^2_{jh}; d^1_{jh}, d^2_{jh} \geq 0 \end{aligned} \tag{10}$$

where the objective or fitness function f depends on the classification accuracy and n represents the set of training objects/samples to be assigned to different classes. The procedure for calculating the fitness function f is described in Algorithm 3.

In this study, PSO and RVNS are utilized to solve the optimization problem described in Eq. (10). The problem dimension D (*i.e.*, the number of search parameters in the optimization problem) is proportional to the number of classes k, prototypes L_h and attributes m in the problem. Because of this hierarchal structure, the elements for each particle position $\mathbf{x}_i$ are updated using:

Algorithm 3. Procedure to calculate objective function f

```
Step 1:
for all $a \in n$ do
    Compute the fuzzy indifference relation $I(a, b_i^h)$ (Eq. (5))
    Evaluate the membership degree $\delta(a, C^h)$ (Eq. (6))
    Assign the object to the class (Eq. (7))
end for
Step 2:
Compare the value of the new class with the true class $C^h$ for all $a \in n$
Calculate the classification accuracy (i.e. the fitness value): $f = \frac{\text{number of correctly classified objects}}{\text{total number of all objects} \in n}$
```

$$x_{i\lambda jbh}(t+1) = x_{i\lambda jbh}(t) + v_{i\lambda jbh}(t+1) \tag{11}$$

where the velocity update $\mathbf{v}_i$ for each element based on $\mathbf{P}_i^{Best}(t)$ and $\mathbf{G}^{Best}(t)$ is formulated as:

$$\begin{aligned} v_{i\lambda jbh}(t+1) = \varpi(t) v_{i\lambda jbh}(t) + \tau_1 \rho_1 (P_{i\lambda jbh}^{Best}(t) - x_{i\lambda jbh}(t)) \\ + \tau_2 \rho_2 (G_{\lambda jbh}^{Best}(t) - x_{i\lambda jbh}(t)) \end{aligned} \tag{12}$$

where $i = 1, \ldots, N_{pop}$, $\lambda = 1, \ldots, D$, $j = 1, \ldots, m$ $b = 1, \ldots, L_h$ and $h = 1, \ldots, k$.

Using RVNS as a local search algorithm, the following equations are considered to update the boundary for the previous solution $\mathbf{x}$ containing $(S_{jh}^1, S_{jh}^2, d_{jh}^1, d_{jh}^2)$ parameters:

$$\begin{aligned} l_{\lambda jbh} &= x_{\lambda jbh} - (k/k_{max}) x_{\lambda jbh} \\ u_{\lambda jbh} &= x_{\lambda jbh} + (k/k_{max}) x_{\lambda jbh} \end{aligned}$$

where $l_{\lambda jbh}$ and $u_{\lambda jbh}$ are the lower and upped bounds for each element $\lambda \in [1, \ldots, D]$. Factor k/k_{max} is used to define the boundary for each element and $x_{\lambda jbh}$ is the previous solution for each element $\lambda \in [1, \ldots, D]$ provided by PSO.

The *shaking* phase to randomly generate the elements of $\mathbf{x}'$ is given by:

$$x'_{\lambda jbh} = l_{\lambda jbh} + (u_{\lambda jbh} - l_{\lambda jbh}).rand[0,1] \tag{13}$$

Accordingly, the *moving* is applied as:

$$\text{If } f'(x'_{\lambda jbh}) > f(x_{\lambda jbh}) \text{ then } x_{\lambda jbh} = x'_{\lambda jbh} \tag{14}$$

The complete classification procedure of the proposed PSOPRO-RVNS is presented in Algorithm 4. After the initialization of all particles $\mathbf{x}$, the optimization is then implemented iteratively. At each iteration, a new fitness value (*i.e.*, classification accuracy) for each particle according to Eq. (10) is calculated. The best global particle $\mathbf{G}^{Best}(t)$ is replaced by its corresponding particle if the latter has better fitness. Furthermore, to enhance the search strategy and to get a better solution, the best solution found so far in each iteration by PSO ($\mathbf{G}^{Best}(t)$) is submitted to RVNS for further exploitation. The search procedure used by RVNs is based on the concept of Algorithm 2 and by applying

Algorithm 4. PSOPRO-RVNS procedure

```
Require:
  $NT$: training data, $NS$: testing data, $m$: number of attributes, $k$: number of classes
  $D$: problem dimension, assign initial values to parameters set ($S^1_{jh}, S^2_{jh}, d^1_{jh}$ and $d^2_{jh}$)
  $N_{pop}, \tau_1, \tau_2, \varpi$: control parameters
  $\mathbf{v}_{max}$: boundary limits for each element in $D$
  Initialization
  for $i = 1$ to $N_{pop}$ do
      Initialize $\mathbf{x}_i$, $\mathbf{v}_i$ and $\mathbf{P}_i^{Best}(t)$ consisting of ($S^1_{jh}, S^2_{jh}, d^1_{jh}$ and $d^2_{jh}$)
      Evaluate fitness value $f(\mathbf{x}_i)$ (the classification accuracy Eq. (10))
  end for
  Obtain the $\mathbf{G}^{Best}(t)$, which contains the best set of ($S^1_{jh}, S^2_{jh}, d^1_{jh}$ and $d^2_{jh}$)
  Optimization stage
  repeat
      while maximum iterations or maximum accuracy is not attained do
          for each particle do
              Update $\mathbf{v}_i$ and $\mathbf{x}_i$ for each particle according to Eqs. (12 and 11)
          end for
          for each particle do
              Calculate fitness value $f(\mathbf{x}_i)$ according to Eq. (10)
              if $f(\mathbf{x}_i) > f(\mathbf{P}_i^{Best}(t))$ then
                  set $\mathbf{P}_i^{Best}(t) = \mathbf{x}_i$
              end if
          end for
          Choose the particle with the best fitness among particles as the $\mathbf{G}^{Best}(t)$
          Apply local search (RVNS) to get a better solution:
            $\mathbf{x}'$ = LocalSearch($\mathbf{G}^{Best}(t)$) (Algorithm 2 and Eqs. (13 and 14))
          if $(f'(\mathbf{x}') > f(\mathbf{G}^{Best}(t))$ then
              $\mathbf{G}^{Best}(t) = \mathbf{x}'$
          end if
      end while
  until (termination criteria are met)
  Apply the classification:
  Submit the best solution $\mathbf{G}^{Best^*}$ along with testing data ($NS$) for evaluation
```

Eqs. (13 and 14). After completing the optimization phase, the best set of parameters $\mathbf{G}^{Best^*}$ is sent, together with the testing data, to PROAFTN to perform classification. The classification procedure based on testing data is carried out using equations (5) to (7).

4 Application and Analysis of PSOPRO-RVNS

The proposed PSOPRO-RVNS algorithm (*i.e.*, Algorithm 4) is implemented in Java and applied to 12 popular datasets: Breast Cancer Wisconsin Original (Bcancer), Transfusion Service Center (Blood), Heart Disease (Heart), Hepatitis, Haberman's Survival (HM), Iris, Liver Disorders (Liver), Mammographic Mass (MM), Pima Indians Diabetes (Pima), Statlog Australian Credit Approval (STAust), Teaching Assistant Evaluation

Table 1. Dataset Description

	Dataset	**Instances**	**Attributes**	**Classes**	$D = dim(\mathbf{x})$
1	BCancer	699	9	2	144
2	Blood	748	4	2	64
3	Heart	270	13	2	208
4	Hepatitis	155	19	2	304
5	HM	306	3	2	48
6	Iris	150	4	3	96
7	Liver	345	6	2	96
8	MM	961	5	2	80
9	Pima	768	8	2	128
10	STAust	690	14	2	224
11	TA	151	5	3	120
12	Wine	178	13	3	312

(TA), and Wine. These datasets are in the public domain and are available at the University of California at Irvine (UCI) Machine Learning Repository database [4]. The details of the dataset description and dimensionality are presented in Table 1. The dimensionality $D = dim(\mathbf{x})$ describes the number of elements of each individual required by PSOPRO-RVNS. Considering two prototypes for each class, and four parameters for each attribute S^1, S^2, d^1, d^2 are needed, the number of components of D for each problem is $2 \times 4 \times k \times m$, where k and m are the number of classes and attributes, respectively.

4.1 Parameters Settings

To apply PSOPRO-RVNS, the following factors are considered:

- The bounds for S^1_{jh} and S^2_{jh} vary between $\mu_{jh} - 6\sigma_{jh}$ and $\mu_{jh} + 6\sigma_{jh}$, where μ_{jh} and σ_{jh} represent mean and standard deviation for each attribute in each class, respectively;
- The bounds for d^1_{jh} and d^2_{jh} vary in the range $[0, 6\sigma_{jh}]$.

During the optimization phase, the parameters $S^1_{jh}, S^2_{jh}, d^1_{jh}, d^2_{jh}$ evolve within the aforementioned boundaries. Also, the following technical factors are considered:

- The control parameters are set as follows: $\tau_1 = 2$, $\tau_2 = 2$ and $\varpi = 1$.
- The size of population is fixed at 80; and the maximum iteration number (gen_{max}) is fixed at 500.

4.2 Results and Analysis

The experimentation work is performed in two stages. First, PSOPRO-RVNS is applied to each dataset, where 10 independent runs are executed over each dataset. In each run, a stratified 10-fold cross-validation [25] is applied, where the percentage of correct classification for each fold is obtained, and then the average of classification accuracy is computed on all 10 folds. Second, to compare the performance of PSOPRO-RVNS against other well-known machine learning classifiers, similar experimental work is performed on the same dataset using *Weka* (the open source platform described in [25]). A stratified 10-fold cross-validation is also used to evaluate the performance of all classifiers.

The second stage of the experimental study includes the evaluation of PSOPRO-RVNS performance against other six machine learning techniques. These algorithms are chosen from different machine learning theories; they are: 1) Tree induction C4.5 (J48) [21], 2) statistical modelling, Naive Bayes (NB) [11], 3) Support Vector machines (SVM), SMO [19], 4) Neural Network (NN), multilayer perceptron (MLP) [23], 5) instance-based learning, IBk with k=3 [1], and 6) rule learning, PART [24]. The default settings in the *Weka* platform are used to run these tests.

Table 2 documents the results of the classification accuracy obtained by PSOPRO-RVNS and the other algorithms based on testing dataset. The best results achieved on each application are marked in bold. PSOPRO-RVNS gives better results on 9 out of 12 datasets. Based on Demšar's recommendation [12], the Friedman test is used in this work to recognize the rank of PSOPRO-RVNS among other classifiers. Based on the classification accuracy in Table 2, the algorithms' ranking results using Friedman test are shown in the last couple of rows in Table 2.

Regarding the execution time of PSOPRO-RVNS, it was noticed that, as expected, the execution time is dependent mainly on the problem size as this affects the number of PROAFTN parameters (D) involved in the training process. Even though the number of PSO iterations was set to 500, PSOPRO-RVNS was able to generate the presented

Table 2. Experimental results based on classification accuracy (in %) to measure the performance of the different classifier compared with PSOPRO-RVNS

	Dataset	C4.5	NB	SVM	NN	k-nn	PART	PSOPRO-RVNS
		J48		SMO	MLP	$Ibk, k = 3$		
1	BCancer	94.56	95.99	96.70	95.56	97.00	94.28	**97.33**
2	Blood	77.81	75.40	76.20	78.74	74.60	78.07	**79.46**
3	Heart	76.60	83.70	84.10	78.10	78.89	73.33	**84.36**
4	Hepatitis	80.00	85.81	83.87	81.94	84.52	82.58	**87.05**
5	HM	71.90	74.83	73.52	72.87	70.26	72.55	**76.27**
6	Iris	96.00	96.00	96.00	**97.33**	95.33	94.00	96.30
7	Liver	68.7	56.52	58.26	**71.59**	61.74	63.77	70.97
8	MM	82.10	78.35	79.24	82.10	77.21	82.21	**84.07**
9	Pima	71.48	75.78	77.08	75.39	73.44	73.05	**77.42**
10	STAust	85.22	77.25	85.51	84.93	83.62	83.62	**86.10**
11	TA	59.60	52.98	54.3	54.3	50.33	58.28	**60.62**
12	Wine	91.55	97.40	**99.35**	97.4	95.45	92.86	96.72
Friedman ranking		4.875	4.375	3.458	3.542	5.292	5.042	**1.417**
Overall ranking		5	4	2	3	7	6	**1**

results in significantly fewer iterations number (*e.g.,* 100 iterations). It was noticed that the execution time of PSOPRO-RVNS compares favorably with the execution time of NN in most cases. The remaining algorithms 3-NN, C4.5, NB, PART and SVM in declining order of speed were relatively faster than PSOPRO-RVNS.

5 Conclusions

In this paper, a new methodology based on the metaheuristic algorithms PSO and RVNS is proposed for training the MCDA classification method PROAFTN. The proposed technique for solving classification problems is named PSOPRO-RVNS. During the learning stage, PSO and RVNS are utilized to induce the classification model for PROAFTN by inferring the best parameters from data with high classification accuracy.

The performance of PSOPRO-RVNS applied to 12 classification dataset demonstrates that PSOPRO-RVNS outperforms the well-known classification methods PART, 3-nn, C4.5, NB, SVM, and NN. PROAFTN requires some parameters and uses the fuzzy approach to assign objects to classes. As a result, there is richer information, more flexibility, and therefore an improved chance of assigning objects to the preferred classes. In this study, using the metaheuristics PSO and RVNS to obtain these parameters proved to be a successful approach for training PROAFTN.

Finally, the utilization of the hybrid PSO-RVNS algorithm demonstrated to be an efficient approach for learning the PROAFTN method. Therefore, it might be a good choice for learning other classification methods from different paradigms.

Acknowledgment

We gratefully acknowledge the support from NSERC's Discovery Award (RGPIN293261-05) granted to Dr. Nabil Belacel.

References

1. Aha, D.: Lazy learning. Kluwer Academic Publishers, Dordrecht (1997)
2. Al-Obeidat, F., Belacel, N., Mahanti, P., Carretero, J.A.: Discretization techniques and genetic algorithm for learning the classification method proaftn. In: Eighth International Conference On Machine Learning and Applications, pp. 685–688. IEEE, Los Alamitos (2009)
3. Alpaydin, E.: Introduction to machine learning. MIT Press, Cambridge (2004)
4. Asuncion, A., Newman, D.J.: UCI machine learning repository (2007)
5. Belacel, N.: Multicriteria Classification methods: Methodology and Medical Applications. PhD thesis, Free University of Brussels, Belgium (1999)
6. Belacel, N.: Multicriteria assignment method PROAFTN: methodology and medical application. European Journal of Operational Research 125(1), 175–183 (2000)
7. Belacel, N., Boulassel, M.: Multicriteria fuzzy assignment method: A useful tool to assist medical diagnosis. Artificial Intelligence in Medicine 21(1-3), 201–207 (2001)
8. Belacel, N., Raval, H., Punnen, A.: Learning multicriteria fuzzy classification method PROAFTN from data. Computers and Operations Research 34(7), 1885–1898 (2007)

9. Belacel, N., Vincke, P., Scheiff, M., Boulassel, M.: Acute leukemia diagnosis aid using multicriteria fuzzy assignment methodology. Computer Methods and Programs in Biomedicine 64(2), 145–151 (2001)
10. Belacel, N., Wang, Q., Richard, R.: Web-integration of PROAFTN methodology for acute leukemia diagnosis. Telemedicine Journal and e-Health 11(6), 652–659 (2005)
11. Cooper, G., Herskovits, E.: A bayesian method for the induction of probabilistic networks from data. Machine Learning 9(4), 309–347 (1992)
12. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)
13. Eberhart, R., Kennedy, J.: Particle swarm optimization. In: Proc. of the 1995 IEEE Int. Conf. on Neural Networks, vol. 4, pp. 1942–1948 (1995)
14. Fenton, N.E., Wang, W.: Risk and confidence analysis for fuzzy multicriteria decision making. Knowledge-Based Systems 19(6), 430–437 (2006)
15. Hansen, P., Mladenovic, N.: Variable neighborhood search for the p-median. Location Science 5(4), 207–226 (1997)
16. Hansen, P., Mladenovic, N.: Variable neighborhood search: Principles and applications. European Journal of Operational Research 130(3), 449–467 (2001)
17. Jabeur, K., Guitouni, A.: A generalized framework for concordance/discordance-based multicriteria classification methods. In: 2007 10th International Conference on Information Fusion, July 2007, pp. 1–8 (2007)
18. Kennedy, J., Eberhart, R.C.: Swarm intelligence. Morgan Kaufmann Pubs., San Francisco (2001)
19. Pang, S., Kim, D., Bang, S.Y.: Face membership authentication using SVM classification tree generated by membership-based lle data partition. IEEE Transactions on Neural Networks 16(2), 436–446 (2005)
20. Poli, R.: Analysis of the publications on the applications of particle swarm optimisation. Journal of Artificial Evolution and Applications 8(2), 1–10 (2008)
21. Quinlan, J.R.: Improved use of continuous attributes in C4.5. Journal of Artificial Intelligence Research 4, 77–90 (1996)
22. Roy, B.: Multicriteria methodology for decision aiding. Kluwer Academic, Dordrecht (1996)
23. Shirvany, Y., Hayati, M., Moradian, R.: Multilayer perceptron neural networks with novel unsupervised training method for numerical solution of the partial differential equations. Applied Soft Computing 9(1), 20–29 (2009)
24. Subramanian, D.K., Ananthanarayana, V.S., Narasimha Murty, M.: Knowledge-based association rule mining using and-or taxonomies. Knowledge-Based Syst. 16(1), 37–45 (2003)
25. Witten, H.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann Series in Data Management Systems (2005)
26. Zopounidis, C., Doumpos, M.: Multicriteria classification and sorting methods: A literature review. European Journal of Operational Research 138(2), 229–246 (2002)

The Ant Search Algorithm: An Ant Colony Optimization Algorithm for the Optimal Searcher Path Problem with Visibility

Michael Morin[1], Luc Lamontagne[2], Irène Abi-Zeid[3], and Patrick Maupin[4]

[1,2] Department of Computer Science and Software Engineering
[3] Department of Operations and Decision Systems
Université Laval, Québec, QC, Canada
Michael.Morin.3@ulaval.ca, Luc.Lamontagne@ift.ulaval.ca,
Irene.Abi-Zeid@osd.ulaval.ca
[4] Defence Research and Development Canada
Valcartier, QC, Canada
Patrick.Maupin@drdc-rddc.gc.ca

Abstract. In the first part of this paper, we present the Optimal Searcher Path problem with Visibility, a novel path planning approach that models inter-region visibility and that uses concepts from search theory to model uncertainty on the goal's (*i.e.*, the search object) detectability and location. In the second part, we introduce the Ant Search algorithm, a solving technique based on ant colony optimization. Our results, when compared with a general mixed-integer programming model and solver, show that Ant Search is a promising technique for handling this particular complex problem.

Keywords: Ant search, optimal searcher path problem, path planning, search theory, ant colony optimization.

1 Introduction

The *optimal searcher path* (OSP) problem is a well-known detection search problem in classical search theory. In short, an OSP problem consists of computing a sequence of regions that maximizes the probability of finding an object of unknown location using available resources and limited capabilities. Search theory and the OSP problem provide a framework to take into account uncertainty on the search object's (*i.e.*, the goal) detectability and location in path planning problems dealing with a physical travelling process: a robot trying to detect an object, a manned patrol searching for the survivors of an aircraft accident, etc. Recently, the *optimal searcher path* problem *with visibility* (OSPV) has been proposed as a variant of the classical OSP problem [12]. The OSPV formulation introduces the notion of inter-region visibility to take into account the fact that a search unit (*i.e.*, a searcher) may search (or scan) visible regions from a distant location.

In this paper, we extend the detection model used in the original OSPV problem formulation [12]. We also describe *ant search* (AS), an application of *ant colony*

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 196–207, 2010.

optimization (ACO) to solve the extended problem and to manage its complexity. We then present the results and compare them to a generic problem solving scheme: a *mixed-integer linear programming* (MILP) model and a general purpose solver. The purpose of the experiment is not to prove the superiority of ACO in comparison to MIP techniques, but to give insights on the AS algorithm performance and applicability to the OSPV problem.

2 The OSP Problem – Background and History

The OSP problem was initially defined in the search theory literature, one of the earliest Operations Research discipline in the United States. B.O. Koopman and the U.S. Navy's operations research group were the first to formalize the detection search problem. Their objective was to enhance the efficiency of naval operations during World War II [10]. Since then, search theory has been applied to surveillance, oil exploration, medicine and search and rescue operations. Recently, its application area has been extended to include unmanned aerial vehicles [7] and robotized search in structured environments [8] [11]. A definition of classical search theory can be found in [14] and [5] and a survey of the classical problems can be found in [1].

For one-sided search problems where the search object's motion model does not depend on the search unit's actions, we may identify two general problem classes in search theory: the *optimal search density* (OSD) problems and the *optimal searcher path* (OSP) problems. The former deals with an unconstrained search unit's motion while the latter, as described in the introduction, involves constrained search unit's motion. As shown in [15], the decision formulation of the OSP problem in discrete time and space with a stationary search object is NP-Complete.

Among the various OSP formulations, Stewart [13] considered a moving search object with a continuous search effort. The problem was solved using a network flow formulation under the assumption of an exponential *probability of detection* (*pod*) function. In the same paper, Stewart considered a discrete and unitary (indivisible) search effort and proposed a depth-first Branch and Bound algorithm. In the discrete (arbitrarily divisible) search effort case, Stewart suggested a sequential allocation or a relaxation of the indivisibility constraint for sufficiently large available search effort amounts. Eagle [4] proposed dynamic programming as a solving technique in the discrete and unitary search effort case. A review of Branch and Bound procedures and heuristics for OSPs before 1998 can be found in [16]. Among the recent developments related to the OSP problem, Lau [11] introduced the OSP problem with non-uniform travel times (OSPT) to use in structured environments.

3 A Formal Definition of the OSPV Problem

The OSPV problem may be characterized as a one-sided detection search problem in discrete time and space that involves one constrained search unit and one search object (moving or not). In the OSPV context, the environment where the search object is hidden is discretized by a set R of N regions numbered from 0 to $N - 1$. The time allocated to the search operation is discretized by a set I of T time steps numbered from 1 to *T*. For convenience purposes, all regions and time steps will be referred to by using their integer identification and $t = 0$ will represent the initialization step of the search operation.

As in OSP-like problems, the search unit's motion is constrained by the environment. For a given search unit, these inter-region accessibility constraints are defined by an *accessibility graph*. Each node of the graph corresponds to a region of the environment and the presence of an edge from region *s* to region *r* implies that it is possible for the search unit to travel from *s* to *r* within one time step. The *accessibility graph* is represented by an accessibility map *A* from the set of regions to the power set of regions (1). At each time step *t*, the search unit can move to an accessible region from its current position noted y_t (2) and y_0 is the initial search unit's position.

$$A : R \rightarrow 2^R . \tag{1}$$

$$\forall t \in I : y_t \in A(y_{t-1}) . \tag{2}$$

Inter-region visibility is defined as a *visibility graph* of regions where an edge from region *s* to region *r* implies that *r* is visible from *s*. Again, the graph is represented by a map (*V*) from the set of regions to the power set of regions (3). At each time step *t*, the search unit allocates a total amount of discrete search effort *Q* to one or many visible regions where $e_t(r)$ represents the amount of discrete search effort (between 0 and *Q*) allocated to region *r* at time step *t* (4). At a given time step *t*, the available search effort can only be allocated to regions that are visible from y_t (5) and the total amount of allocated effort must be less than or equal to *Q* (6). Moreover, the amount of effort allocated to any region *r* at time step 0 is equal to 0 (7).

$$V : R \rightarrow 2^R . \tag{3}$$

$$\forall t \in I : \forall r \in R : e_t(r) \in \{0,...,Q\} . \tag{4}$$

$$\forall t \in I : \forall r \in R : e_t(r) > 0 \Rightarrow r \in V(y_t) . \tag{5}$$

$$\forall t \in I : \sum_{r \in R} e_t(r) \leq Q . \tag{6}$$

$$\forall r \in R : e_0(r) = 0 . \tag{7}$$

The accessibility and visibility graphs share the same set of nodes but have different sets of edges. They can be seen as roadmaps of the original continuous environment (see [2] for roadmaps examples).

The *probability of containment* (*poc*) distribution over all the regions defines our knowledge of the search object's position before and during the search operation. The updated local *poc* value in each region and at each time step is conditional to the fact that the search object has not been detected before. If we assume that the search object is located somewhere in the environment, then the initial *poc* distribution (poc_0) will total 1.0 over the environment (8).

$$\sum_{r \in R} poc_0(r) = 1.0 \ . \tag{8}$$

At each time step, this distribution evolves according to the motion model of the search object and to the search unit's effort allocations (9).

$$\forall t \in I : \forall r \in R : poc_t(r) = \sum_{s \in R} d(s,r)\left[poc_{t-1}(s) - pos_{t-1}(s)\right], \tag{9}$$

where $d(s,r)$ is the probability that the search object moves from region s to region r within one time step and where $pos_{t\text{-}1}(s)$ is the local probability of success in region s at the previous time step (defined in (11)). Recalling that $poc_t(r)$ is the probability that the search object is located in region r at time step t and given that it has not been detected before time step t, the resulting sum over all the regions s may be interpreted as the remaining "mass of containment" in region r at time step t. The motion model is such that d may be represented as a matrix where each row is a source region s and where each column is a destination region r. The sum of each row is 1.0 (10) since we assume that the search object cannot escape from the environment.

$$\forall s \in R : \sum_{r \in R} d(s,r) = 1.0\,. \tag{10}$$

The probability of success $pos_t(r)$ (local to a given region r at a given time step t) is conditional to the fact that the search object has not been detected earlier. This probability is the product of the local probability of containment in r at time step t and of the conditional local probability of detection in r at time step t (11). Note that $pos_0(r)$ is equal to 0 for all regions r since the search operation begins at time step 1.

$$\forall t \in I : \forall r \in R : pos_t(r) = poc_t(r) \times pod_t(y_t, r, e_t(r)), \tag{11}$$

where $pod_t(y_t,r,e_t(r))$ is the probability of detection at time t in region r conditional to the fact that the search object is in region r at time t. Moreover, the pod function varies according to y_t due to inter-region visibility and as a function of $e_t(r)$, the amount of effort allocated in r at time step t.

In this paper, we assume that the detection model follows the exponential detection law of equation (12). The detectability index ($W_t(s,r)$) varies as a function of the source region s, of the destination region r and of time due to several factors (*e.g.*, weather).

$$\forall t \in I : \forall s,r \in R : \forall e \in \{0,\ldots,Q\} : pod_t(s,r,e) = 1 - \exp(-W_t(s,r) \times e)\,. \tag{12}$$

Our objective is to obtain a search plan P (a path and a sequence of effort allocations) (13) that maximizes the cumulative overall probability of success (COS) defined as the sum of the local probabilities of success across regions and time steps (14).

$$P = \left\langle [y_1, y_2, \ldots, y_T] \quad [e_1, e_2, \ldots, e_T] \right\rangle. \tag{13}$$

$$COS(P) = \sum_{t \in I} \sum_{r \in R} pos_t(r)\,. \tag{14}$$

4 Searching with Ants

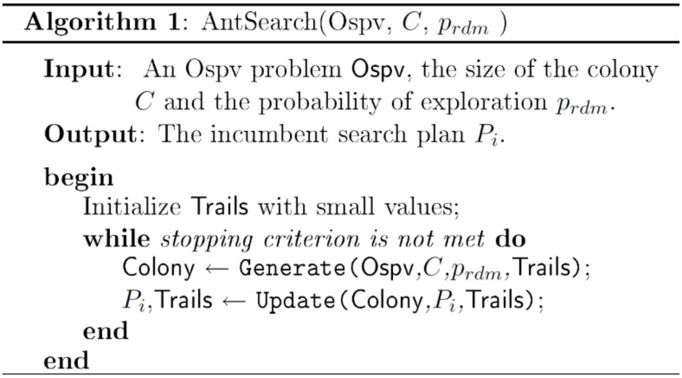
Ant colony optimization (ACO) is a general stochastic local search technique. As defined in [9], it constructs, at each iteration of the algorithm, a population of candidate solutions according to a common "memory" (the *pheromone trails*) to iteratively find better incumbent solutions. In view of the ACO metaheuristic metaphor, each candidate solution is built by an ant that stochastically chooses the solution's components based on their associated *pheromone value* and on a heuristic function that corresponds to the greedy "appeal" of this local decision. A typical definition for the probability of choosing a given action (or component) is shown in equation (15) where v_{act} is the pheromone value of the action *act* weighted by α , h_{act} is the heuristic value of the action *act* weighted by β and p_{act} is the resulting probability of choosing the action *act*.

$$p_{act} = \frac{(v_{act})^{\alpha}(h_{act})^{\beta}}{\sum (v_{act})^{\alpha}(h_{act})^{\beta}} \, . \tag{15}$$

Usually, the pheromone value of a given decision is updated proportionally to its quality and an evaporation factor decreases the pheromone values at each iteration in order to avoid getting stuck in a local optimum. In [3], Dorigo and Blum present a survey of ACO techniques.

4.1 The Ant Search Algorithm

Our ant search (AS) algorithm applies the idea of the ACO metaheuristic to the OSPV problem. This section defines the general algorithm and the *pheromone model* used to store and to update the trails. The outline of AS is presented on Figure 1.



```
Algorithm 1: AntSearch(Ospv, $C$, $p_{rdm}$ )

Input: An Ospv problem Ospv, the size of the colony
       $C$ and the probability of exploration $p_{rdm}$.
Output: The incumbent search plan $P_i$.
begin
    Initialize Trails with small values;
    while stopping criterion is not met do
        Colony ← Generate(Ospv,$C$,$p_{rdm}$,Trails);
        $P_i$,Trails ← Update(Colony,$P_i$,Trails);
    end
end
```

Fig. 1. The outline of the Ant Search algorithm

Given an OSPV problem, the total number of candidate search plans (solutions) to generate at each iteration, and a small probability of exploration, the algorithm first sets the pheromone trails to small values. Then it iteratively generates a set of candidate solutions and updates the trails and the incumbent according to its pheromone model (as defined in sections 4.1.1 and 4.1.2).

4.1.1 Generating the Colony

The *Generate* function constructs each solution of the colony by choosing the search unit's actions for each time step. There are two search unit's action types: moving from one region to another, and allocating one unit of effort in a visible region from its new position. Each ant can be seen as a search unit that tries to construct the best possible search plan using the colony's experience (defined as the pheromone trails). First, an ant chooses a region that is accessible from its current position. Then it sequentially allocates the Q units of available effort to visible regions. The process is repeated for each time step. In our implementation, the pheromone values correspond to the local probability of success (11) and to the overall probability of success at time t (16).

$$\forall t \in I : os_t = \sum\nolimits_{r \in R} pos_t(r)\,. \tag{16}$$

The local probability of success is used to select the effort allocation and the overall probability of success is used to select the path (see section 4.1.2).

In classical ACO algorithms, the ant's choices are guided by the pheromone values and by a heuristic function. Our first implementation of AS involved a greedy heuristic. However, the time required for computing the value of a greedy move based on the overall probability of success was quite high: for each accessible region, each possible combination with repetitions of visible regions must be evaluated. Thus we chose not use a heuristic in the implemented pheromone model and we replaced the usual equation (15) with equation (17). The main benefit of such an approach is to reduce the time consumed by an ant during its stochastic choices. Another benefit to our choice is the reduction of the number of parameters since there is no need for α and β nor for any other heuristic parameters.

$$p_{act} = v_{act} \Big/ \sum v_{act}\,. \tag{17}$$

In order to avoid stagnation, at each time step, the ant has a small probability p_{rdm} of choosing a random move and a random search effort allocation. In the AS context, this random choice can help diversify the search since the ants rely entirely on the pheromone trails. At the end of the *Generate* function, a colony of C candidate solutions has evolved.

4.1.2 Updating the Trails and the Incumbent Solution

At each time step, each ant performs two action types: it moves and it allocates the search effort. The first challenge was to store the pheromone values in relatively small data structures while keeping enough information. The solution we retained was to create 2 $T \times N$ tables (one per action type) and to store the pheromone by pairs of time steps and regions. The second challenge was to choose the values to store in the tables. As mentioned in section 4.4.1, we used the overall probability of success for the *path table* and the local probability of success for the *allocation table*.

Two versions of the algorithm were tested. In the first variant (called *Egalitarian AS*), the pheromone is updated for all candidate solutions of the colony. In the second variant (called *Elitist AS*), the pheromone is updated only when the candidate solution is better than the incumbent (many updates may occur for the same colony of search plans depending on their evaluation order). In all cases, a small evaporation factor is applied at each iteration to decrease the pheromone values. The most promising approach is Elitist AS. As a result, we present the results of the Elitist AS algorithm only.

5 Experimentation

The goal of our experiment was to compare the results of Elitist AS over a small set of instances to the results of a well known and well-founded solution scheme that guarantees the optimality of its solutions (given that it has enough time and resources to do so): ILOG CPLEX and a MILP model. The experiment was run on the following hardware: an Intel Core2 Quad Q6600 processor with 3 GB of RAM. All the code was developed using C++. The following assumptions hold for each instance:

- All environments are grids containing cells of 5 × 5 distance units.
- The search object's motion model (*d*) is random on rows and each row sums up to 1.0.
- The initial *poc* distribution and the initial search unit's position are randomly generated (using a uniform distribution).
- The accessibility graph is defined using a maximum accessibility range a_{rng} equal to 5.01 distance units.
- The visibility graph is defined using a maximum visibility range v_{rng} equal to 8.0 distance units.
- A region *r* is accessible (resp. visible) from another region *s* if the distance between the centers of *s* and *r* is less than a_{rng} (resp. v_{rng}).
- The *pod* function corresponds to equation (12) but is kept constant over time and $W_t(s,r)$ is such that

$$\forall t \in I : \forall s, r \in R : W_t(s,r) = \begin{cases} \dfrac{(v_{rng} - dist(s,r))}{area(r)}, & \forall s, r \in R : r \in V(s) \\ 0, & \forall s, r \in R : r \notin V(s) \end{cases}, \qquad (18)$$

where *dist*(*s*,*r*) is the distance between the center of region *s* and the center of region *r* (in distance units) and *area*(*r*) is the area of region *r* (in square distance units).

Table 1 presents the 7 problems used for the tests. While these environments are relatively small, the overall complexity of the problem is high since a search plan is both a path of *T* regions and a sequence of *T* combinations with repetitions of *Q* visible regions. The evaluation metrics are the *COS* measure of the final incumbent solution and the approximate time (in seconds) needed to obtain the resulting search plan which corresponds to the last incumbent update time. By using these metrics we avoid including the time used by CPLEX to prove the optimality of its incumbent solution. The maximum allowed resolution time is 7200 seconds.

Table 1. The environments

No	Height	Width	*T*	*Q*
0	2	2	4	5
1	3	3	9	5
2	4	4	16	5
3	5	5	25	5
4	6	6	36	5
5	7	7	49	5
6	10	10	100	5

5.1 A Generic Solver

By looking at the formal OSPV model we note that the *pod* function, the initial *poc* distribution, the search object's motion model and the search unit's initial position are a priori known data. Moreover, the detection model implements a *pod* function that varies as a function of a discrete search effort. Thus the formal model can be reformulated as a MILP. The advantage of such a reformulation is that algorithms for solving MILP to global optimality are in the public domain (*e.g.*, [6]) and that MILP solvers implementing programmable libraries are available (*e.g.*, ILOG CPLEX 11.2 and Concert Technology 2.7). In the version used for the tests (*i.e.*, 11.2), CPLEX uses Branch and Cut algorithms variants such as dynamic search to solve MILP models. Branch and Cut is a cutting plane algorithm and its solving process is not too far from the usual Branch and Bound procedure used for OSP problems (*e.g.*, [13]).

We present the results of 5 different configurations of CPLEX. The first one (*Default*) involves the default parameters. The second (*ScaleUp*) involves a MILP model where all the probabilities are multiplied by a large factor (100000). The third (*Feasibility*) puts the emphasis on finding feasible solutions. The fourth (*BestEst*) implements a best-first search approach based on the estimate of the *COS* value given by the heuristics of CPLEX. The fifth (*DepthFirst*) implements a depth-first search approach.

5.2 Ant Search Configurations

In order to estimate the impact of the parameters on the performance of Elitist AS, the algorithm is evaluated on all the 7 problems using the following configurations: a colony size C in {1, 100, 250, 500, 1000, 2500, 5000} with an exploration probability p_{rdm} of 0.1 and p_{rdm} in {0.0, 0.001, 0.01, 0.1, 0.25, 0.5, 0.75, 1.0} with C equal to 100. Then, to derive the statistics, the algorithm is evaluated on the 7 problems for 10 runs with $C = 1000$ and $p_{rdm} = 0.001$. For all tests, the evaporation factor is set to 0.1 and the initial pheromone values are set to 0.01. Moreover, Elitist AS is configured to stop if no improvement occurs within 900 seconds of the last incumbent's update.

6 Results and Discussion

Figure 2 presents the last incumbent's *COS* as a function of various configurations for problems 4 to 6. Other problems (while having the same general tendency) are not shown on the figure to avoid overcrowding it. The first graph shows the *COS* as a function of increasing colony size (C) with an exploration probability (p_{rdm}) of 0.1, the second graph shows the *COS* as a function of an increasing exploration probability (p_{rdm}) with a colony size (C) of 100. For the evaluated OSPV instances and the evaluated configurations, the general tendency shows an increase in the solution's quality among larger colonies. We see (in the second graph) that a small exploration probability (p_{rdm}) has a positive impact on the incumbent's quality of problems 5 and 6 while the highest p_{rdm} values have a negative impact on the last incumbent's *COS*. When p_{rdm} tends toward 1.0, the algorithm roughly corresponds to a random search of the environment and this is why the performance usually decreases for the larger values of p_{rdm}. This tendency shows the positive impact of using our pheromone model instead of a random search as it guides the exploration of the solutions' space.

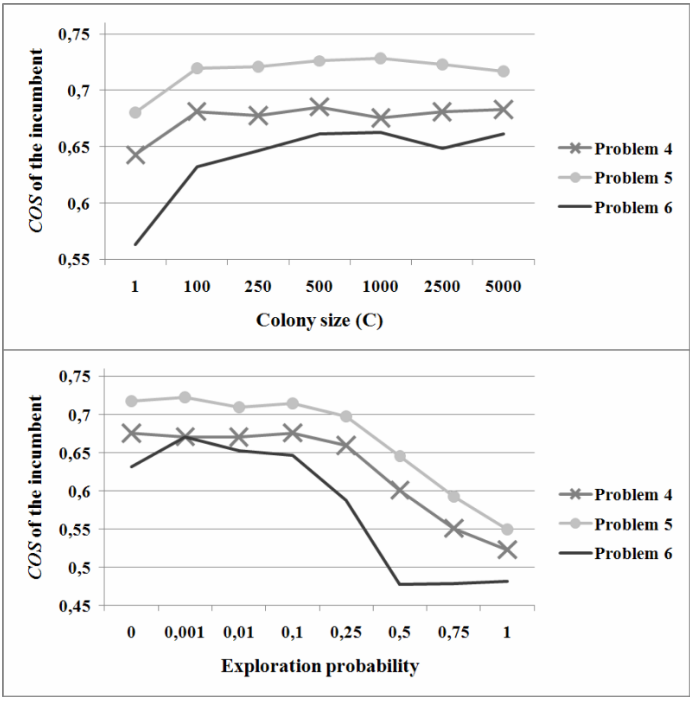


Fig. 2. For problems 4 to 6, the *COS* value of the incumbent for *C* in {1, 100, 250, 500, 1000, 2500, 5000} with $p_{rdm} = 0.1$ (above) and for p_{rdm} in {0.0, 0.001, 0.01, 0.1, 0.25, 0.5, 0.75, 1.0} with *C* = 100 (below)

Figure 3 presents the time (in seconds) spent to obtain the last incumbent's search plan as well as the corresponding *COS* value for problems 0 to 6. The time to last incumbent differs from the stopping time. This metric is preferred over the stopping time since it does not consider the time used by CPLEX to prove the optimality of its solution. The values displayed for Elitist AS are the average of 10 runs with *C* = 1000 and $p_{rdm} = 0.001$ and the values displayed for CPLEX are the best of the 5 tested configurations. A lower time value and a higher *COS* value imply a better performance. In all cases, CPLEX failed to prove the optimality of its solution within 2 hours. The MILP model of problem 6 is too large to fit in memory (considering the current hardware and software configuration of CPLEX). As a result CPLEX has failed to find a feasible solution. Moreover, the average *COS* of the last incumbent found by Elitist AS is equal or higher than the best *COS* of the last incumbent found by CPLEX for problems 0, 2, 4, 5 and 6. The exceptions are problem 1 and 3 where the average *COS* obtained by Elitist AS is slightly below the best one of CPLEX. Finally, we see that the average times to the last incumbent of Elitist AS are lower than the ones of

CPLEX. The exceptions are problem 4 where CPLEX found its first and last incumbent after approximately 550 seconds and problem 6 where CPLEX failed to start its solving process due to a lack of memory. The time to the last incumbent obtained by CPLEX on problems 4, 5 and 6 are irrelevant due to the poor solution's quality: the best *COS* values are approximately equal to 0.187, 0.053 and 0.0. These results are positive when we consider the fact that the Elitist AS algorithm was stopped 900 seconds after the last incumbent's update (if no improvement occurred). Introducing a mechanism to perturb the pheromone values after a specific delay of non improvement may yield to better results.

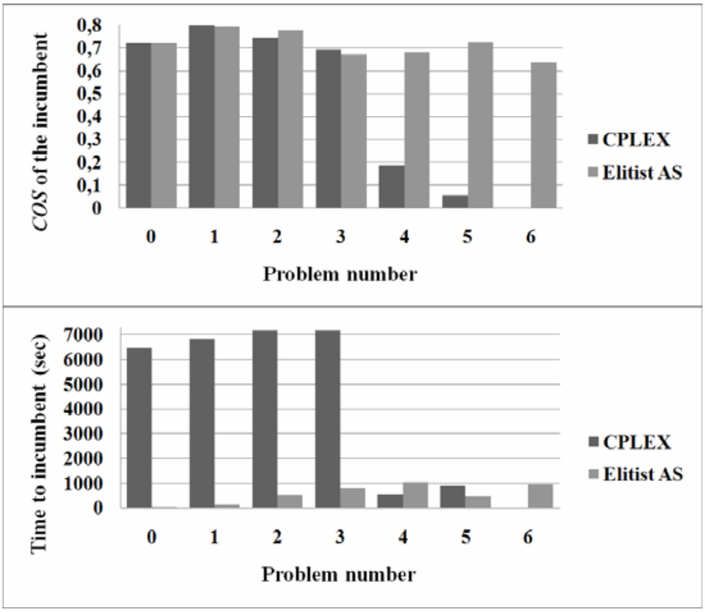


Fig. 3. The best *COS* obtained by CPLEX and the average *COS* obtained by Elitist AS with $C = 1000$ and $p_{rdm} = 0.001$ over 10 runs (above). The best time to the last incumbent obtained by CPLEX and the average time obtained by Elitist AS over the same 10 runs (below).

Figure 4 shows the distribution of the last incumbents' *COS* value obtained by Elitist AS for 10 runs on problems 0 to 6. For all problems, the *COS* values of the 10 runs are similar. There are few outliers considering an inter-quartile range (IQR) of 1.5. While providing *COS* values that are superior in average to the results obtained with our MILP model, Elitist AS is relatively constant in its incumbent quality (in view of the evaluated problems). Considering our 10 runs on problem 1 to 5, the *COS* values are within 0.005 of the mean in 95% of the cases. For problem 6, the confidence interval is of 0.03 with all the values and of 0.01 when removing the extreme *COS* value of 0.494. For problem 0, it is equal to 0.0. Considering the outlier obtained for problem 6 (*COS* = 0.494), further tests using the same random numbers with $p_{rdm} = 0.1$ suggest that this is a local optimum that can be avoided using a higher probability of exploration.

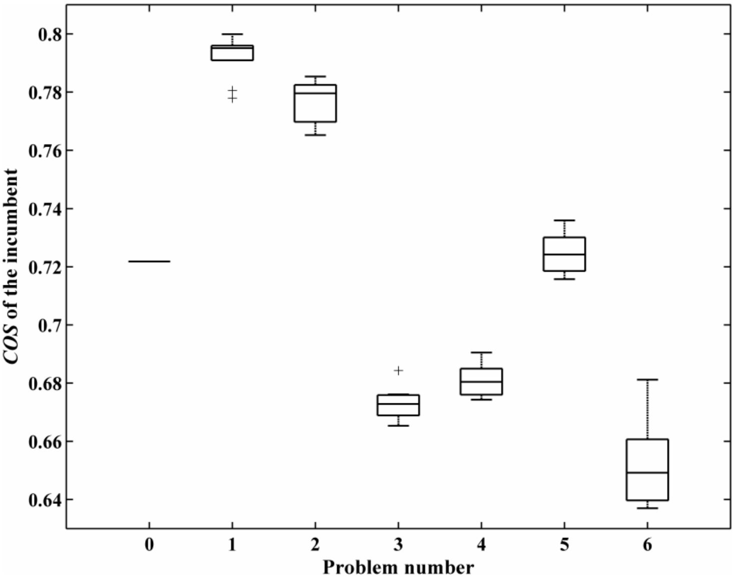


Fig. 4. For problem 0 to 6, a box plot (IQR = 1.5) illustrating the distribution of the last incumbent *COS* on 10 runs of Elitist AS. All the outliers are shown except for the value of approximately 0.494 obtained on problem 6 that was removed for clearness purpose.

7 Conclusion and Further Research

In this paper, we introduced the novel OSPV problem that models inter-region visibility and that uses concepts from search theory to model uncertainty on the search object's detectability and location. Moreover, we showed that our implementation of the ACO metaheuristic in the context of the OSPV problem (AS) is a promising technique to obtain search plans. Even if AS shares the main disadvantage of other local search techniques (it cannot guarantee the optimality of its last incumbent), it performed well in finding high quality solutions (in terms of *COS* when compared to our MILP formulation) to each tested instance. In all the experimental cases, the ratios of the last incumbent's *COS* to the time to the last incumbent for Elitist AS stayed high in comparison to the ones obtained by our general solving scheme involving CPLEX, a widely used MILP solver. Since practical OSP-like problems usually involve larger environments than those presented in our experiment (*e.g.*, a thousand of regions in the case of search and rescue operations), the AS methodology still needs to be enhanced. Further work and experiments include the development of fast heuristic functions, the development of better pheromone models to handle the environment's size, the reevaluation of Egalitarian AS and the development of algorithms based on other metaheuristics and on constrained programming.

References

1. Benkoski, S., Weisinger, J.R., Monticino, M.G.: A Survey of the Search Theory Literature. Naval Research Logistics 38, 469–494 (1991)
2. de Berg, M., Cheong, O., van Kreveld, M., Overmars, M.: Computational Geometry, Algorithms and Applications, 3rd edn. Springer, Berlin (2008)
3. Dorigo, M., Blum, C.: Ant Colony Optimization Theory: a Survey. Theoretical Computer Science 344, 243–278 (2005)
4. Eagle, J.N.: The Optimal Search for a Moving Target when the Search Path is Constrained. Operations Research 32(5), 1107–1115 (1984)
5. Frost, J.R.: Principles of Search Theory, part I-IV (2000)
6. Garfinkel, R.S., Nemhauser, G.L.: Integer Programming. John Wiley & Sons, New York (1972)
7. Hansen, S.R.: Applications of Search Theory to Coordinated Searching by Unmanned Aerial Vehicles. Master's thesis. Dept. of Mechanical Engineering, Brigham Young Univ., Provo, Utah, USA (2007)
8. Hollinger, G.A.: Search in the Physical World. Proposal for Ph.D. Thesis, Robotics Institute, Carnegie Mellon Univ., Pittsburgh, PA, USA (2008)
9. Hoos, H.H., Stützle, T.: Stochastic Local Search: Foundations and Applications. Elsevier, The Netherlands (2004)
10. Koopman, B.O.: Search and Screening: General Principles with Historical Applications. Pergamon Press, New York (1980)
11. Lau, H.: Optimal Search in Structured Environments. Ph.D. Thesis. The University of Technology, Sydney, Australia (2007)
12. Morin, M., Lamontagne, L., Abi-Zeid, I., Lang, P., Maupin, P.: The Optimal Searcher Path Problem with a Visibility Criterion in Discrete Time and Space. In: Proceedings of the 12th International Conference on Information Fusion, pp. 2217–2224. ISIF IEEE (2009)
13. Stewart, T.J.: Search for a Moving Target when the Searcher Motion Is Restricted. Computers and Operations Research 6, 129–140 (1979)
14. Stone, L.D.: Theory of Optimal Search. Topics in Operations Research, INFORMS (2004)
15. Trummel, K.E., Weisinger, J.R.: The Complexity of the Optimal Searcher Path Problem. Operations Research 34(2), 324–327 (1986)
16. Washburn, A.R.: Branch and Bound Methods for a Search Problem. Naval Research Logistics 45, 243–257 (1998)

Automatic Discovery of Network Applications: A Hybrid Approach

Mahbod Tavallaee[1], Wei Lu[2], Ebrahim Bagheri[3], and Ali A. Ghorbani[1]

[1] Information Security Centre of Excellence, University of New Brunswick
[2] Q1 Labs Inc., Fredericton, New Brunswick, Canada
[3] National Research Council Canada - IIT and Athabasca University - SCIS

Abstract. Automatic discovery of network applications is a very challenging task which has received a lot of attentions due to its importance in many areas such as network security, QoS provisioning, and network management. In this paper, we propose an online hybrid mechanism for the classification of network flows, in which we employ a signature-based classifier in the first level, and then using the weighted unigram model we improve the performance of the system by labeling the unknown portion. Our evaluation on two real networks shows between 5% and 9% performance improvement applying the genetic algorithm based scheme to find the appropriate weights for the unigram model.

1 Introduction

Accurate classification of network traffic has received a lot of attentions due to its important roles in many subjects such as network security, QoS provisioning, network planning and class of service mapping, to name a few. Traditionally, traffic classification relied to a large extent on the transport layer port numbers, which was an effective way in the early days of the Internet. Port numbers, however, provide very limited information nowadays due to the increase of HTTP tunnel applications, the constant emergence of new protocols and the domination of peer-to-peer (P2P) networking applications. An alternative way is to examine the payload of network flows and then create signatures for each application. It is important to notice that this approach is limited by the fact that it fails to detect 20% to 40% of the network flows because of the following reasons:

- Lack of regular update to support new releases of applications.
- Not being aware of all types of applications being developed around the world.
- Great deal of variation in P2P applications.
- The emergence of encrypted network traffic.

Observing daily traffic on a large-scale WiFi ISP[1] network, Fred-eZone[2], over a half year period (from June 2007 to December 2007), we found that there are about 40% of network flows that cannot be classified into specific applications by the state-of-the-art signature-based classifiers, i.e., 40% network flows are labeled as unknown applications.

[1] Internet Service Provider.

[2] http://www.fred-ezone.com

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 208–219, 2010.

Addressing the limitations of the signature-based approach, we propose a hybrid mechanism for online classification of network flows, in which we apply two signature-based methods sequentially. In the first level, we employ a powerful traffic classification software, MeterFlow[3], to detect the network flows containing known applications signatures. We then apply a machine learning based approach to examine the payloads of network flows. However, instead of searching for the exact signatures, we extract the characteristics of payload contents using the Weighted Unigram Payload Model. The main contributions of this paper are:

- Employing the unigram of packet payloads to extract the characteristics of network flows. This way similar packets can be identified using the frequencies of distinct ASCII characters in the payload.
- Applying a machine learning algorithm to classify flows into their corresponding application groups. For the experiments we used the J48 decision tree, the Weka implementation of C4.5 [1], since it demonstrated to have high accuracy while maintaining a reasonable learning time. In addition, decision trees are shown to have a reasonable height and need a few comparisons to reach a leaf which is the final label, and therefore have a reasonably short classification time.
- Assigning different weights to the payload bytes to calculate the unigram distribution. We believe that depending on the applications those bytes that include signatures, should be given higher weights.
- Applying genetic algorithm to find the optimal weights that results in improvement in the accuracy of the proposed method.

The rest of the paper is organized as follows. Section 2 introduces the related work, in which we provide a brief overview of signature-based traffic classiffication approaches. Our proposed hybrid traffic classification scheme will be explained in Section 3. Section 4 summarizes the unigram payload model. In Section 5, we formally define our problem and explain how we apply genetic algorithms to improve the accuracy of our proposed traffic classification method. Section 6 presents the experimental evaluation of our approach and discusses the obtained results. Finally, in Section 7, we draw conclusions.

2 Related Work

Early common techniques for identifying network applications rely on the association of a particular port with a particular protocol [2]. Such a port number based traffic classification approach has been proved to be ineffective due to: 1) the constant emergence of new peer-to-peer networking applications that IANA[4] does not define the corresponding port numbers; 2) the dynamic port number assignment for some applications (e.g. FTP); and 3) the encapsulation of different services into a same application (e.g. chat or steaming can be encapsulated into the same HTTP protocol). To overcome this issue, there have been recently significant contributions towards traffic classification. The

[3] http://www.hifn.com

[4] Internet Assigned Numbers Authority (http://www.iana.org/assignments/port-numbers)

most currently successful approach is to inspect the content of payloads and seek the deterministic character strings for modeling the applications. For most applications, their initial protocol handshake steps are usually different and thus can be used for classification. Moreover, the protocol signatures can be modeled through either public documents such as RFCs or empirical analysis for deriving the distinct bit strings on both TCP and UDP traffic.

In [3], Gummadi et al. develop a signature model for KaZaA workload characterization through analyzing a 200-day trace of over 20 tera bytes of Kazaa P2P traffic collected on a campus network. In [4], Sen et al. analyze the application layer protocols and then generate the signatures of a few P2P applications. Although the protocol semantic analysis improves the accuracy of signatures, it makes the real-time analysis of the backbone traffic impossible since the underlying assumption is that every packet is being inspected. In their consequent work [5], Sen et al. examine available specification and packet-level traffic traces for constructing application layer signatures, and then based on these signatures, P2P traffic are filtered and tracked on high-speed network links. Evaluation results show that their approach obtains less than 5% false positives and false negatives.

To have a better understanding of this approach, Table 1 illustrates the signatures of 11 typical applications in which to print the signatures, alphanumeric characters are represented in the normal form, while non-alphanumeric ones are shown in the hex form starting with "0x".

Table 1. Payload signatures of some typical network applications

Application	Payload	Offset	Signatures
Bit Torrent	source	1	BitTorrent
HTTP Image Transfer	destination	4	image/
HTTP Web	source	0	GET
Secure Web	destination	0	0x16 0x03
MSN Messenger	source	0	MSG
MS-SQL	destination	0	0x04 0x01 0x00 0x25 0x00 0x00 0x01 0x00 0x00 0x00 0x15 0x00 0x06 0x01 0x00 0x1B
POP	destination	0	+OK
SMTP	source	0	EHLO
Windows File Sharing	destination	4	|FF|SMB
Yahoo! Messenger	destination	0	YMSG

As can be seen in Table 1, each application has a unique set of characters to be identified. *Secure Web* traffic can be identified by searching the hex string "1603" in the beginning of a flow payload. Similarly *Yahoo! Messenger* traffic can be detected by finding the ASCII string of "YMSG" in the payload. However, these signatures do not necessarily start from the beginning of the payload. For example, to identify *HTTP Image Transfer* traffic, we should search for the string "image/" starting from the 5^{th} byte of the payload. This starting point is referred as *offset* in Table 1. Moreover, it is important to know which side of the connection, client or server, produce the signature. For example, the signature to detect *HTTP Web* is in the source payload, i.e., the ASCII string "GET" is sent by the client, initiator of the connection, to the server. This information helps signature-based methods to improve their performance by looking at a fewer number of signatures in either the source or destination payloads.

Although this approach has a high accuracy for identifying network applications, it fails to detect 20% to 40% of the network flows. Some of the reasons that cause this problem are:

- Whenever a newer version of an application is released, all the signatures should be updated which usually cannot be done immediately.
- There are hundreds of applications being developed by small groups around the world which networking companies are not aware of.
- Peer-to-peer (P2P) applications employ lots of protocols with different signatures to prevent their traffic being filtered by organizations. Usually these signatures are variants of famous P2P applications (e.g., Bit Torrent, Gnutella, eDonkey) with small changes and kept confidential by crime organizations.
- Encrypted traffic is one the biggest challenges traffic classifiers are facing. Data encryption is becoming more popular with the growth of commercial services on the Internet. Besides, it is widely used to hide malicious activities such as botnet traffic.

In the next section, we propose our hybrid method which is capable of detecting new releases and variants of existing applications. Furthermore, newly created applications and different variant types of P2P software will be classified in a relevant category with similar characteristics. However, detecting encrypted traffic still remains as a challenging issue.

3 Hybrid Traffic Classification Scheme

As explained in the previous section, to overcome the shortcomings of port-based traffic classification, researchers have proposed new approaches based on the examination of payload content. These signature-based methods are able to detect a wide range of applications providing the specific signatures. Besides, They are fairly accurate and generate almost no false positives. These advantages have made the signature-based traffic classifiers very popular. However, conducting a thorough analysis of state-of-the-art signature based classifiers, we observed that depending on the type of network (e.g. Universities, ISPs, Enterprises) between 20% to 40% of the traffic is still unknown. This unknown traffic mainly consists of new applications, variation of old applications or encrypted traffic. To overcome this issue, we propose a hybrid mechanism for online classification of network flows, in which we apply two signature-based methods sequentially. In the first level, we employ a powerful traffic classification software, MeterFlow, to detect the network flows containing known applications signatures. We then apply a machine learning based approach to examine the payloads of network flows. However, instead of searching for the exact signatures, we extract the characteristics of payload contents using the Weighted Unigram Model.

In order to achieve reliable results, the machine learning based classifier needs to be provided with an up-to-date training set. To this end, we have defined learning time intervals, e.g. 1 day, at the end of which the classifier will be trained by the latest training set. This training set is composed of known flows labeled by the signature-based traffic classifier in the previous time interval. Figure 1 illustrates the structure of the hybrid traffic classifier. In the first interval which we do not have any training set, we only rely on the labels from MeterFlow. The flows with known labels will be used as the training set for the weighted unigram classifier in the next time interval. During the second time interval, flows are given to MeterFlow to do the labeling. The unknown flows will then be given to the weighted unigram classifier to be labeled based on the learned information in the previous time interval. Finally, the known flows from MeterFlow and the labeled flows by the weighted unigram classifier will be mixed together and reported to the administrator as the application labels.

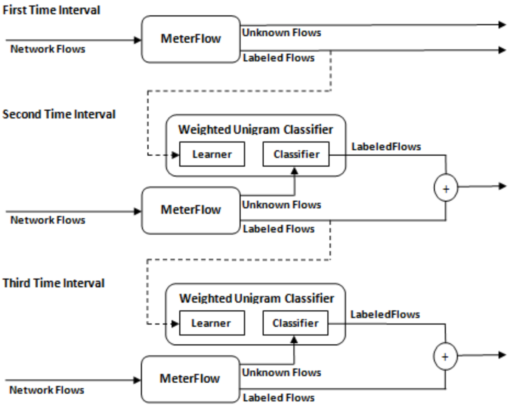


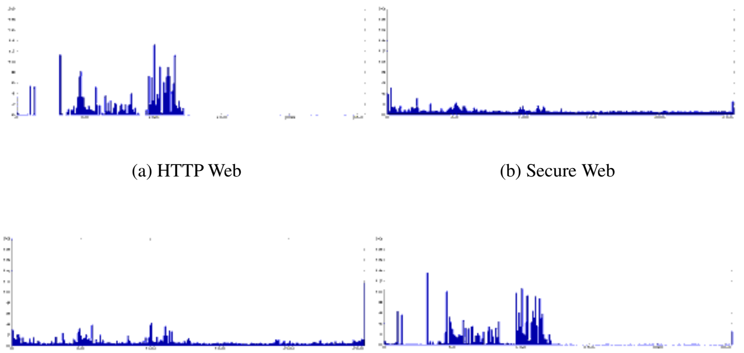
Fig. 1. General structure of the hybrid traffic classifier

4 Unigram Payload Model

N-grams are language-independent means of gauging topical similarity in text documents. Traditionally, the n-grams technique refers to passing a sliding window of *n* characters over a text document and counting the occurrence of each n-gram. This method is widely employed in many language analysis tasks as well as network security.

Applying the same idea on network packets, one can consider unigram (1-gram) of a network packet as a sequence of ASCII characters ranging from 0 to 255. This way similar packets can be identified using the frequencies of distinct ASCII characters in the payload. In order to construct the unigram payload distribution model, we extract the first M bytes of the payload and count the occurrence frequency of each ASCII character. However, since some of the applications bear their signatures in the source payload such as *HTTP Web* and some of them have their signatures in the destination payload like *Secure Web*, we consider source and destination payloads as separate pieces of information. In other words, each ASCII character has two frequency values, one for the source payload (data sent by the client, initiator of the connection, toward the server) and one for the destination payload (data received by the client from the server).

By observing and analyzing the known network traffic applications, labeled by a signature-base classifier called MeterFlow, over a long period on large-scale WiFi ISP network, we found that the unigram distribution of source and destination payloads



(a) HTTP Web

(b) Secure Web

(c) Bit Torrent

(d) SMTP

Fig. 2. Average unigram distribution of source payload

can be used as a powerful tool to detect the applications. Figure 2 shows the unigram distribution of several applications namely, HTTP Web, Secure Web, SMTP, POP, Bit Torrent, Oracle. The x axis in the figures is the ASCII characters from 0 to 255, and the y axis shows the frequency of each ASCII character in the first 256 bytes of the source payload. As it can be seen in Figure 2, text-based applications (e.g., *HTTP Web*) can be easily distinguished from binary applications (e.g., *Bit Torrent*), as well as encrypted applications (e.g., *Secure Web*) since they use only a portion of ASCII characters, especially alphanumeric ones. Moreover, having an exact investigation of similarly behaving protocols such as *HTTP Web* and *SMTP*, we can still separate them based on the most frequent characters.

After applying the unigram distribution model on network flows, we can map each flow to a 512-tuple feature space $\langle f_0, f_1, ..., f_{511} \rangle$ such that f_0 to f_{255} shows the frequencies of corresponding ASCII characters in the source payload. Similarly, f_{256} to f_{511} hold the frequencies in the destination payload.

Having specified the applied network features, we focus on the selection of a high-performance classifier. In order to choose an appropriate classifier, we selected some of the most popular classification methods implemented by Weka [6] and performed some evaluations to compare the accuracy, learning time, and classification time. We finally selected the J48 decision tree, the Weka implementation of C4.5 [1], since it has a high accuracy while maintaining a reasonable learning time. In addition, decision trees are shown to have a reasonable height and need fewer comparisons to reach a leaf which is the final label, and therefore have a very short classification time. Taking advantage of the J48 decision tree as a traffic classifier, we evaluated our proposed method on two real networks.

Although our evaluations on the two networks showed promising results, we still believe that the performance can be improved by assigning different weights to the payload bytes based on the degree of importance. However, finding the appropriate weights is a challenging task. To this end, we employ a genetic algorithm based scheme to find the weights.

In the next section, we formally define our problem and explain how we apply genetic algorithms to improve the accuracy of our proposed traffic classification method.

5 Problem Formulation

In this section, we formally describe how the network application discovery problem can be performed through the combination of genetic algorithms and decision trees. Essentially, we formulate the network application discovery problem as a classification problem, i.e., given the values for a specific set of features extracted from the network flows, we identify the possible application that has generated this payload using the decision trees.

As was mentioned earlier, payload can be viewed as a multidimensional vector, where in the context of a unigram analysis of the payload, the frequency of each ASCII character in the payload represents one feature in the vector space. For instance, if the character '∞' is repeated 20 times in the payload, the value for the 236^{th} dimension[5] of the representative vector would be 20. So with this simple yet intuitive formulation of the features of the payload vector space, it is possible to construct a learning classification machine that operates over features that are the frequency of each ASCII character in the payload. We formally describe this as follows:

Definition 1. *Let $\kappa_0^s, ..., \kappa_{255}^s$ be the set of 256 available ASCII characters in the source payload and $\kappa_0^d, ..., \kappa_{255}^d$ be those in the destination payload, and Ψ be a given payload generated by an application $\lambda \in \Lambda$, Λ being the set of all known network applications. We denote $\nu_{\kappa_i^{s/d}}(\Psi)$ as the frequency of $\kappa_i^{s/d}$ in Ψ. Further, $\Im : \Psi \rightarrow \Lambda$ is a learning classification machine that maps a payload Ψ with frequency ν to a network application such as λ.*

Based on this definition, we need to employ a classifier that infers the correct network application label given the features from the payload. To achieve this we employ the J48 decision tree learning algorithm [1] that builds a decision tree classifier from a set of sample payloads and their corresponding network application labels. The learned decision tree classifier will enable us to find the possible network application label for a payload from an unknown network application. Here, the learned J48 classifier is our required $\Im$ mapping function.

Now lets consider the *HTTP Web* application payload. The payload starts with `GET` and continues with further detailed information about that specific connection such as the protocol version, host and others. So for this specific application, i.e. *HTTP Web*, the first few characters are representative enough of the application; therefore, by just looking at these 3 characters we are able to identify the signature for this application and easily assert that this payload has been generated by the *HTTP Web* network application. Similarly, other network applications have specific signatures that are present in some designated positions in the payload. So, it can be inferred that some positions in the payload are more *discriminative* in the process of classifying the payloads for their generating network applications. For this reason, it is important to place more weight

[5] 236 being the index of ∞ in the ASCII character list.

on the features that appear in these more important positions. We achieve this through a weighted scheme over the features.

Definition 2. *Suppose Ψ is a given payload of length η. We let Ψ_p, $p \in [0, \eta]$ denote the p^{th} position in the payload Ψ, and $\omega(\Psi_p)$ denote the weight (importance) of position p in the payload. The weighted feature of the vector space is defined as:*

$$\omega\nu_{\kappa_i^{s/d}}(\Psi) = \sum_{p \in P} \omega(\Psi_p) \tag{1}$$

where P represents the positions in which $\kappa_i^{s/d}$ has appeared in Ψ.

Simply, this definition defines that each position in the payload has its own significance in the application discovery process; therefore, some of the features may be more discriminative of the applications and hence should receive a higher weight. Based on this weighting scheme, we now revise the dimensions of our vector space such that the frequency of each ASCII character observed in the payload is multiplied by the weight of the position that it was located. For instance, if ∞ is observed in positions 5 and 210, and the weight for 5 and 210 are 1 and 8, respectively, the value for the 236^{th} dimension of the representative vector would be 9 rather than simply being 2. Accordingly, we have:

Definition 3 (Extends Definition 1). $\Im^{\omega} : \Psi \rightarrow \Lambda$ *is a learning classification machine that maps a payload Ψ with frequency $\omega\nu$ to a network application such as $\lambda \in \Lambda$.*

Now, since $\Im^{\omega}$ is sensitive to the weights given to the positions in the payload, a method needs to be devised to compute the appropriate weights for the positions. For this purpose, we employ a genetic algorithm based process to find the weights.

Briefly stated, genetic algorithms are non-deterministic and chaotic search methods that use real world models to solve complex and at times intractable problems. In this method the optimal solution is found by searching through a population of different feasible solutions in several iterations. After the population is studied in each iteration, the best solutions are selected and are moved to the next generation through the applications of genetic operators. After adequate number of generations, better solutions dominate the search space therefore the population converges towards the optimal solution. We employ this process over the weights of the positions in the payload to find the optimal weights for each location. The process that we employ is explained in the following:

- The objective is that for a payload Ψ with length of η the optimal corresponding weight vector ω_O is found. So, initially a pool of random weight vectors of length η are generated: $\omega_1, ...\omega_n$;
- For each ω_i a learning classification machine ($\Im^{\omega_i}$) is developed given a set of learning instances;
- $\Im^{\omega_i}$ is evaluated based on its performance accuracy over a given set of test instances. The accuracy of $\Im^{\omega_i}$ represents the genetic fitness function; therefore, the accuracy of $\Im^{\omega_i}$ is the fitness of ω_i among the other weight vectors in the genetic algorithm pool of possible solutions;
- The best set of weights based on their fitness value are selected and are moved onto the next generation and the rest of the weight vectors are discarded;

- The genetic operators, i.e., the mutation and crossover operators, are applied on the weight vectors present in the genetic algorithm pool of possible solutions and new weight vector solution instances are generated;
- The process of the genetic algorithm is repeated for Gen generations;
- Once the algorithm reaches a steady state and stops, the weight vector with the best fitness is selected and will be used as the most appropriate weight vector (ω_O) for the application discovery process using $\Im^{\omega_O}$.

In the above process, the weight vectors that are needed for the payload are defined using the genes in the genetic algorithm and the fitness function is defined as the accuracy of the learning classification machine developed based on that specific weight vector (gene). The outcome of the genetic algorithm process provides us with the optimal set of weights for the positions of the ASCII characters in the payloads. This optimal set of weights can be used to learn a classifier that can best find the network application for new payloads whose generating application is not known.

To implement this process, we employed the JGAP (Java Genetic Algorithms Package) software package [7]. In our experiments, reported in the following section, we apply the default crossover technique implemented in JGAP with the population size of 500 evolving for 50 generations. The default crossover rate is [population size/2], and the value for mutation rate is 1/15. Moreover, we consider the first 256 bytes of source and destination payloads since the rest of the payload does not contain any signature and decreases the classifier performance by including noise data.

6 Experiments

To evaluate our proposed method we prepared two data set from various networks. The first data set is prepared using the traffic captured in the Information Security Center of Excellence (ISCX) in the University of New Brunswick during 10 days. Having passed this traffic through MeterFlow we ended up with 28% remaining as unknown. For our second data set we used 3-hour captured traffic from a large-scale ISP network[6]. This data set is composed of 35% unknown flows since it is an ISP network and contains lots of peer to peer applications which are really hard to detect by the state of the art signature-based traffic classifiers.

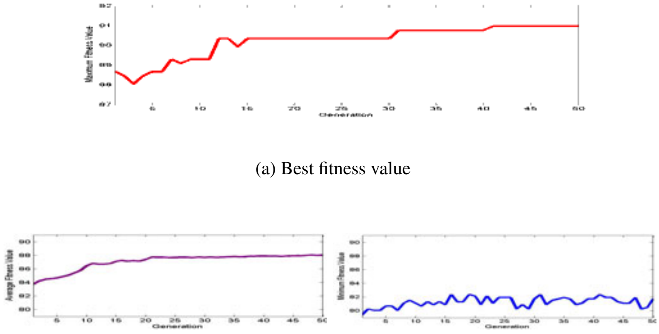
Table 2. Accuracy of the proposed method for the ISCX and ISP networks

	ISCX Network	ISP Network
Base Accuracy	81.93%	81.72%
First Generation Accuracy	83.69%	80.74
Optimal Accuracy	90.97%	86.55%

For the evaluation of the weighted unigram model, we filtered out the unknown flows from our data sets and used the rest as the training and testing set. We then extracted the unigram features of the source and destination payloads to evaluate our decision

[6] Due to privacy issues we do not reveal the name of this ISP.

tree based classifier. However, to have enough samples of each application for training, we decided to only keep those with more than 200 records in the data set. As a result, in our first data set we ended up with 7 applications namely, *FTP*, *HTTP Image Transfer*, *HTTP Web*, *DNS*, *Secure Web*, *SSH*, *Web Media Documents*. Applying the same approach for the second data set we left with 14 applications namely, *Bit Torrent, HTTP Image Transfer, HTTP Web, DNS, Secure Web, MSN Messenger, MS-SQL, NTP, Oracle, POP, SMTP, Windows File Sharing, Yahoo Messenger, Web Media Documents*.



(a) Best fitness value

(b) Average fitness value

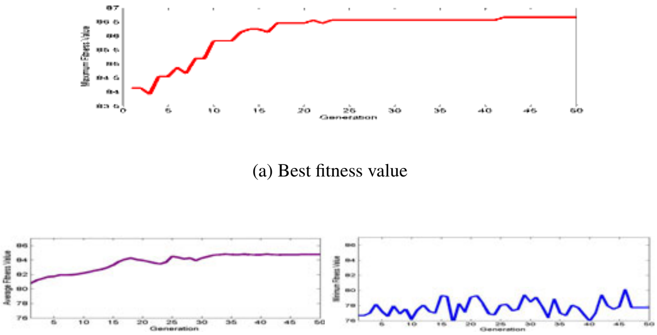
(c) Worst fitness value

Fig. 3. Fitness value vs. generation number calculated for the ISCX network

For the experiments we applied J48 decision tree classifier. In the first step we evaluated our classifier using the payload unigram features with equal weights. For the evaluation we employed 10-fold cross-validation to obtain reliable result. We then applied the genetic algorithm technique to find the appropriate weights to obtain higher accuracy. The experiment took approximately five days to complete since the classifier model should be updated during each run of the fitness function which takes a few seconds. As illustrated in Table 2, the best fitness value, *optimal accuracy*, returned is 90.97% and 86.55% for ISCX and ISP networks, respectively. This means that we have achieved about 9% and 5% performance increase in our applied data sets, respectively. Moreover, *base accuracy* shows the performance of the classifier using equal weights and *first generation accuracy* is the average fitness value in the first generation.

Figures 3 and 4 show the evolution of each generation in the experiments on the ISCX and ISP networks, respectively. In the first data set, Figure 3(a), the fitness value of the best individual for each generation has a noticeable increase until generation 15. However, in the next 35 generations there is only about 1% increase in the fitness value. Similarly, as it is illustrated in 4(a), for the ISP network there is an approximate steady increase until generation 17, at which point it is apparent that almost the best possible weights have been achieved.

Although the decision tree based classifier achieves a high classification accuracy during the experimental evaluation with cross-validation, the results might be misleading due to the fact that the hybrid classifier should deal with real unknown flows. However, having no information about the unknown flows, we experienced some difficulties to evaluate the weighted unigram classifier.



(a) Best fitness value

(b) Average fitness value

(c) Worst fitness value

Fig. 4. Fitness value vs. generation number calculated for the ISP network

To have a better estimation of the accuracy of our method, we captured corresponding traffic of five different applications separately. We then pass the traffic throuhg MeterFlow and used the known flows as a training set to detect the unknown portion. Table 3 illustrates the total accuracy of our proposed hybrid classifier for each specific applications. As can be seen in the table, the weighted unigram approach has increased the performance of signature-based approach by detecting 61% (31 out of 51) of the unknown flows.

Table 3. Accuracy of the proposed method for the ISCX and ISP networks

Applications	Total Flows	Unknown Flows	Successfully Classified	Total Accuracy
MSN Messenger	53	0	0	100%
Bit Torrent	103	27	18	91.26%
HTTP Web	121	12	6	95.04%
SMTP	74	4	2	97.30%
Windows File Sharing	91	8	5	96.70%
Total	442	51	31	95.48%

7 Conclusions

In this paper, we propose a hybrid mechanism for online classification of network flows, in which we apply two signature-based methods sequentially. In the first level, we employ a powerful traffic classification software, MeterFlow, to detect the network flows containing known applications signatures. We then apply a machine learning based approach to examine the payloads of network flows. However, instead of searching for the exact signatures, we extract the characteristics of payload contents using the Unigram Payload Model. Having a detail analysis of application signatures, we observed that the signatures are present in some designated positions in the payload, and it is important to place more weight on the features that appear in these more important positions. This is achieved through a weighted scheme over the unigram features. However, finding the appropriate weights is a challenging task. To this end, we employ a genetic algorithm based scheme to find the weights. Our evaluation on two real networks, showed promising results compared to other methods in detecting tunneled applications and variation of existing ones while maintaining a comparable accuracy in detecting old applications.

Acknowledgements

The authors graciously acknowledge the funding from the Atlantic Canada Opportunity Agency (ACOA) through the Atlantic Innovation Fund (AIF) to Dr. Ghorbani.

References

1. Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco (1993)
2. Moore, D., Keys, K., Koga, R., Lagache, E., Claffy, K.: The CoralReef Software Suite as a Tool for System and Network Administrators. In: Proceedings of the 15th USENIX conference on System administration, pp. 133–144 (2001)
3. Gummadi, K., Dunn, R., Saroiu, S., Gribble, S., Levy, H., Zahorjan, J.: Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. ACM SIGOPS Operating Systems Review 37(5), 314–329 (2003)
4. Sen, S., Wang, J.: Analyzing peer-to-peer traffic across large networks. In: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment, pp. 137–150 (2002)
5. Sen, S., Spatscheck, O., Wang, D.: Accurate, scalable in-network identification of p2p traffic using application signatures. In: Proceedings of the 13th international conference on World Wide Web, pp. 512–521 (2004)
6. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11(1) (2009)
7. Meffert, K., Rotstan, N., Knowles, C., Sangiorgi, U.: JGAP–Java Genetic Algorithms and Genetic Programming Package, http://jgap.sf.net

Overlap versus Imbalance

Misha Denil and Thomas Trappenberg

Faculty of Computer Science
Dalhousie University
6050 University Avenue
Halifax, NS, Canada
B3H 1W5
denil@cs.dal.ca,
tt@cs.dal.ca

Abstract. In this paper we give a systematic analysis of the relationship between imbalance and overlap as factors influencing classifier performance. We demonstrate that these two factors have interdependent effects and that we cannot form a full understanding of their effects by considering them only in isolation. Although the imbalance problem can be considered a symptom of the small disjuncts problem which is solved by using larger training sets, the overlap problem is of a fundamentally different character and the performance of learned classifiers can actually be made worse by using more training data when overlap is present. We also examine the effects of overlap and imbalance on the complexity of the learned model and demonstrate that overlap is a far more serious factor than imbalance in this respect.

1 Introduction

The imbalance problem occurs when the available training data contains significantly more representatives from one class compared to the other. Many classifiers have been shown to give poor performance in identifying the minority class in cases where there is a large imbalance [1]. The machine learning literature acknowledges the imbalance problem as a major obstacle to building accurate classifiers and there has been significant effort invested in searching for solutions [2,3,4,5,6], as well as some investigation into possible root causes of the problem itself [2]. Recent work shows that imbalance is not a problem when the overall size of the training set is sufficiently large [7,8]. These findings suggest that it is instead the problem of small disjuncts—exacerbated by imbalance and small training sets—which is the real cause of poor performance in these cases.

The overlap problem occurs when a region of the data space contains a similar number of training data for each class. This leads to the inference of near equal estimates for the prior probabilities of each class in the overlapping region and makes it difficult or impossible to distinguish between the two classes. There has been comparatively little work done on the overlap problem [5,9,10]; however, recent findings by Garcia et al. [1] have shown that overlap can play an even larger role in determining classifier performance than imbalance. The performance of

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 220–231, 2010.

Support Vector Machines (SVMs) in this area is of special interest due to the findings by Japkowicz et al. which suggest that SVMs are not sensitive to the imbalance problem in cases where the classes are separable [6].

Previous investigations of the overlap and imbalance problems have taken place largely in isolation. Although some authors have performed experiments in the presence of both factors, the nature of their interaction is still not well understood. Our work demonstrates that these two problems acting in concert cause difficulties that are more severe than one would expect by examining their effects in isolation. This finding demonstrates that we cannot achieve a full understanding of these problems without considering their effects in tandem.

In this paper we provide a systematic study of the interaction between the overlap and imbalance problems as well as their relationship to the size of the training set. We outline a method for testing the hypothesis that these two factors influence classifier performance independently and show through experimentation that this hypothesis is false for the SVM classifier. Finally, we illustrate a connection between model complexity and model performance in the presence of overlap and imbalance.

We have chosen to focus our investigation on SVMs since they have been shown to be particularly robust in the presence of the factors we wish to investigate; however, since our method does not rely on the particulars of the SVM formulation we expect the results reported here to generalize to other classification algorithms. In fact it is likely the case that the interdependence of overlap and imbalance is even stronger in algorithms which are more sensitive to these factors.

2 Detection of Interdependence

In this section we outline a method to test the hypothesis that overlap and imbalance have independent effects on classifier performance. Let us take μ as a measure of overlap between the classes and α as a measure of the between class imbalance[1]. If these two factors act independently we would expect the performance surface with respect to μ and α to follow the relation

$$\mathrm{d}P(\mu,\alpha) = f'(\mu)\,\mathrm{d}\mu + g'(\alpha)\,\mathrm{d}\alpha, \tag{1}$$

where f' and g' are unknown functions. That is, we would expect the total derivative of performance to be separable into the components contributed by each of μ and α. This hypothesis of independence leads us to expect that we can consider the partial derivatives independently, i.e.

$$\frac{\partial}{\partial\mu}P = f'(\mu), \tag{2}$$

$$\frac{\partial}{\partial\alpha}P = g'(\alpha). \tag{3}$$

[1] We provide a concrete method for assigning values to μ and α in Sect. 3, but for the moment we leave the details of this assignment intentionally vague.

The functions f' and g' may not have simple or obvious functional forms meaning that we cannot compute f' and g' directly; however, if f' and g' were known we could find a predicted value for $P(\alpha, \mu)$, up to an additive constant, by evaluating

$$P(\mu, \alpha) = \int f'(\mu)\,\mathrm{d}\mu + \int g'(\alpha)\,\mathrm{d}\alpha + C. \tag{4}$$

Specific values for $P(\mu, \alpha)$ can be computed numerically by training a classifier on a data set with the appropriate level of overlap and imbalance. This requires the use of synthetic data sets since there is no general method to measure the level of class overlap in real data. The use of synthetic data allows us to ensure that other confounding factors, such as problem complexity, remain constant throughout our tests.

Since we expect the partial derivatives of $P(\mu, \alpha)$ to be independent we can compute values for f' by evaluating $P(\mu, \alpha)$ for several values of μ while holding α constant and taking a numerical derivative. Values for g' can be computed in a similar manner by holding μ constant and varying α. These values can then be combined into predicted values for $P(\mu, \alpha)$ using (4). Comparing the predicted values for $P(\mu, \alpha)$ to the observed values will allow us to determine if our hypothesis of independence is sound.

The above method only estimates $P(\mu, \alpha)$ up to the additive constant C. To obtain a value for C we simply need to compute the value of $P(\mu, \alpha)$ for a single point where we have computed both f' and g'.

3 Experiment

In this section we present an experiment designed to test the hypothesis of independence using the procedure outlined in Sect. 2. The data sets we generate for this experiment are a collection of "backbone" models in two dimensions. To generate a data set we sample points from the region $[0, 1] \times [0, 1]$. The range along the first dimension is divided into four regions with alternating class membership (two regions for each class) while the two classes are indistinguishable in the second dimension. To change the overlap level of the classes we allow adjacent regions to overlap. The overlap level is parametrized such that when $\mu = 0$ the two classes are completely separable and when $\mu = 1$ both classes are distributed uniformly over the entire domain. Changing the imbalance level is done by sampling more data points from one class than the other. The imbalance level is parameterized such that α is the proportion of the total data set belonging to the majority class. The total number of samples is kept fixed as α varies. Some example data sets are shown in Fig. 1. These domains are simple enough to be readily visualized yet the optimal decision surface is sufficiently non-linear to cause interesting effects to emerge.

We measure classifier performance over three collections of data sets generated in the manner described above.

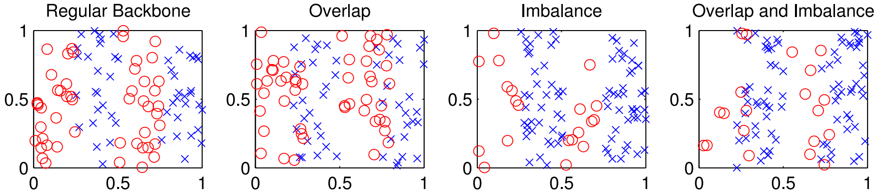


Fig. 1. Sample Data Sets

1. **Varying Overlap** — For these data sets we fix $\mu = 0$ value and α varies over the range $[0.5, 0.95]$.
2. **Varying Imbalance** — For these data sets we vary μ over the range $[0, 1]$ with $\alpha = 0.5$ fixed.
3. **Varying Both** — For these data sets we vary μ and α simultaneously over the ranges $[0, 1]$ and $[0.5, 0.95]$ respectively.

Evaluating our classifier on the first two collections gives us enough information to evaluate (4). Comparing this with the results generated by testing on the third collection of data sets will allow us to determine if overlap and imbalance have independent effects on classifier accuracy.

For each level of imbalance and overlap we measure the classifier performance using several different training set sizes. To build a training set we first select the overlap and imbalance levels as well as the size of the training set and then sample the selected number of points according to the generative distribution defined by the chosen overlap and imbalance. All of our tests are repeated for several training sets sizes varying from 25 to 6400 samples. Testing is done by generating new samples from the same distribution used for training.

We assess classifier performance using the F_1-score of the classifier trained on each data set where the minority class is taken as the positive class. The F_1-score is the harmonic mean of the precision and recall of a classifier and is a commonly used scalar measurement of performance. Our choice of positive class reflects the state of affairs present in many real world problems where it is difficult to obtain samples from the class of interest. The F_1-score is one of the family of F_β-scores which treats precision and recall as equally important.

4 Results

We trained several SVM classifiers on the data sets described in Sect. 3. For each level of overlap, imbalance and training set size we built several classifiers in order to track the variance as well as the overall performance. Parameter values for the SVMs were chosen by selecting a few data sets from our domain of interest and running simulated annealing following the method described in [11] to select the optimal parameters. The optimal parameter values from these tests showed very little variation so we selected a constellation of representative values and left them unchanged for all of our tests. We used the SVM Radial Basis Function kernel for all our tests since it is the most popular non-linear SVM kernel used in practice.

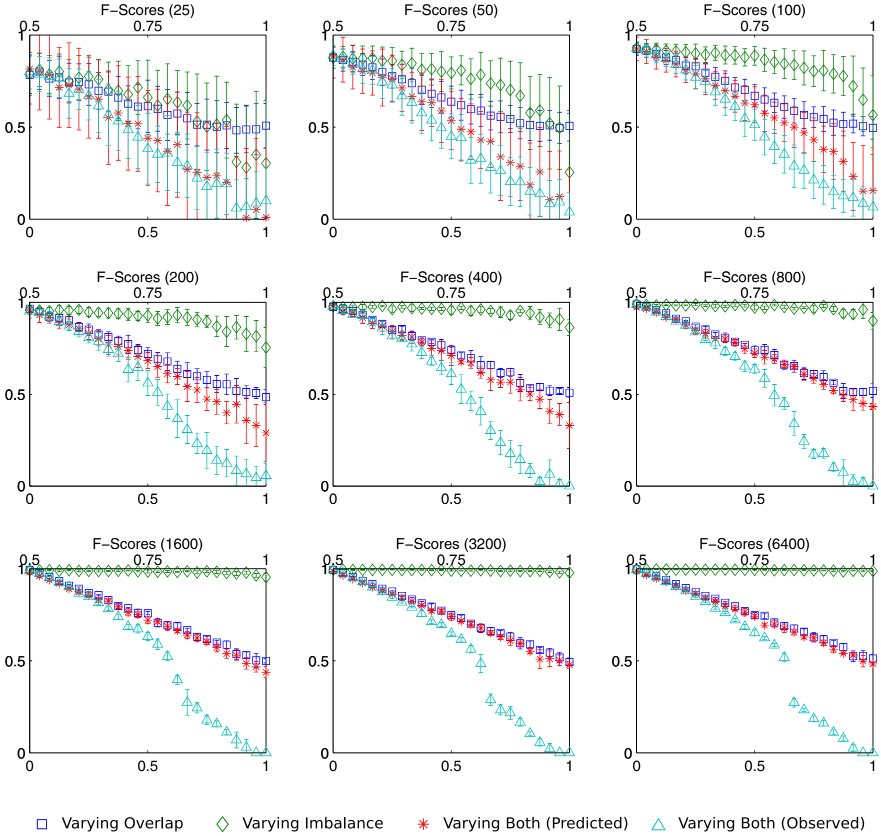


Fig. 2. F_1-scores of several SVM classifiers at different data set sizes. The lower horizontal axis shows the level of overlap and the upper horizontal axis shows the level of imbalance. The vertical axis shows the corresponding F_1-score. Error bars show one standard deviation around the mean.

The performance results from our experiments are shown in Fig. 2. These results clearly show that when the training set size is large the performance predicted by assuming that overlap and imbalance are independent is very different than what is observed. On the other hand, when the training set is small our model is quite accurate, showing only a minor deviation from the observed results.

When the training set size is reasonably large we observe that the class imbalance has very little effect on the classifier performance. This result agrees with previous investigations [6] which suggested that SVMs are not sensitive to class imbalance. When the imbalance level is very high, or when there are few training examples, we still see a drop in performance. This is what we would expect from the existence of small disjuncts in these domains [7,9].

In addition to the F_1-scores we also recorded the number of support vectors from each run. These data are recorded in Fig. 3. Our model of independence does not make predictions about the number of support vectors so these results cannot be used to test our hypothesis; however, the number of the support vectors can be used as a measure of model complexity.

These results also support the idea that SVMs are not significantly effected by class imbalance when there is sufficient training data. When only imbalance is present we observe that a very small proportion of the total training data is retained as support vectors. This indicates that the SVM has found a highly parsimonious model for the data and, since the corresponding F_1-scores in Fig. 2 are high, we see that these models generalize well. Conversely, when there is class overlap in the training data the number of support vectors rises quickly. This indicates that the SVM has difficulty finding a parsimonious solution despite the fact that there is no increase in complexity of the optimal decision surface. It is interesting to notice that, provided the training set is sufficiently large, the proportion of the training set retained as support vectors shows very little variation across differently sized training sets. A constant proportion of support vectors corresponds to a massive increase in the complexity of the learned model as the training set size is increased.

5 Analysis

5.1 Is Independence Ever a Good Model?

We mentioned previously that for small training set sizes, as well as for small levels of overlap and imbalance, the performance predicted by our model of independence appears to give good predictions for the observed accuracy. Conversely, for high levels of combined overlap and imbalance the predictions given by our model appear to be very poor. In this section we provide a more systematic assessment of the quality of the independence model to determine when, if ever, it might be reasonable to treat these effects independently.

To assess the quality of our model's predictions we preform a two tailed t-test to determine if our predictions differ significantly from the observed results. For these tests we take as our null hypothesis the assumption that overlap and imbalance influence classifier performance independently and compute when it is possible to reject this hypothesis with >99% confidence. Results from these tests are shown in Fig. 4.

For the smallest training set size we see no strong evidence to reject our hypothesis of independence; however, when there is sufficient training data we see that it is highly unlikely that our hypothesis of independence is correct. It is interesting to note even for very large training sets there is a region of the parameter space where we cannot confidently reject our hypothesis of independence; however, with a large training set this region is quite small and outside it the evidence against independence is quite strong. We also note that the size of this region decreases as the training set size is increased. This is notable since the performance degradation from overlap alone is decreased by using more training data. If the two factors were independent we would expect the combined performance to be very close to the performance in the presence of overlap alone

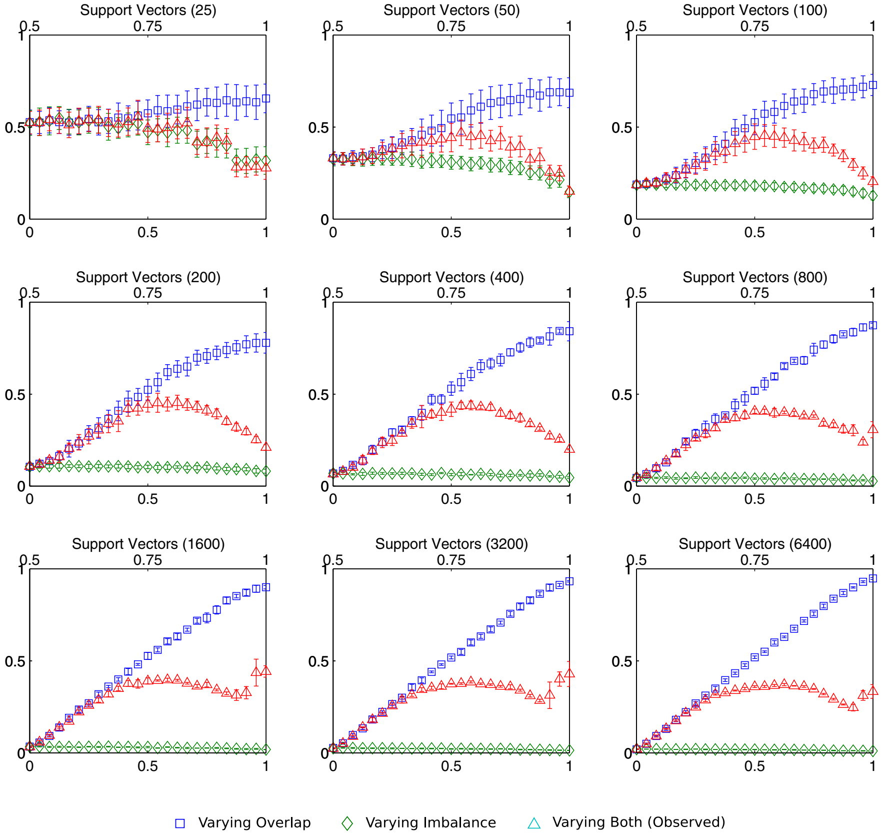


Fig. 3. Proportion of the training set retained as support vectors by several SVM classifiers at different training set sizes. The lower horizontal axis shows the level of overlap and the upper horizontal axis shows the level of imbalance. The vertical axis shows the corresponding proportion of the training set retained as support vectors. Error bars show one standard deviation around the mean.

since with large training sets the degradation from imbalance alone is negligible; however, this is not the case. In light of these observations it is reasonable to conclude that the hypothesis is false in general and we speculate that our inability to reject the model in all cases is merely a case of lack of data.

These results confirm that overlap and imbalance do not have independent effects on performance. We can also see from Fig. 2 that the combined contribution from both factors is made stronger as the training set size increases. Although the contribution from imbalance alone is negligible it cannot be ignored since its presence combined with overlap causes additional degradation beyond the level caused by overlap alone.

From this analysis we see that if independence is ever a good model it can only be when the training set size is very small; however, as we have already

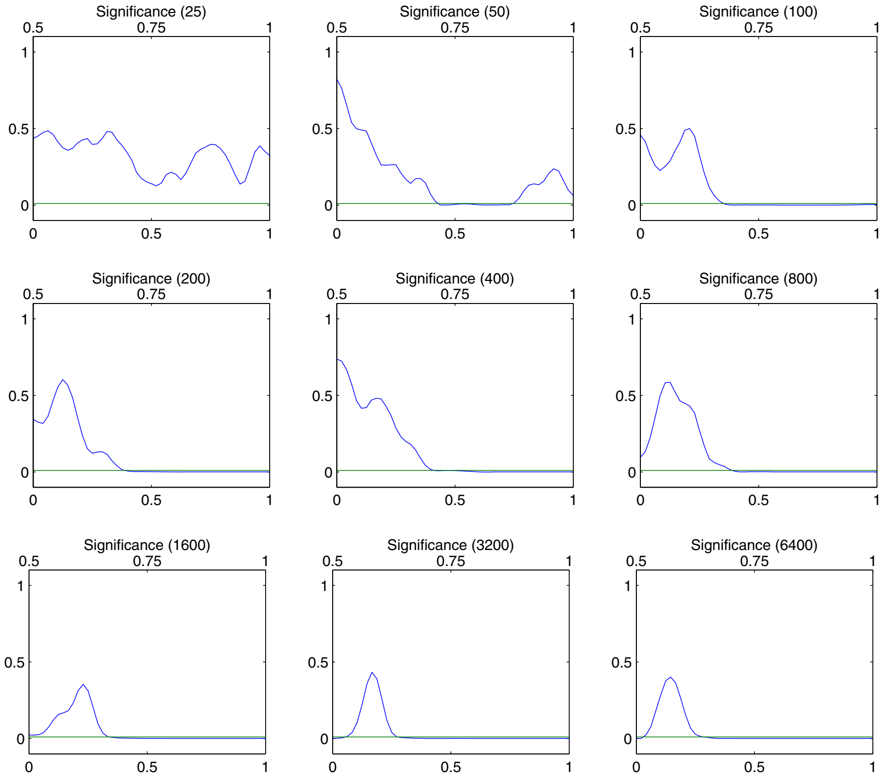


Fig. 4. The p-values for our significance tests. Small p-values indicate a statistically significant deviation between the observed and predicted results for combined overlap and imbalance. The lower horizontal axis shows the level of overlap and the upper horizontal axis shows the level of imbalance. The vertical axis shows the p-value for the associated hypothesis test; also shown is the 99% confidence threshold. These data have been smoothed for readability.

seen, this situation causes other problems and should generally be avoided. We have also not directly shown that the model is good under these conditions, only that we cannot confidently say it is poor.

5.2 Which Is Worse?

We have shown that overlap and imbalance are not independent factors, but the question still remains: Which factor has a more profound effect on the classifier on its own? To answer this we refer again to Fig. 2 which shows classifier performance with overlap and no imbalance as well as with imbalance and no overlap.

When the training set is small, high levels of imbalance cause a dramatic drop in classifier performance; however, with the use of larger training sets this effect disappears almost entirely. When the training set is large, even an imbalance level of 95% has a barely noticeable effect on performance. This observation is consistent with previous work which showed that problems typically associated with imbalanced data can be better explained by the presence of small disjuncts. As the size of the training set grows the number of training data in each cluster is increased for both the minority and the majority classes. Once there are sufficiently many points in each of the minority class clusters the SVM has no trouble identifying them despite even very high levels of imbalance.

Referring to Fig. 3 we see that more imbalanced training sets actually produce less complex models; i.e. the proportion of the training set retained as support vectors actually drops as the imbalance level is increased. The drop is quite small, however, and the overall proportion of support vectors retained from just imbalance is dwarfed by the proportion retained in the overlapping or combined cases. It is likely that this drop is an artifact of there simply being fewer data available along the margin in the minority class rather than a meaningful reduction in complexity.

Contrasting the above to the effects from overlap, we see from Fig. 2 that overlapping classes cause a consistent drop in performance regardless of the size of the training set. The drop in performance from overlap is linear and performance drops from nearly perfect to $\sim$0.5 as the overlap level is increased. It should be noted that this is exactly what we would expect to happen, even with a perfect classifier. When the classes are overlapping and not imbalanced there are ambiguous regions in the data space where even an optimal classifier with prior knowledge of the generative distributions would not be able to predict the class labels better than chance. The SVM performance in the presence of overlap alone follows exactly the profile we would expect from an optimal classifier in these cases.

It is far more interesting to examine the model complexity in terms of overlap as shown in Fig. 3. Despite the fact that the complexity of the optimal solution remains constant throughout all of our tests, as overlap increases the number of training data retained by the model increases dramatically. This means that although the SVM is able to find a solution which performs comparably to the optimal classifier, the solution it finds becomes progressively more complex as the level of overlap increases.

5.3 Correlating Performance and Model Complexity

We have seen that for sufficiently large training sets there is a sharp drop in performance beyond a certain level of combined overlap and imbalance and that this effect is only seen when both factors are present simultaneously. We also saw that when the two factors are combined the number of support vectors in the resulting model reaches its maximum at an intermediate level of imbalance and overlap. In this section we illustrate the connection between these two observations.

The peak in the number of support vectors (and hence model complexity) is highly correlated with the sharp drop in performance we see with sufficiently

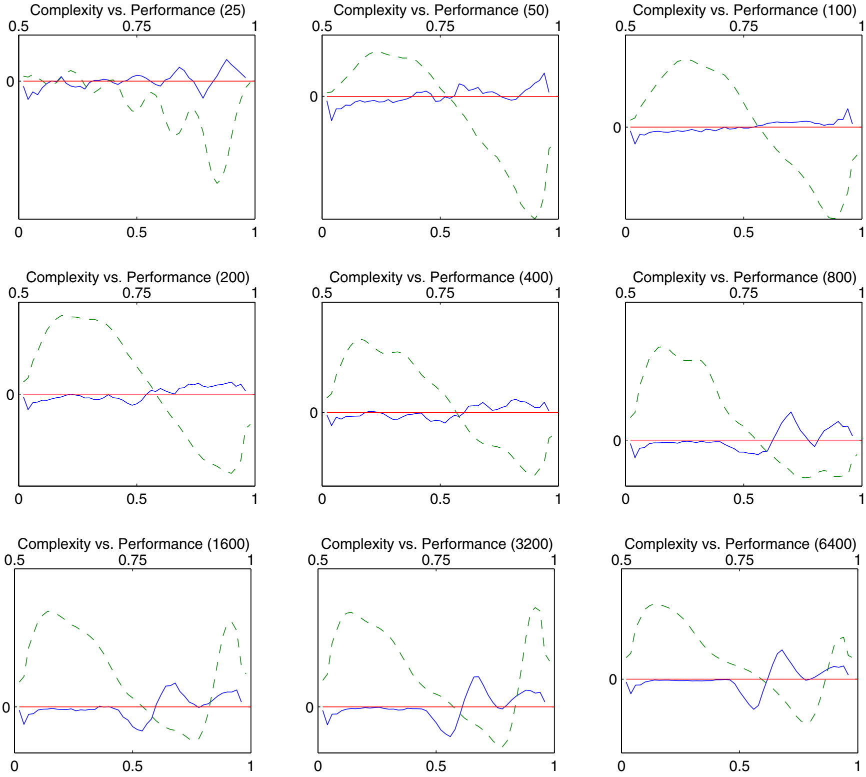


Fig. 5. Comparison between model complexity and performance. The lower horizontal axis shows the level of overlap and the upper horizontal axis shows the level of imbalance. The solid line shows the second derivative of the F_1-score and the dashed line shows the first derivative of the number of support vectors. These data have been smoothed for readability.

large training sets. This correlation is illustrated in Fig. 5 by showing the second derivative of the combined F_1-score and the first derivative of the number of support vectors. The data shown in Fig. 5 has been scaled vertically and smoothed in order to make the plots readable; the important feature to note is where the two lines cross the x-axis. We see that for all but the smallest training set sizes both plots cross the x-axis at approximately $\mu = 0.6$ and $\alpha = 0.78$. The points where the support vector and performance curves cross the x-axis correspond to the peak model complexity and the inflection point in performance respectively. If the overlap and imbalance are increased beyond this point the performance of the trained classifier drops rapidly. Since this effect does not occur when only one of the two factors are present it is clearly an artifact of the combined contribution.

Interestingly, the location of this crossing varies very little as the number of training examples is increased. When there is sufficient training data for the effect to emerge it is consistently present and its location is relatively unchanged by varying the size of the training set. This suggests that we are observing a type of breaking point—a point where we transition from being able to extract a (somewhat) meaningful representation from the data to a regime where the data representation is not sufficient to build an effective classifier. This observation is supported by a reexamination of Fig. 2 where we can see that performance before the drop is higher in cases where we have used large training sets, but that the performance after the drop is consistently poor regardless of the size of the training set.

6 Conclusion

We have shown that classifier performance varies with overlap and imbalance in a manner that necessitates an interrelationship between these two factors. Comparing the observed performance in cases of combined overlap and imbalance to the performance levels predicted by a model of independence shows that when the two factors are combined the classifier performance is degraded significantly beyond what the model predicts.

Our analysis is consistent with previous results which show that the imbalance problem is properly understood as a problem of small disjuncts in the minority class. When sufficiently many training data are available imbalanced distributions do not impede classification even when the imbalance level is very high. Despite this the imbalance problem cannot be considered solved since levels of imbalance which, in isolation, cause no significant degradation of performance can have a large impact on performance when overlapping classes are also present.

Our analysis of the overlap problem shows that, in isolation, it is a much more serious issue than imbalance. Although the SVM is able to achieve performance comparable to the optimal classifier in the presence of overlap the model complexity tells a different story. Despite the fact that the complexity of the optimal solution remains constant, the complexity of the SVM solution grows proportional to the overlap level and the training set size. This result is important since it shows that more training data—which is often regarded as a panacea for poor performance—can have a detrimental effect on the quality of the learned model.

We have also shown that SVMs have a breaking point where, if the overlap and imbalanced levels are too high we cannot achieve good performance regardless of amount of available training data. We have shown that this breaking point is strongly correlated with the peak model complexity. This effect is notable for several reasons. First, it only appears when both overlap and imbalance are present in tandem, which demonstrates directly that there are effects that we miss by examining overlap and imbalance separately. Second, the insensitivity of this effect to the training set size indicates that it is the result of a systematic weakness of the SVM classifier in the presence of overlap and imbalance rather than a problem with the data. Finally, this finding suggests an avenue for further research into the interaction between overlap and imbalance.

References

1. García, V., Sánchez, J.S., Mollineda, R.A.: An empirical study of the behavior of classifiers on imbalanced and overlapped data sets. In: Rueda, L., Mery, D., Kittler, J. (eds.) CIARP 2007. LNCS, vol. 4756, pp. 397–406. Springer, Heidelberg (2007)
2. Akbani, R., Kwek, S., Japkowicz, N.: Applying support vector machines to imbalanced datasets. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 39–50. Springer, Heidelberg (2004)
3. Batista, G.E.A.P.A., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explorations 6(1), 20–29 (2004)
4. Bosch, A.V.D., Weijters, T., Herik, H.J.V.D., Daelemans, W.: When small disjuncts abound, try lazy learning: A case study. In: Seventh Benelern Conference, pp. 109–118 (1997)
5. Yaohua, T., Jinghuai, G.: Improved classification for problem involving overlapping patterns. IEICE Transactions on Information and Systems 90(111), 1787–1795 (2007)
6. Japkowicz, N., Stephen, S.: The class imbalance problem: A systematic study. Intelligent Data Analysis 6, 429–449 (2002)
7. Japkowicz, N.: Class imbalances: are we focusing on the right issue. In: Workshop on Learning from Imbalanced Data Sets II, pp. 17–23 (2003)
8. Jo, T., Japkowicz, N.: Class imbalances versus small disjuncts. ACM SIGKDD Explorations Newsletter 6(1), 40–49 (2004)
9. Prati, R.C., Batista, G.E.A.P.A., Monard, M.C.: Class imbalances versus class overlapping: An analysis of a learning system behavior. LNCS, pp. 312–321. Springer, Heidelberg (2004)
10. Visa, S., Ralescu, A.: Learning imbalanced and overlapping classes using fuzzy sets. In: ICML 2003 Workshop on Learning from Imbalanced Data Sets II, vol. 3 (2003)
11. Boardman, M., Trappenberg, T.: A Heuristic for Free Parameter Optimization with Support Vector Machines. In: International Joint Conference on Neural Networks, IJCNN 2006, pp. 610–617 (2006)

Robustness of Classifiers to Changing Environments

Houman Abbasian[1], Chris Drummond[2],
Nathalie Japkowicz[1], and Stan Matwin[1,3]

[1] School of Information Technology and Engineering,
University of Ottawa
Ottawa, Ontario, Canada, K1N 6N5
habba057@uottawa.ca, nat@site.uottawa.ca, stan@site.uottawa.ca
[2] Institute for Information Technology,
National Research Council of Canada,
Ottawa, Ontario, Canada, K1A 0R6
Chris.Drummond@nrc-cnrc.gc.ca
[3] Institute of Computer Science,
Polish Academy of Sciences,
Warsaw, Poland

Abstract. In this paper, we test some of the most commonly used classifiers to identify which ones are the most robust to changing environments. The environment may change over time due to some contextual or definitional changes. The environment may change with location. It would be surprising if the performance of common classifiers did not degrade with these changes. The question, we address here, is whether or not some types of classifier are inherently more immune than others to these effects. In this study, we simulate the changing of environment by reducing the influence on the class of the most significant attributes. Based on our analysis, K-Nearest Neighbor and Artificial Neural Networks are the most robust learners, ensemble algorithms are somewhat robust, whereas Naive Bayes, Logistic Regression and particularly Decision Trees are the most affected.

Keywords: Classifier evaluation, changing environments, classifier robustness.

1 Introduction

In this paper, we test some of the most popular, and commonly used, classifiers to identify which are the most robust to changing environments. The environment is the state of the world when the data set is first collected. Essentially, the data set is a single snapshot of the world, capturing its state. This state may change over time, as time passes from first deployment of the classifier in the field. This state may even change in the time from when the classifier is trained to when it is deployed. This state may change with situation: different locations of the same application will have appreciable differences. It would be surprising if

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 232–243, 2010.

the performance of a classifier did not degrade with these changes. The question we address here is whether or not some classifiers are inherently more immune than others to such changes. Although there has been considerable work on the effect of changing the distribution of the class attribute on classifier performance [8,9,5,4], only more recently has research looked at the effect of changes in the distribution of other attributes [1,7]. This is an area of growing importance, as evidenced by a recent book on the topic [11].

For the purposes of discussion, let us divide the remaining attributes into three types: conditions, outcome and context. This division captures the causal structure of many problems. Conditions cause the class. This is particularly apparent in medicine where they are called risk factors. A change in risk factor causes a change in the frequency of the disease. For example, the population, in most western countries, is aging and putting on weight yet smoking less and eating more carefully. To what extent will a classifier, say predicting heart problems, be impervious to such changes? The class causes the outcomes. In medicine these would be the symptoms. We might expect, however, that the symptoms of a disease would remain constant. Nevertheless, we could see the concept change over time, i.e. concept drift [14], e.g. as our understanding of the implications of the disease is clarified. Context causes changes in other attributes but not the class [13,12]. For example, the increase in the prevalence of other diseases in the population may mask the symptoms of a heart attack, and therefore degrade any classifier's performance.

We discuss here the changes in the influence of a particular attribute on the class, which can be measured by information gain. We think this is most likely to occur with changes in conditions, although changes in context and outcome may also have an effect. A good example of changes in risk factors, and with them changes in the influence of attributes, is type-2 diabetes. This disease, originally called adult onset diabetes, has lately turned up in alarming numbers in children. Age might have originally been a strongly predictive attribute but it is no longer. Smoking and lung cancer are strongly related, but as more people quit smoking the influence of other factors will begin to dominate. Simply put, the predictive power of attributes will change with time.

This research is related to previous work by one of the authors that also addressed the robustness of classifiers [2]. However, we would contend that the issue of the robustness of classifiers is a significant and growing area and many aspects warrant investigation. This paper differs in the type of change investigated – the previous work changed the distribution of the data in the test set while in this work we change the influence of the most significant attributes – and the simulation of change based on real data sets rather than an artificial one. More importantly, it differs in that it focuses on algorithms that learn more complex representations. We surmise that algorithms that produce classifiers reliant on a large number of attributes should be more robust to such changes than the ones reliant on a small number. This is, in general, the conclusion we draw from our experiments although there are subtleties which we discuss later in the paper. Based on our analysis, K-Nearest Neighbor and Artificial Neural Networks are the most robust learners, ensemble algorithms are somewhat robust, whereas Logistic Regression, Naive Bayes and particularly Decision Trees fare badly.

The remainder of this paper is organized as follows. Section 2 describes our testing methodology and Section 3 the experimental results. Section 4 is a discussion of why some learners are more robust than others and suggestions for future work.

2 Testing Methodology

Our experiments are designed to answer two questions: (a) Which learners are the most robust to changing environments? (b) Does the reliance on a large number of attributes make a classifier more robust?

To empirically answer the above questions, we selected a good variety of data sets. We used 6 data sets from the UCI repository [3]: Adult, Letter, Nursery, Connect-4, Breast Cancer and Vote. The number of classes ranges from 2 to 26; the number of instances from 569 to 67557[1]; the number of attributes from 8 to 42 and the types are both continuous and categorical. All algorithms come from the Weka data mining and machine learning tool [16]. The classifiers produced range from those that are reliant on a small number of attributes such as the decision tree algorithm J48, through those of ensemble algorithms that rely on a larger number – Random Forest, Bagging, AdaBoost – to ones that use all attributes such as Naive Bayes, Logistic Regression, Artificial Neural Networks, K-Nearest Neighbor. In this study, the five most significant attributes are determined using the “Gain Ratio Attribute Evaluator” of Weka. For all of the experiments, we changed the influence of the attributes by changing its information gain ratio, as discussed in the next section. We train each algorithm using the original data and test the classifier on data where the influence of different attributes on the class is progressively decreased.

2.1 Changing an Attribute’s Influence

One measure for the influence of an attribute on the class is information gain, defined as the expected reduction in entropy caused by partitioning the examples according to that attribute. To decrease the influence of an attribute, we reduce the information gain by adding noise to that attribute. Equation 1 shows the information gain of attribute A relative to a collection of examples with class labels C.

$$Gain(C,A) = Entropy(C) - \sum_{v \in Values(A)} \frac{|C_v|}{|C|} Entropy(C_v) \tag{1}$$

$$Entropy(C_v) = -\sum_{i=1}^{c} p_i log_2 p_i$$

[1] For the experiments, we use a sub-sample of 10% for all data sets except the smallest.

$Values(A)$ is the set of all possible values for attribute A; C_v is the subset of C for which attribute A has value v; p_i is the proportion of C_v belonging to class i. In equation 1, adding noise to attributes does not change C but will change C_v. The noise randomly changes attribute values of selected instances. So, the proportion p_i of each class associated with a particular value will move closer to the proportion of classes in the whole data set. Therefore, $Entropy(C_v)$ moves towards $Entropy(C)$ resulting in a lower $Gain(C, A)$.

In this paper, we used a slightly modified version of information gain called *gain ratio*[2]. This measure includes a term called *split information* which is sensitive to how broadly the attribute splits the data and defined by equation 2. By adding noise to attributes, the $|C_i|$ will move closer to each other. Therefore *SplitInfo* in equation 2 will become larger and as a result the gain ratio will be lower.

$$GainRatio(C, A) = \frac{Gain(C,A)}{SplitInfo(C,A)} \tag{2}$$

$$SplitInfo(C, A) = -\sum_{i=1}^{c} \frac{|C_i|}{|C|} log_2 \frac{|C_i|}{|C|}$$

In the following, rather than record the actual gain ratio, we record change level. This is the percentage of test instances whose attribute values have been changed. Thus 100% means all values have been altered. If the attribute is nominal, the existing value is replaced by one randomly selected from the remaining values of that attribute. If the attribute is numeric, it is replaced by a randomly generated number drawn from the uniform distribution, spanning the maximal and minimal values of that attribute.

3 Experimental Results

In this section, we first determine which attributes have the largest effect on performance. We then compare classifiers to see how they fare as the influence of the attributes decreased.

3.1 The Influence of Each Attribute on Accuracy

Some attributes will have a strong influence on the class; others will have a much weaker influence. To determine the strength, we found the five most influential attributes, based on the information gain ratio, for each data set. We then used 10-fold cross validation to calculate the performance of each learner, averaged over all change levels. Table 1 shows the different degrees of influence that the attributes have on the class. The $*$ in this table indicates that, using a paired t-test with an alpha of 0.05, the accuracy of the learner for corrupted data is significantly different from that for the original data.

[2] The use of information gain as a way of selecting significant attributes has a long history in machine learning. So does this modified version, which is used to select attributes in C4.5 [10].

Table 1. Accuracy with decreasing the gain ratio

Data set	Attribute	J48	RF	NB	BG	ADB	KNN	LR	ANN
Adult	11	53.22*	70.19*	49.2*	74.4*	73.44*	80.35	57.09*	80.25
	12	83.3	82.52	68.7	82.6	83.22	80.75	81.83	80.47
	6	82.04	82.32	81.4	82.7	80.62	80.18	82.1	80.37
	8	83.17	81.54	81.5	78.3	83.22	80.52	83.05	80.73
	10	83.87	82.63	81.6	82.7	83.22	80.7	83.43	81.13
	Original Data	82.64	82.48	82	82.3	83.37	80.7	83.5	80.87
Letter	14	49.05*	63.69*	49.2*	55.5*	6.749	60.52*	51.35*	21.21*
	13	61.88*	73.7*	54.1*	70*	6.749	66.71*	61.84*	24.27*
	11	49.56*	75.81*	52.7*	67.9*	6.749	68.94*	58.4*	25.2*
	7	63.63*	73.03*	54.7*	67*	6.749	70.66*	65.98*	26.01*
	12	59.95*	72.42*	50.8*	63.5*	6.749	64.15*	60.08*	27.1*
	Original Data	70.94	80.46	62	76.4	6.916	75.34	74.82	29.48
Nursery	8	63.38*	65.31*	64.4*	63.8*	46.68*	63.06*	64.85*	65.92*
	2	79.86*	79.98*	77.1*	78.7*	65.89	79.23*	80.3*	80.35*
	1	84.33*	85.06*	84.7*	84*	65.89	82.46*	84.92*	85.5*
	7	88.41	90.15	87.9	88.5	65.89	86.67	89.18	89.92
	5	88.9	90.26	88.1	88	65.89	85.68	89.99	90.32
	Original Data	90.02	91.66	88.6	88.5	65.51	87.89	91.66	91.72
Connect-4	36	79.8	80.63	58.8*	81.4	72.79	78.13	58.34	83.61
	35	79.8	80.62	57.9*	81.4	72.79	78.13	64.47	83.42
	21	70.59*	76.3*	65.1*	72.6*	69.1*	76.26	54.12*	76.86*
	18	79.8	80.63	59.8*	81.4	72.79	78.12	65.24	83.57
	41	79.8	80.64	58.1*	81.4	72.79	78.13	64.89	83.49
	Original Data	80.02	80.33	71.7	81.8	73.25	78.05	65.11	82.02
Breast Cancer	5	65.71*	68.61	72.3	69.8	73.93	72.95	71.13	74.74
	4	73.24	70.06	71.8	69.2	73.07	73.94	71.04	74.72
	6	70.95	68.24	73.6	69.2	69.41	72.66	70.96	72.79
	9	74	70.11	73.7	70.2	74.78	73.82	72.02	75.57
	3	74.07	67.77	73.7	72.1	74.98	74.8	71.52	75.74
	Original Data	76.44	68.6	74.6	70.4	75.91	75.87	71.42	75.28
Vote	4	67.91*	72.12*	85.7*	67.7*	73.47*	89.91	78.33*	74.87*
	3	96.07	95.35	90.5	96.7	96.26	91.04	88.54	94.44
	5	96.07	95.51	89.7	96.7	96.7	92.03	83.36	95.13
	12	95.94	96.45	90.1	96.5	96.33	92.06	91.71	95.25
	14	96.1	95.64	89.7	96.7	96.6	91.59	90.85	95.72
	Original Data	96.11	95.17	89.6	96.4	96.11	92.15	92.84	94.69

For the Adult, Vote and Connect-4 data sets, only one attribute has a substantial influence. For others, more attributes are influential: three attributes for Nursery, and all five for the Letter data set. For the Breast Cancer data set, only for the Decision Tree is there a statistically significant difference and then only for the first attribute. However, as the percentage of positive samples is 30%, the accuracy of all algorithms on the original data is very close to 70%, that of the default classifier. Thus corrupting the attributes has little impact. We will drop this data set from further consideration.

For the Letter data set, AdaBoost uses a Decision Stump as its base classifier. The performance of a Decision Stump, on this data set, is close to that of default classifier, so corrupting the attributes has little effect on its accuracy. The performance of the Neural network is also low, yet higher than that of AdaBoost and the change level does affect its performance significantly. The surprising point for connect-4 data set is that the first and the second most influential attributes do not have an impact on the accuracy that is statistically significant for any learner. However, the third most influential attribute does. Exactly why this occurs will need further investigation.

3.2 Ranking the Algorithms

For each learner and each data set, we use the attributes that significantly affect the accuracy of the learner. First, we trained each learner using the original data set. We then tested it on data where the influence of the attributes has been decreased. We decreased the influence of each attribute using different change levels (20%, 40%, 60%, 80%, and 100%). The performance of each learner was averaged over all attributes for each change level. It should be noted that in Table 2 smaller is better, i.e. there is less change in performance. The best value in each row is in bold type. The average value of the best performing learner is underlined if it is significantly better than the value from the second best learner, using a paired t-test with a significance level of 0.05.

Let us go through the results of Table 2 by data set:

Adult: The performances on clean and corrupted data are essentially indistinguishable for Nearest Neighbor and the Neural Network. The small negative numbers are likely caused by random errors. So, we conclude, on average across all change levels, the Neural Network and Nearest Neighbor are the most robust learners.

Letter: AdaBoost appears initially to be the most robust but, as explained earlier, the accuracy of AdaBoost on the original data set is very low and changing attributes has little effect. Excluding AdaBoost, Neural Network is the most robust at all levels. On average across all change levels, the Neural Network is the most robust learner and Random Forest is second.

Nursery: Bagging is the most robust at lower change levels, while Nearest Neighbor is the best at higher levels. In addition, on average Nearest Neighbor is the most robust method but, using the t-test, it is not significantly different from the second best learner, Bagging.

Connect-4: Nearest Neighbor is the best learning algorithm at all change levels. On average across all levels, Nearest Neighbor is the best learner and AdaBoost the second best.

Vote: Nearest Neighbor is the best model at all change levels. On average across all levels, Naive Bayes is the second best.

Not surprisingly perhaps, a t-test for a significant difference between the best and the second best classifier does not give us all the information we require. The results are further validated for statistical significance using a one way analysis

Table 2. The impact of change level on the difference in performance

Data set	Change Level	J48	RF	NB	BG	ADB	KNN	LR	ANN
Adult	20%	9.25	3.83	11.29	2.42	3.56	**-0.35**	8.04	**-0.86**
	40%	18.94	8.12	22.35	5.42	7.30	**-0.55**	17.88	**-0.36**
	60%	29.25	12.90	31.43	7.61	10.04	**-0.25**	26.11	**-0.16**
	80%	38.47	16.20	44.35	11.00	12.39	**0.22**	34.83	**-0.31**
	100%	51.20	20.41	54.38	13.08	16.38	**-0.46**	44.69	**-0.20**
	Ave	29.42	12.29	32.76	7.91	9.94	**-0.28**	26.31	**-0.38**
Letter	20%	5.10	2.64	3.33	4.68	0.17	2.85	4.53	**2.63**
	40%	9.57	5.95	6.63	8.43	0.17	5.97	9.53	**3.64**
	60%	14.12	8.67	9.18	11.59	0.17	9.36	15.47	**4.54**
	80%	18.66	11.74	13.02	15.32	0.17	12.54	21.18	**6.10**
	100%	23.18	14.65	16.34	18.29	0.17	15.02	25.72	**6.69**
	Ave	14.12	8.73	9.70	11.66	0.17	9.15	15.29	**4.72**
Nursery	20%	4.97	4.59	3.80	**3.12**	5.52	3.78	4.85	4.38
	40%	9.69	9.53	9.16	**8.14**	13.07	8.49	9.34	9.27
	60%	14.23	15.23	13.27	**12.67**	20.09	13.08	15.55	14.18
	80%	18.93	19.60	17.90	18.23	24.57	**17.86**	20.33	19.45
	100%	23.01	25.41	22.12	22.74	30.87	**21.65**	24.79	25.00
	Ave	14.16	14.87	13.25	12.98	18.82	**12.97**	14.98	14.46
Connect-4	20%	4.05	0.78	5.08	3.96	1.54	**0.78**	8.23	0.98
	40%	7.09	2.69	8.61	6.50	2.63	**1.65**	10.00	2.98
	60%	8.47	3.48	11.98	8.93	3.85	**1.79**	10.54	5.03
	80%	12.17	5.89	14.89	13.66	6.06	**2.54**	13.26	7.79
	100%	15.35	7.36	18.35	12.84	6.65	**2.20**	12.89	9.06
	Ave	9.43	4.03	11.78	9.18	4.15	**1.79**	10.99	5.17
Vote	20%	7.98	7.55	0.67	8.29	6.12	**0.33**	6.59	8.29
	40%	16.67	11.89	1.61	19.83	16.97	**1.09**	9.58	11.49
	60%	32.15	28.82	5.05	29.72	18.61	**1.12**	15.16	19.80
	80%	40.65	30.18	3.97	35.74	32.15	**3.52**	17.42	27.78
	100%	43.52	36.81	5.66	50.16	39.38	**5.14**	23.83	31.74
	Ave	28.20	23.05	3.39	28.75	22.64	**2.24**	14.52	19.82

of variance followed by what is termed a post hoc analysis [15]. First, the learners are tested to see if the average difference in performance, on the original and changed data, of the 8 learners are equal, across the five change levels and the most significant attributes respectively. This is the null hypothesis in ANOVA; the alternative hypothesis is that at least one learner is different. The results of

Table 3. ANOVA with their corresponding F-Value and P-Value

Adult		Letter		Nursery		Connect-4		Vote	
F-Value	P-Value	F-Value	P-Value	F-Value	P-Value	F-Value	P-Value	F-Value	P-Value
72.3	<0.0001	32.46	<0.0001	2.6	0.012	11.42	<0.0001	25.86	<0.0001

the ANOVA are given in Table 3 and allow us to reject the null hypothesis, but it does not tell us how to rank the classifiers.

To achieve this we use the post hoc analysis. We apply the Fisher's Least Significant Difference (LSD) test, with an individual error rate of 0.05. Table 4 provides the average values of each metric for each learner, as well as their significant ranks, the columns labeled **Ave** and **R** respectively in this table. Note that if two or more instances have the same letter, then their performances are not significantly different. Table 4 shows the overall impact of changing the influence of all significant attributes over all change levels. For the Adult data set, the Neural Network and Nearest Neighbor are the most robust learners; the ensemble learners Bagging, AdaBoost and Random Forest are next. For the Letter data set, although AdaBoost is the most robust learner the original accuracy of AdaBoost is close to the default classifier. Excluding AdaBoost from this data set, the Neural Network is the most robust and Random Forest, Nearest Neighbor, Naive Bayes, Bagging are the next. Nearest Neighbor, Random Forests, AdaBoost and the Neural Network are the most robust model, for the Connect-4 data set. Next are the Decision Tree, Bagging, Logistic Regression and Naive Bayes. For the Nursery data set, the post hoc test is not able to differentiate among the learners very well. The mean difference value of the learners is close to one another and the variance is high. In this data set all learners except AdaBoost are placed in the first level. For the Vote data set, the best learners are Nearest Neighbor and Naive Bayes, all other learners are in the next group.

From table 4, due to the high variance among learners in each data set, the post hoc tests does not differentiate among the learners very well. For the Adult data set, the post hoc test does not differentiate among the learners of the third group indicated by letter C. For the Letter data set, it also does not differentiate among the third group of the learners. This is repeated across the individual data sets.

To improve differentiation of the robustness of algorithms, in the last column of Table 4, we give the average values for each learner, and significant ranks, for all data sets combined. Here, Nearest Neighbor is the clear winner with the Neural Network in second place. Learners reliant on a smaller number of attributes such

Table 4. Overall impact of decreasing attribute influence

Adult			Letter			Connect-4			Nursery			Vote			All data sets		
Alg	Ave	R	Alg	Ave	R	Alg	Ave	R	Alg	Ave	R	Alg	Ave	R	Alg	Ave	R
ANN	-0.38	A	ADB	0.17	A	KNN	1.79	A	KNN	12.97	A	KNN	2.24	A	KNN	5.18	A
KNN	-0.28	A	ANN	4.72	B	RF	4.03	A	BG	12.98	A	NB	3.39	A	ANN	8.76	A
BG	7.91	B	RF	8.73	C	ADB	4.15	A	NB	13.25	A	LR	14.52	B	ADB	11.14	B
ADB	9.94	B	KNN	9.15	C	ANN	5.17	A	J48	14.16	A	ANN	19.82	B	RF	12.59	B
RF	12.29	B	NB	9.70	C	J48	9.43	B	ANN	14.46	A	ADB	22.64	B	BG	14.1	B
LR	26.31	C	BG	11.66	C	BG	9.18	B	RF	14.87	A	RF	23.05	B	NB	14.18	B
J48	29.42	C	J48	14.12	D	LR	10.99	B	LR	14.98	A	J48	28.22	B	LR	16.42	C
NB	32.76	C	LR	15.29	D	NB	11.98	B	ADB	18.82	B	BG	28.75	B	J48	19.07	D

as Random Forests, AdaBoost and Bagging are next. AdaBoost is the best but, as it did poorly in terms of accuracy on a couple of original data sets, this robustness comes at some price. The Decision Tree comes firmly in last place. In general, our experimental results show that learners reliant on more attributes, tend to be more robust to the changes of the influence of some of them.

4 Discussion and Future Work

In this section, we discuss some hypotheses for why some learners are more robust to changing the influence of attributes than others. These need verification and will be the subject of future work. We will also discuss other directions for future research.

For Artificial Neural Networks robustness to attribute changes is dependent on a decision surface using all attributes of the data simultaneously. Thus, the changing of one attribute of an instance is unlikely to cause that instance to be misclassified, unless the weight of that attribute is very much larger than that of others. Likewise, for Nearest Neighbor all attributes are used to find the distance between two instances. If the influence of one attribute decreases, other attributes can still be predictive. For decision trees such as J48, where the complexity of the tree depends on the data set we use, often only a few attributes define the decision boundary. Thus, if the influence of one of these attributes on one particular instance on the class is changed, then that instance may well cross the decision boundary.

For example, Figure 1(a) is the decision tree and Figure 1(b) is the data and decision boundary formed by the tree (solid lines) and that formed by the Neural Network (dashed line). Note that x is the most significant attribute for decision tree. Now suppose that by changing the influence of this attribute, the instance at the top of Figure 1 moves in the x direction crossing the decision boundary for

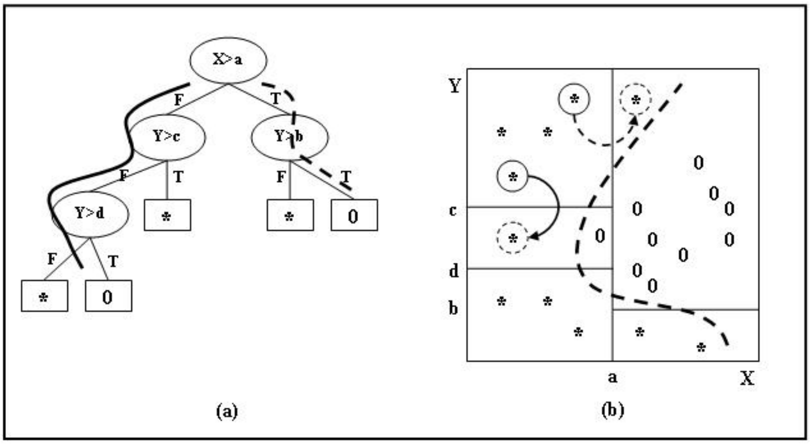


Fig. 1. The impact of noise on decision tree and Artificial neural network

the tree. That instance will now travel down the right branch of the decision tree as shown by the dashed line in Figure 1(a), classifying this instance incorrectly. This instance does not cross the decision boundary for the Neural Network. This is because this boundary uses both attributes and is much smoother. The same reasoning can be applied to the instance that crosses the y=c boundary in Figure 1(b). If by changing y, the second most influential attribute, the instance crosses the boundary y=c then the left solid branch on the tree classifies this instance as 0 but the Neural Network correctly classifies it.

Ensemble algorithms, as they consist of multiple classifiers, tend to rely on many, though not all, attributes. Where they differ, from each other, is primarily in the way the individual classifiers are constructed. We expect that Random Forests, by deliberately selecting classifiers based on different attributes, would be the most robust. That is supported by our experimental results in that it does well on a couple of data sets. It is, however, beaten overall by AdaBoost, although the poor performance of AdaBoost on some original data sets accounts for much of this. Further experiments are needed to determine whether or not there is any real difference between ensemble algorithms in terms of robustness. If there is, we aim to find out what is at the root these differences. What is clear, however, is that they are generally better than the base classifier they use.

We had concerns that the statistical tests, which showed AdaBoost to be very robust, might be misleading due to its poor performance on some of the original data sets. So, we claim that for an algorithm to be considered robust it must not only have a small difference in performance, but also the performance on the original data set must be good, or at least competitive with other algorithms. As future work, it would be worth exploring if a single metric might capture these two concerns. An alternative would be to plot points showing how they trade-off on a two-dimensional graph, similar to ROC curve or cost curves [9,5]. Another concern with the experiments is that the measure we used, information gain ratio, is also used by the decision tree algorithm to chose the attributes at each branch. So, it may have a larger impact on the performance of decision trees than on that of other algorithms. We will therefore explore the effect of using different measures for selecting the most significant attributes in the future experiments. This can be quite easily realized within Weka as it has many attribute evaluators: Info Gain, Relief Attribute Evaluator, Principal Components, and OneR.

As other future work, the experiments will look at a greater range of algorithms. It would be worth exploring if there are other general characteristics, apart from the number of attributes used, that affect robustness. Another simple extension of current research is to include more data sets. We will also use the entire number of instances of the present data sets instead of just a 10% fraction of them. Some of the data sets we chose only had one significant attribute. This property has been noticed before, simple classifiers often do well on UCI data sets [6]. It would be worth doing experiments on data sets with a wide spread in the number of significant attributes, to see how this affects robustness. We will also experiment with changing the influence of a combination of attributes instead of one attribute at a time and with changing attributes in other ways.

We believe this paper has given some insight into what makes a classifier robust to changing environments. Nevertheless we have not explained by any means all the factors. We need to determine why Naive Bayes and Logistic Regression, which use all attributes, are not robust. Further experiments will be needed to expose the other differences between classifiers.

5 Conclusions

The objective of this study was to investigate the robustness of a variety of commonly used learning algorithms to changing environments. The Neural Network and Nearest Neighbor are the most robust, learners reliant on smaller number of attributes such as Random Forest, AdaBoost and Bagging are located in the second place and finally the Decision Tree is in last place. In general, we conclude that learners reliant on more attributes tend to be more robust. This is clearly not the whole story, however, as Naive Bayes and Logistic Regression are not robust, future work will investigate this issue further.

Acknowledgments

We would like to thank the Natural Sciences and Engineering Research Council of Canada for the financial support that made this research possible.

References

1. Alaiz-Rodríguez, R., Guerrero-Curieses, A., Cid-Sueiro, J.: Minimax regret classifier for imprecise class distributions. Journal of Machine Learning Research 8, 103–130 (2007)
2. Alaiz-Rodríguez, R., Japkowicz, N.: Assessing the impact of changing environments on classifier performance. In: Proceedings of the 21st Canadian Conference in Artificial Intelligence, pp. 13–24 (2008)
3. Blake, C., Merz, C.: UCI repository of machine learning databases. Univ. of California, Irvine (1998), http://www.ics.uci.edu/~mlearn/MLRepository.html
4. Drummond, C.: Discriminative vs. generative classifiers for cost sensitive learning. In: Lamontagne, L., Marchand, M. (eds.) Canadian AI 2006. LNCS (LNAI), vol. 4013, pp. 481–492. Springer, Heidelberg (2006)
5. Drummond, C., Holte, R.C.: Cost curves: An improved method for visualizing classifier performance. Machine Learning 65(1), 95–130 (2006)
6. Holte, R.C.: Very simple classification rules perform well on most commonly used datasets. Machine Learning 11(1), 63–91 (1993)
7. Huang, J., Smola, A.J., retton, A.G., Borgwardt, K.M., Schölkopf, B.: Correcting sample selection bias by unlabeled data. In: Schölkopf, B., Platt, J., Hoffman, T. (eds.) Advances in Neural Information Processing Systems 19, pp. 601–608. MIT Press, Cambridge (2007)
8. Provost, F., Fawcett, T.: Robust classification for imprecise environments. Machine Learning 42, 203–231 (2001)

9. Provost, F., Fawcett, T., Kohavi, R.: The case against accuracy estimation for comparing induction algorithms. In: Proceedings of the Fifteenth International Conference on Machine Learning, pp. 43–48. Morgan Kaufmann, San Francisco (1998)
10. Ross, J.: Quinlan. C4.5 Programs for Machine Learning. Morgan Kaufmann, San Mateo (1993)
11. Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N.D.: Dataset Shift in Machine Learning. The MIT Press, Cambridge (2009)
12. Turney, P.D.: Exploiting context when learning to classify. In: Brazdil, P.B. (ed.) ECML 1993. LNCS, vol. 667, pp. 402–407. Springer, Heidelberg (1993)
13. Turney, P.D.: The management of context-sensitive features: A review of strategies. In: Proceedings of the 13th International Conference on Machine Learning: Workshop on Learning in Context-Sensitive Domains, pp. 60–66 (1996)
14. Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden contexts. Machine Learning 23, 69–101 (1996)
15. Wikipedia, http://en.wikipedia.org/wiki/Post-hoc_analysis
16. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, San Francisco (2005)

Belief Revision of Product-Based Causal Possibilistic Networks

Salem Benferhat[1] and Karim Tabia[2]

[1] CRIL UMR CNRS 8188 - Université d'Artois
benferhat@cril.fr
[2] LINA/COD CNRS UMR 6241 - Ecole Polytechnique de Nantes
Karim.Tabia@univ-nantes.fr

Abstract. Belief revision is an important task for designing intelligent systems. In the possibility theory framework, considerable work has addressed revising beliefs in a possibilistic logic framework while only few works have addressed a possibilistic revision process in graphical-based frameworks. In particular, belief revision of causal product-based possibilistic networks which are the possibilistic counterparts of probabilistic causal networks has not yet been addressed. This paper is concerned with revising causal possibilistic networks in presence of two kinds of information: observations and interventions (which are external actions forcing some variables to some specific values). It contains two contributions: we first propose an efficient method for integrating and accepting new observations by directly transforming the initial graph. Then we highlight important issues related to belief revision of causal networks with sets of observations and interventions.

Keywords: Causal possibilistic networks, belief revision, interventions.

1 Introduction

Belief representation and revision are important tasks for designing intelligent systems in different areas of artificial intelligence. Belief revision is the process that consists in modifying a set of initial beliefs in order to integrate a new and sure piece of information called *input*. In this paper, we address belief revision of graphical representations of beliefs called possibilistic networks [4][1] which are the possibilistic counterparts of probabilistic networks [6][13]. Like probabilistic networks, possibilistic ones allow an easy and compact encoding of joint possibility distributions specifying agents' beliefs and offer interesting inference capabilities. In possibility theory, considerable works addressed belief revision in possibilistic logic [9][16][18] while only few works address revising possibilistic networks [3]. In possibilistic logic belief revision, the input is a fully certain propositional formula that refines our beliefs. When we deal with belief graphical representations, the input can be of two forms: a set of evidences (observations) which are results of *testing* some variables, and a set of interventions [14] which represent external actions that force some variables to have some specific values.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 244–255, 2010.

Observations concern static worlds while interventions concern dynamic worlds. Handling interventions is close to the concept of updating [15] while handling observations is more close to belief revision in [10]. Note that in the possibilistic logic framework there is to the best of our knowledge only one work [2] where the authors address handling interventions.

In this paper, we are interested in revising causal possibilistic networks in presence of observations and interventions. Causal graphical models are very useful tools for causal ascription. The beliefs encoded by a causal networks are revised with two kinds of information implying different treatments: *observations* and *interventions*. Observations denote sure pieces of information specifying the precise values taken by some domain variables. Belief revision is needed here to infer possible consequences or causes from these observations. However, interventions are external actions forcing some variables to specific values and belief revision is performed only to determine the consequences of such actions. The real problem is how to efficiently revise the beliefs encoded by a causal possibilistic network with a single observation/intervention or with a set of both observations and interventions. The main contribution of this paper is an efficient solution consisting in handling belief change induced by the observations and interventions graphically: only the beliefs of the needed variables are revised. Hence, instead of revising a joint possibility distribution to take into account the new information, we revise only some local distributions while guaranteeing the same results as if we revised the joint one. The paper also highlights some issues relating revising causal possibilistic networks with belief revision of possibilistic epistemic states and possibilistic knowledge bases.

The rest of this paper is organized as follows: Section 2 briefly presents the basic backgrounds on possibility theory and causal possibilistic networks. In section 3, we briefly present the related works on belief revision in the possibilistic logic frameworks. Section 4 presents our solution for revising possibilistic causal networks with observations. In section 5, we deal with revising possibilistic networks with sets of observations and interventions. Finally, section 6 concludes this work.

2 Basic Background on Possibility Theory and Causal Networks

This section provides the basic background on possibilistic graphical models. But let us first fix the notations that will be used in the rest of this paper.

X_1, X_2,..,X_n denote the set of domain variables and x_i denotes an instance (value) of variable X_i while D_{X_i} denotes the domain of variable X_i. U_i denotes the parents of variable X_i. X (in bold face) denotes a subset of variables and $D_X = \times_{X_i \in X} D_{X_i}$ represents the cartesian product of the variables X_i involved in the subset X. x denotes an instance of X (namely, $x \in D_X$).

2.1 Product-Based Possibility Theory

Possibility theory is an uncertainty theory [8] suitable for representing and reasoning with agents' beliefs. It constitutes a powerful and simple alternative to probability theory in particular for dealing with some types of uncertainty. The fundamental concept of possibility distribution π is a mapping from the universe of all possible states of the world Ω (universe of discourse) to the unit interval $[0,1]$. A possibility degree $\pi(w_i)$ expresses to what extent the state of the world w_i (also called *interpretation*, *state* or *model*) can be the actual state of affairs. By convention, $\pi(w_i)=1$ means that w_i is totally possible (unsurprising) while $\pi(w_i)=0$ denotes an impossible world. The statement $\pi(w_i)>\pi(w_j)$ means that the state w_i is more possible than w_j. The second fundamental notion in possibility theory is the one of possibility measure $\Pi(\phi)$ which assesses the possibility degree of an arbitrary event $\phi\subseteq\Omega$. $\Pi(\phi)$ is defined as follows:

$$\Pi(\phi)=\max_{w_i\in\phi}(\pi(w_i)) \tag{1}$$

The certainty entailed by the current knowledge of the world is assessed by the necessity measure, defined as follows:

$$N(\phi)=1-\Pi(\overline{\phi})=1-\max_{w_i\notin\phi}(\pi(w_i)), \tag{2}$$

where $\overline{\phi}$ denotes the complementary of ϕ (namely $\Omega-\phi$). The necessity degree of ϕ is inversely proportional to the possibility degree of events other than ϕ.

The other fundamental notion in possibility theory is the one of conditioning [11] which is concerned with updating the current beliefs encoded by the possibility distribution π having observed a completely sure event (commonly called *evidence* or *observation*). Note that there are several definitions of the possibilistic conditioning [11] and we focus in this paper only on the well-known product-based conditioning defined as follows:

$$\pi(w_i|\phi)=\begin{cases}\frac{\pi(w_i)}{\Pi(\phi)} & \text{if } w_i\in\phi;\\ 0 & \text{otherwise.}\end{cases} \tag{3}$$

In the product-based setting, the effect of conditioning is to exclude every interpretation w_i which does not satisfy the evidence ϕ while the the other interpretations are proportionally upgraded. Note that revising possibilistic beliefs (encoded for instance by a possibility distribution π) in the presence of an observation e (or evidence) is traditionally performed by conditioning the current beliefs by the evidence, namely the revised beliefs π' are such that $\forall\omega\in\Omega$, $\pi'(\omega)=\pi(\omega|e)$.

2.2 Possibilistic Causal Networks

A possibilistic network [4] [1] is specified by two components:

1. **A graphical component:** it consists in a directed acyclic graph (DAG) encoding the direct influence relationships between domain variables.
2. **A numerical component:** it involves a set of local conditional possibility distributions quantifying the influence endured by each domain variable A_i in the context of its parents U_{A_i}.

In order to guarantee that the joint possibility distribution encoded by a possibilistic network is normalized, all local possibility distribution relative to network nodes should satisfy the normalization condition expressed as follows:

$$\max_{x_i \in D_{X_i}} (\pi(x_i|u_i)) = 1. \tag{4}$$

A possibilistic network encodes a unique joint possibility distribution defined as follows:

$$\Pi(x_1, x_2, .., x_n) = \prod_{i=1}^{n} (\pi(x_i|u_i)). \tag{5}$$

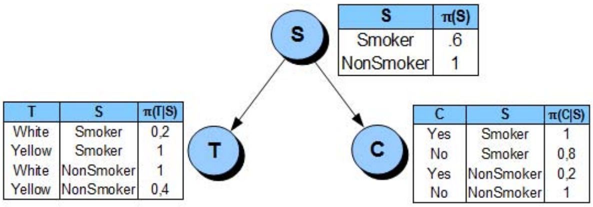
In a causal possibilistic network, the influence relationships existing between the domain variable encode causal relationships (cause-effect relationships). Hence, the parent set U_i of a node X_i in a causal graph represents all the direct causes of X_i while this latter stands for a direct consequence of U_i. In the rest of this paper, our contributions will be illustrated on the following example:

Example

In this academic and simplified example, we focus on the the lung cancer/smoking problem. We define the following variables:

- C for lung *cancer* and takes its values in the domain $D_C = \{Yes, No\}$.
- S for *smoking* and takes its values in the domain $D_S = \{Smoker, NonSmoker\}$.
- T for *teeth* color and takes its values in the domain $D_T = \{White, Yellow\}$.

The causal possibilistic network encoding an agent's beliefs is given in Figure 1.



S	π(S)
Smoker	.6
NonSmoker	1

T	S	π(T|S)
White	Smoker	0,2
Yellow	Smoker	1
White	NonSmoker	1
Yellow	NonSmoker	0,4

C	S	π(C|S)
Yes	Smoker	1
No	Smoker	0,8
Yes	NonSmoker	0,2
No	NonSmoker	1

Fig. 1. Causal possibilistic network G of the example

The example of Figure 1 gives the agent's beliefs about the lung cancer/smoking problem represented graphically. For instance, the prior beliefs state that the most common state for variable S is that the individual is *not smoker* while the statement *smoker* is less plausible. The statement teeth are *yellow* is the most common state in case where the individual is *smoker.* As for the lung cancer variable, the statement an individual *has lung cancer* decease is the most common state in case where this individual is *smoker.*

3 Belief Revision in Possibilistic Frameworks

There are several settings for representing and revising agents' beliefs in the possibilistic framework. For instance, an agent's beliefs can be specified in the form of a possibility distribution over the set of possible worlds Ω or by a possibilistic logic base. Beliefs are revised according to the semantics, role and priority of the input information with respect to the current beliefs. For example, the standard conditioning (see Equation 3) is used to change the current beliefs in the presence of a new evidence in the product-based possibilistic setting when beliefs are represented by a possibility distribution. In case where the input information is uncertain, one can use the possibilistic counterpart of Jeffrey's rule of conditioning [9] to update the current beliefs.

In this paper, belief revision is understood as the belief change process allowing to completely accept a new piece of information (in the sense that this input information will be completely preserved in the posterior beliefs). As we will see in the next section, handling observations and interventions which is the focus of this paper can be viewed as belief change operations constrained by the inputs. Indeed, observations and interventions constitute strong constraints on the belief change operations. Note that on one hand most works on belief change in the possibilistic framework addressed only revision of possibilistic knowledge bases and on the other hand a possibility distribution (hence a possibilistic network) can be encoded in the form of a possibilistic logic base and vice versa. Let us in the following focus on revising possibilistic logic bases with observations and interventions.

3.1 Belief Revision in Possibilistic Logic

Possibilistic logic [7] extends the standard logic formulas to a weighted logic for handling qualitative uncertainty and priorities. In this logic, each formula (a well formed proposition in a propositional language) is associated with a degree of certainty or priority (it often is the necessity measure that associates a degree of certainty α_i to each formula of the base p_i). Regarding interventions, the authors in [2] addressed handling interventions on possibilistic logic bases where they pointed out that simple interventions $do(x_i)$ forcing a propositional variable X_i to $True$ can be handled only if there already exists a world ω in the base satisfying $X_i{=}True$ and $\pi(\omega){=}1$. Otherwise, the intervention will create inconsistency in the possibilistic logic base under intervention. Moreover, the authors pointed out that handling interventions in possibilistic logic bases cannot fully catch the handling of interventions in causal graphical models where an intervention disconnects the variable under intervention from its direct causes to express the fact that the intervention is due to an external action.

Clearly, causal graphical models offer a natural framework for causal ascription as interventions are more appropriately handled. In the following, we present our contribution for revising graphical representations of possibilistic beliefs.

4 Revising Product-Based Causal Possibilistic Networks

In the following, we present our method for revising with observations and interventions.

4.1 Revising by Observations

An observation is the fact of having obtained the value of one or several domain variable at a given time. Belief revision with observations is traditionally done by simple conditioning. Namely, if π_G is the joint possibility distribution encoded by the possibilistic network G and $X_i{=}x_i$ is an observation, then the revised beliefs are $\pi'_G{=}\pi_G(.|X_i{=}x_i)$. Clearly, revising the beliefs encoded by a possibilistic network G by revising the joint possibility distribution π_G encoded by G is untractable. The solution we propose in this paper handles observations graphically (by transforming the initial network G). This solution proposes a graphical counterpart for the conditioning operation.

This graphical counterpart views conditioning (see Equation 3) as i) a *combination operation* followed by ii) a normalization operation. The combination operation combines the original possibility distribution with the one associated with the observation $X_i{=}x_i$ while the normalization operation re-normalizes the possibility distribution obtained after the combination step in case where this latter becomes sub-normalized.

Let G be the causal possibilistic network encoding the initial beliefs and π_G be the possibility distribution encoded by G (π_G is obtained form G using the chain rule of Equation 5). In order to perform the combination operation, let us define the possibility distribution encoding the observation $X_i{=}x_i$ as follows:

$$\forall \omega \in \Omega, \pi_{X_i=x_i}(\omega) = \begin{cases} 1 \; if \;\; w[X_i] = x_i \\ 0 \; otherwise \end{cases} \tag{6}$$

Clearly, in $\pi_{X_i=x_i}$ only the observed value x_i is totally possible while all the remaining ones are completely impossible which fully corresponds to the definition of an observation. Now, the combination of the initial beliefs with observation (namely, combining the possibility distribution π_G with $\pi_{X_i=x_i}$) can be defined as follows:

$$\forall \omega \in \Omega, \pi_{G2}(\omega) = \pi_G(\omega) * \pi_{X_i=x_i}(\omega). \tag{7}$$

In Equation 7, the possibility distribution π_{G2} is obtained from π_G by considering as completely impossible every interpretation ω where the value of X_i is different from x_i (namely, $\forall\omega{\in}\Omega$ $\pi_{G2}(\omega){=}0$ if $\omega[X_i]{\neq}x_i$), and preserving unchanged the possibility degrees of all interpretations ω where the value of X_i is x_i. After the combination step, the possibility distribution π_{G2} may be sub-normalized. Let us define the normalization operation as follows:

$$\forall \omega \in \Omega, \pi_{G3}(\omega) = \frac{\pi_{G2}(\omega)}{\max_{\omega\in\Omega} \pi_{G2}(\omega)}. \tag{8}$$

The normalization operation of Equation 8 upgrades the possibility degrees obtained after the combination operation such that the most plausible interpretation in π_{G2} becomes totally possible. Hence, using the combination (see

Equation 7) and normalization formulas (see Equation 8), the product-based conditioning given by Equation (3) can be redefined as follows:

$$\forall \omega \in \Omega, \pi_G(\omega | X_i = x_i) = \pi_{G3}(\omega). \tag{9}$$

In the following, we propose the graphical counterparts for the combination and normalization operations. For simplicity's sake, we deal in this paper with causal possibilistic networks with tree structures (a node can have at most one parent).

Graphical counterpart of the combination operation. Let us use $G2$ to denote the result of integrating the new observation $X_i{=}x_i^*$ in the network G, namely the network associated with the possibility distribution given by Equation 7. $G2$ is specified as follows:

Proposition 1. The possibilistic network $G2$ associated with the possibility distribution given by Equation 7 is obtained form network G as follows:

- the DAG of $G2$ is obtained from the one of G by deleting the arcs from every variable in U_i (the parents of X_i) to X_i.
- the local possibility distribution of any variable X_j in $G2$ different from X_i and U_i is identical to X_j's local distribution in G. The variables X_i and its parent denoted X_p (X_p is the only variable of U_i), the new local distributions are defined as follows:
  - $\forall x_i{\in}D_{X_i}$,
  $$\pi_{G2}(x_{ij}) = \begin{cases} 1 \; if \; x_i = x_i^* \\ 0 \; otherwise \end{cases}$$
  - Let X_q be the parent of X_p in G, then $\forall x_p{\in}D_{X_p}$, $\forall x_q{\in}D_{X_q}$
  $$\pi_{G2}(x_p|x_q) = \pi_2(x_p \mid x_q) * \pi_G(x_i \mid x_p)$$

Note that the new local possibility distribution relative to the variable under observation X_i ensures that only the instance x_i is fully possible and all the other instances are completely impossible. As for the possibility distribution relative to variable X_p (parent of X_i in G), it is altered in order to ensure that possibility degrees of every interpretation ω satisfying x_i are identical in π_G and π_{G2}. Now, since the value of the observed variable X_i is completely fixed, the arc from X_p (the parent of X_i) to X_i can be removed and guaranteeing that $\forall\omega{\in}\Omega$, $\pi_{G2}(\omega){=}\pi_G(\omega)^*\pi_{X_i=x_i}(\omega)$. Let us continue our example in order to illustrate this operation:

Example(continued)

We continue with the example of Figure 1 where network G encodes the initial beliefs. $G2$ of Figure 2 is the network obtained after combining G with the observation $T{=}Yellow$.

As shown on node S of network $G2$ of Figure 2, the new local distribution of the parent of the observed variable may be sub-normalized after the combination operation. We deal with this problem in the normalization step.

Graphical counterpart of the normalization operation. The combination step alters the local distribution of the parent of the observed one X_i in order to satisfy Equation 9. Let us use X_p to denote the parent of the observed variable X_i and X_q the parent of X_p if any. The possibility distribution of X_p may be sub-normalized after the combination step. Namely, it may exists an instance x_q of X_q such that $\max_{x_p \in D_{X_p}}(\pi_{G2}(x_p|x_q))=\beta$ with $\beta<1$. Hence, we need to compute a new possibilistic network $G3$ satisfying Equation 8.
and all of the local possibility distributions in $G3$ become normalized. The network $G3$ is obtained by progressively normalizing local distributions for each variable. There are two case to be considered:

1. **Case 1: The parent of the observed variable is a root variable**
   We study here the case where only the parent of the observed variable is root and its local possibility distribution is sub-normalized:

   Proposition 2. Let $G2$ be the network obtained from G after the combination step. Assume that only the root variable X_p is sub-normalized. Let $\max_{x_p \in D_{X_p}}(\pi_{G3}(x_p))=\beta$ with $0<\beta<1$. The re-normalized network $G3$ is such that:
   – The DAG of $G3$ is exactly identical to the one of $G2$,
   – $\forall X_j,\ X_j \neq X_p,\ \pi_{G3}(x_j|u_j)=\pi_{G2}(x_j|u_j)$,
   – $\forall x_p \in D_{X_p},\ \pi_{G3}(x_p)=\frac{\pi_{G2}(x_p)}{\beta}$.

   The transformation of Proposition 2 ensures that the local distribution relative to X_p is re-normalized while the joint possibility distributions encoded by the network $G3$ satisfies Equation 8. Let us illustrate the transformation of Proposition on our example.

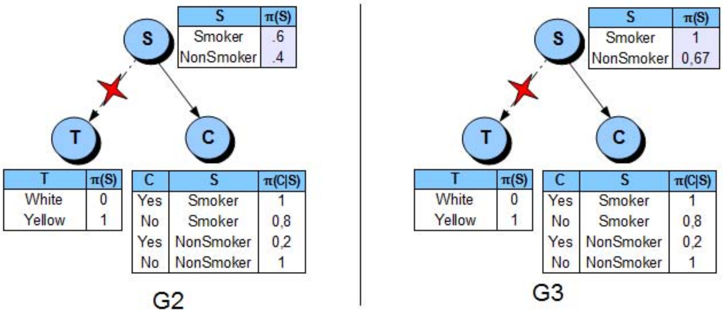
   Example(continued)

   In the network $G2$ (obtained after the combination of the initial network G with the observation $T=Yellow$) of Figure 2, the local possibility distribution relative to root node S is sub-normalized. The normalization of this distribution according to Proposition 2 gives the network $G3$ of Figure 2. One can easily check that the joint possibility distributions encoded by network $G3$ satisfies Equation 8.

   Let us now deal with the case where the sub-normalized distribution is relative to a variable X_p which is not a root.

2. **Case 2: The parent of the observed variable is not a root variable**
   In this case, the parent X_p of the observed variable X_i is not root and its possibility distribution has become sub-normalized after the combination step. Let us denote by X_q the parent of X_p. Hence, the possibility distribution of X_q must be adjusted when normalizing the distribution of X_p in order to keep unchanged the underlying joint possibility distribution. The normalization of a non root variable X_p is performed using Proposition 3 without changing the global possibility distribution:

   Proposition 3. Let $G2$ be the network obtained from the combination step. Let X_p denote the variable whose possibility distribution is sub-normalized. Let X_q be the parent variable of X_p and x_q^* be the value of X_q such that $\max_{x_p \in D_{X_p}}(\pi_{G2}(x_p \mid x_q))=\beta$ with $\beta<1$. Network $G3$ is such that it has exactly the same DAG as $G2$ and,



S	π(S)
Smoker	.6
NonSmoker	.4

T	π(S)
White	0
Yellow	1

C	S	π(C|S)
Yes	Smoker	1
No	Smoker	0,8
Yes	NonSmoker	0,2
No	NonSmoker	1

S	π(S)
Smoker	1
NonSmoker	0,67

T	π(S)
White	0
Yellow	1

C	S	π(C|S)
Yes	Smoker	1
No	Smoker	0,8
Yes	NonSmoker	0,2
No	NonSmoker	1

Fig. 2. Network $G2$ (resp. $G3$) obtained after the combination (resp. normalization) step

- $\forall X_j$, $X_j{\neq}X_p$ and $X_j{\neq}X_q$, $\pi_{G3}(x_j \mid u_{X_j}){=}\pi_{G2}(x_j \mid u_{X_j})$,
- $\forall x_p{\in}D_{X_p}$, $\forall x_q{\in}D_{X_q}$,

$$\pi_{G3}(x_p|x_q) = \begin{cases} \frac{\pi_{G2}(x_p|x_q)}{\beta} & if x_q = x_q^* \\ \pi_{G2}(x_p|x_q) & otherwise \end{cases}$$

- $\forall x_q{\in}D_{X_q}$, $\forall u_{X_q}{\in}D_{U_{X_q}}$,

$$\pi_{G3}(x_q|u_{x_p}) = \begin{cases} \pi_{G2}(x_q|u_{x_q}) * \beta \; if x_q = x_q^* \\ \pi_{G2}(x_q|u_{x_q}) \quad\quad otherwise \end{cases}$$

Proposition 3 ensures that the networks $G2$ and $G3$ encode the same joint possibility distribution. Now, if after the re-normalization of X_p, its parent X_q can become in turn sub-normalized. The normalization process should then be repeated until reaching a root variable. Once a root is reached, it is enough to re-normalize according to Proposition 2 to obtain a possibilistic network where all the local possibility distributions are normalized.

4.2 Revising by Interventions

There are mainly two equivalent methods for handling interventions in causal probabilistic graphical models: *graph mutilation* [19] and *graph augmentation* [17]. In [3], the authors proposed possibilistic counterparts for the mutilation and augmentation methods. An intervention on a variable X_i denoted $do(x_i)$ ensures that our beliefs on U_i should remain unchanged. This is achieved in the mutilation method by deleting all the arcs from each variable composing U_i to X_i and preserving the rest of the graph unmodified [14]. The obtained graph is called the *mutilated graph* and denoted G_m such that $\pi(\omega|do(x_i)){=}\pi_{G_m}(\omega|x_i)$, where π_{G_m} is the possibility distribution associated with the mutilated graph G_m. The effect of the intervention $do(x_i)$ on the rest of the initial graph G is determined by simple conditioning on the mutilated graph G_m after having observed the event $X_i{=}x_i$. Hence, the effect of this intervention on the joint possibility distribution is given by $\forall\omega$, $\pi(\omega){=}\pi_G(\omega)$ is represented by the new joint possibility distribution $\forall\omega$, $\pi(\omega|do(x_i)){=}\pi_{G_m}(\omega|do(x_i))$ (for more details, see [3]).

Note that the mutilation method ensures the same results as if the joint possibility distribution were revised using a possibilistic counterpart of Jeffrey's rule to encode the intervention. Jeffrey's rule [12] is an extension of the standard probabilistic conditioning to the case where the evidence is uncertain [5]. This revision rule imposes two strong constraints corresponding respectively to the way the uncertain evidence is specified (the uncertainty is of the form $(\lambda_i,\ \alpha_i)$ meaning that after the revision operation, the revised possibility degree of each event λ_i must be equal to α_i, the set $\{\lambda_1, .., \lambda_n\}$ induces a partition of Ω) and the way the prior beliefs are revised (minimize belief change according to kinematics principles). Similarly to conditioning, interventions can also be viewed as a belief change process that transforms an a priori possibility distribution π encoding the initial beliefs into a new one π' constrained by the intervention. Assume that an intervention $do(x_i)$ forces the variable X_i to take the value x_i. Assuming that the event $X_i{=}x_i$ is somewhat plausible in π (namely, $\Pi(x_i){\neq}0$) and letting U_i be the set of direct parents of X_i in the causal graph G encoding π and D_{U_i} be the domain associated with U_i. Handling such an intervention in the new possibility distribution π' should satisfy that i) the event $X_i{=}x_i$ is a sure piece of information (consequently, any interpretation $w{\in}\Omega$ where X_i is different from x_i is considered as completely impossible) and ii) the beliefs on the direct causes of X_i remain unchanged. Namely, $\forall X_j{\in}U_j$, $\forall x_j{\in}D_{X_j}$ $\Pi(x_j|do(x_i)){=}\Pi(x_j)$.

Clearly, interventions can be naturally encoded using the possibilistic counterpart of Jeffrey's rule. Indeed, by specifying the inputs as follows:

$\mu{=}\{((x_i, u_{x_i}), \pi(x_i, u_{x_i}))\colon x_i{\in}D_{X_i}$ and $u_{x_i}{\in}D_{U_i}\}{\cup}\{(x_j, 0)\colon x_j{\in}D_{X_i}$ and $x_j{\neq}x_i\}$.

5 Revising with Sets of Observations and Interventions

Using the graph mutilation method (or equivalently the graph augmentation) for handling interventions and the graphical counterpart of conditioning we presented in Section 4.1, one can efficiently revise a causal possibilistic network with any set of interventions, observations or both. This method guarantees the same results as if the joint distribution is revised using conditioning (for handling observations) and Jeffrey's rule (for handling interventions).

5.1 Revising with Sets of Observations

When revising with sets of observations, it is well-known that the order in which observations are treated does not matter. For instance, if we revise the beliefs encoded by the network G first with the observation $C{=}True$ then with another observation $T{=}Yellow$, then we will obtain exactly the same a posteriori beliefs as if we first revise G with the observation $T{=}Yellow$ then with $C{=}True$. It is important to note that revising with an observation $X_i{=}x_i$ followed by another observation excluding x_i will not result in a contradiction. However, revising with an observation that excludes x_i followed by the observation $X_i{=}x_i$ will lead to a contradiction due to the well-known problem of conditioning on an impossible event.

5.2 Revising with Sets of Interventions

Revising with sets of interventions also does not take into account the order in which interventions are performed. Namely, given an initial causal possibilistic network G, revising G with the intervention $do(x_i)$ then with a second intervention $do(x_j)$ is always equivalent in terms of obtained revised beliefs to revising G with $do(x_j)$ then with $do(x_i)$.

5.3 Revising with Sets of Observations and Interventions

Contrary to handling sets of only observations (resp. interventions) where the order of observations (resp. interventions) does not matter, the situation is different when revising by a sets involving both observations and interventions. Note that most works dealing with handling interventions [14][3] do not explicitly take into account the order in which the observations and interventions are reported. Indeed, revising a possibilistic networks by an observation followed by an observation does not give the same results as if the intervention were reported before the observation. For instance, in the example of Figure 1, if we first observe that the teeth are *yellow*, then we will change our prior beliefs on the variable S (the statement *smoker* will become more plausible). After this observation, if the teeth are further yellowed by an intervention then we will not change our beliefs about the statement *smoker* as the most plausible state for S. Consider now another scenario where first the teeth are yellowed. After this intervention, we observe without surprise that the teeth are *yellow*. However, we will not change our prior beliefs regarding the fact that the statement *nonsmoker* is the most common state of affairs for the smoking variable S (the statement *nonsmoker* is the most plausible state of the smoking variable S according to the network Figure 1). Clearly, these intuitive scenarios show that belief revision should take into account the order to arrival of observations and interventions.

6 Conclusions

This paper dealt with an important issue consisting in belief revision of causal possibilistic networks. More precisely, we dealt with revising product-based causal possibilistic networks by observations and interventions. We proposed an efficient method for handling sets of observations and interventions which views standard conditioning traditionally used for revising beliefs when new evidence is available as a belief change process consisting in a belief combination operation followed by a normalization operation. In this paper, we proposed graphical counterparts for the combination and normalization operations allowing to handle observations in polynomial time. We also pointed out some important issues related to handling sets of observations and interventions. Future works will address belief revision of qualitative causal possibilistic networks which have significant differences with product-based and probabilistic causal graphs.

References

1. Ben-Amor, N., Benferhat, S., Mellouli, K.: A two-steps algorithm for min-based possibilistic causal networks. In: Benferhat, S., Besnard, P. (eds.) ECSQARU 2001. LNCS (LNAI), vol. 2143, pp. 266–277. Springer, Heidelberg (2001)
2. Benferhat, S., Dubois, D., Prade, H.: Interventions in possibilistic logic. In: Godo, L., Pugliese, A. (eds.) SUM 2009. LNCS, vol. 5785, pp. 40–54. Springer, Heidelberg (2009)
3. Benferhat, S., Smaoui, S.: Possibilistic causal networks for handling interventions: A new propagation algorithm. In: The 22nd AAAI Conference on Artificial Intelligence, pp. 373–378 (2007)
4. Borgelt, C., Kruse, R.: Graphical Models: Methods for Data Analysis and Mining. John Wiley and Sons, Inc., USA (2002)
5. Chan, H., Darwiche, A.: On the revision of probabilistic beliefs using uncertain evidence. Artificial Intelligence 163(1), 67–90 (2005)
6. Darwiche, A.: Modeling and Reasoning With Bayesian Networks. Cambridge University Press ELT, New York (April 2009)
7. Dubois, D., Lang, J., Prade, H.: Possibilistic logic, pp. 439–513 (1994)
8. Dubois, D., Prade, H.: Possibility theory. Plenum Press, New-York (1988)
9. Dubois, D., Prade, H.: A synthetic view of belief revision with uncertain inputs in the framework of possibility theory. Int. J. of Approximate Reasoning 17(2-3), 295–324 (1997)
10. Gärdenfors, P., Makinson, D.: Revisions of knowledge systems using epistemic entrenchment. In: Proceedings of the 2nd Conference on Theoretical Aspects of Reasoning about Knowledge, pp. 83–95. Morgan Kaufmann Publishers Inc., San Francisco (1988)
11. Hisdal, E.: Conditional possibilities independence and non interaction. Fuzzy Sets and Systems, 283–297 (1978)
12. Jeffrey, R.C.: The Logic of Decision. McGraw Hill, NY (1965)
13. Jensen, F.V., Nielsen, T.D.: Bayesian Networks and Decision Graphs (Information Science and Statistics). Springer, Heidelberg (June 2007)
14. Judea, P.: Causality: Models, Reasoning, and Inference. Cambridge University Press, Cambridge (March 2000)
15. Katsuno, H., Mendelzon, A.O.: Propositional knowledge base revision and minimal change. Artif. Intell. 52(3), 263–294 (1991)
16. Ma, J., Liu, W.: A general model for epistemic state revision using plausibility measures. In: Proceeding of the 2008 conference on ECAI 2008, pp. 356–360 (2008)
17. Pearl, J.: [bayesian analysis in expert systems]: Comment: Graphical models, causality and intervention. Statistical Science 8(3), 266–269 (1993)
18. Prade, H., Benferhat, S., Dubois, D., Williams, M.-A.: A general framework for revising belief bases using qualitative jeffrey's rule. In: Ninth International Symposium on Logical Formalizations of Commonsense Reasoning, Toronto (2009)
19. Verma, T., Pearl, J.: Equivalence and synthesis of causal models. In: UAI 1990: Proc. of 6th Annual Conf. on Uncertainty in Artificial Intelligence, NY, USA, pp. 255–270. Elsevier Science Inc., Amsterdam (1991)

A Survey on Statistical Relational Learning

Hassan Khosravi and Bahareh Bina

School of Computing Science, Simon Fraser University,
Burnaby, B.C., Canada V5A 1S6
{hkhosrav,bba18}@cs.sfu.ca

Abstract. The vast majority of work in Machine Learning has focused on propositional data which is assumed to be identically and independently distributed, however, many real world datasets are relational and most real world applications are characterized by the presence of uncertainty and complex relational structure where the data distribution is neither identical nor independent. An emerging research area, Statistical Relational Learning(SRL), attempts to represent, model, and learn in relational domain. Currently, SRL is still at a primitive stage in Canada, which motivates us to conduct this survey as an attempt to raise more attention to this field. Our survey presents a brief introduction to SRL and a comparison with conventional learning approaches. In this survey we review four SRL models(PRMs, MLNs, RDNs, and BLPs) and compare them theoretically with respect to their representation, structure learning, parameter learning, and inference methods. We conclude with a discussion on limitations of current methods.

1 Introduction

The vast majority of work in learning has focused on propositional data which consists of identically structured entities that are assumed to be independent. However, many real world datasets are relational. Relational data consists of different types of entities where each entity is characterized with a different set of attributes. Relational data are more complex and better suited with our surroundings where examples are given as multiple related tables. The structure of relational data provides an opportunity for objects to carry additional information via their links and enables the model to show correlations among objects and their relationships.

Statistical Relational Learning(SRL) is a new branch of machine learning that tries to model a joint distribution over relational data[9]. SRL is a combination of statistical learning which addresses uncertainty in data and relational learning which deals with complex relational structures. A statistical relational model for a given database shows not only the correlations between attributes of each table, but also dependencies among attributes of different tables. Statistical relational models are usually represented with graphical models [22] and are different in methods of representation, learning , and inference.

SRL models have been extensively researched and many applications have been proposed for them[13, 25, 23, 27], but they have lacked popularity among researchers in Canada. Due to this, we try to motivate SRL and their importance in this survey. In Section 2 we cover some background information required for studying this paper and introduce a running example to define some of the new concepts. We emphasize on some

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 256–268, 2010.

of the main differences of relational data and propositional data which rationalizes the need of new methods for learning on relational data in Section 3. Section 4 shows how propositional and relational learners use graphical models for their representation. We review four of the proposed models in Section 5; the methods of representing, learning, and inference are discussed. Studying current state-of-the-art methods provides a realistic view of the current capabilities for SRL.

2 Background and Notation

Entity-Relationship model. We assume that tables in the relational schema can be divided into *entity tables* and *relationship tables.* This is the case whenever a relational schema is derived from an entity-relationship model (ER model) [24, Ch.2.2]. Symbol E refers to entity tables or objects. Symbol R refers to relationship tables or links. Symbol T refers to generic tables.

To better explain the concepts and demonstrate the notations, we define a running example of a university database, which contains three objects or entity tables: $Student$, $Course$, and $Professor$, and two relationship tables: $Registered$, with foreign key pointers to the $Student$ and $Course$ tables, whose tuples indicate students have registered in which courses, and RA, with foreign key pointers to the $Student$ and $Professor$ tables, whose tuples indicate the RAship of students for professors. Table 1 shows the relational schema of university database. Relationships refer to their related objects using *reference slots*. Each table in the relational database is seen as a class that has some descriptive attributes. A schema is instantiated when actual objects are assigned to each table and the references between them are specified.

Table 1. A relational schema for a University model. Key fields are underlined. The schema has three entities and two relationships.

Student(*student_id*: integer, *intelligence*: string, *ranking*: string)
Course(*course_id*: integer, *difficulty*: string, *rating*: string)
Professor (*professor_id*, *teaching_ability*: string, *popularity*: string)
Registered (*student_id*: integer, *Course_id*: integer, *grade*: string, *satisfaction*: string)
RA (*student_id*: integer, *professor_id*: integer, *salary*: string, *capability*: string)

Graphical models. Graphical models [17, 14] are a popular tool for modeling both statistical models and statistical relational models. They provide a principled approach to dealing with uncertainty and relational data through probability theory. The goal of graphical models is to represent a joint distribution over a set of random variables. A random variable is a pair $X = \langle dom(X), P_X \rangle$ where $dom(X)$ is a set of possible values for X called the **domain** of X and $P_X : dom(X) \rightarrow [0, 1]$ is a probability distribution over these values. We assume in this paper that all random variables have finite domains (i.e., discrete or categorical variables). An **atomic assignment** assigns a value $X = x$ to random variable x, where $x \in dom(X)$. A **joint distribution** P assigns a probability to each conjunction of atomic assignments. The two most common classes of graphical models are Bayesian networks and Markov Networks [22].

A **Bayes net structure** is a directed acyclic graph (DAG) G, whose nodes comprise a set of random variables denoted by X. When discussing a Bayes net, we refer interchangeably to its nodes or its variables. A Bayes net(BN) is a pair $\langle G, \theta_G \rangle$ where θ_G is a set of parameter values that specify the probability distributions of each node conditional on instantiations of their parents. These conditional probabilities are specified in a **conditional probability table** (CP-table) for variable X. A BN $\langle G, \theta_G \rangle$ defines a joint probability distribution over $V = \{v_1, .., v_n\}$. The joint probability of an assignment is obtained by multiplying the conditional probabilities of each node value assignment given its parent value assignments.

A **Markov network** is a model for the joint distribution of a set of variables $X = (X_1, X_2, \dots X_n)$. It is composed of an undirected graph G and a set of potential functions ϕ_k. The graph has a node for each variable, and the model has a potential function for each clique in the graph. A potential function is a non-negative real-valued function of the state of the corresponding clique. The joint distribution over an assignment $X = x$ represented by a Markov network is given by

$$P(X = x) = \frac{1}{Z} \prod_k \phi_k(x_{\{k\}}) \tag{1}$$

where $(x_{\{k\}})$ is the state of the kth clique. Z, known as the partition function, is given by $Z = \sum_{x \in X} \prod_k \phi_k(x_{\{k\}})$

3 Propositional Data and Relational Data

Most real world data is relational which provides more information about an object via its links; . However, relational data has several major differences to propositional data that makes it more challenging to learn models. Several characteristics of relational data and some of its main differences to propositional data are the following.

1. The representation method for relational data and propositional data is different. A relational database stores data in multiple tables that represent different types of entities and relationships between them. Propositional data is stored in a single table.
2. Propositional data consists of identically structured entities, typically assumed to be independently and identically distributed (iid); however, relational data consists of different entities of different types which may be related to each other. For example, all tuples are of type student in propositional data and we assume all students are independent of each other. Relational data has tuples of different types (students and courses) and they may be statically dependent on each other.
3. In modeling propositional data, the number of different states is exponential in the number of attributes, e.g. with n binary attributes, the number of states is $O(2^n)$. When modeling joint distribution of relational data, the number of different states is exponential in the product of the attributes and objects, e.g. with m objects and n binary attributes in total, the number of states is $O(2^{nm})$.
4. The presence of **autocorrelation** is a feature of relational data which augments the complexity of relational learning. Correlations between values of attributes of

objects of the same type were dubbed autocorrelations by [12]. A relationship between objects of the same entity is required to have autocorrelation. For example, $Friends(s_1, s_2) = true$ is a relationship on entity $Student$ where s_1 and s_2 are friends. There may be a pattern between the intelligence of friends, ie, the attribute value $Intelligence(s_1)$ and the attribute value $Intelligence(s_2)$.

In our example, in order to do classification on a student, statistical relational learners may not only look at the courses that student has taken (i.e. the links of the object itself), but also other students who have taken those courses (i.e. the links of the linked objects).

5. One to many and many to many relationships in which an object is in relation with a set of objects is a character of relational data. For example a student may be registered in a set of courses where the $ranking$ of the student is in correlation with the $difficulty$ of a set of courses. Dealing with many to many relationships is a challenging problem for many SRL models.

Using relational data, not only leads to more accurate results on traditional tasks like classification and prediction, but also introduces some new tasks on relational data. The most popular tasks introduced on relational data are the following.

- Collective classification [13] is an extension over relational classification in presence of autocorrelation. Relational classification is the task of predicting the class label of an object given its attributes, links. In collective classification, the class labels of the links may be unknown.
- Linked based clustering [25] groups together objects that have similar characteristics based on their own attributes [7] and more importantly, the attributes of their links.
- Link prediction [23] determines whether a relation exists between two objects from the attributes of the objects and their links.

4 Statistical Relational Learners

Due to the underlying complicated multi-table structure and correlations of relational data, statistical learning methods have a different representation to propositional learners. SRL models use three graphical structures in order to fully define their model.

1. Graph $G_D(V_D, E_D)$ is used to show the schema of the database and instances. Nodes of G_D, are the objects of the database, and the edges between them represent the relationships between the objects. A vector is assigned to each node and each edge to keep the information of the attributes of objects and links. This graph presents an instantiated schema of the database used for learning. Figure 1 shows $G_D(V_D, E_D)$ for an instance of Table 1. The graph is not required for propositional learners as the single table provides all the existing information.
2. Graph $G_M(V_M, E_M)$ represents the class model which encodes probabilistic relationships among a set of random variables in relational data. Random variables in $G_M(V_M, E_M)$ are usually descriptive attributes of the dataset and edges between variables is a sign of their correlation. Figure 2 shows an example of a $G_M(V_M, E_M)$ for the instance in Figure 1.

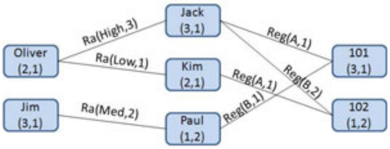


Fig. 1. Graph $G_D(V_D, E_D)$ for an instance of Table 1. Variables represent objects of the schema and links represent the relationships. An array is assigned to nodes and edges that carry the information of the objects and the relationships.

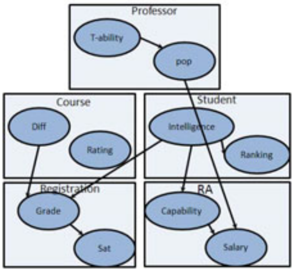


Fig. 2. An example of a graph $G_M(V_M, E_M)$ for G_D given in Figure 1. The variables represent descriptive attributes of tables of the schema and edges show dependencies between attributes.

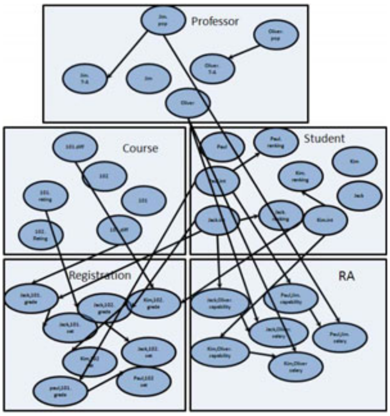


Fig. 3. The graph G_I for G_M in Figure 2 rolled out over the schema in Figure 1. The variables of the graph show descriptive attributes of each object and edges connect attributes of objects using the template from G_M.

3. A third graph G_I used for inference in Relational data mining Models is much larger than the other two graphs. The structure of G_I is determined by G_M and G_D. Let $G'_D(V_D, E_D)$ be the graph of the test data constructed similarly to $G_D(V_D, E_D)$ of the instance and $G_M(V_M, E_M)$ be the model. Then, $G_I(V_I, E_I)$ takes the template from G_M and the relations in $G'_D(V_D, E_D)$ constrains the way that G_I is rolled out. Figure 3 shows G_I for G_D in Figure 1 using the template G_M from Figure 2. The most common query type is the standard conditional probability

query $p(Y = y|E = e)$ where the evidence, E, is a subset of the random variables with instantiation e to those variables and the query nodes.

5 Statistical Relational Learning Models

In this section we review four of state-of-the-art models of statistical relational learning.

5.1 Probabilistic Relational Models (PRM)

Probabilistic relational models (PRMs) [8] are a rich representation language for statistical models which are among the first successful methods proposed for statistical relational learning (SRL). PRMs combine logical representation with probabilistic semantics based on directed graphical models. PRMs extend Bayesian networks with the concept of objects to deal with relational data.

Representation. A PRM, like a Bayesian network, has two components: the dependency structure and the parameters associated with it. For most cases in $G_M(V_M, E_M)$ where random variables correspond to attributes from different tables, an edge between them shows a correlation between sets. For example, ranking of a student may depend on the grades of courses she has taken. PRMs use the notion of aggregation from database theory to summerize information from different links (*mode, mean, median, maximum, and minimum*) We consider two types of PRMs

- PRMs in which both the objects and their relationships are fixed and the only uncertainty is over descriptive attributes of entities and relationships. Such a PRM and a database of objects with their relationships define a probability distribution over the descriptive attributes of the objects.
- PRMs with structural uncertainty in which objects are fixed but there is uncertainty over objects to which relationships correspond to. For example the tables for $student$, $course$, and $registration$ are available, however the foreign key attributes in the registration table which determines the student and the course that the information in the tuple refers to is missing.

Parameter Learning. The key feature in parameter learning is the likelihood function which is defined as the probability of the data given the model. Let $G_M^{\{V\}}(o)$ be an assignment of values to attributes related to object o over a model m. For example, $G_M^{intelligence,ranking}(Jack) = \{3, 1\}$. Formula 2 shows the likelihood of data given the model for PRMs.

$$l(\theta_{G_M}|G_D, G_M) = \sum_{v \in V_M} \sum_{o} \log p(G_M^v(o)|G_M^{pa(v)}(o)) \tag{2}$$

Where θ_{G_M} indicates the parameters for G_M, o is the set objects. In order to do parameter learning, the well understood theory of learning maximum likelihood(ML) parameter estimation may be used. Using ML, formula 2 can be decomposed into summation terms that each may be maximized separately, where $C_{[v,u]}$ is the number of times $G_M^V(o) = v$ and $G_M^{pa(V)}(o) = u$ occurs, and $v|u$ is the conditional probability of $G_M^V(o) = v$ given $G_M^{pa(V)}(o) = u$.

$$l(\theta_{G_M}|G_D, G_M) = \sum_{v \in V_M} \sum_{k \in D(v)} \sum_{u \in D(pa(v))} C_{[v,u]} \times \log \theta_{G_M(v|u)} \tag{3}$$

Structure Learning. Structure learning is more challenging task than parameter learning in graphical models. The general task of structure learning in PRMs is to find the set of edges E_M in G_M. The "goodness" of different structures must be comparable in order to allow preference of a model over another. For evaluating different structures, maximum a posteriori (MAP) or score functions like BIC [10] are used. For Bayesian networks the task of finding the best structure is NP hard. PRMs use greedy algorithms that iteratively modifies the structure to increase the score. Three operations used in each step are adding, removing or reversing an edge. In each step, all possible transformations using these operations are considered. The structure with highest score is chosen as the next candidate.

Inference. PRMs in few cases, when either the skeleton is small or the tree width is low, can use exact inference on the $G_I(V_I, E_I)$ graph. Unfortunately exact inference is usually not applicable with real world data. Inference in PRMS requires inference over the ground network defined by an instantiated PRM for a specific skeleton. Because $G_I(V_I, E_I)$ is usually very large, efficient inference is very complicated. The approximate algorithm used for inference in PRMs is a variant of belief propagation [19, 26].

5.2 Relational Dependency Network (RDN)

Relational Dependency Networks(RDNs)[20], an extension of Dependency Networks (DNs)[11], are a class of graphical models that approximate a joint distribution using a bidirected graph with conditional probability tables for variables.DNs have several characteristics which make them favorable for relational data: Their unique representation provides the ability to represent cyclic dependencies, simple methods for parameter estimation, and efficient structure learning techniques. The strength of RDNs is mostly due to the use of pseudo-likelihood [1] learning algorithm that estimates an acceptable approximation of the joint distribution.

Representation. RDNs extend the graphical model of DNs to the relational setting. The model encodes probabilistic relationships among a set of random variables with a bidirected graph $G_M(V_M, E_M)$ where conditional independence is interpreted using graph separation as in undirected models. Although conditional independence is inferred using an undirected view of the graph, bidirected edges are used to define the set of neighbors of a node used in their CPT. Each node has a probability distribution conditional on its neighbors as in directed models.The nodes of the model are similar to the variables in PRMs. Each node $v \in V$ corresponds to a descriptive attribute from the entity or relationship tables. An Edge between two nodes correspond to correlations between attributes of the dataset. However, the conditional probability distributions do not factor the joint probability so its impossible to calculate the joint probability directly as in PRMs.

Learning. RDNs extend the learning of DNs to a relational setting. The set of the conditional probability tables CPTs describe both the structure and the parameters of the model. RDNs use pseudo-likelihood techniques [1] to avoid the complexities of estimating the partition function. Instead of optimizing the log-likelihood of the joint distribution, RDNs optimize the pseudo-likelihood for each node separately conditioned on all its neighbors. Formula 4 computes the pseudo-likelihood for each node in G_M seperately, where θ_{G_M} is the parameters for G_M, $D(v)$ is the domain of values for variable v, and $D(p(v))$ is the domain of values for parents of v.

$$Pl(\theta_{G_M}|G_D, G_M) = \sum_{v \in V_M} \sum_{k \in D(v)} \sum_{u \in D(pa(v))} P(v = k|pa(v) = u) \tag{4}$$

where, $P(v = k|pa(v) = u)$is computed by two main relational learners

- Relational Bayesian classifier is a non selective model that treats heterogeneous relational subgraphs as a homogeneous set of attribute multisets. The classifier assumes that each value in the multiset is drawn independently from the same multinomial distribution. For example, the *difficulty* of the courses taken by a student form a multiset {Hard, Hard, Easy, Medium}. RBC selects values independently from the multiset distribution.
- Relational probability trees are a selective model that extend traditional classification trees to relational settings. Relational probability trees also treat heterogeneous relational subgraphs as a set of attribute multisets; however, instead of treating the values like an independent set, Relational probability trees use aggregation functions to map the set of values into a single value.

Inference. RDNs use Gibbs sampling for inference on G_I. The values of unobserved variables are initialized with their prior distribution and are iteratively relabeled using the current state on the model and the CPT of the node. Gibbs sampling is generally an inefficient approach to estimate the joint probability of the model, however, it is reasonably fast to estimate conditional probabilities for each node given its parents.

5.3 Bayesian Logic Programming (BLP)

Bayesian Logic programs [15] are a model based on Bayesian networks. BLPs use logic programming [18] to unify Bayesian networks with logic programming. This unification overcomes the propositional character of Bayesian networks and logical programs. BLPs use Bayesian clauses that use a conditional probability table to present the distribution of the head of the clause conditional on its body, and use combining rules to unite the information on a single literal that is the head of several clauses. BLPs are implemented in a software called BALIOS [15] and are considered as one of successful models of SRL.

Representation. BLPs are produced from logical programs. A logical program is a set of clauses of the form $A : B_1, B_2, \ldots B_n$ where A and B_i are universally quantified atoms. We call A the head and B_is the body of the clause. The head of the clause is considered true in the model if the body of the of the clause is entailed. BLPs use

Bayesian clauses which differ from logical clauses. Bayesian clauses uses a conditional probability table to keep the probability of the head of the clause conditioned on its body, whereas logical clauses have a deterministic value. It is possible to have several clauses with the same variable in the head of the clause. Since each clause has its own local probability distribution, a variable may have several local probability distributions with possibly different sets of parents. To obtain a single conditional probability distribution for the variable that includes the union of all parents, *combining rules* are used. A combining rule is a function that maps finite sets of conditional probability distributions $\{ P(A|A_{i1} \ldots A_{in_i}), i = 1 \ldots m \}$ on to one combined conditional probability distribution $P(A|B_1 \ldots B_k)$ with $P(A|B_1 \ldots B_k) \subseteq \bigcup_{i=1}^{m} \{A_{i1} \ldots A_{in_i}\}$.

Learning. Learning in BLPs, as in MLNs [4], is a probabilistic extension of learning in inductive logic programing [18] and is formulated as follows. "Given a set of Bayesian logic programs H, G_D, and a scoring function F; find a acyclic candidate $H^* \in H$ such that H^* matches G_D best according to F ". The score function F is used to evaluate how good the clauses are. To adapt traditional techniques used for parameter estimation of Bayesian networks such as Expectation maximization algorithm, combining rules are required to be decomposable ; most common combining rules for Bayesian networks such as "noisy or" are decomposable. The best match refers to those parameters of the associated conditional probability distributions that maximize the scoring function where the score function is based on maximum likelihood [5]. Structure learning in BLNs follows the procedure of rule learning in ILP systems [21] which have operators such as adding and deleting logically valid literals, flipping, instantiating variables, unifying variables on literals or clauses. BLNs speed up the learning procedure executing several operations simultaneously.

Inference. Inference, as in other SRL methods, is intractable in BLNs and is proceeded via grounding the clauses of Bayesian logic Program. Each Bayesian logic Program species a propositional Bayes net, where inference is done using standard Bayes net learning algorithms

5.4 Markov Logic Networks (MLN)

Markov Logic Networks (MLNs) [4] are among the most well known methods proposed for statistical relational learning. Syntactically MLNs extend first-order logic and put a weight for each formula. Semantically, they can represent a probability distribution over possible worlds using formulas and their coressponding weights.

Representation. Formally, a Markov Logic Network is a set of pairs of formulas and their corresponding weights (F_i, w_i) where formulas are in first order logic and the weights are any real number. The set of formulas in MLNs correspond to the class model G_M. An MLN with a finite set of objects in G_D defines a ground Markov network G_I. A grounding is defined as assigning a value to a variable from its domain. G_I has a binary node for each ground predicate in the MLN. The value of the node is 1 if the ground atom is true and 0 otherwise. Also, there is an edge between two nodes if the ground predicates appear together in at least one grounding of a formula. An MLN is a template for the ground Markov network and the size of the model is a function of the number of objects. The ground network has regularities in structure and parameters which are forced by the MLN. All terms in a formula form a clique.

A world is an assignment of truth values to all possible ground atoms. Each state of the Markov network presents a possible world. The probability distribution over possible worlds x specified by the ground network is calculated by Formula 5 where $n_i(x)$ is the number of true groundings for F_i in x and Z is the partition function that is used to make the summation of all possible groundings adds up to one.

$$P(X = x) = \frac{1}{Z}\exp(\sum_i w_i n_i(x)) \tag{5}$$

Parameter Learning. Finding the weight of the formulas in MLNs is equivalent to computing parameters in other models. The weights are learned from the relational database. Assuming the network has n ground atoms, a database has up to n entries indicating the true facts. MLNs use the close-world assumption [6] that if a ground atom is absent in the database, it is assumed to be false. In MLNs, the weight are computed by maximizing the log likelihood of the data. Formula uses the derivative of Formula 5 with respect to its weights.

$$\frac{\partial}{\partial w_i}\log p_w(x) = n_i(x) - \sum_{x'} P_w(x')n_i(x') \tag{6}$$

Where the sum is over all possible databases x', and $P_w(X = x')$ is $P(X = x')$ computed using the current weight vector. However, Formula 6 is NP hard to compute. An approximation for calculating probability of a world using pseudo-likelihood [1] is used that omits the use of the partition function. Pseudo-likelihood is a measure in statistics that serves as an approximation of the distribution of x based on its Markov blanket instead of all other x'.

Structure Learning. Structure learning in MLNs is very similar to BLPs. They use the CLAUDIEN [3] system which is able to learn first order formulas and not just horn clauses.

Inference. Inference has two main phases in MLNs. In the first phase, a minimal subset of G_I the ground Markov network is selected. Many predicates that are independent of the predicates of the query may be filtered in this phase. As a result inference carried out over a smaller Markov network. In the second phase inference is performed on the Markov network using Gibbs sampling [2] where the evidence nodes are observed and are set to their values. Gibbs sampling first randomly initialize and orders unobserved variables in the network, i.e. $\{X_1 = x_1 \ldots X_n = x_n\}$, and then iterate through the variables. In step 1 a new value x'_1 for variable X_1 is sampled conditional on all the other variables $P(X_1|X_2 = x_2 \ldots X_n = x_n)$, and in step i a new value x'_i for variable x_i is sampled conditional on all the other variables $P(X_i|X_1 = x'_1 \ldots X_{i-1} = x'_{i-1}, X_{i+1} = x_{i+1} \ldots X_n = x_n)$. Conditional independence eases the computation as V_i conditional on its immediate neighbors, Markov Blanket, is independent of all other variables.

6 Limitations of Current Methods and Conclusion

Statistical relational models(SRL) methods have generally been very successful. Table 2 summarizes comparison among different SRL models based on various dimensions of importance in Statistical-Relational Learning. Many different problems have been defined over SRL models and good results have already been achieved; however, we believe limitations on current models show the necessity for more research on the field. In this section we point out some of the limitations of most SRL models.

Table 2. A comparison of Probabilistic Relational Models (PRMs), Markov Logic Networks (MLNs), Relational Dependency Netoworks (RDNs), and Bayes Logic Networks (BLNs) along various dimensions of importance in Statistical-Relational Learning

	PRM	**MLN**	**RDN**	**BLN**
Class level model	Directed GM	Logical clauses	Bidirected GM	bipartite directed GM
Parameter Estimation	ML to fill CPTs	ML to learn Weights	CR Learners to learn CPTs	ML to fill CPTs
Structure learning	Score Based learning	ILP methods	Use CR learners	ILP Methods
Inference Graph	Bayesian network	Markov Model	undirected model	Bayesian network
Inference Method	belief propagation	Pseudo-likelihood	Pseudo-likelihood	stand BN inference
Autocorrelation	self-loops in class-level model	additional variables needed	Yes	Not discussed
X-many relationships	Require Aggregation	No requirements	No requirements	Require combination rules

The computational complexity of inference is probably the biggest limitation shared between most SRL methods. The size of the graph G_I is proportional to the number of descriptive attributes and objects, which limits the scalability for many realistic datasets. PRMs and BLNs suffer from inference as they use standard complex Bayesian network inference algorithms on the G_I graphs. MLNs share the same problem as they use an undirected model as their ground network. It is impractical to do exact inference on large Markov models because of the computations on the partition function. MLNs are forced into using approximation techniques for inference on generative models. Pseudo-likelihood fails to give significant results when querying on variables that are distant in the model [4]. Inference is quicker in RDNS, because they approximate a joint distribution using a bi-directed graph with conditional probability tables for variables; however, the conditional probability distributions do not factor the joint probability so it is impossible to calculate the joint probability directly as in most other SRL methods. A lot of research is currently being done on lifted inference [16]. Lifted inference aims to do exact inference without materializing the ground inference graph.

Autocorrelation causes significant representation and computational difficulties for most SRL models. PRMs and BLNs add a single random variable to their class model for every descriptive attribute in the dataset. Due to this fact, autocorrelation is shown with self loops in their class model. It is possible to achieve acyclic ground models even with the existence of self loops in the class model, however additional information on the dataset is required which complicates the model and is usually unknown or does not exist. A well known example is the correlation between blood type of parents and their children. The parent relationship does not introduce loops so, showing the correlation does not have any cycles in the ground model, however, the descriptive attribute of blood type requires a self loop in the class level model. Neville and Jensen conclude that the acyclicity constraints of directed PRMs precludes the learning of arbitrary autocorrelation dependencies and thus severely limits the applicability of these models in relational domains[20].

The underlying complicated multi-table structure results in large datasets, which leads to difficulties of scalability and efficiency in structure learning of SRL. Structure learning in MLNs is very similar to ILP methods which are usually not scalable and very inefficient for large datasets. Some new approaches for structure learning in MLNs are being considered, but they have only been tested on small to medium sized datasets. PRMS and BLNs use score based learning that leads to local maxima solutions. RDNs use two relational learners as part of their model and both relational learners have its own problems. Relational Bayesian classifiers are non selective and choose a single value independently from the multinomial distribution of values of the link to be a representer, but this weakens the model. Relational probability trees use an informative aggregation functions to describe a set of values with one value, but this increases the complexity of the learning procedure.

Acknowledgments. This article is the result of a part of our research under the supervision of Dr Oliver Schulte. We thanks Dr Schulte for his guidance and interesting discussions. We also thank Tong Man and Xiaoyuan Xu for helpful comments.

References

[1] Besag, J.: Statistical analysis of non-lattice data. The Statistician 24(3), 179–195 (1975)

[2] Casella, G., George, E.I.: Explaining the gibbs sampler. The American Statistician 46(3), 167–174 (1992)

[3] De Raedt, L., Dehaspe, L.: Clausal discovery. Mach. Learn. 26(2-3), 99–146 (1997)

[4] Domingos, P., Richardson, M.: Markov logic: A unifying framework for statistical relational learning. In: Introduction to Statistical Relational Learning [9], ch. 12, pp. 339–367 (2007)

[5] Edgeworth, F.Y.: On the probable errors of frequency-constants (contd.). Journal of the Royal Statistical Society 71(3), 499–512 (1908)

[6] Eiter, T., Gottlob, G.: Propositional circumscription and extended closed world reasoning are π_2^p-complete. Theoretical Computer Science 114, 231–245 (1993)

[7] Flake, G.W., Lawrence, S., Lee Giles, C.: Efficient identification of web communities. In: KDD 2000: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 150–160. ACM, New York (2000)

[8] Getoor, L., Friedman, N., Koller, D., Pfeffer, A., Taskar, B.: Probabilistic relational models. In: Introduction to Statistical Relational Learning [9]

[9] Getoor, L., Tasker, B.: Introduction to statistical relational learning. MIT Press, Cambridge (2007)

[10] Heckerman, D.: A tutorial on learning with bayesian networks. In: NATO ASI on Learning in graphical models, pp. 301–354 (1998)

[11] Heckerman, D., Chickering, D.M., Meek, C., Rounthwaite, R., Kadie, C., Kaelbling, P.: Dependency networks for inference, collaborative filtering, and data visualization. Journal of Machine Learning Research 1, 49–75 (2000)

[12] Jensen, D., Neville, J.: Linkage and autocorrelation cause feature selection bias in relational learning (2002). In: Proceedings of the 19th International Conference on Machine Learning (2002)

[13] Jensen, D., Neville, J., Gallagher, B.: Why collective inference improves relational classification. In: Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 593–598 (2004)

[14] Jordan, M.: Graphical models. Statistical Science (Special Issue on Bayesian Statistics) 19, 140–155 (2004)

[15] Kersting, K., de Raedt, L.: Bayesian logic programming: Theory and tool. In: Introduction to Statistical Relational Learning [9]

[16] Zettlemoyer, L.S., Leslie, M.H., Kristian, P.K., Kersting, B.M.: Reasoning about large populations with lifted probabilistic inference. In: NIPS Workshop (2008)

[17] Lauritzen, S.L.: Graphical Models. Oxford Statistical Science Series. Oxford University Press, USA (July 1996)

[18] Muggleton, S.: Inductive logic programming. New Gen. Comput. 8(4), 295–318 (1991)

[19] Murphy, K.P., Weiss, Y., Jordan, M.I.: Loopy belief propagation for approximate inference: An empirical study. In: Proceedings of Uncertainty in AI, pp. 467–475 (1999)

[20] Nevile, J., Jensen, D.: Relational dependency networks. In: An Introduction to Statistical Relational Learning [9]

[21] Nienhuys-Cheng, S.-H., de Wolf, R. (eds.): Foundations of Inductive Logic Programming. LNCS, vol. 1228. Springer, Heidelberg (1997)

[22] Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, San Francisco (1988)

[23] Taskar, B., Wong, M., Abbeel, P., Koller, D.: Link prediction in relational data (2004)

[24] Ullman, J.D.: Principles of database systems, Vol. 2. Computer Science Press, Rockville (1982)

[25] Wang, Y., Wang, Y., Kitsuregawa, M.: Link based clustering of web search results. LNCS, pp. 225–236. Springer, Heidelberg (2001)

[26] Weiss, Y.: Correctness of local probability propagation in graphical models with loops. Neural Comput. 12(1), 1–41 (2000)

[27] Winkler, W.: The state of record linkage and current research problems (1999)

Exploiting Frame Information for Prepositional Phrase Semantic Role Labeling

Dunwei Wen and Qing Dou

School of Computing & Information Systems, Athabasca University,
Athabasca, Alberta T9S 3A3, Canada
dunweiw@athabascau.ca, qdou@cs.ualberta.ca

Abstract. Semantic role expresses the underlying relations that an argument has with its governing predicate. Prepositional phrase semantic role labeling concentrates on such relations indicated by prepositional phrases. Previously, the problem has been formulated as a word sense disambiguation (WSD) problem and contextual words are used as important features. In the past years, there has been a growing interests in general semantic role labeling (SRL). Therefore, it would be interesting to compare the previous contextual features with argument related features specifically designed for semantic role labeling. In experiments, we showed that the argument related features are much better than the contextual features, improving classification accuracy from 84.96% to 90.25% on a 6 role task and 71.47% to 75.93% on a 33 role task. To further investigate dependency between frame elements, we also introduced new features based on semantic frame that consider the governing predicate, preposition, and content phrase at the same time. The use of frame based features further improves the accuracy to 91.25% and 83.48% on both tasks respectively. In the end, we found that by treating prepositional phrases carefully, the overall performance of a semantic role labeling system can be improved significantly.

1 Introduction

Prepositional phrases (PPs) prevail in languages. Labeling of semantic role for prepositional phrases is important. It has been used to help construction of semantic lexicons [1]. More importantly, it is critical to the overall performance of semantic role labeling system. To demonstrate the importance of preposition for the task of semantic role labeling (SRL), we counted the number of arguments that are prepositional phrases in the test data of CoNLL 2005 Shared Task and observed that prepositional phrases prevail in a wide range of semantic roles. Particularly, the percentage of prepositional phrases increases from ARG0 to ARG5. For ARG3, ARG4, and ARG5, the percentage of prepositional phrases is over 50%.

Given the prevalence of prepositional phrases in the SRL task, there is an increasing interest in the use of prepositions for SRL [2,3]. Although it has been shown that the semantic role of a prepositional phrase can be predicted by using some general contextual information of prepositions in question [1,2], the types of roles are only limited to those that generalize well across different predicates, such as time and locations. This might explain why previous work is not able to achieve impressive improvement in the

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 269–272, 2010.

general SRL task. we believe that information in predicate frames should be explored. To capture the content information, we apply some features designed for general semantic role labeling. To catch dependency between the frame elements, we introduce some new features that consider the whole frame (governing predicate, preposition, and content). We use Support Vector Machines (SVMs) to utilize a much larger number of features than the previous WSD features used in [2]. In experiments, we show that the carefully selected content and governing predicate related features are much better than the general WSD features. In addition to that, by adding our new frame based features, we show significant improvements in a holistic SRL system, especially on roles where prepositional phrases dominate. To our knowledge, this is the first time that the use of preposition improves the overall performance of SRL.

2 Semantic Role Labeling of Prepositional Phrases

Since our objective is to improve a holistic SRL system, instead of using roles defined in Penn TreeBank annotation, we focus on roles annotated in the PropBank. The former covers 9 coarse and non verb-specific semantic roles for prepositional phrases. In contrast, the latter includes 33 different roles related to prepositional phrases, some of which are verb specific (e.g. ARG0-ARG5).

We formulate the SRL of prepositional phrases as a multi-class classification problem using Support Vector Machines (SVMs) [4]. We choose SVMs because it has been shown to work well in similar problems, such as text classification, where data is represented with a large number of sparse features. We train an SVM classifier for each class.

New Features. Through our observations, we found that semantic roles in FrameNet and PropBank are defined in frames, where the governing predicate, preposition, and phrase content interact with each other. Therefore, the entire frame should be considered[1]. Due to data sparse problem, we break down the frame by enumerating all possible combinations of any two components. We call this type of new features frame based features and list them in Table 1. All the predicates and content words are in their lemma forms. Although it is also possible to use a polynomial kernel to cover some of our frame based features, it also generates a large number of nonsensical combined features and significantly increases training time.

Since the task of semantic role labeling has been studied extensivly in recent years, it is also interesting to compare those commonly used argument related features used in [5,6].

3 Prepositional Phrase SRL Experiment

The training examples for prepositional phrase SRL are extracted from the data used in CoNLL 2005 shared task. The data is based on PropBank and corresponds to the Wall Street sections 02-21 as training set, section 24 as development set, and section 22-23 as

[1] Note, in a normal setting, one would not be able to know the governing verb. We take advantage of how our data is annotated and show the importance of this information through our experiments.

Table 1. Frame Based Features

Predicate-Preposition-Content Word
Predicate-Preposition-POS of Content Word
Predicate-Preposition
Predicate-Content Word
Predicate-POS of Content Word
Preposition-Content Word
Preposition-POS of Content Word

Table 2. Prepositional phrase SRL using different features. WSDFeat: uses standard word sense disambiguation features in [1]. ArgFeat: uses the argument structure features described in [5]. FrameFeat: uses collocation features described in Table 1. Arg+Frame: uses both Argument structure features and frame based features.

	33 Roles	6 Roles
WSDFeat	71.47 %	84.96%
ArgFeat	75.93%	90.59%
FrameFeat	80.74%	88.40%
Arg+Frame	83.48%	91.25%

testing set. To make a cross comparison with previous approaches, we selected 6 roles, AM-EXT, AM-MNR, AM-TMP, AM-DIR, AM-LOC, AM-CAU, which are closest related to those evaluated in [1,2]. It is worth noticing that all the 6 roles and the 7 roles considered in [1,2] are all verb independent, which means their definitions generalize well across different verbs. Same as previously reported results, we evaluate the systems using 5 fold cross validation and report the accuracy of classification.

Table 2 provides results on the SRL of prepositional phrases all 33 roles and 6 selected roles. ArgFeat achieves 75.93% accuracy on 33-role task, although the features it uses is not designed specifically for classifying prepositional phrases. In contrast, WSDFeat only has 71.47% accuracy. Most noticeably, FrameFeat, which considers the governing verb and content word at the same time, increases the accuracy to 80.74%, a 32% error reduction compared with WSD features. When the frame based features are combined with argument structure features, the accuracy is further improved to 83.48%.

In the 6-role challenge, Arg+Frame achieves the highest accuracy 91.25%, followed by ArgFeat 90.59%. FrameFeat alone obtains 88.40% accuracy. All are much higher than the accuracy of WSDFeat, 84.96%. The reason that ArgFeat outperforms FrameFeat might be that, in the 6-role task, all the roles are not verb-specific. Therefore, the governing verb might not be as discriminative as it is in the 33-role task. Previously, Tom O'Hara and Janyce Wiebe [1] reported 85% accuracy and Patrick Ye and Timothy Baldwin [2] reported 87% accuracy on a 7-role task using Penn TreeBank annotation.

4 Combined Holistic SRL Experiment

Given the improvement we have gained in the prepositional phrase SRL task, we hope the frame based features introduced for disambiguating the semantic roles of prepositional phrases can help a holistic SRL system.

In general, our baseline system BASELINE achieves 75.92 overall F1 score, which is slightly worse than 76.45 reported in [5]. Nonetheless, it is still a very strong baseline. With the new frame based features , the F1 score of our system PREPSRL is increased to 77.23, 1.3 higher compared to our baseline and 0.8 higher than the Surdeanu's system. To our knowledge, this is the first time that the use of prepositions shows help on such a wide range of different roles and overall F1 score. To make a further assessment, we compare the performance of PREPSRL and BASELINE on 7 semantic roles where the proportion of prepositional phrases is over 40%.

5 Conclusion

We compared the use of general WSD features and features designed for semantic role labeling in the task of prepositional phrase SRL and showed that the performance can be greatly improved by carefully selecting specific context information. The use of SVMs allowed us to include a large number of collocation features that consider interactions between governing predicates, prepositions and phrase content simultaneously, which is very helpful for disambiguating verb dependent roles. When the new frame based features were added to a state-of-the-art SRL model, we gained significant improvements over a wide range of semantic roles. In the future, we would like to investigate automatic detecting of attachment of predicate and prepositional phrase and use that to further improve SRL.

Acknowledgments

The authors acknowledge the support of Mission Critical Research (MCR) grant and Research Incentive Grant (RIG) by Athabasca University.

References

1. O'Hara, T., Wiebe, J.: Preposition semantic classification via Treebank and FrameNet. In: Proceedings of Computational Natural Language Learning (CoNLL 2003), Edmonton (2003)
2. Ye, P., Baldwin, T.: Semantic role labelling of prepositional phrases. ACM Transactions on Asian Language Information Processing 5(3), 228–244 (2006)
3. Dahlmeier, D., Ng, H.T., Schultz, T.: Joint learning of preposition senses and semantic roles of prepositional phrases. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pp. 450–458 (2009)
4. Vapnik, V.: Statistical Learning Theory. John Wiley and Sons Inc., Chichester (1998)
5. Surdeanu, M., Turmo, J.: Semantic role labeling using complete syntactic analysis. In: Proceedings of the CoNLL share task: semantic role labeling (2005)
6. Gildea, D., Jurafsky, D.: Automatic labeling of semantic roles. Computational Linguistics 28(3), 245–288 (2002)

Phrase-Based Statistical Machine Translation for a Low-Density Language Pair*

Maxim Roy and Fred Popowich

School of Computing Science, Simon Fraser University
Burnaby, BC, Canada V5A1S6
maximr@cs.sfu.ca, popowich@cs.sfu.ca

Abstract. We present a phrase-based statistical machine translation (SMT) system for Bangla to English that incorporates a novel transliteration module, and a specialized component for handling prepositions and Bangla compound words. We evaluate our components through their impact on the BLEU score for the phrase-based SMT system. According to the experimental results, the transliteration component has the most significant impact on the BLEU score. We also provide a new test set with multiple references between Bangla and English for MT evaluation purposes. Finally we propose a new manual evaluation approach for the MT community and evaluate our components using the new manual evaluation approach.

1 Introduction

Machine translation (MT) from Bangla to English has recently become a priority task for Bangla Natural Language Processing (NLP) community. In SMT massive amounts of parallel text between the source and target language are required to achieve high quality translation. However, there are a large number of languages that are considered low-density, either because the population speaking the language is not very large, or if there is insufficient bilingual text involving that language. Bangla is one such language. It still lacks significant research in the area of NLP specifically in MT ([4], [5]).

In this paper we describe the first phrase-based SMT system from Bangla to English which incorporates transliteration, compound word and prepositional module in order to deal with limited resources. Further we extended a Bangla-English test set which contained only a single reference translation by adding multiple reference translations to the test set, and propose a new manual evaluation approach for evaluating MT output which does not require source language knowledge.

2 Bangla Specific Modules

Below we describe three modules added to our SMT system which are the Bangla transliteration module, preposition module and compound module. The Bangla

* This research was partially supported by a discovery grant from NSERC, Canada.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 273–277, 2010.

compound module was added as pre-processing step and transliteration and preposition module were added as post-processing steps.

2.1 Transliteration

We used a generative approach for transliteration inspired by the work done by [1] for their statistical transliteration system from English to Arabic in the context of Cross Lingual Information Retrieval (CLIR). Our transliteration module looks for the best candidates for transliteration of a given Bangla named entity using the Viterbi algorithm in a Hidden Markov Model (HMM) framework. The approach treats each letter and each word as word and sentence respectively. The transliteration module is trained using GIZA++[1] on a parallel name list of about 580000 names collected from the West Bengal election website[2]. The language model is built using the SRILM toolkit over a list of names in English collected from the US census bureau[3] and the west Bengal election website. Given the output of our SMT system, the transliteration module first identifies all the untranslated named entities. Then the transliteration module is applied to find the best English matches for all the explicitly written letters of named entities in Bangla. The generated transliterations are validated against an English monolingual dictionary by string comparison and finally the untranslated words are replaced with these transliterations in the output of our SMT system.

2.2 Handling Prepositions

In Bangla, there are no prepositions. English prepositions are handled in Bangla by using inflections attached to the referenced objects and/or post-positional words inserted after the referenced objects. Our analysis based on the output of the SMT system shows that some words were not translated due to the presence of inflections. In the SMT system we handle several Bangla inflections -r, -er, -yer, -ke, and -e which can be translated into any of the English prepositions “in”, “of”, “to” and “at” based on several rules. Our Bangla preposition handling module works according to the following steps: First it considers all words that have any of the following inflections: (r / er / yer / ke/ e). It then removes the inflection and looks for the base word in the bilingual dictionary to check if the word exists. If the word matches, it applies the appropriate rule based on the inflection. Finally, it proposes the English preposition and the translation of the word from the bilingual dictionary lookup as a correct resolution of preposition.

2.3 Handling Bangla Compound Words

Bangla has a large number of compound words. Almost all combinations of noun, pronoun and adjectives can be combined with each other joined optionally

[1] GIZA++ is a tool that performs alignment between two parallel aligned corpora.

[2] http://www.indian-elections.com/assembly-elections/west-bengal/

[3] http://www.census.gov/

by hyphens, to form a compound. For example, “ma-baba” in Bangla means “mother and father”. Koehn and Knight [2] presented an empirical splitting algorithm that is used to improve translation from German to English which we used as the basis for our approach. Our compound splitting approach can be described in the following few steps. We first handle compound words joined by a hyphen (‘-’) since in Bangla most of the time the meaning of these compound words is the composition of the meaning of each root-word. For all the words with hyphens, remove the hyphen, and look for each component in a monolingual Bangla corpus. If both words are found in the corpus, we propose the two words as a replacement for the compound word. So after the first step, some compound words containing hyphens will have been replaced. We then consider all possible splits of a word into known words that exist in the monolingual corpora. We restrict known words to be of at least of length three. Then for all splitting options we compute their frequency in the monolingual corpora and compute the arithmetic mean for each proposed compound split based on their frequency. The proposed compound split with the highest arithmetic mean is selected as correct compound split.

3 Dataset and Experimental Results

The corpus we used for training the system was provided by the Linguistic Data Consortium[4](LDC) containing approximately 11,000 sentences of newswire text taken from the BBC Asian Network and some other South Asian news websites. A bilingual Bangla-English dictionary collected from different websites was also used as part of the training set which contains around 55,000 words. For our language model we used data from the English section of EuroParl[5] combined with the LDC training set. The LDC development and test sets contain 600 and 1000 sentences respectively. The test set provided by the LDC contains only single reference translations between Bangla and English. We extended the LDC test set by adding two new English reference translation sets. The SMT system we used in our experiments is Moses [3]. We evaluate the performance of the three modules by showing the improved performance for the whole MT system using BLEU, WER (word error rate), and PER (position independent word error rate). The results of these different approaches are described in table 1, where all the training data from LDC corpus was used to train the system and LDC test set was used to evaluate the system. Our system obtains a BLEU score of 10.1 when all three modules are incorporated which is a 1.1 increase in BLEU score over the baseline system. Next we evaluated the translation quality of our SMT system on a new extended test set which was described earlier. We achieve on average around 1.4 and 2.5 increase in BLEU score with two and three reference test sets respectively compared to the single reference test set provided by the LDC corpus.

[4] LDC Catalog No.: LDC2008E29.

[5] Distributed for the shared task in the NAACL 2006 workshop on SMT (WSMT06).

4 Manual Evaluation

In addition to MT evaluation using BLEU, PER and WER we also conducted a manual MT quality evaluation. Since manual evaluation is time consuming, we are only comparing our overall SMT system containing transliteration, prepositional and compound word modules with the baseline system. We randomly selected 20 sentences from the test set and translated with our overall system, and the baseline system. We then created a survey with 20 questions where each question is the reference sentence and each answer contains three options baseline output, our overall system output or a statement saying both outputs are similar. Then we ask humans to choose the option which is most appropriate for the reference sentence. We sent the survey to 22 participants and compiled their responses for the 20 questions of the survey. Participants preferred the baseline system 22 %, our system 52 % and both similar 25 %. Based on the 20 questions of the survey we performed further analysis of the results and concluded that the baseline system was slightly better for shorter sentences, however our system outperforms the baseline system in handling longer sentences.

Table 1. Impact of Transliteration, compound word and prepositional module on SMT

System	BLEU (%)	WER (%)	PER (%)
Baseline(with dictionary)	8.0	82.5	62.4
Baseline+Transliteration	8.6	81.1	61.5
Baseline+Compound words	8.3	81.9	61.9
Baseline+Preposition handling	8.2	82.1	62.2
Baseline+Preposition handling+Transliteration	8.9	80.7	61.2
Baseline+All	9.1	80.5	61.0

5 Conclusion and Future Work

In this paper, we have produced the first phrase based SMT system for Bangla to English with minimal resources. We incorporated three modules into our baseline MT system to improve the translation accuracy of Bangla to English SMT. Additionally we have contributed a better test set with three reference test sets for evaluation of translation quality between Bangla and English SMT systems. We also proposed a new manual evaluation approach for evaluating SMT output, since automatic evaluation like BLEU is not always sufficient to reflect a genuine improvement in translation quality.

References

1. AbdulJaleel, N., Larkey, L.S.: Statistical Transliteration for English-Arabic Cross Language Information Retrieval. In: Proc. of CIKM 2003 (2003)
2. Koehn, P., Knight, K.: Empirical Methods for Compound Splitting. In: EACL 2003, pp. 187–194 (2003)

3. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open source toolkit for statistical machine translation. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics (2007)
4. Naskar, S.K., Bandyopadhyay, S.: A Phrasal EBMT System for Translating English to Bangla. In MT Summit X (2005)
5. Rahman, A., Islam, S., Alim, A., Hasan, K.: A Rule Based English-Bangla MT System for Compound Sentences. In: Proceedings of NCCPB (2005)

Coordination of Subject Markers in Arabic and Typed Categorial Logic

Adel Jebali, Ismaïl Biskri, and Louisette Emirkanian

Concordia University, Université du Québec à Trois-Rivières,
Université du Québec à Montréal
ajebali@alcor.concordia.ca, ismail.biskri@uqtr.ca,
emirkanian.louisette@uqam.ca

Abstract. In Standard Arabic, object markers and subject markers behave differently, although they share some properties. We are concerned here by their morphosyntactic status; whether they are arguments or agreement markers. The status of object markers is not an issue in itself, but the status of subject markers is one. The agreement asymmetries lead us to stipulate that, when subject markers are doubled by NPs whether these NPs are coordinated or not, they are agreement markers and not arguments. This analysis is implemented by means of an applicative combinatory categorical grammar (ACCG).

Keywords: Typed Categorial Grammar, Arabic Subject Markers, Coordination, ACCG.

1 Introduction

In standard Arabic, there are two sets of elements that can be qualified as pronominal forms: independent forms and conjunctive forms. Conjunctive forms, which will be referred to as argument markers, borrowed from [2], are prosodically deficient morphemes that encode the argument properties with which they are associated or that they replace.

In spite of their resemblances, subject and object markers present several differences that allow them to be split into two distinct groups.

In this paper, we will mainly focus on subject markers, which have only one possible host (i.e. verbs). Furthermore, purely morphologically speaking, only those markers are included in the morphological boundaries of their hosts. We address the question concerning the morphosyntactic status of these subject markers: are they arguments or agreement markers? In order to answer this question, it is necessary to observe the behaviour of these units in contexts where they co-occur with noun phrases.

2 Morphosyntactic Status of Subject Markers

Let us first clarify what we mean by agreement marker and argument. We define the agreement marker as an element devoid of a proper syntactic function. This element refers to another with which it shares certain grammatical features in a given

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 278–281, 2010.

configuration [2]. The agreement marker is therefore usually part of a word (or a phrase) that itself satisfies a syntactic function. We define a syntactic argument here as being a linguistic object selected by, and so complement of, another constituent.

There are several contexts that allow us to address the question of the morphosyntactic status of argument markers. The cases of interest are either those of doubling or dislocation. Even if the dislocation is tied in with the doubling in terms common features, it is distinguished by an intonational break (or pause) which places the dislocated element in a peripheral position that produces certain pragmatic effects, such as emphasis.

The following examples illustrate cases of dislocation (1) and doubling (2).

(1) ?ar- rijaal -u jaa? **-uu**
the- men -NOM come.PER -3MP
"The men, they came."
(2) jaa? **-at** ?al- banaat -u
come.PER -3F the- girls -NOM
"The girls came."

In (1) the agreement is made in gender and number (rich agreement). In (2), and this is particularly interesting, the doubling has an effect on the agreement. As can be seen in (2), the agreement is only made in terms of gender (the subject marker *-at* is feminine singular). It is therefore a case of impoverished agreement.

The tests proposed by [4] and applied to our data [5] lead us to conclude that subject markers in this context are agreement markers and not arguments. To corroborate this hypothesis and our observations on agreement asymmetries, it is appropriate to examine the conjoined structures that introduce, in a certain way, an extra level of agreement.

In the case of the dislocation in (3), agreement is made in gender and number with the sum of the values of the conjuncts as in (1).

(3) jaa? **-aa,** ?al- bint -u wa l- walad -u
come.PER -3MD, the- girl -NOM and the- boy -NOM
“They came, the girl and the boy.”

Example (4) illustrates a particular behaviour. Here, the subject marker *–at* (3F) attached to the verb *xaraj* (to go out) agrees only in gender, as in (2). We furthermore notice that the agreement is made only with the first conjunct[1].

(4) xaraj **-at** al- banaat -u wa al- ?awlaad -u
leave.PER -3F the- girls -NOM and the- boys -NOM
“The girls and the boys left.”

The data on coordinate structures show us that subject markers enter into the same type of impoverished agreement relationship in the case of doubling as the postposed nominal, be it a single element or two coordinated elements.

[1] [1] proposed that the structure is VSm NP [and NP], which would explain how the agreement is only made with the first conjunct.

In light of the tests proposed by [4] to account for the anaphoric agreement/ grammatical agreement opposition on the one hand, and on the other hand, the behaviour of subject markers in a doubling context (with coordinate NPs or not), we can conclude that they are indeed agreement markers.

3 Implementation by Means of the Applicative Combinatory Categorial Grammar (ACCG)

The applicative combinatory categorial grammar [3] model is founded on explicit logical rules, substituting a purely surface linguistic analysis for an inferential logical calculation. Relying more on the notion of surface structure, it leads, with the possible introduction of a combinatory, to a logical form in order to express meaning. This model has the advantage of being able to represent the intricacies of phrasal units by way of the operation of the application of an operator to its operand, a universal representation itself. The premise of each applicative combinatory categorical rule is the concatenation (noted by -) of linguistic units with categorial types. The consequence of each rule is a typed applicative expression with the possible introduction of a combinator.

[X/Y : u1] - [Y : u2] ------------------------< [X : (u1 u2)] Forward applicative rule	[Y : u1] - [X\Y : u2] ------------------------< [X : (u2 u1)] Applicative rule	[X : u] ------------------------**<T** [Y\(Y/X) : (**C*** u)] Type raising rule	[Y\Z : u1]-[X\Y : u2] ------------------------**<B** [X\Z : (**B** u2 u1)] Functional composition rule

Let us show, how we can give an account, by means of ACCG, of subject markers and the coordination of subject markers in Arabic. The major challenge is the addition of morphological information on gender and number to categories.

?ar-rijaal –u	jaa?	-uu	Jaa?	-at	al-banaat –u
----------------	------	------	-------	-------	----------------
N^s_{3pm}	S/N^s	$(S\backslash N^s_{3pm})\backslash(S/N^s)$	S/N^s	$(S/N^s_{3f})\backslash(S/N^s)$	N^s_{3pf}
	------------------------------------<		--<		
	$(S\backslash N^s_{3pm})$		S/N^s_{3f}		
---------------------------------------<			--->		
S			S		
	(a)			(b)	

Note that N^s (respectively N^o) is a class name acting as a subject (object, respectively). Category N^s_{3pm} typifies a noun acting as subject of 3rd person masculine plural. N^s_{3pf} typifies a noun acting as subject of 3rd person plural feminine. The sentence is syntactically correct because of the obtained type S. In the case of the examples (b) and (c) *-at* is considered as an operator which operates on the verb *xaraj* in order to construct a complex operator whose operand is a subject of the 3rd person and of feminine gender. In the case of the example (d) *-aa* is considered as an operator which operates on the verb *jaa?* in order to construct a complex operator whose operand is a subject of the 3rd couple person.

Xaraj -at al-banaat-u wa al-awlaad-u ------ ----- ------------- ----- -------------- S/N^s $(S/N^s_{3f})\backslash(S/N^s)$ N^s_{3fp} $(X\backslash X)/X$ N^s_{3mp} --------<T $S\backslash(S/N^s_{3fp})$ ---------------------------<B $S\backslash(S/N^s)$ -----------<T $S\backslash(S/N^s_{3mp})$ ---(X\X)/X $S\backslash(S/N^s_{3mp})$ ------------------------< S (c)	Jaa? -aa ?al-bint-u wa l-walad-u ------ ----- ------------- ----- -------------- S/N^s $(S/N^s_{3c})\backslash(S/N^s)$ N^s_{3fs} $(X\backslash X)/X$ N^s_{3ms} --------<T $S\backslash(S/N^s_{3fs})$ -----<T $S\backslash(S/N^s_{3ms})$ ---------------------------(X\X)/X $S\backslash(S/N^s_{3c})$ ---------------------------< S/N^s_{3c} ---------------------------------------< S (d)

4 Conclusion

We have shown that the behavior of subject markers changes according to whether they appear in a dislocation or a doubling configuration. In the first case, we observe a rich agreement. In the second case, the resulting agreement is impoverished. Thus we are in presence of a type of grammatical and non-anaphoric agreement, and the subject markers in this case are agreement markers, whether the NPs are coordinated or not.

An implementation by means of the applicative combinatory categorial grammar has been done. We have proved that in the general framework of categorial grammars it is possible to give an account for the analysis of subject markers in Arabic.

References

1. Aoun, N.J., Benmamoun, E., Sportiche, D.: Agreement, Word Order, and Conjugation in Some Varieties of Arabic. Linguistic Inquiry 25(2), 195–220 (1994)
2. Auger, J.: Pronominal Clitics in Québec Colloquial French: A Morphological Analysis. PhD dissertation, University of Pennsylvania, Philadelphia (1994)
3. Biskri, I., Desclés, J.P.: Coordination de Catégories Différentes en Français. Faits de Langue, N. 28 (2006)
4. Bresnan, J., Mchombo, S.A.: Topic, Pronoun and Agreement in Chichewa. Language 63(4), 741–782 (1987)
5. Jebali, A.: La modélisation des marqueurs d'Arguments de l'arabe standard dans le cadre des grammaires à base de contraintes. PhD dissertation. Université du Québec à Montréal, Canada (2009)

Word Reordering Approaches for Bangla-English Statistical Machine Translation[*]

Maxim Roy and Fred Popowich

School of Computing Science, Simon Fraser University
Burnaby, BC, Canada V5A1S6
maximr@cs.sfu.ca, popowich@cs.sfu.ca

Abstract. We apply several word reordering techniques to a Bangla-English Statistical Machine Translation (SMT) system. We evaluate the approaches through their impact on the BLEU score for the phrase-based Bangla-English SMT system. According to the experimental results, automatic reordering rules have the most significant impact on the BLEU score. We also provide a new test set with multiple references between Bangla and English for SMT evaluation purposes and evaluate the reordering approaches on the extended test set.

1 Introduction

While SMT systems have continued to improve in the quality of translations they provide, these systems are still challenged when providing translations between languages with different word orders. A range of reordering techniques can be used to overcome this challenge and improve translation accuracy. Reordering techniques can be applied to Bangla-English translation since Bangla grammar generally follows the Subject Object Verb(SOV) structure and English follows the Subject Verb Object(SVO) structure. While there has been research into reordering approaches that pre-process the source language input in SMT systems, there has not been a specific investigation of how these techniques can be applied to translation from Bangla to English. Crego and Mario [1] proposed an approach which extracted syntactic reordering rules from a parallel training corpus with a tagged source side. Rottmann and Vogel [5] also extracted rules from word alignments and source language POS tags for the Spanish-English and German-English language pairs.

Since SMT systems have relatively limited potential to model word-order differences, we apply an approach that attempts to modify the source language (e.g. Bangla) in such a way that its word order is very similar to that seen in the target language (e.g. English) based on the automatically learnt reordering rules from a part of speech (POS) tagged source language text. We also propose several manual rules based on POS tags to reorder the source sentences and experiment with lexicalized reordering implemented in Moses [2]. Finally, in order to evaluate the translation quality we provide a new test set with multiple reference translations between Bangla and English.

[*] This research was partially supported by a discovery grant from NSERC, Canada.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 282–285, 2010.

2 Reordering Techniques

Although statistical word alignments work rather well at capturing differences in word order and a number of strategies for non-monotonic search have been developed, differences in word order between the source and the target language are still one of the main causes of translation errors. Below we describe the automatic and manual reordering techniques.

2.1 Automatic Reordering Rules

The automatic reordering rules extraction approach that we used is similar to [1] which also requires POS tagged source text. The automatic reordering rules are learned from an aligned corpus, containing word-to-word alignments, for which the POS information of the source sentences is available. Given a sentence pair with source word $f_1^J$and target words e_1^I and the alignment a_1^J, a POS-based reordering rule is extracted whenever the alignment contains a crossing, i.e. whenever there is i and j with $i < j$ and $a_i > a_j$.

The next step is to calculate the frequencies of the rules. The frequencies are calculated by the number of times any particular rule is observed in the source side of the training data. Using the frequencies of the reordering rules we filter out rules that are observed less than 10 times in order to obtain only the most frequent rules. We choose the rules that were observed more than 10 times based on the experiments we conducted on different settings of the frequency of rules. The table 1 shows some of the most frequent reordering rules extracted from the training data and their frequency.

Table 1. Automatic Reordering rules

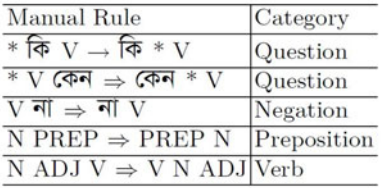
Automatic Rule	Freq.
N ADJ ⇒ ADJ N	256
N ADJ V ⇒ V N ADJ	85
NAME N ADJ V ⇒ V NAME N ADJ	70
ADJ N PREP⇒ PREP ADJ N	52

2.2 Manual Reordering Rules

Our Bangla language expert examined the Bangla sentences of the training data tagged with POS tags and manually proposed several reordering rules. The rules are divided into four categories to handle negation, question, prepositional and verb reordering in Bangla. Since English and Bangla differ in these four categories, these manual rules help reorder Bangla source sentences so the source and target sentences are closer to each other in word order. Below in table 2 we describe some of these manually extracted reordering rules. Some of the rules are a mix of words, POS tags and * to indicate arbitrary number of words or POS tags.

The main difference between the manual and automatic reordering rules is that the manual rules are linguistically motivated and created by a Bangla language expert, whereas automatic reordering rules are based on cross alignment

Table 2. Manual Reordering rules



Manual Rule	Category
* কি V → কি * V	Question
* V কেন ⇒ কেন * V	Question
V না ⇒ না V	Negation
N PREP ⇒ PREP N	Preposition
N ADJ V ⇒ V N ADJ	Verb

and might not always be correct. We have automatically extracted 6350 automatic reordering rules and after applying filtering we have 3120 rules along with 20 manual reordering rules. We also experimented with lexicalized reordering implemented in the Moses [2] system.

3 Dataset and Experimental Results

The corpus we used for training the system was provided by the Linguistic Data Consortium[1](LDC) containing approximately 11,000 sentences of newswire text taken from the BBC Asian Network and some other South Asian news websites. A bilingual Bangla-English dictionary collected from different websites was also used as part of the training set which contains around 55,000 words. For our language model we used data from the English section of EuroParl[2] combined with the LDC training set. The LDC development and test sets contain 600 and 1000 sentences respectively. The test set provided by the LDC contains only single reference translations between Bangla and English. We extended the LDC test set by adding two new English reference translation sets. The SMT system we used in our experiments is Moses [2].

We evaluated the translation quality of our SMT system on the three reordering approaches: lexicalized reordering, automatic reordering approach and manual reordering approach. We have two baselines, the first one (baseline-1) is trained on 11,000 sentences from the training corpora and the second baseline (baseline-2) is trained on 11,000 sentences and the bilingual dictionary. Below in table 3 we present the results of the reordering approaches evaluated on a single reference test set. We report the results in BLEU [4], PER(position independent word error rate) and WER(word error rate) score. Reordering the sentences using automatic reordering rules contributed the most improvement in our Bangla-English SMT system which achieves an improvement of 1.3 BLEU score over baseline-2. Manual reordering approach achieves higher BLEU score over lexicalized reordering and the baseline approach.

Below in table 4 we report the results of the reordering approaches evaluated on our newly extended reference test set. We only present the BLEU score here however the extended test set decreases the PER and WER score as well.

[1] LDC Catalog No.: LDC2008E29.

[2] Distributed for the shared task in the NAACL 2006 workshop on statistical machine translation (WSMT06).

Table 3. Impact of Reordering rules on SMT system

method	BLEU (%)	PER (%)	WER (%)
Baseline-1	7.2	65.6	84.7
Baseline-2	8.0	62.4	82.5
Lexicalized reordering	8.2	61.2	81.3
Manual reordering	8.4	61.9	81.8
Automatic reordering	9.3	60.1	80.5

Table 4. Impact of Reordering rules on SMT system using two and three test references

method	BLEU % (2 ref)	BLEU % (3 ref)
Baseline-1	10.3	12.2
Baseline-2	11.1	13.4
Lexicalized reordering	11.2	13.6
Manual reordering	11.5	13.7
Automatic reordering	12.3	15.1

4 Conclusion

We applied three reordering techniques to the source language in a Bangla-English SMT system and demonstrated that applying reordering approaches improves translation accuracy. We demonstrated that for low-density language like Bangla where sufficient bilingual data is not available, reordering approaches can improve translation quality. Also we contributed to the research community by building a better test set with three reference test sets for evaluation of translation quality between Bangla-English SMT systems. A richer reference test set permits better evaluation of SMT systems.

References

1. Crego, J.M., Marino, J.B.: Syntax-enhanced ngram- based smt. In: Proceedings of the 11th MT Summit (2007)
2. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open source toolkit for statistical machine translation. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics (2007)
3. Och, F.J.: Minimum Error Rate Training in Statistical Machine Translation. In: ACL 2003, pp. 160–167 (2003)
4. Papineni, K., Roukos, S., Ward, T., Zhu, W.: BLEU: A method for Automatic Evaluation of Machine Translation. In: Proceedings of the 20th Annual Meeting of the Association for Computational Linguistics (2002)
5. Rottmann, K., Vogel, S.: Word Reordering in Statistical Machine Translation with a POS-based Distortion Model. In: TMI 2007: 11th International Conference on Theoretical and Methodological Issues in MT, Skvde, Sweden (2007)
6. Stolcke, A.: SRILMan extensible language modeling toolkit. In: Hansen, J.H.L., Pellom, B. (eds.) Proc. ICSLP, September 2002, vol. 2, pp. 901–904 (2002)

Comparison of Feature Selection Methods for Sentiment Analysis

Chris Nicholls and Fei Song

Dept. of Computing and Information Science
University of Guelph
Guelph, Ontario, Canada N1G 2W1
{cnicholl,fsong}@uoguelph.ca

Abstract. Sentiment analysis is a sub-field of Natural Language Processing and involves automatically classifying input text according to the sentiment expressed in it. Sentiment analysis is similar to topical text classification but has a significant contextual difference that needs to be handled. Based on this observation we propose a new feature selection method called Document Frequency Difference to automatically identify the words which are more useful for classifying sentiment. We further compare it to three other feature selection methods and show that it can help improve sentiment classification performance.

Keywords: Sentiment Analysis, Feature Selection, Text Classification, Natural Language Processing, Maximum Entropy Modeling.

1 Introduction

Sentiment analysis (SA) is concerned with automatically classifying pieces of text according to the opinions expressed in them – positive or negative. With the growth of user generated content on the web, the need for SA has increased. Popular applications of SA are summarization of online customer reviews or social media monitoring and analytics software that help organizations manage their reputation.

SA is often viewed as a special case of topical text classification [1, 2] and machine learning techniques can be used to classify documents. However, there are contextual differences that need to be modeled in order for any SA solution to work effectively. An important difference is that people tend not to repeat the same sentiment-carrying words in the same context. For example, when people write reviews for a digital camera they are less likely to write something like "The camera is good. The LCD screen is good and it takes good pictures" but more likely to write something like "The camera is great. The LCD screen is clear and it takes stunning pictures".

In this paper, we are focused mainly on the representation of documents for classification. We explore different methods for feature selection that have had success for topical text classification and propose a new method specifically for SA.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 286–289, 2010.

2 Feature Selection

Feature Selection (FS) ranks all features based on a metric of how much they contribute to a class and removes all features below a specified threshold. The reduced set of features will not only improve the efficiency of the training and testing procedures, but also increase the classification performance since the least relevant features have been removed. In our study, we compare three existing FS methods: χ^2 [3], Optimal Orthogonal Centroid [4], and Count Difference [5]. We also propose a new metric based on our observations of polar (*positive* or *negative*) text.

Our proposed FS metric, called Document Frequency Difference (DFD), is calculated as follows:

$$score_t = \frac{|DF_+^t - DF_-^t|}{D}.$$

where DF_+^t is the number of documents in the *positive* class that term *t* occurs in, DF_-^t is the number of documents in the *negative* class that *t* occurs in and *D* is the number of documents in the training set.

An advantage of DFD is that the value it produces is normalized on a scale of 0 to 1. This means that scores will be proportional to other features, but furthermore, we feel that this makes the score suitable to be used as the feature weight itself. DFD will also not score rare terms highly. A drawback is that it requires an equal, or nearly equal, number of documents in each of the *positive* and *negative* classes. However, since we are only working with two classes, we feel that using the same number of documents in each class for training is necessary anyway.

3 Experiments and Results

3.1 Dataset and Evaluation

We use a publicly available dataset for our experiments. It is a set of movie reviews produced by Pang et al. [1, 2], which includes 1,000 positive and 1,000 negative review. This dataset can be seen as homogeneous since all of the reviews are from the movie domain despite genre differences.

To test our feature selection and weighting methods for SA, we implemented a classifier based on Maximum Entropy Modeling (MEM) as described in [6]. We prefer MEM to other machine learning methods because it has been shown to work well for topical text classification [5] and SA [1].

Since our dataset is relatively small, we employ cross-validation to make full use of them. The entire dataset is first partitioned into five folds. One fold is used as the validation set while the other four folds are joined and re-partitioned into ten-folds: nine for training and one for testing. So there are a total of 50 runs for each cross-validation process. The classification performance is computed on the validation set after every iteration of the IIS (Improved Iterative Scaling) algorithm for MEM. Training is terminated when all parameters have converged or a performance decrease is seen on the validation set.

Our baseline results are obtained by using all of the features seen in the training dataset. They are measured by P (precision), R (recall), and F (F-measure). For the positive class, P, R, and F are 0.791, 0.817, and 0.802, respectively, and for the negative class, P, R, and F are 0.811, 0.782, and 0.795, respectively. So the average-F for both classes is 0.799. Note that for both baseline and subsequent experiments, we filtered out all words that are not nouns, verbs, adjectives or adverbs along with the removal of a small list of 40 stop-words.

3.2 Feature Selection Results

To evaluate the performance of the FS ranking methods, we iterate over different feature cut-off thresholds and observe the performance for each one. Fig. 1 plots the results for the movie dataset. The best performance is achieved using the DFD metric with an average F of 0.851 with 900 or 1000 features. The CD metric achieves an average F of 0.848 with each of 800, 1000 and 1200 features. We perform the one-tailed two-sample z-test on these results and find that the performance increase over the baseline (0.848 or 0.851 vs 0.799) is statistically significant with over 99% certainty.

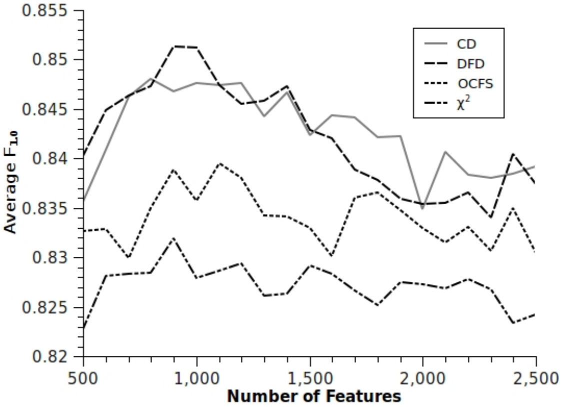


Fig. 1. Plot of feature selection results using the movie reviews dataset. The feature cut-offs in the range of 100-2,500 are plotted on the horizontal axis and the average F on the vertical axis.

3.3 Feature Score Weighting Results

In an attempt to weigh terms on their relevance, we further tried using the actual *score* computed by each FS method as the feature weight. The MEM feature function thus takes the form[7]:

$$f_{i,c'}(d,c) = \begin{cases} \frac{score_i}{\sum_{j\in d} score_j}, & c = c' \\ 0, & \text{otherwise} \end{cases}.$$

The average-F for DFD, CD, OCFS, and ÷^{2} are 0.869, 0.862, 0.809, and 0.857, respectively. All of them are better than the baseline of 0.799 and interestingly, using the DFD and CD scores in this fashion further increases classification performance beyond that achieved using them for feature selection. We did try to couple the feature score weights with feature selection, but the performance actually decreased slightly. We believe this is because we are already getting the most information that we can get out of the feature ranking metrics by using them as the scores.

4 Conclusions

We compared four FS methods and showed that they can help the classification performance of SA. The CD metric and our proposed DFD metric, both of which focus on the document frequency of a feature, have been shown to work well for SA. This matches our observation that for SA, frequently occurring terms in a particular review are not necessarily more helpful as is the case for topical text classification. This is further evidenced by the fact that when using the DFD score as the value of the feature function, performance was increased even higher. Although the movie dataset is about one domain, the automatic FS methods we have experimented with can be easily adapted to new domains.

References

1. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment classification using machine learning techniques. In: Proceedings of the ACL 2002 Conference on Empirical Methods in Natural Language Processing, vol. 10, pp. 79–86 (2002)
2. Pang, B., Lee, L.: A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (2004)
3. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: Machine Learning, pp. 412–420 (1997)
4. Yan, J., Liu, N., Zhang, B., Yan, S., Chen, Z., Cheng, Q., Fan, W., Ma, W.Y.: OCFS: Optimal orthogonal centroid feature selection for text categorization. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 122–129 (2005)
5. Cai, J., Song, F.: Maximum Entropy Modeling with Feature Selection for Text Categorization. In: Li, H., Liu, T., Ma, W.-Y., Sakai, T., Wong, K.-F., Zhou, G. (eds.) AIRS 2008. LNCS, vol. 4993, pp. 549–554. Springer, Heidelberg (2008)
6. Nigam, K., Lafferty, J., McCallum, A.: Using maximum entropy for text classification. In: IJCAI 1999 Workshop on Machine Learning for Information Filtering (1999)
7. Nicholls, C., Song, F.: Improving sentiment analysis with part-of-speech weighting. In: Proceedings of the International Conference on Machine Learning and Cybernetics (2009)

Unsupervised Extraction of Appraisal Expressions

Kenneth Bloom and Shlomo Argamon

Illinois Institute of Technology, Chicago, IL 60616, USA
kbloom1@iit.edu, argamon@iit.edu

1 Introduction

The goal of sentiment analysis is to characterize texts in terms of the opinions and evaluations they express. As such, a wide variety of different tasks have been addressed in the field. However, there is not yet a clear consensus on how to formalize the notion of "sentiment" or "subjective language". The most commonly studied kind of subjective language in sentimant analysis is *evaluative language*, that which gives a positive or negative evaluation of some target. (Although *positioning language*, which relates the position of one opinion holder with respect to those of other opinion holders and *intentional language* and some aspects of modality have also been included.)

Most research in sentiment analysis has focused on application-level tasks such as positive/negative text classification or opinion summarization, rather than on on developing precise methods for dealing with primitive linguistic representations of sentiment, which makes it nearly impossible to determine what factors explain the success of (e.g.) a sentiment-based text classification system—the underlying appraisal extraction methods, the semantic formalism, the classification learning algorithm, or something else?

We therefore study evaluative language independently, evaluating directly on a corpus tagged to indicate appraisal expressions. This requires identifying and classifying the components of each specific appraisal expression, which is more difficult than the usual task of classifying texts as positive or negative.

An appraisal expression is the fundamental textual unit for expressing an evaluation. An *appraisal expression* has three main parts: a *source* denoting the individual making an evaluation, an *attitude* denoting the nature of the evaluation, and a *target* denoting what has been evaluated. Sometimes other components may be present, such as a second target when the attitude is comparative (e.g. 'better'), or a phrase denoting which aspect of the target is being evaluated. Our focus here is on extracting and characterizing attitudes and their targets where they co-occur in the same sentence (which is usually the case [4]).

2 System Operation

FLAG extracts appraisal expressions in two main steps: (a) extracting attitude groups, and then (b) finding targets associated with these attitude groups.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 290–294, 2010.

First FLAG finds *attitude groups* using a lexicon-based shallow parser, which also determines the values of attributes (e.g. the *orientation*) characterizing the attitude. FLAG find phrases such as “not very happy”, “somewhat excited”, “more sophisticated”, or “not a major headache” which convey an evaluative stance, using a method from our previous work [6]. This method uses a lexicon of nominal and adjectival appraisal head words and phrases, giving initial attribute values for each attitude. Given occurrences of head words in the text, FLAG looks leftward to attach modifiers to the words, updating values of the attributes.

To find a target for each attitude, we assume that such targets are syntactically related to extracted attitudes, and that targets are represented by noun phrases. FLAG finds target phrases by following paths through a dependency parse of each sentence containing an attitude. This is done via *linkage specifications*, each a possible dependency path defining an sequence of syntactic relations connecting an attitude and target. For each attitude it finds, the system looks for a linkage matching some linkage specification which connects a word in the attitude group to a word that can be the head of a target phrase. Upon finding such a target word, shallow parsing is used to identify a compact phrase naming the target. If multiple linkages are found for a given attitude, FLAG prefers linkages that connect to likely candidate targets over those that do not. If none do, or if multiple linkages connect to candidate targets, FLAG selects the linkage from the highest-ranked linkage specification.

Both the target finder and the linkage learner require *likely candidate targets* independent of syntactic position. Our earlier work [2,1] chose likely targets using a hand-built lexicon of common target entities for each domain. In this study, we automatically identified likely targets, choosing any noun or adjective appearing in at least 1% of the sentences for a given product/topic. (For MPQA, which does not have topics, we used Lemur to cluster documents into 10 clusters).

Constructing good specifications by hand is difficult. Therefore, we find linkage specifications automatically from an untagged corpus, (extending [1]). Starting with lists of attitudes and likely candidate targets, FLAG finds all syntactic paths that connect attitude/candidate-target pairs in the same sentence.

Each possible linkage specification is assigned a “goodness score;” the 50 linkage specifications are kept. Higher scores should be given to frequent paths, though lower scores should be given those with lower confidence; favoring more general linkage specifications but penalizing links to many unlikely targets. FLAG computes two counts for each possible specification: B, the number of times it links an attitude group with a candidate target, and T, the number of times the specification links an attitude to text not in a candidate target. We compute the goodness score using the “log-n ranking metric”: $B[p]/\big(\ln(B[p] + T[p])\big)^n$. The single parameter n is tuned for different corpora sizes (a value of 3.5 works well for corpora with several hundred documents).

3 Evaluation Metrics

We evaluated different configurations of FLAG against the UIC Review Corpus [3] and the MPQA 2.0 corpus [7]. The goal of evaluation on MPQA was to determine how effectively FLAG could identify attitudes, targets, and full appraisal expressions with the attitude and target correctly matched. For the UIC Corpus, where only product features are annotated, we could only check how well FLAG identified targets and their orientations in context.

For MPQA, we evaluated finding attitudes, finding targets independent of their attitudes, and extracting full appraisal expressions (attitudes and their targets) for corpus annotations of types sentiment and arguing (Table 2). Because the MPQA corpus systematically contains much longer attitudes and targets than FLAG extracts, we considered an extracted attitude or target to be correct if it was a substring of the tagged ground truth.

In the UIC Review Corpus, only targets are annotated, with shorter and more focused phrases chosen. Hence, we evaluated FLAG via multiple match criteria, including exact matching as well as allowing ground truth to be a subsequence of found targets and vice versa (Table 1). Attitude groups are often tagged in UIC as features, so we also evaluated performance matching ground truth targets to both attitudes and targets found by FLAG.

We further compared results for our hand-built lexicons with results of using that of Turney and Littman [5], in which orientations of words in the General Inquirer were automatically determined from web-based statistics. We also evaluated different linkage specification lists, including the manual linkage specification list and lists learned using the Log-n ranking metric, for the n performing best on the development subset for each corpus. The baseline method accepted

Table 1. Results on the UIC Review Corpus, microaveraged over 13 products. Matching criteria: Exact Match, GT in FLAG (ground truth substring of FLAG's target), FLAG in GT (the reverse), and With Attitudes (attitudes match as targets). Base$_0$ is the baseline of using all candidate targets, Base$_{Att}$ the baseline of using all candidate targets in a sentence with an extracted attitude.

Linkage	**Exact Match**			**GT in FLAG**			**FLAG in GT**			**With Attitudes**			**Ori**
Specs	Prec	Rec	F_1	Prec	Rec	F_1	Prec	Rec	F_1	Prec	Rec	F_1	%
No lexicon — just use all candidate targets													
Base$_0$	0.07	0.37	0.12	0.08	0.41	0.13	0.11	0.56	0.18	—	—	—	—
FLAG lexicon													
Manual	0.13	0.15	**0.14**	0.20	0.23	**0.21**	0.15	0.18	0.16	0.20	**0.25**	0.22	**85%**
Log-3.5	**0.14**	0.14	**0.14**	**0.21**	0.22	**0.21**	**0.16**	0.17	0.16	**0.22**	0.23	0.22	**85%**
Base$_{Att}$	0.08	**0.27**	0.12	0.09	**0.30**	0.14	0.13	**0.40**	**0.19**	—	—	—	—
Turney's lexicon													
Manual	0.13	0.14	0.13	0.19	0.21	0.20	0.15	0.17	0.16	0.21	**0.23**	**0.22**	84%
Log-4.5	**0.15**	0.13	**0.14**	**0.23**	0.19	**0.21**	**0.18**	0.15	0.17	**0.25**	0.21	**0.22**	83%
Base$_{Att}$	0.08	**0.23**	0.12	0.09	**0.26**	0.13	0.13	**0.36**	**0.19**	—	—	—	—

Table 2. Results for the MPQA corpus, as above; see text for more details

Linkage	**Targets**			**Ori**	**Attitudes**			**Expressions**			**Ori**
Specs	Prec	Rec	F_1	%	Prec	Rec	F_1	Prec	Rec	F_1	%
No lexicon — just use all candidate targets											
$Base_0$	0.09	0.58	0.15	—	—	—	—	—	—	—	—
FLAG lexicon											
Manual	0.14	0.17	0.15	73%	0.27	0.32	0.29	**0.06**	**0.07**	0.06	**80%**
Log-3.5	**0.14**	0.13	0.14	**76%**	0.27	0.32	0.29	0.05	0.06	0.06	78%
$Base_{Att}$	0.11	**0.37**	**0.17**	—	—	—	—	—	—	—	—
Turney's lexicon											
Manual	0.14	0.14	0.14	71%	**0.27**	0.25	0.25	0.05	0.05	0.05	76%
Log-3.5	**0.15**	0.12	0.13	71%	0.26	0.25	0.25	0.05	0.05	0.05	76%
$Base_{Att}$	0.10	**0.30**	**0.15**	—	—	—	—	—	—	—	—

all likely candidate targets as targets. A variant baseline only accepted targets in the same sentence as an extracted attitude.

4 Results

Best overall performance at target finding on the UIC corpus was achieved by using FLAG's lexicon with the linkage specifications learned using the Log-3.5 metric. Allowing attitude groups to be considered as targets did improve accuracy slightly. Orientations of extracted targets were quite good, at 85%, though as expected not approaching accuracies for orientation classification of longer (hence more informative) texts. There was little difference in performance between using the manual lexicon and Turney's lexicon to find attitudes.

While FLAG had noticeably better precision than baseline on the MPQA corpus, recall was much lower, and therefore so was F_1. This was due to errors and incompleteness on the part of FLAG as well as issues with the annotation of the corpus. Examination of a random sample of both false-positive and false-negative errors showed six main categories of errors: (i) FLAG finding spurious appraisal expressions, (ii) FLAG finding the wrong target for a true attitude, by following the wrong linkage, (iii) appraisal expressed through verbs (not yet handled by FLAG), (iv) appraisal in the corpus tagged correctly but that is evoked indirectly through figurative language (hence much harder to detect), (v) non-sentiment attitudes labeled as sentiment in the corpus (thereby testing FLAG against expression types it is not designed to handle), and (vi) some erroneous annotations in the corpus.

References

1. Bloom, K., Argamon, S.: Automated learning of appraisal extraction patterns. In: Gries, S.T., Wulff, S., Davies, M. (eds.) Corpus Linguistic Applications: Current Studies, New Directions. Rodopi, Amsterdam (2009)

2. Bloom, K., Garg, N., Argamon, S.: Extracting appraisal expressions. In: Proceedings of HLT/NAACL (2007)
3. Ding, X., Liu, B., Yu, P.S.: A holistic lexicon-based approach to opinion mining. In: Najork, M., Broder, A.Z., Chakrabarti, S. (eds.) First ACM International Conference on Web Search and Data Mining (WSDM), pp. 231–240. ACM, New York (2008)
4. Hunston, S., Sinclair, J.: A local grammar of evaluation. In: Hunston, S., Thompson, G. (eds.) Evaluation in Text: authorial stance and the construction of discourse, pp. 74–101. Oxford University Press, Oxford (2000)
5. Turney, P.D., Littman, M.L.: Measuring praise and criticism: Inference of semantic orientation from association. ACM Trans. Inf. Syst. 21(4), 315–346 (2003)
6. Whitelaw, C., Garg, N., Argamon, S.: Using appraisal taxonomies for sentiment analysis. In: SIGIR (2005)
7. Wilson, T.A.: Fine-grained Subjectivity and Sentiment Analysis: Recognizing the Intensity, Polarity, and Attitudes of Private States. Ph.D. thesis, University of Pittsburgh (2008)

Automatic Text Segmentation for Movie Subtitles

Martin Scaiano, Diana Inkpen, Robert Laganière, and Adele Reinhartz

University of Ottawa
mscai056@uottawa.ca, {diana,laganier}@site.uottawa.ca,
areinhar@uottawa.ca

Abstract. To improve information retrieval from films we attempt to segment movies into scenes using the subtitles. Film subtitles differ significantly in nature from other texts; we describe some of the challenges of working with movie subtitles. We test a few modifications to the TextTiling algorithm, in order to get an effective segmentation.

Keywords: TextTiling, Segmentation, Subtitles, Film.

1 Introduction

This research is a part of a larger video information retrieval project. Our work in segmenting movies based on subtitles will later be combined with analysis of visual information. The segments should provide good units that can be returned in response to information retrieval queries.

Typical segmentation data is expository, where communicating information efficiently and effectively is key; thus, expository text often has a high frequency of meaningful content words. Movie dialog tends to have much less frequent content words; also with characters speaking back and forth, many of the sentences are short.

Our work has focused on a limited selection of films; we manually segmented *Shawshank Redemption* and *3:10 to Yuma* into scenes, in order to have a gold standard, for evaluation purposes. The first movie was used for development; the second one only for testing. We used the strict scene definition from [1].

2 TextTiling

TextTiling is an algorithm for text segmentation developed by Marti Hearst [2]. The method considers all possible segment boundaries (usually between sentences), and evaluates how a window of words preceding the segment boundary correlates with a window of words following the segment boundary. This creates a graph representing the correlation over positions; as the correlation increases, we are likely in the middle of a segment; as it decreases and approaches a valley, we are likely reaching the segment's end. We use this method as a starting point and enhance certain steps.

Lexical chains [3,4,5] are a popular and effective alternative to TextTiling, but we expected they may not meet some of our long term requirements.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 295–298, 2010.

3 Segmentation Data

Movie Subtitles

We had to create our own segmentation standard for our test films. Human judges familiar with the project were given our strict scene definition [1]; they each watched one of the two movies mentioned in the introduction, marking the time in seconds when a new scene begins.

The definition of a scene is fairly objective and can be summarized as follows: a scene lasts until the location or the time changes, with a few exceptions. Even with this fairly objective definition, there were still inconsistencies in marking scenes arising from subjectively determining the scope of references to locations and of temporal references.

The number of chapters on DVD is often less than the number of scenes our scene definition produces: *Shawshank Redemption* contains 123 scenes with an average length of 65 seconds and *3:10 to Yuma* contains 30 scenes with an average length of 3 minutes and 54 seconds.

Physics and AI lectures

We also evaluated our methods on the Physics and AI lectures used to evaluate the Minimum Cut Model for text segmentation from [6]. This data consists in transcriptions of spoken lectures and on average contains between 500 and 700 sentences. Segments roughly correspond to what was spoken during one PowerPoint slide. While this data is in fact generated from speech, similar to subtitles, it is mostly expository and does not contain casual dialog between multiple parties.

DUC 2002 Texts

This data consists in 10 documents from the DUC 2002 conferences (AP880911, AP891018, AP890922, AP880314, AP880817, AP890323, FT9235589, AP900621, AP890925, AP900103). Each text was manually split into segments by a professional linguist. Each text contains between 10 and 30 sentences. Segments vary in length from 1 sentence to 5 sentences. This data is also expository and comes from a variety of news sources. The best segmentation result we know of for this data set uses lexical chains and produces a WindowDiff value of 0.47.

4 Method

We started with a basic TextTiling algorithm with a cosine comparison as the correlation measure between windows. This provided a baseline method for comparison. We enhanced the cosine similarity with a WordNet-based method [7], that we expect to work more effectively on our sparse data.

We used a vector of synsets instead of a vector of words with the cosine similarity. If a word has several senses (associated synsets in WordNet), it contributes a proportional weight to each synset. For example, if a word has four senses, it contributes 0.25 weight to each synset. This saves us the need to do word sense disambiguation, which is usually time-consuming and prone to errors. Furthermore, to increase overlap for sparse data, we can also iteratively add any synsets related to the ones in the vectors. In our experiments, we iteratively added relatives three times.

The ideal shape of the correlation graph would be something sinoidal, where the peaks are the centres of the segments and the valleys are changes of segments. In practice, many of the graphs are far too noisy (with many local minima and maxima) for simple methods.

We propose two methods for reducing the noise in the correlation graph: using a local average of points, and using a minimum difference threshold between peaks and valleys. If the difference between a peak and valley is greater than the threshold, we consider this to be a valid segment. A factor F will provide a context-aware threshold, in the form of the following equation (note that the factor should be greater than 1):

*Minimum Correlation Peak Threshold = Valley Correlation Value * F*

We tested two methods to specify segment boundaries: the lowest correlation value in a valley and the centre of the valley calculated based on the area over the curve (AOC); we are effectively finding the centre of mass for the valley. The AOC method has the advantage that it selects a particular second as the boundary, while the lowest correlation value must generically select an inter-subtitle time period.

5 Evaluation

We use WindowDiff measure [9] to evaluate our results. The WindowDiff algorithm is designed to evaluate segmentation, while fairly rewarding or penalizing slight misplacements or shifts of the segment boundary. The method works by comparing the number of expected segment boundaries to the number of experimental segment boundaries in a sliding window. Note that for subtitles we consider each second as a position and not the usual inter-sentence positions. Determination of the exact position of scene boundaries is a visual process. Slight misalignments (shifted from the expected position) can be corrected later through the addition of visual techniques.

Table 1. Results for the Cosine and the WordNet similarity measure, with and without local averaging of 3 points, and with thresholding using the WordNet similarity measure

Text	Cos	Cos + Avg	WN	WN + Avg	Threshold + WN
Shawshank	1.18	0.78	1.2	0.75	0.50
3:10 to Yuma	2.34	1.32	2.4	1.13	0.57
AI Lect	112.26	112.26	4.61	4.61	4.59
Physics Lect	302.94	302.94	271.66	271.66	26.32
DUC	4.21	3.97	107.87	107.87	13.63

Table 1 shows that local averaging is indeed a good method for reducing noise, which is useful for datasets with many sentences in each segment. The effectiveness of the WordNet similarity method is not clearly determined, though when coupled with averaging it seems to show improvement. Due to space limitations we only present the results for the optimal parameters.

Using a threshold generally shows the best results, except on extremely short segments, and can be used to control the balance between retrieving correct boundaries and missing some boundaries. Note that as the threshold increases above 1, an optimal point is reached where increasing the threshold increases the correctness but misses to many segment boundaries.

The area over curve method for selection of scene boundary consistently shows equivalent or better results than the lowest correlation value method (as we noticed in additional experiments not shown here), but the improvements are small, which is to be expected when only shifting the boundaries slightly.

6 Conclusions

We find working with movie subtitles an interesting new challenge. The nature of data is very different than the data used in most segmentation and NLP tasks. Our methods improved the results of TextTiling in our domain, though when applied to other domains our methods are not as effective.

In future, we hope to develop an objective definition for topical segmentation of movies. We will also be combining our results with visual analysis to determine scene boundaries. Finally, future work will include investigation into how the unique nature of dialog in movies can be leveraged to assist in NLP tasks.

Acknowledgments. We thank Emma Tamaianu-Morita for the manual segmentation of the 10 texts from DUC 2002, and Doina Tatar for sharing this data with us. We thank our human judges that helped producing the manual segmentation for the movies: Andrew Thompson and Jason Shim.

References

1. Truong, B.T., Dorai, C., Venkatesh, S.: Automatic Scene Extraction in Motion Pictures. Technical Report 1/2001, School of Computing, Curtin University of Technology, Perth, Western Australia (2001)
2. Hearst, M.A.: Multi-Paragraph Segmentation of Expository Text. In: 32nd Annual meeting on Conference on ACL, pp. 9–16 (1994)
3. Manabu, O., Takeo, H.: Word sense disambiguation and text segmentation based on lexical cohesion. In: 15th Conference on Computational Linguistics (1994)
4. Jarmasz, M., Szpakowicz, S.: Not as Easy as It Seems: Automating the Construction of Lexical Chains Using Roget's Thesaurus. LNCS. Springer, Heidelberg (2003)
5. Tatar, D., Tamaianu-Morita, E., Czibula, G.: Segmenting Text By Lexical Chains Distribution. In: KEPT 2009 (2009)
6. Malioutov, I., Barzilay, R.: Minimum Cut Model for Spoken Lecture Segmentation. In: 21st International Conference on Computational Linguistics, pp. 25–32 (2006)
7. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
8. Pevzner, L., Hearst, M.A.: A Critique and Improvement of an Evaluation Metric for Text Segmentation. Computational Linguistics 28(1), 19–36 (2002)

Using Classifier Performance Visualization to Improve Collective Ranking Techniques for Biomedical Abstracts Classification

Alexandre Kouznetsov[1] and Nathalie Japkowicz[2]

[1] Department of Computer Science and Applied Statistics,
University of New Brunswick Saint John
[2] School of Information Technology and Engineering, University of Ottawa

Abstract. The purpose of this work is to improve on the selection of algorithms for classifier committees applied to reducing the workload of human experts in building systematic reviews used in evidence-based medicine. We focus on clustering pre-selected classifiers based on a multi-measure prediction performance evaluation expressed in terms of a projection from a high-dimensional space to a visualizable two-dimensional one. The best classifier was selected from each cluster and included in the committee. We applied the committee of classifiers to rank biomedical abstracts based on the predicted relevance to the topic under review. We identified a subset of abstracts that represents the bottom of the ranked list (predicted as irrelevant). We used False Negatives (relevant articles mistakenly ranked at the bottom) as a final performance measure. Our early experiments demonstrate that the classifier committee built using our new approach outperformed committees of classifiers arbitrary created from the same list of pre-selected classifiers.

Keywords: Machine Learning, Automatic Text Classification, Systematic Reviews, Ranking Algorithms, Scientific Visualization.

1 Introduction

Evidence-based medicine (EBM) is an approach to medical research and practice that attempts to provide better care with better outcomes by basing clinical decisions on solid scientific evidence [1]. Systematic Reviews (SR) are one of the main tools of EBM. Building SRs is a process of reviewing literature on a specific topic with the goal of distilling a targeted subset of data. Usually, the reviewed data includes titles and abstracts of biomedical articles that could be relevant to the topic. SR can be seen as a text classification problem with two classes: a positive class containing articles relevant to the topic of review and a negative class for articles that are not relevant.

Previous work by Kouznetsov et al. [2] proposed an algorithm to reduce the workload of building SRs while maintaining the required performance of the existing manual workflow.

Since the approach in [2] is based on using a committee of classifiers to rank biomedical abstracts based on the predicted relevance to the review topic, selecting

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 299–303, 2010.

the right classifiers to be included in the committee could be a key feature in improving prediction performance.

In this work we propose an approach to selecting classifiers based on a Projection-Based Framework for Classifier Performance Evaluation with respect to Multiple Metrics and Multiple Domains [3], [4], [5].

2 Method

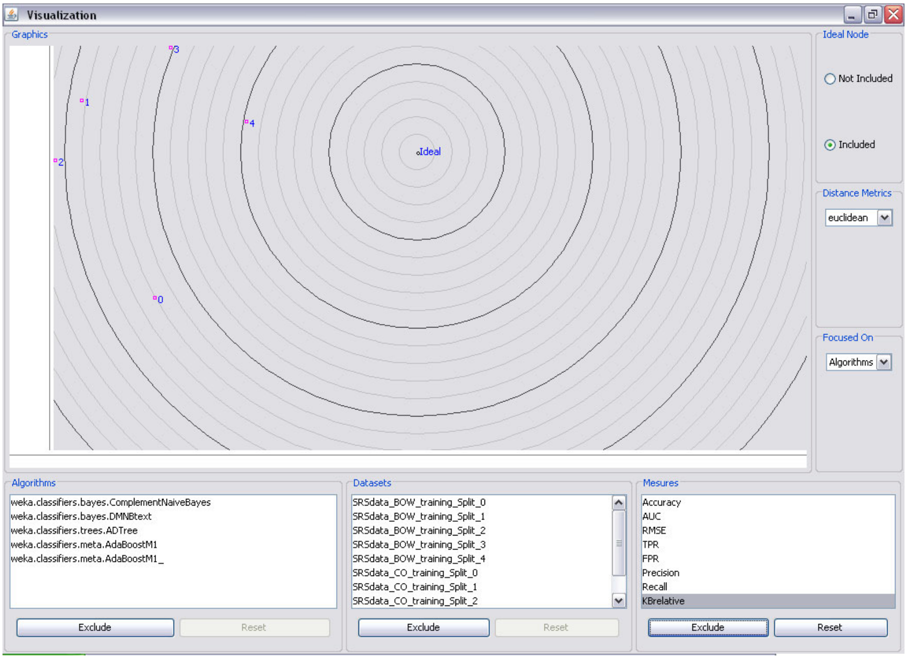
Ranking Method. We used the Ranking Algorithm, presented in the work of Kouznetsov et al. [2]. This approach is based on using committees of classification algorithms to rank instances based on their relevance to the topic of the review. It is a two-step ranking algorithm. While the first step, called local ranking, is used to rank instances based on a single classifier output, the second step, named collective ranking, integrates the local ranking results of individual classifiers and sorts instances based on all local ranking results. Finally, we get the collective rank which is assigned to each article in the test set. An instance with a higher collective rank is more relevant to the topic under review than another instance with a lower collective rank.

The classification decision of the committee is based on the collective ordered list of instances. The work in [2] provides ML techniques to establish the bottom threshold (number of instances at the bottom to be classified as negative with respect to the required level of prediction confidence). The articles with rank below the bottom threshold are predicted as not relevant and excluded from the review (in an attempt to reduce workload).

Visualizing method. Applying a ranking method assumes having a committee of classifiers that first needs to be selected. We propose a method for doing so that uses a **Visualizing Classifier Performance Tool (VCPT)** that was previously introduced in [3], [4], [5] and integrated with WEKA [7] in the context of this study. VCPT [9] includes our modification of the WEKA package to extend the functionality of its Experimenter Module integrated with the R Statistical package.

VCTL implements the following pipeline: All the classifiers involved in the study are run on all the data sets considered. The performance measures associated with one classifier on each dataset are organized into a single vector (any involved data set as well as any involved measure would produce a new dimension). The *Multidimensional Scaling* MDS [8] projection method (with Euclidean distance measure) is used to project high dimensional vectors into two dimensional vectors. Finally, projected vectors are visually presented on the VCTL graphical panel and users can visually separate classifiers into a few different clusters based on the performance achieved by each algorithm. We consider as a possibly good combination one where the best classifier is selected from each cluster and included in the team.

The visualization method we applied is a special case of the method proposed in the work of Japkowicz et al [5]. The purpose of this approach is to summarize the results obtained by classifiers on different data sets, using various performance measures. However, rather than summarizing these results numerically, we do so visually, exploiting the human visual skill, in the hope of retaining more information than we would by using a numerical summary.



Ideal Node

○ Not Included

● Included

Distance Metrics: euclidean

Focused On: Algorithms

Algorithms

weka.classifiers.bayes.ComplementNaiveBayes
weka.classifiers.bayes.DMNBtext
weka.classifiers.trees.ADTree
weka.classifiers.meta.AdaBoostM1
weka.classifiers.meta.AdaBoostM1_

Datasets

SRSdata_BOW_training_Split_0
SRSdata_BOW_training_Split_1
SRSdata_BOW_training_Split_2
SRSdata_BOW_training_Split_3
SRSdata_BOW_training_Split_4
SRSdata_CO_training_Split_0
SRSdata_CO_training_Split_1
SRSdata_CO_training_Split_2

Mesures

Accuracy
AUC
RMSE
TPR
FPR
Precision
Recall
KBrelative

Fig. 1. Visualizing Classifiers Performance with VCPT

3 Experiments

The work presented here was done on the same data that was used in [2]. The source data includes 23,334 medical articles pre-selected for the review topic "Assessment of health care delivery and outcomes in children and youth with special needs". The data contained only article titles and abstracts. While 19,637 articles have a title and an abstract, 3,697 articles have only a title. The data is available at http://www.site.uottawa.ca/~stan/public/

A stratified repeated random sampling scheme was applied to validate the experimental results. The data was randomly split into a training set and a testing set five times. On each split, the training set included 7,000 articles (626 positive and 6,374 negative) or about 30% of the entire data set, while the testing set included 16,334 articles (1,461 positive and 14,873 negative) or about 70% of the entire data set.

We applied two data representation schemes to build document-term matrices: the Bag-of- words (BOW) and the Second Order Soft Co-Occurrences (SOSCO) [6] representations.

The five classifiers included in the Classifier Committee used in [2] are considered as our pre-selected short list[1], namely: (0) Complement Naïve Bayes (CNB); (1) Discriminative Multinomial Naïve Bayes (DMNB); (2) Alternating Decision Tree (ADT); (3) AdaBoost [Logistic Regression] (AB/LR); (4) AdaBoost [j48] (AB/J48)];

[1] We are using this list just to illustrate the concept of building a classification team over a pre-selected set. In a real world setting, we would expect a wider list of algorithms, such as 10-20 algorithms.

Visualizing Classifier Performance Experiments. We ran classifiers on both the BOW data representation and the SOSCO data representation. Therefore we have 10 data sets (5 BOW data sets+5 SOSCO data sets). Since we used the 10 fold cross validation scheme each 7000-article set was randomly split into a new training set (6,300 instances) and a new testing set (700 instances), following a stratified strategy. The following eight performance evaluation metrics were included in each experiment: Accuracy, AUC, RMSE, True Positive Rate, False Positive rate, Precision, Recall, Kononenko & Bratko Relative Information score.

Committee Validation Experiments. The objective of these experiments is to compare the performance of committees of classifiers selected with VCPT against the performance of other possible committees. In these experiments, we are using both the training sets (7,000 articles on each split) and the testing sets (16,334 articles on each split). We consider the bottom threshold as 4000 and run the ranking algorithm to predict 4000 not relevant (negative) articles on the testing set. The paired two tailed t-test was applied to validate the statistical significance of the obtained results.

4 Results

The VCPT visual panel output is presented in Figure 1. We visually observed the obtained output and divided the classifiers into 3 clusters. The clusters are presented in Table 1.

We selected the following classifiers: Complement Naïve Bayes, Discriminative Multinomial Naïve Bayes, Ada Boosted J48. We call this committee the 0-1-4 team.

In order to evaluate our approach, we built two validation teams that include the same number of members: (1) Validation team 2-3-4 includes: Alternating Tree, Ada Boosted Logistic Regression, Ada Boosted J48; (2) Validation team 1-2-3 includes: Discriminative Multinomial Naïve Bayes, Alternating Tree,Ada Boosted Logistic Regression.

We built the validation committees based on a different approach from the one that yielded committee 0-1-4. While committee 0-1-4 includes only one classifier from every cluster, both committees 2-3-4 and 1-2-3 include two classifiers taken from the same cluster. These validation committees also used the best classifiers in order to maximize their performance.

Table 2 demonstrates the performance obtained with the ranking method by each classification committee. The performance measure is the number of False Negatives (FN) (taken as averages over 5 splits) that occurred for each classification committee over 4000 bottom ranked articles. False Negatives mean articles that are relevant to the SR topic but mistakenly ranked at the bottom of the ranking list.

The results we obtained have demonstrated that the committee completed with the proposed approach (0-1-4 team) outperformed both validation committees on the negative tail, namely the average FN of Team 0-1-4's output is around 48% less than the average FN of Team 2-3-4 (FN 11.2 vs. FN 21.4) and the average FN of Team 0-1-4's output is around 45% less than the average FN of Team 1-2-3 (FN 11.2 vs. FN 20.2). The applied t-test demonstrates that the achieved differences are statistically significant.

Table 1. Clusters visually built from VCPT outputs

Cluster #1	Cluster #2	Cluster #3
(3) AB/LR, (4) AB/J48	(1) DMNB, (2) ADT	(0) CNB

Table 2. False Negatives on Negative prediction zone (means over 5 folds)

Team 0-1-4	Team 2-3-4	Team 1-2-3
11.2	21.4	20.2

5 Conclusion and Future work

Our results demonstrate that using our approach can improve on the performance of classifier committees employed on Systematic Reviews. Based on our early experiments' output, the classifier committee formed by applying the projection method of classifier evaluation significantly overperformed the validation committees that consist of the same number of algorithms arbitrary included from the same list of pre-selected classifiers.

As a possible topic for future work we are planning more experimentation on different SR data sets and larger numbers of classifiers, to test whether our method can scale up.

References

1. Sackett, D., Rosenberg, W., Gray, J., Haynes, R., Richardson, W.: Evidence based medicine: what it is and what it isn't. BMJ 312 (7023): 71-2. PMID 8555924 (1996)
2. Kouznetsov, A., Matwin, S., Inkpen, D., Razavi, A., Frunza, O., Sehatkar, M., Seaward, L., O'Blenis, P.: Classifying Biomedical Abstracts Using Committees of Classifiers and Collective Ranking Techniques. In: Canadian Artificial Intelligence Conference (2009)
3. Alaiz-Rodriguez, R., Japkowicz, N., Tischer, P.: Visualizing Classifier Performance. In: Proceedings of the 20th IEEE International Conference on Tools for Artificial Intelligence, ICTAI 2008 (2008)
4. Alaiz-Rodriguez, R., Japkowicz, N., Tischer, P.: A Visualization-Based Exploratory Tool for Classifier Comparison with respect to Multiple Metrics and Multiple Domains. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part II. LNCS (LNAI), vol. 5212, pp. 660–665. Springer, Heidelberg (2008)
5. Japkowicz, N., Sanghi, P., Tischer, P.: A Projection-Based Framework for Classifier Performance Evaluation. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part I. LNCS (LNAI), vol. 5211, pp. 548–563. Springer, Heidelberg (2008)
6. Razavi, A.H., Matwin, S., Inkpen, D., Kouznetsov, A.: Parameterized Contrast in Second Order Soft Co-Occurrences: A Novel Text Representation Technique in Text Mining and Knowledge Extraction. In: Second International Workshop on Semantic Aspects in Data Mining (SADM 2009), USA, Miami (2009)
7. Software package Weka, `http://www.cs.waikato.ac.nz/ml/weka/`
8. Cox, T., Cox, M.: Multidimensional Scaling. Chapman and Hall, Boca Raton (October 1994)
9. Visualization Software for Clasifier Evaluation, `http://www.site.uottawa.ca/~nat/Visualization_Software/visualization.html`

Annotation Concept Synthesis and Enrichment Analysis

Mikhail Jiline[1], Stan Matwin[1,2], and Marcel Turcotte[1]

[1] University of Ottawa, Canada
{mjiline,stan,turcotte}@site.uottawa.ca
[2] Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland

Abstract. Annotation Enrichment Analysis (AEA) is a widely used analytical approach to process data generated by high-throughput genomic and proteomic experiments such as gene expression microarrays. We briefly review ideas behind AEA, identify some limitations and propose a novel logic-based Annotation Concept Synthesis and Enrichment Analysis (ACSEA) approach. ACSEA fuses inductive logic reasoning with statistical inference to uncover more complex phenomena captured by the experiments. The results of the evaluation suggest that ACSEA can boost effectiveness of the processing of high-throughput experiments.

1 Introduction

One of the main challenges of modern bioinformatics is to develop methods and techniques that can help inferring knowledge from accumulated datasets and large-scale experimental data such as expression microarrays. High-throughput experimental techniques typically generate sets of genes that require further investigation.

Recently a number of algorithms have been developed to help experts interpreting experimental data. One of the most popular approaches is Annotation Enrichment Analysis (AEA) [1]. AEA algorithms extract known descriptive information (called gene annotations) characterizing each gene and compare the statistical distributions of gene annotations between the gene set of interest (known as study set) and the rest of the genome or the rest of the microarray (known as universe set).

A variety of algorithms and tools are based on the AEA framework. They differ by the knowledge bases used as source for annotations (GO, KEGG, BIND, etc.), statistical hypothesis testing models (χ^2, Fisher's exact test, Binomial distribution, Hypergeometric distribution, etc.), types and organization of annotation terms, and sets of reference genes [1].

The data representation model used in Annotation Enrichment Analysis is bag-of-annotations which associates a set of annotation terms with each gene [2]. While bag-of-annotations is a very popular and efficient model allowing natural application of statistical inference methods, it has a number of disadvantages. The main weakness of the model is the limitation in the types of annotation terms and relations that may be used as well as the types and the complexity of enriched phenomena that can be discovered and described.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 304–308, 2010.

Table 1. Microarray Datasets

Name	Description	Microarray
ALL	Data of T- and B-cell Acute Lymphocytic Leukemia from the Ritz Laboratory at the DFCI	Affymetrix HGU95Av2
GSEA Gender	Transcriptional profiles from male and female lymphoblastoid cell lines	Affymetrix HGU133A
GSEA p53	Transcriptional profiles from p53+ and p53 mutant cancer cell lines	Affymetrix HGU95Av2
GSEA Diabetes	Transcriptional profiles of smooth muscle biopsies of diabetic and normal individuals	Affymetrix HGU133A
GSEA Leukemia	Transcriptional profiles from leukemias - ALL and AML	Affymetrix HGU95Av2
GSEA Lung Cancer	Transcriptional profiles from lung cancer outcome datasets	Affymetrix HGU95Av2

To overcome the analytical challenges posed by the bag-of-annotations model, we propose a new paradigm: Annotation Concept Synthesis and Enrichment Analysis (ACSEA). ACSEA utilizes a logic-based data representation model and a fusion of inductive logic reasoning and statistical inference in the general framework of Annotation Enrichment Analysis. The results of ACSEA's evaluation suggest that it is a very potent technique capable of increasing the efficiency (i.e. the ease of data analysis by a human expert) and effectiveness (i.e. the quality and quantity of the obtained knowledge) of the processing of high-throughput experiments.

2 Annotation Concept Synthesis and Enrichment Analysis

The corner stone of Annotation Concept Synthesis and Enrichment Analysis is a logic based representation and mining model. In this model, all readily available information about genes is represented by logic statements. Inductive logic reasoning together with statistical inference is then applied to synthesize logic formulas (called annotation concepts) discriminating genes belonging to the study set from genes belonging to the universe set. Then, following AEA approach, constructed annotation concepts are sorted according to their P-value and the best of them are presented to the biology expert.

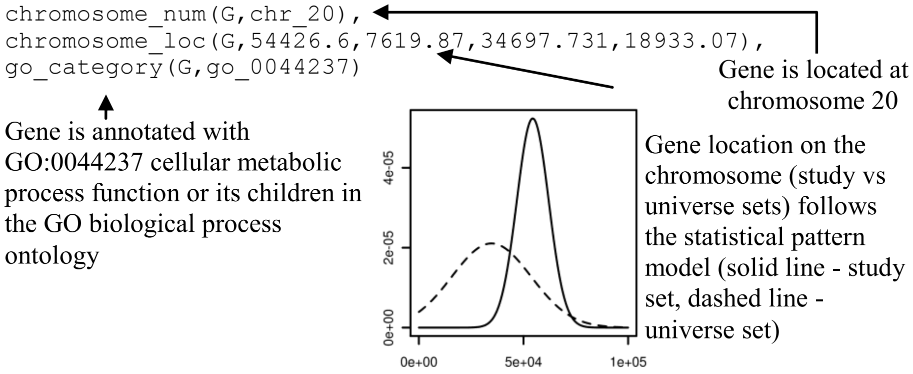
The heart of Annotation Concept Synthesis and Enrichment Analysis is the Annotation Concept Inference algorithm. The algorithm fuses Inductive Logic Programming (ILP) [3] and Statistical Inference [4] approaches. By fusing the inductive logic reasoning and the statistical inference approaches we obtain an inference algorithm capable of mining complex knowledge structures while tolerant to noise and data incompleteness. The ACSEA approach consists of the following key elements: annotation concept synthesis, a hypothesis fitness measure, a theory building strategy, integration of specialized algorithms, and methods for controlling the quality of the theory.

3 Results

To evaluate the performance of Annotation Concept Synthesis and Enrichment Analysis we selected six well known microarray datasets listed in Table 1.

Table 2. Quantitative Performace Evaluation of AEA and ACSEA. Lesser is better.

		PvAvr_1		PvAvr_5		PvAvr_{10}		PvAvr_{25}	
		AEA	ACSEA	AEA	ACSEA	AEA	ACSEA	AEA	ACSEA
ALL	GO	7.62e-06	**1.48e-08**	1.74e-05	**5.55e-08**	4.10e-05	**1.09e-07**	5.50e-04	**2.72e-07**
	GCM	1.43e-03	**1.27e-03**	1.46e-02	**2.91e-03**	3.64e-02	**6.45e-03**	1.19e-01	**3.93e-02**
	GO+GCM	1.50e-05	**1.89e-08**	2.70e-05	**3.87e-08**	1.09e-04	**6.58e-08**	7.44e-04	**1.58e-07**
GSEA Gender	GO	9.04e-08	**4.30e-10**	2.50e-06	**1.04e-09**	3.03e-05	**1.72e-09**	8.29e-04	**3.04e-09**
	GCM	3.90e-07	3.90e-07	1.22e-03	**7.19e-05**	5.99e-02	**1.61e-02**	3.31e-01	**1.00e-01**
	GO+GCM	8.59e-08	**1.34e-10**	6.75e-07	**3.33e-10**	7.49e-06	**6.44e-10**	4.23e-04	**1.55e-09**
GSEA p53	GO	2.16e-03	**2.59e-06**	2.52e-03	**7.46e-06**	3.88e-03	**8.20e-06**	9.25e-03	**1.43e-05**
	GCM	1.97e-06	**3.24e-07**	2.44e-03	**7.86e-04**	6.57e-03	**2.43e-03**	2.25e-01	**1.85e-02**
	GO+GCM	1.98e-06	**3.24e-07**	1.82e-03	**1.33e-06**	2.48e-03	**2.19e-06**	5.84e-03	**4.91e-06**
GSEA Diabetes	GO	3.48e-04	**1.65e-05**	2.55e-03	**2.08e-05**	3.55e-03	**2.55e-05**	6.27e-03	**4.18e-05**
	GCM	7.16e-03	**1.08e-03**	2.59e-02	**6.31e-03**	1.09e-01	**1.33e-02**	2.90e-01	**1.17e-01**
	GO+GCM	3.53e-04	**4.49e-07**	2.62e-03	**9.20e-07**	3.62e-03	**2.41e-06**	6.21e-03	**6.12e-06**
GSEA Leukemia	GO	5.62e-06	**2.81e-06**	7.39e-05	**3.12e-06**	2.28e-04	**3.56e-06**	2.44e-03	**4.35e-06**
	GCM	2.07e-03	2.07e-03	1.80e-02	**4.49e-03**	3.55e-02	**7.59e-03**	6.32e-02	**1.61e-02**
	GO+GCM	9.48e-06	**2.53e-06**	9.68e-05	**3.05e-06**	2.57e-04	**3.91e-06**	2.64e-03	**5.32e-06**
GSEA Lung Cancer	GO	3.05e-05	**9.15e-07**	1.83e-04	**5.27e-06**	3.40e-04	**8.93e-06**	1.25e-03	**2.14e-05**
	GCM	5.82e-08	5.82e-08	2.13e-03	**1.87e-04**	1.20e-02	**1.11e-03**	5.94e-02	**6.76e-03**
	GO+GCM	5.84e-08	**1.70e-08**	1.09e-04	**3.65e-08**	2.79e-04	**4.81e-08**	1.08e-03	**8.18e-08**



```
chromosome_num(G,chr_20),
chromosome_loc(G,54426.6,7619.87,34697.731,18933.07),
go_category(G,go_0044237)
```

Gene is located at chromosome 20

Gene is annotated with GO:0044237 cellular metabolic process function or its children in the GO biological process ontology

Gene location on the chromosome (study vs universe sets) follows the statistical pattern model (solid line - study set, dashed line - universe set)

Fig. 1. An example of a synthesized annotation concept

For each dataset, we applied non-specific filtering, removing genes without sufficient variability to be informative. Next, we applied the standard t-test to identify differentially expressed genes which formed the study set, while all genes left after the non-specific filtering formed the universe set. Each dataset was analyzed with the following annotation sources GO (Gene Ontology,Gene Ontology Annotation for Human), GCM (Gene to Chromosome Mapping: chromosome, chromosome band, and start/end base pairs from Ensembl 56 database), GO+GCM (a combination of the two annotation sources above).

We defined a family of performance measures to evaluate the proposed approach. The measures are based on assessing the P-values for the set of generated hypotheses.

$$PvAvr_n(T) = \frac{1}{n}\sum_{i=1}^{n} \mathrm{Pvalue}(\mathrm{t_i}), \tag{1}$$

where T is a theory consisting of a list of hypotheses $\{t_i\}$ sorted in ascending order by their P-values, and n is the number of the top hypotheses included into evaluation.

We compared the ACSEA approach to the AEA approach represented by Bioconductor's GOstats algorithm. The quantitative performance evaluation results are presented in Table 2. The significance levels are 0.16, 0.02, 0.04, 0.02 for n=1, 5, 10, and 25. The quality of constructed annotations and the level of the integrative information analysis performed by ACSEA can be illustrated by the annotation in Figure 1.

4 Conclusion

Annotation Enrichment Analysis (AEA) is becoming the dominant technique for the secondary processing of data generated by high-throughput experimental techniques. In this paper, we present a novel paradigm, Annotation Concept Synthesis and Enrichment Analysis, which relies on a logic-based representation of annotations and employs a fusion of inductive logic inference and statistical inference.

The methodological advantage of Annotation Concept Synthesis and Enrichment Analysis is five-fold. Firstly, it is easier to represent complex, structural annotation information. Secondly, it is possible to synthesize and analyze complex annotation concepts. Thirdly, it is possible to perform the enrichment analysis for sets of aggregate objects (such as sets of protein-protein interactions or sets of protein complexes). Fourthly, annotation concepts are straightforward to interpret by a human expert. Fifthly, the logic data model and logic induction are a common platform that can integrate specialized analytical tools.

Our results demonstrate that the proposed approach synthesizes higher quality integrated interpretation of biological phenomena captured by biological experiments.

For future work, we plan to pursue the following research directions. We will evaluate our approach on protein-protein interaction screens. We will advance the theory consolidation algorithm that can remove “synonymic” annotations based on coverage and elements of theorem proving. Inductive annotation construction can be extended by abduction [5] to directly suggest annotations missing in the annotation databases that can be inferred based on new experimental data.

References

1. Huang, D.W., Sherman, B.T., Lempicki, R.A.: Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. Nucleic Acids Research 37(1), 1–13 (2009)

2. Beißbarth, T., Speed, T.P.: GOstat: find statistically overrepresented Gene Ontologies within a group of genes. Bioinformatics 20(9), 1464–1465 (2004)
3. Muggleton, S.H., De Raedt, L.: Inductive Logic Programming: Theory and Methods. Jnl. Logic Programming 19(20), 629–679 (1994)
4. Rivals, I., Personnaz, L., Taing, L., Potier, M.: Enrichment or depletion of a GO category within a class of genes: which test? Bioinformatics 23(4), 401–407 (2007)
5. Kakas, A.C., Kowalski, R.A., Toni, F.: Abductive Logic Programming. Journal of Logic and Computation 2(6), 719–770 (1993)

Using Learned PSR Model for Planning under Uncertainty

Yunlong Liu[1,*], Guoli Ji[1], and Zijiang Yang[2]

[1] Department of Automation, Xiamen University, 361005 Xiamen, China
ylliu@xmu.edu.cn, glji@xmu.edu.cn
[2] School of Information Technology, York University, Toronto, Canada M3J 1P3
zyang@mathstat.yorku.ca

Abstract. As an alternative to partially observable Markov decision processes (POMDPs), Predictive State Representations (PSRs) is a recently developed method to model controlled dynamical systems. While POMDPs and PSRs provide general frameworks for solving the problem of planning under uncertainty, they rely crucially on having a known and accurate model of the environment. However, in real-world applications it can be extremely difficult to build an accurate model. In this paper, we use learned PSR model for planning under uncertainty, where the PSR model is learned from samples and may be inaccurate. We demonstrate the effectiveness of our algorithm on a standard set of POMDP test problems. Empirical results show that the algorithm we proposed is effective.

Keywords: Partially observable Markov decision processes, Predictive State Representations, Learned PSR model, Planning under uncertainty.

1 Introduction

One of the central and challenging problems of artificial intelligence is concerned with agent operating in environments that are partially observable and stochastic, i.e., how an agent can plan and act optimally in such environments. A commonly used technique for solving this problem is to model the system firstly, and then solve it by using the obtained model.

Predictive state representations (PSRs)[1] are a recently proposed framework for modeling such controlled dynamical systems, which models a system by defining and operating the PSR state. The PSR state is denoted by a vector of predictions or outcome probabilities for tests that could be done on the system, where a test is a sequence of action-observation pairs. The PSR state summarizes all information regarding the past and the use of it allows one to transform the original POMDP into a Markov decision process (MDP). It has been proved that the PSR state involves only the predictions of core tests, rather than the predictions of all tests.

[*] Corresponding author.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 309–314, 2010.

In recent years, some work for solving the problem of planning under uncertainty has been proposed in the PSR literature [2, 3]. But to the best of our knowledge, all the work assumes an accurate model of the environment. However, in real-world applications, this is rarely the case and it is extremely difficult to build an accurate model of an environment.

As stated above, the use of PSR state allows one to transform the original POMDP into a MDP. Therefore, the original POMDP can be solved by corresponding MDP techniques. However, when the PSR model is inaccurate, even for the same PSR state, the prediction vectors at different time steps used to denote this PSR state can hardly be the same, which in turn cause many PSR states, denoted by the prediction vector, seem like appearing only once. But as we know, the techniques used for solving the MDP problem usually require that the states can be visited more than once.

In this paper, we develop a framework that makes use of a crude PSR model learned from samples for planning under uncertainty. The key idea behind our algorithm is to use discriminant function analysis [4] to track the PSR state at every time step. Given an inaccurate PSR model, the PSR state is identified using discriminant function analysis under the assumption that the prediction vectors corresponding to the same PSR state follow the multivariate normal distribution. Then the partial observable problem is converted into a deterministic finite MDP.

2 Predictive State Representations

As mentioned above, PSR-based models represent the state of a system as a vector of predictions for core tests. A set of tests $Q = \{q_1, q_2, \cdots, q_k\}$ constitutes a PSRs if its prediction vector, $p(Q \mid h) = [p(q_1 \mid h), p(q_2 \mid h), \cdots, p(q_k \mid h)]^T$, is a sufficient statistic for all histories. The definition of history is similar to the definition of test in that they are both a sequence of action-observation pairs. Nevertheless, a history is constrained to start from the beginning of time and is used to describe the full sequence of past events. The tests in Q are called core tests, $p(Q \mid h)$ is called prediction vector at history h and the PSR state at history h is denoted by $p(Q \mid h)$. In general, f_t can be linear or non-linear. If f_t is linear, the PSRs are called linear PSRs; otherwise, it is called non-linear PSRs. In this paper, we only consider linear PSRs.

For linear PSRs, after action a is taken and observation o is observed from history h, the prediction vector of Q can be updated as follows [1]: for each $q_i \in Q$

$$p(q_i \mid hao) = \frac{p(aoq_i \mid h)}{p(ao \mid h)} = \frac{p(Q \mid h)^T m_{aoq_i}}{p(Q \mid h)^T m_{ao}}$$

where m_{aoq_i} is the weight vector corresponding to test aoq_i and m_{ao} is the weight vector corresponding to test ao. Therefore, the prediction vector of hao is [1]:

$$p(Q \mid hao) = \left(\frac{p(Q \mid h)^T M_{ao}}{p(Q \mid h)^T m_{ao}} \right)^T$$

where M_{ao} is a kxk matrix with its i^{th} column equals m_{aoq_i} .

The above formula shows that the prediction vector at any history can be calculated trivially when the model parameters M_{ao} , m_{ao} and the initial prediction vector $p(Q \mid \phi)$ are given.

3 Planning under Uncertainty in Real-World Domains

We have introduced the PSR framework and shown the difficulties that exist in using a learned PSR model for planning under uncertainty. Below we describe our algorithm.

Controlled and uncontrolled dynamical systems can be described mathematically by the system-dynamics matrix Z [5], and given an accurate Z , every row in it has the following property: $\forall k \ \forall \overline{a} \in A^k, \sum_{t \in T(\overline{a})} p(t) = 1$, where $T(\overline{a})$ is the set of tests whose action sequences equal to $\overline{a}$ and A^k is the set of actions whose length is k [5]. So if any two rows in Z are not equal, they are linearly independent and vice versa.

Definition of state history: A set of state histories for a system is any maximal set of histories such that the rows of Z corresponding to these histories are pairwise linearly independent. The element of this set is called state history.

According to the definition of state history, the set of state histories include all the unique rows in the system-dynamics matrix (except for the zero row, the histories corresponding to it are not reachable) and it can be used to describe all reachable PSR states of a system. It also holds true that for a system the mapping between the element in one set of state histories and the element in the set of reachable PSR states is one-to-one. In this paper, we just use state history to represent reachable PSR state. i.e., state history i represents PSR state i .

In reality, there is generally no way to build an accurate PSR model and the model is usually learned from samples. Most existing algorithms for learning the PSR model of a system use Monte Carlo approaches to estimate any entry of the system-dynamics matrix[6, 7]. Under such condition, it can be assumed that all rows corresponding to the same PSR state in a noise system-dynamics matrix are multivariate normal distributed. If the underlying system has finite k reachable PSR states, then for a noise system-dynamics matrix, there exist k groups, and each group contains all the rows corresponding to the same PSR state. Therefore, under the assumption that the mean vector and the covariance matrix of each group are known, if we have a new prediction vector, the group it belongs to can be determined using discriminant function analysis[4]. Then the question becomes how to obtain samples to estimate every group's mean vector and covariance matrix.

In the process of discovering and learning the PSR model of a system, a set of estimated prediction vectors at different histories will be obtained [6]. We divide these vectors into submatrices by partitioning the set of histories to make the prediction vectors in the same submatrix pairwise linearly dependent, i.e., the rank of the submatrix is 1. And the prediction vectors from different submatrices are pairwise linearly independent. (Note: because the sampled prediction vectors are noise, in this paper the rank of a matrix is calculated using the rank calculation procedure developed in the PSR learning algorithm by James et al.[7]). As suggested above, if two rows are linearly dependent, then they belong to the same PSR state; otherwise, they belong to different PSR states. As a result, the prediction vectors in the same submatrix correspond to the same PSR state, i.e., the prediction vectors in the same submatrix constitute a sample. If there exist N submatrices, then there will be N samples from different groups. Group i 's mean vector $u_{(i)}$ and covariance matrix $\sum_{(i)}$ are estimated using sample i . For all the histories in one sample, if a null history exists, it is labeled as a state history for the purpose of reserving the initial state vector; otherwise, the history occurred most times is labeled as the state history. Then a learned PSR is defined and operated by the following:

- a set of PSR states $S = \{s_1, s_2, \cdots, s_{|SH|}\}$, where SH is the obtained set of state history. According to the method of obtaining state history, the prediction vector at state history i can be considered as the closest one to the true prediction vector.
- a set of actions $A = \{a_1, a_2, \cdots, a_{|A|}\}$
- a set of observations $O = \{o_1, o_2, \cdots, o_{|O|}\}$
- a set of transition probabilities $T(s_i, ao, s_j) = 1$, where s_j is determined using the following formula:

$$s_j \leftarrow \arg\min_k [(\frac{p(Q \mid s_i)^T M_{ao}}{p(Q \mid s_i)^T m_{ao}})^T - u_{(k)}]^T \sum_{(k)}{}^{-1} [(\frac{p(Q \mid s_i)^T M_{ao}}{p(Q \mid s_i)^T m_{ao}})^T - u_{(k)}] \quad (k = 1, \cdots, N)$$

- a set of rewards $R : S \text{ x } AO$
- a discount factor $\gamma \in [0,1]$
- an initial prediction vector $p(Q \mid \phi)$

From the frame listed above, it can be seen that the partial observable problem is converted into a deterministic finite MDP, which in turn can be solved by corresponding MDP techniques, such as value iteration, reinforcement learning methods.

4 Experiments and Results

Based on the framework we proposed in Section 3, the PSR state at each time can be determined. Accordingly, the Q-learning algorithm can be implemented straightforwardly using the PSR state as the state representation.

In order to evaluate the effectiveness of our algorithm, we compared it against the Q-learning with CMAC function approximation on PSRs algorithm[3], using two standard POMDP dynamical systems available at Cassandra[8].

Figure 1 and Figure 2 show the empirical results of our algorithm and the Q-learning with CMAC algorithm. These graphs show the average reward per time step given by the learned policies. The y-axis on all graphs is the average reward, and the x-axis is the number of steps taken by the learning algorithms. As seen from the results, our algorithm performed quite well and outperforms the Q-learning with CMAC algorithm.

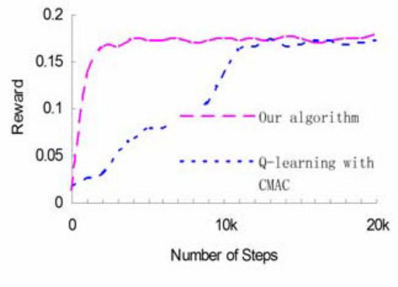


Fig. 1. Cheese Maze

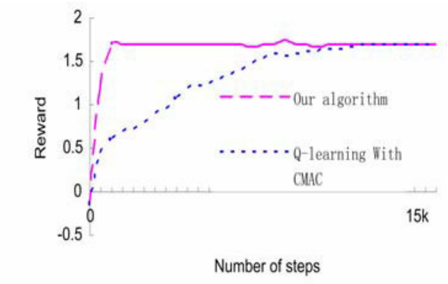


Fig. 2. Shuttle

5 Conclusion

Most approaches for planning under uncertainty assume access to an accurate model of the environment. Such models are generally difficult and expensive to acquire in reality. PSRs is a recently developed method of modeling controlled dynamical systems. This paper has proposed an approach for planning under uncertainty while taking the uncertainty of the learned PSR model into account; something that current state of the art planning algorithms with PSRs are unable to do.

Acknowledgments

This project was funded by grants from the National Natural Science Foundation of China (No. 60774033), Specialized Research Fund for the Doctoral Program of Higher Education of China (No. 20070384003 and 20090121110022), and Xiamen University's National 211 Project 3rd period (No. 0630-E62000). This project is also partially supported by Natural Sciences and Engineering Research Council of Canada.

References

1. Littman, M., Sutton, R., Singh, S.: Predictive Representations of State. In: Neural Information Processing Systems (NIPS). MIT Press, Vancouver (2002)
2. Izadi, M.T., Precup, D.: A Planning Algorithm for Predictive State Representations. In: Eighteenth International Joint Conference on Artificial Intelligence, IJCAI (2003)
3. James, M., Singh, S., Littman, M.: Planning with Predictive State Representations. In: Proceedings of the International Conference on Machine Learning and Applications (ICMLA). IEEE Press, Louisville (2004)

4. Manly, B.F.J.: Multivariate Statistical methods: A PRIMER. Chapman and Hall, New York (1986)
5. Singh, S., James, M., Rudary, M.: Predictive State Representations: A New Theory for Modeling Dynamical Systems. In: Proceedings of the Twentieth Conference on Uncertainty in Artificial Intelligence (UAI). AUAI Press, Banff (2004)
6. Yun-Long, L., Ren-Hou, L.: Discovery and Learning of Models with Predictive State Representations for Dynamical Systems without Reset. Knowledge-Based Systems 22(8), 557–561 (2009)
7. James, M., Singh, S.: Learning and Discovery of Predictive State Representations in Dynamical Systems with Reset. In: Proceedings of the Twenty-First International Conference on Machine Learning (2004)
8. Cassandra, A.: Tony's pomdp file repository page (1999), `http://www.cs.brown.edu/research/ai/pomdp/examples/index.html` (cited 2010)

Notes on Generating Satisfiable SAT Instances Using Random Subgraph Isomorphism

Călin Anton[1,2] and Christopher Neal[2]

[1] Grant MacEwan University, Edmonton, Alberta, Canada
[2] Augustana Faculty, University of Alberta, Camrose, Alberta, Canada

Abstract. We present a preprocessing procedure for the Subgraph Isomorphism problem, and report empirical results of applying it to Generating Satisfiable SAT instances using the Random Subgraph Isomorphism model. The experimental results show that the preprocessor makes the easy-hard-easy pattern of empirical hardness clear for all SAT solvers tested. Moreover, the use of the procedure does not significantly influence the location or the empirical hardness of the instances at the hardness peak, thus preserving the model's main characteristics.

1 Introduction

SRSGI, a new model for generating hard Satisfiable SAT instances based on Random Subgraph Isomorphism was proposed in [1]. The variation of the empirical hardness of SRSGI instances, exhibits an easy-hard-easy pattern, which is important as it helps identifying the hardest instances and allows a fine tuning of the instances' hardness. It has been noticed that in some cases the pattern of hardness variation of SRSGI instances is rather hard-harder-easy. In this short paper we present a preprocessing procedure for the Subgraph Isomorphism Problem, and investigate its consequences for the above mentioned model. We show that the procedure enhances the easy-hard-easy pattern while preserving the other characteristics of the model. This is beneficial for the model because it reduces the unnecessary hardness of some instances, and maintains the other hard instances. The empirical results confirm the hypothesis stated in [1] that "some form of preprocessing done by the solvers" is the main reason for the exceptions which did not exhibit the easy-hard-easy pattern.

2 Definitions

A graph is a pair $G = (V_G, E_G)$, where V_G is the set of vertices, and $E_G \subset V_G X V_G$ is the set of edges. The number of vertices of a graph is referred to as the order of the graph and the number of edges is referred to as the size of the graph. Vertices x and y are neighbors if (x, y) in an edge. The degree of a vertex is the cardinality of its set of neighbors. The distance between two vertices is the length of the shortest path connecting them, or ∞ if there is no path between the vertices. A graph $G = (V_G, E_G)$ is a subgraph of a graph $H = (V_H, E_H)$, if

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 315–318, 2010.

$V_G \subseteq V_H$, and $E_G \subseteq E_H$. If $E_G = E_H \cap V_G \mathrm{X} V_G$ then G is the induced subgraph of H. The neighbor sub-graph of a vertex x in G, is the induced subgraph of G having the neighbors of x as the set of vertices. Two graphs are isomorphic, if and only if there exists a bijective mapping of their vertices which preserves the neighbor relationship among vertices. Given two graphs H and G the subgraph isomorphism problem (SGI) asks if G is isomorphic to a subgraph of H. For $n \in \mathbb{N}^*$, $m \in \{0 \dots \binom{n}{2}\}$, a random graph of order n, and size m is generated by randomly choosing without replacement m edges from the set of all possible $\binom{n}{2}$ edges. For $n \leq m \in \mathbb{N}$, $q \in \{0 \dots \binom{m}{2}\}$, and $p \in (0, 1)$ a satisfiable random (n, m, p, q) SGI (SRSGI)[1] consists of a random graph H, of order m and size q, and a graph G obtained by the following steps: (1) randomly select an order n induced subgraph G' of H; (2) randomly remove $\lfloor p|E_{G'}|\rfloor$ edges of G'. SRSGI instances are then encoded to SAT using the straightforward direct encoding.

3 A Preprocessing Procedure for SGI

Any solution of an SGI instance consisting of a graph H and a subgraph G, in which vertex x of G is mapped into vertex x' of[1] H, and vertex y of G is mapped into vertex y' of H, must satisfy the following properties: (1) The degree of x in G is less than or equal to the degree of x' in H; (2) The size of the neighbor subgraph of x in G is less than or equal to the size of the neighbor subgraph of x' in H; (3) The number of common neighbors of x and y in G must be less than or equal to the number of common neighbors of x' and y' in H; (4) The distance in G between x and x' is greater than or equal to the distance in H between y and y'. Properties (1), (2) and (3) are immediate consequences of the feature of SGI which allows mapping non-edges of G into edges of H. Property (4) is based on the same feature and the observation that the distance between vertices decreases as the density of the graph increases. We used the above properties to define a preprocessing procedure (PP) for SGI. It eliminates all mapping pairs $x - x'$ which violate any of the above conditions. It also enforces arc consistency among mapping pairs and checks that they satisfy the all different constraint. PP is applied repeatedly until no more mapping pairs can be eliminated. The number of eliminated mapping pairs is used as a measure of PP's efficiency. All other parameters being fixed, it is presumable that PP's efficiency: (a) increases as n_G increases - as there are more conditions to check, and so more chances to eliminate mapping pairs; (b) decreases as m_H increases - because as H becomes denser, it is less likely that any of the above properties will be violated.

4 Empirical Results

We intend to investigate PP's influence on the characteristics of SAT encoded SRSGI. In order to allow comparisons, we used the experimental framework of [1]: (1)BerkMin, minisat, picosat, and Rsat were the complete solvers, and

[1] We will use mapping pair to denote such a pair x-x'.

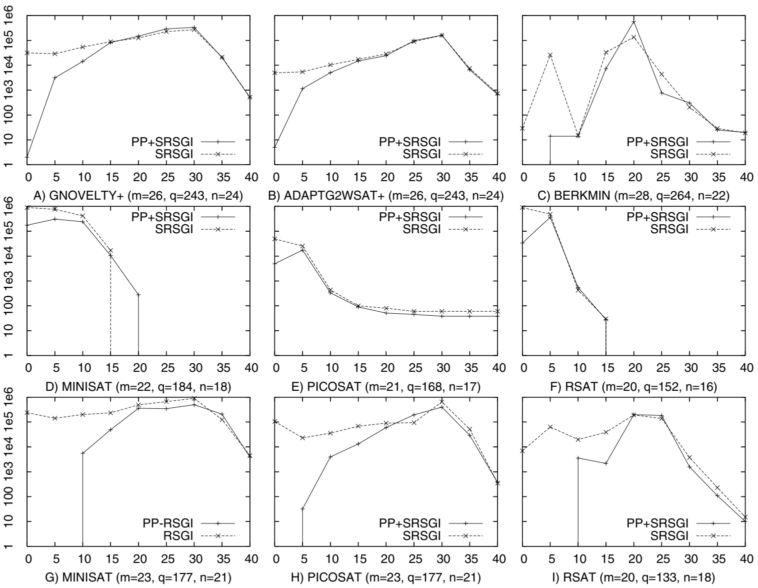
adaptg2wsat+ and gnovelty+ were the incomplete solvers used in the experiment; (2) m was set to 20, 21, 22, 23 and 24; for BerkMin and the incomplete solvers, experiments were performed for m set to 26 and 28; (3) n varied from $m - 6$ to $m - 2$; (4) q varied between $0.65\binom{m}{2}$ and $0.80\binom{m}{2}$ in increments of $0.05\binom{m}{2}$; (5) p varied between 0% and 40% in increments of 5%; (6) 10 instances were generated for each combination of values of m, q, n and p; (7) The cutoff time limit was set to 900 seconds.

The experimental results confirm the hypothesis made about the efficiency of PP. For all solvers, the efficiency of PP increases with n, and decreases as q increases. The results indicate that PP is more efficient for small values of p ($0\% - 10\%$) and it becomes less and less efficient as p increases - see Fig. 1. A possible explanation is that as p increases, the subgraph (G) becomes sparser, and so the probability of violating the stated properties decreases, and so does PP's efficiency. Hence, it follows that the overall efficiency of PP is positively correlated with n , and negatively correlated with q and p respectively.

An immediate consequence is that PP is not very effective for small values of n and large values of q. For such cases it was noticed in [1] that "RSat, minisat and picosat found the instances with $p = 0$ to be the hardest and ... they did not exhibit the easy-hard-easy pattern." Despite the limited efficiency of PP for these combinations, it still reduced the number of cases showing these "irregularities", see Fig. 1 D, E and F. In these cases the addition of PP changed the pattern of empirical hardness by revealing a peak in hardness, which is consistent among all solvers. This implies that the cause of the "irregularities" is indeed "some form of preprocessing done by the solvers" as suggested in [1].

SAT encoded SRSGI instances exhibit an easy-hard-easy pattern of empirical hardness but as stated in [1] for "incomplete solvers the pattern may appear as hard-harder-easy". Given that PP is most effective for $p = 0$, its usage should reduce the hardness of the instances with small p, thus making the pattern of empirical hardness become "easy-hard-easy" for all cases. As showed in Fig. 1 A and B, this is indeed the case. PP has a similar effect for the complete solvers, as showed in Fig. 1 C, G, H and I.

For large n the hardness of the SRSGI instances peaks at values of p larger than 10. Because PP's efficiency decreases as p increases, the use of PP does not significantly change the hardness of these instances (see Fig. 1). Moreover, it is not always the case that PP decreases the empirical hardness of the hardest instances; in some cases the hardest instances become harder when PP is applied to them. Not only that the PP does not have an important influence on the empirical hardness of the instances at the hardness peak, but also it very rarely changes their location. Even in the few cases when it does, it only shifts the location by one increment (5%). Furthermore, the location is sometimes shifted to the left and sometimes shifted to the right, therefore the overall effect of PP on the location of the hardest instances is irrelevant. For example, for each value of m and for all solvers, the combinations of parameters which produce the hardest instances for PP+SRSGI and those which produces the hardest instances for SRSGI are the same. Hence, applying PP to SAT encoded SRSGI preserves the



A) GNOVELTY+ (m=26, q=243, n=24)
B) ADAPTG2WSAT+ (m=26, q=243, n=24)
C) BERKMIN (m=28, q=264, n=22)
D) MINISAT (m=22, q=184, n=18)
E) PICOSAT (m=21, q=168, n=17)
F) RSAT (m=20, q=152, n=16)
G) MINISAT (m=23, q=177, n=21)
H) PICOSAT (m=23, q=177, n=21)
I) RSAT (m=20, q=133, n=18)

Fig. 1. Comparison of SRSGI and PP+SRSGI instances. p is represented on the x-axis and time(ms) is represented on the (logarithmic) y-axis.

characteristics of the model regarding the hardness peak, such as: the exponential growth rate of the hardest instances and the correlations between parameters. Therefore, we believe that PP improves the SRSGI model, and its SAT encoding, as it preserves their important and desirable characteristics, while making the easy-hard-easy pattern of the empirical hardness more evident.

5 Conclusions

We investigated the influence of a Subgraph Isomorphism preprocessing procedure on the hardness of a generator of satisfiable SAT instances, based on random subgraph isomorphism. We noticed that the use of the procedure renders more clear the easy-hard-easy pattern of the evolution of the empirical hardness of the instances, while preserving all the other characteristics of the model. The experiments support the hypothesis stated in [1] that the perceived hardness of the instances generated for small values of p, is a consequence of some form of preprocessing done by the SAT solvers. The use of the procedure enhances the characteristics of the generator, therefore making it an even better candidate for generating satisfiable SAT instances.

Reference

1. Anton, C., Olson, L.: Generating satisfiable SAT instances using random subgraph isomorphism. In: Proceeding of Canadian Conference on AI 2009, pp. 16–26 (2009)

A Model for Reasoning about Interaction with Users in Dynamic, Time Critical Environments for the Application of Hospital Decision Making

Hyunggu Jung and Robin Cohen

Cheriton School of Computer Science
University of Waterloo
{h3jung,rcohen}@uwaterloo.ca

1 Overview

In this paper, we present a model for reasoning about interaction with users in dynamic, time critical environments in a way that is sensitive to the cost of bother. We project the model into the scenario of decision making in hospital emergency room settings, providing a framework for modeling the doctors in that environment, to determine whom to ask to attend to a current patient. A simulation of our model demonstrates that it offers valuable improvements due to its reasoning about bother.

2 A Model for Dynamic, Time Critical Scenarios

Our research aims to develop a model that can be used for scenarios where an agent is reasoning about which human users to enlist to perform decision making, in an environment where decisions need to be made under critical time constraints and where the parameters that serve to model the human users are changing dynamically, to a significant extent.We offer a hybrid transfer of control strategy that takes as its starting point the model of Cheng [1], which includes reasoning about interaction (partial transfers of control or PTOCs) as well as about full transfers of control of the decision making (FTOCs) to another entity.

Each possible transfer control strategy is generated in order to select the strategy that offers the highest expected utility after period of time. A transfer of control chain includes provision to ask a different entity after waiting a determined period of time. Distinct from Cheng's original model, attempts at FTOCs are in framed as PTOCs with the question Q: "Can you take over the decision making?". This then enables either a "yes" response, which results in an

FTOC[1] or a "no" response or silence.

The "no" and silence responses ultimately lead to a new method aimed at coping with the dynamics of the environment and the criticality of time. As in Cheng's model, a TOC strategy will arrange for alternate users to query,

[1] As a simplication, we assume that a "Yes" response results in the user successfully assuming control of the decision.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 319–323, 2010.

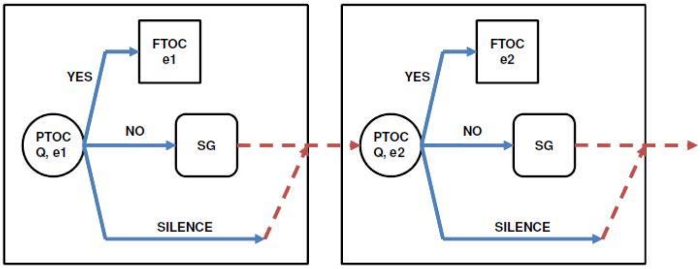


Fig. 1. Visual representation of strategy with the FTOCs and PTOCs; each world occupies one square

once the current respondent has not volunteered to assist. We first make the simplification that the strategies do not ask different entities within the same chain. This is because we are limiting ourself to only one question, that of asking the expert to help. Then, at the end of this chain of attempts, we inject a final decision of strategy regeneration. Strategy regeneration will then allow for an updating of parameter values. This new generation of possible strategies allows us to reason at this point in time regarding the users who are available to help and their expected quality of decision, cost of bother, etc. with information that is no longer stale; this is of particular value in circumstances where choices that are less than optimal can be undesirable, to a dramatic extent.

A diagram outlining the FTOCs and the PTOCs that we envisage is presented in Figure 1 where an arrow with a solid line means the stream of time, but a dotted line means there is no break by the end of the arrow. In addition, we introduce a concept of *world* to facilitate the computation of the utility of any given strategy.

One *world* consists of one PTOC, one FTOC, and one SG (strategy regeneration) node and includes all the parameters currently used to calculate benefits and costs to reason about interaction with entities. Therefore, when the current *world* is moved to the next step, our system asks a new entity. The number of worlds is equivalent to the number of entities that will be asked.

There are n FTOC nodes, n PTOC nodes, n SG nodes, and one virtual node in the overall framework with n worlds. We obtain the overall EU of strategy s by summing up n EU values for FTOC nodes, n EU values for SG nodes and one EU value for the virtual node as follows:

$$EU(s) = EU_n(dfl) + \sum_{j=1}^{n}(EU_j(fn_l) + EU_j(sg)) \tag{1}$$

where dfl[2] reflects a virtual node, n denotes the number of worlds, $EU(fn_l)$ reflects the utility of ending in a FTOC, and $EU(sg)$ reflects the utility of ending in SG node.

[2] The leaf node for the silence response is set to sg.

3 Selection of Experts for Medical Decision Making

We are also interested in determining an appropriate reasoning strategy to find the right person, at the right time, to assist with the care of patients who are arriving at a hospital emergency room. Typically in these settings, patients who appear to require further assistance than can be immediately provided (what we could call "a decision") require soliciting aid form a particular specialist.

In order for the human first clinical assistants (FCAs) to make the best decisions about which specialists to bring in to assist the patients that are arriving, the proposal is to have our multiagent reasoning system running in the background, operating with current parameter values to suggest to the medical professionals who exactly they should contact to assist the current patient. These experts are then the entities $\{e_1,\ e_2, \ldots,\ e_n\}$ that are considered in our reasoning about interaction, with the FCA contacting the experts according to the optimal strategy our model generates (who first, waiting for how long, before contacting who next, etc.)

We propose the addition of one new parameter as part of the user modeling for the bother cost, a $Lack\_of\_ExpertiseFactor$. This parameter is used to help to record the general level of expertise of each doctor (i.e. medical specialist), with respect to the kind of medical problem that the patient is exhibiting.

We also adjust the calculations proposed by Cheng for estimating bother cost, in order to reduce the number of parameter values that need to be acquired or solicited (for our time-critical scenarios).

We assume that a user's willingness is simply determined by their attentional state factor and their expertise level. We also assume that a user's probability of response is determined by their willingness. Some factors which affect bother cost in hospital settings are thus as follows.

- *User_Unwillingness_Factor=Attention_State_Factor+Lack_of_Expertise_Factor*
- $Init = User\_Unwillingness\_Factor \times Attention\_State\_Factor \times TOC\_Base\_Bother\_Cost$
- $BSF\ (Bother\,So\,Far) = \sum_{toc \in PastTOC} TOC\_Base\_Bother\_Cost(toc) \times \beta^{t(toc)}$
- $BotherCost\ (BC) = Init + BC\_Inc\_Fn(BSF, User\_Unwillingness\_Factor)$

We introduce another new parameter, *task criticality (TC)*, to affect the reasoning about interaction. TC is used to enable the expected quality of a decision to be weighted more heavily in the overall calculation of expected utility, when the case at hand is very critical. This parameter may also be adjusted, dynamically. When a patient has high task criticality, strong expertise is required because the patient's problem may become much more serious if not treated intensively.

With bother to an expert being modeled in detail, this leads to strategies that are less likely to ask an entity who will simply fail to respond. Our detailed bother modeling for time critical environments is an advance on other bother models such as [2], which focuses on modeling annoyance.

4 Validation

Our validation measures performance of our model reflecting dynamic and time critical aspects by comparing with that of a model missing the calculation of bother cost. In the setting of our validation simulating hospital emergency scenarios, there are four entities on the entity list and five patients on the waiting list. Every patient has a task criticality for the specific medical problem and the task criticality of each patient is changed dynamically as time passes. Our simulation first selects the patient whose task criticality is highest among those of patients.

We then obtain a strategy chain by calculating formulae reflecting our model with information of each patient. After choosing a entity in the chain, we ask him/her to treat the current patient and update the task criticality of patients who have been treated by entities, as well as those remaining on the waiting list. When there is no more patients on the waiting list, we finally count the number of dead patients[3]. By comparing the number of dead patients simulated by our model with bother cost and without bother cost, we can validate whether our model reflects dynamic and time critical domains effectively. Our current results demonstrate valuable improvements with our model.

5 Discussion

Our work contrasts with those of user modeling researchers such as [2], who focus on modeling a user's plans in order to determine whether to interact, whereas we focus on deriving the best expected utility from the interaction. The user modeling approach to our research coincides well with those of others such as [3], who advocates that values of variables in user models be determined as a combination of modeling the specific user, the class to which that user belongs and the traits that are typical of all users. For future work, it would be valuable to explore a counterpart to Fleming's stereotypical classes, in order to gain greater insights into how to set the values of the user modeling parameters. For future work, we should also lift the various default parameters suggested in Section 3 and explore methods for learning about our users, over time. One possible starting point for this work is the research of [4], which advocates the use of active learning, to involve the user in the process of progressively determining appropriate parameter values for interruptions.

References

1. Cheng, M., Cohen, R.: A hybrid transfer of control model for adjustable autonomy multiagent systems. In: Proceedings of AAMAS 2005 (2005)
2. Raskutti, B., Zukerman, I.: Generating queries and replies during information seeking interactions. Int. J. of Human Computer Studies 47(6), 689–734 (1997)

[3] This is a rather drastic statistic. We are exploring other measures of system performance.

3. Fleming, M.: The use of increasingly specific user models in the design of mixed-initiative systems. In: Proceedings of the 17th Canadian Conference on AI, pp. 434–438 (2004)
4. Kapoor, A., Horvitz, E.: Experience sampling for building predictive user models: a comparative study. In: CHI 2008: Proceeding of the 26th annual SIGCHI conference on Human factors in computing systems, pp. 657–666. ACM, New York (2008)

A Novel Approach for Recommending Ranked User-Generated Reviews

Richong Zhang and Thomas T. Tran

School of Information Technology and Engineering,
University of Ottawa
800 King Edward Ave. Ottawa,
ON, K1N 6N5, Canada
{rzhan025,ttran}@site.uottawa.ca

Abstract. User-generated reviews play an important role for potential consumers in making purchase decisions. However, the quality and helpfulness of user-generated reviews are unavailable unless consumers read through them. Existing helpfulness assessing models make use of the positive vote fraction as a benchmark. This benchmark methodology ignores the voter population size and the uncertainty of the helpfulness estimation. In this paper, we propose a user-generated review recommendation model based on the probability density of the review's helpfulness. Our experimental results confirm that our approach can effectively assess the helpfulness of user-generated reviews and recommend the most helpful ones to consumers.

Keywords: Helpfulness Ranking, User-Generated Review.

1 Introduction

The conventional helpfulness assessing approaches [1] [2] [3] are to learn a helpfulness function from a set of user-generated contents with helpfulness values which is defined as the observed positive vote fractions. The positive vote fraction is an empirical estimation of the true helpfulness value and the same estimated helpfulness value can be achieved from a small population of votes or a large set of votes. When the user-generated reviews only correspond to a small number of votes, the uncertainty of point estimate will be increased. This motivates us to model the helpfulness of user-generated reviews probabilistically to account for the uncertainty. In this paper, we propose a probabilistic approach to model the helpfulness distribution of review documents and recommend helpful reviews to the possible consumers based on the obtained helpfulness distribution, which contains both the information of how helpful of a review and how voters will vote the review. Furthermore, we introduce the helpfulness bias formulation based on which our model can rank user-generated reviews. We apply our algorithm to a review document data set from Amazon.com and the experimental results show that our proposed algorithm can effectively predict the helpfulness distribution of user-generated reviews and recommend the most helpful contents to potential consumers.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 324–327, 2010.

2 Helpfulness Distribution Discovering Model

Formally, we use i to index the review documents set D. Let α_i denote the helpfulness value of a review document D_i. The existing approaches toward learning the helpfulness of document merely consider utilizing the positive vote fraction as the helpfulness benchmark and finding a point estimate to the helpfulness value. However, the positive vote fraction ignores the voter population size and treat all the reviews with the same helpful vote ratio as the same degree of helpful. This benchmark does not take the uncertainty of the helpfulness estimation into account. Therefore, we propose a probabilistic model to find the distribution $P(\alpha_i)$ for a review document D_i and rank user-generated reviews based on the resulted helpfulness distribution.

We use I to denote the index set of all review documents. Let Γ denote the set of all voters' opinion for D and $\Gamma(i)$ index the set of all voters' opinion on the review document D_i. Let $V_{\Gamma(i)}$ denote the collection of all votes on the i^{th} document. We also defined $V_{\Gamma(i)} = \{V^j_{\Gamma(i)} : j \in \Gamma(i)\}$ and $V_\Gamma = \cup V_\Gamma(i)$, $i \in I$. The helpfulness α_i of a review document D_i can be defined as the probability that a random chosen voter j, where $j \in \Gamma(i)$, will vote positively on the review document D_i. Namely, $P(V^j_{\Gamma(i)} = 1|D_i) = \alpha_i$ and $P(V^j_{\Gamma(i)} = 0|D_i) = 1 - \alpha_i$.

Given a review document D_i and a collection of voters' opinions $V_{\Gamma(i)}$, the helpfulness is the subjective opinion of the voter population in the statistic sense. When the voter population is given, the helpfulness value is an inherent property of the review document.

$$P(V^j_{\Gamma(i)}|D_i, \alpha_i) = P(V^j_{\Gamma(i)}|\alpha_i) \tag{1}$$

Furthermore, we assume that votes are independent for any $P(V^j_{\Gamma(i)}|D_i)$ and it is possible to determine $P(V_\Gamma|D_I)$ according to:

$$P(V_\Gamma|D_I) = \prod_{i \in I} P(V_{\Gamma(i)}|D_I) = \prod_{i \in I} \prod_{j \in \Gamma(i)} P(V^j_{\Gamma(i)}|D_i).$$

Under the formulation of helpfulness, the objective of inferring the helpfulness is to learn α_i based on a sample of documents and a collection of votes on these documents. Therefore, the objective of helpfulness inference is to select a distribution such that $P(V_\Gamma|D)$ is maximized, namely, to determine

$$P^*(\alpha|D) = \underset{P(\alpha|D)}{\operatorname{argmax}} \prod_{i \in I} \prod_{j \in \Gamma(i)} \int P(V^j_{\Gamma(i)}|\alpha_i) P(\alpha_i|D_i) d\alpha_i \tag{2}$$

Assuming that each feature appears independently in a review, then α_i can be calculated by the linear summation model of:

$$\alpha_i = F_i A + z \tag{3}$$

where the variances of the Gaussian noise term $z \sim \mathcal{N}(0, \sigma^2)$.

Consider the generative process for a review document containing features F_i, the probability density function of α_i is:

$$P(\alpha_i|F_i, A, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(\alpha_i - F_i A^T)^2}{2\sigma^2}) \tag{4}$$

Then it leads to the following optimization problem to find a parameters pair of A, σ^2 to maximize

$$\sum_{i \in I} \log \int_{\alpha_i} \prod_{i \in I} \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(\alpha_i - F_i A^T)^2}{2\sigma^2}) \prod_{j \in \Gamma(i)} \alpha_i^{V^j_{\Gamma(i)}} (1 - \alpha_i)^{V^j_{\Gamma(i)}} d\alpha_i \tag{5}$$

Once the helpfulness distribution $P(\alpha_i|F_i)$ is determined, we define a ranking metric from $P(\alpha_i|F_i)$ as:

$$r_i := P(\alpha_i > 0.5|F_i) = \int_{0.5}^{1} P(\alpha_i|F_i) d\alpha_i. \tag{6}$$

We call this metric as *helpfulness bias*, which describes the helpfulness distribution bias towards 0 or 1. The helpfulness bias of the "Oracle", which carries the prior knowledge of voters' opinions, can be formulated as the integral of the posterior density over [0.5, 1], namely,

$$r_i^* := P(\alpha_i > 0.5|F_i) = \int_{0.5}^{1} \prod_{j \in V_{\Gamma(i)}} \alpha_i^{V^j_{\Gamma(i)}} (1 - \alpha_i)^{1 - V^j_{\Gamma(i)}} d\alpha_i. \tag{7}$$

This value can be seen as the helpfulness benchmark of the user-generated review D_i.

3 Experimental Results

This section provides the preliminary empirical evidence that our probabilistic model can effectively rank the helpfulness of the user-generated reviews.

3.1 Data Set

We crawled 501 user-generated LCD HDTV reviews from Amazon.com. These documents have been evaluated by at least 10 consumers as helpful or not helpful by the consumers. We use the bag-of-words model to represent text and to build our language model. Each feature is a non-stop stemmed word and the value of this feature is a boolean value of the occurrence of the word on the review. After the parsing and stemming of all the reviews, we receive a document term matrix associated with the helpfulness value of each review document. In total, we have 501 reviews with a vocabulary size of 4207 stemmed, none-stop terms. Because all online product review consists of terms, we consider to build a document-term matrix and apply PCA to map the input data into a lower dimensional space.

To obtain a reliable result, we perform our probabilistic model on the PCA-projected data with 10-fold cross-validation and report the average performance of those 10 folds of experimentations. The helpfulness bias of "Oracle" can be calculated from the training set and be compared with the resulted helpfulness bias of our algorithm. The ranking performance of different approaches is analyzed by take the average Spearman rank correlation coefficient. Table 1 shows the experimental results of our model at different by fixing σ from 0.1 to 0.5.

Table 1. Ranking correlation of our probabilistic model

Algorithm	Rank Correlation
Probabilistic Model and $\sigma = 0.1$	0.57104
Probabilistic Model and $\sigma = 0.2$	**0.59280**
Probabilistic Model and $\sigma = 0.3$	0.59022
Probabilistic Model and $\sigma = 0.4$	0.58832
Probabilistic Model and $\sigma = 0.5$	0.58794

From Table 1, we find that our probabilistic model achieves the best ranking performance of 0.593 at $\sigma = 0.2$ (marked by bold fonts). The results demonstrate that the resulted helpfulness bias is correlated significantly with the observed Oracle's helpfulness bias. Therefore, we can rank available reviews based on the helpfulness bias and the most helpful reviews can be recommended to the potential consumers.

4 Conclusion

We have proposed a probabilistic approach to predict the helpfulness distribution of online product review. The helpfulness bias has been introduced to rank user-generated reviews based on the obtained predicted distribution. Furthermore, we have executed a empirical study on the reviews of Amazon.com and the experimental results show that our model can effectively predict the review's helpfulness.

References

1. Kim, S.M., Pantel, P., Chklovski, T., Pennacchiotti, M.: Automatically assessing review helpfulness. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, July 2006, pp. 423–430 (2006)
2. Liu, Y., Huang, X., An, A., Yu, X.: Modeling and predicting the helpfulness of online reviews. In: Proceedings of the 18th IEEE International Conference on Data Mining, December 2008, pp. 443–452 (2008)
3. Weimer, M., Gurevych, I., Mühlhäuser, M.: Automatically assessing the post quality in online discussions on software. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, June 2007, pp. 125–128 (2007)

A Semantic Model for Social Recommender Systems

Heung-Nam Kim[1,2], Andrew Roczniak[1], Pierre Lévy[1],
and Abdulmotaleb El-Saddik[2]

[1] Collective Intelligence Lab, University of Ottawa
{hnkim,roczniak}@mcrlab.uottawa.ca, pierre.levy@mac.com
[2] Multimedia Communication Research Lab, University of Ottawa
abed@mcrlab.uottawa.ca

Abstract. Social recommender systems, which have emerged in response to the problem of information overload, provide users with recommendations of items suited to their needs. To provide proper recommendations to users, social recommender systems require accurate models of characteristics, interests and needs for each user. In this paper, we introduce a new model capturing semantics of user-generated tags and propose a social recommender system that is incorporated with the semantics of the tags. Our approach first determines semantically similar items by utilizing semantic-oriented tags and secondly discovers semantically relevant items that are more likely to fit users' needs.

Keywords: Social Recommender System, IEML Semantic Model.

1 Introduction

With the popularity of social tagging (also known as folksonomies), recently a number of researchers have concentrated on recommender systems with social tagging. Recommender systems incorporated with the tags can provide promising possibilities to better generate personalized recommendations [3]. However, if the systems do not take into consideration the semantics of tags themselves, they suffer from fundamental problems: polysemy and synonymy of the tags. Without the semantics of the tags used by users, the systems cannot differentiate the various social interests of the users from the same tags.

To address the discussed issues, we introduce a new model capturing semantics of user-generated tags and propose a social recommender system that is incorporated with the semantics of the tags. Our approach first determines similarities between items by utilizing semantic-oriented tags that is associated to tags that users collectively annotate, called *Uniform Semantic Locator* (*USL*) [4] and subsequently identifies semantically similar items for each item. Finally, we recommend items (*e.g.*, text, picture, video) based on the semantically similar items. The main contributions of this study toward social recommender systems can be summarized as follows: 1) We present a model for semantic-oriented social tagging by using IEML [4]. We illustrate how the model can be adapted and applied to existing social tagging systems. 2) We propose a method in semantic space that aims to find semantically similar items and discover (recommend) items semantically relevant to users' needs.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 328–331, 2010.

2 A Semantic Model for Social Recommender Systems

In this paper, we exploit Information Economy MetaLanguage (IEML) [4] for social recommender systems. Formally, folksonomy F is a tuple $F = \langle \tilde{U}, \check{T}, \tilde{I}, Y \rangle$ where $\tilde{U}$ is a set of users, $\check{T}$ is a set of tags, $\tilde{I}$ is a set of items, and $Y \subseteq \tilde{U} \times \check{T} \times \tilde{I}$ is a ternary relationship called tag assignments, respectively [2]. Beyond the tagging space, in our study, there is another space where the tags are connected to *USL*s according to their semantics. We label this space the IEML semantic space. According to IEML model, a *USL* is composed of a set of semantic categories of different layers, called catset [4]. Therefore an extended formal definition of the folksonomy, called *semantic folksonomy*, is defined as follows:

Definition 1 (Semantic Folksonomy). Let $Ł$ be the whole IEML semantic space. A *semantic folksonomy* is a tuple $SF = \langle \tilde{U}, \check{T}, \tilde{I}, Y, \check{N} \rangle$ where $\check{N}$ is a ternary relationship such as $\check{N} \subseteq \tilde{U} \times \check{T} \times Ł$.

From *semantic folksonomies*, we present a formal description of a semantic item model as follows:

Definition 2 (Semantic Item Model). Given an item $i \in \tilde{I}$, a formal description of a semantic item model for item i, $M(i)$, follows: $M(i) = \langle \check{T}(i), \check{N}(i) \rangle$, where $\check{T}(i) = \{(u, t) \in \tilde{U} \times \check{T} \mid (u, t, i) \in Y\}$ and $\check{N}(i) = \{(t, \upsilon) \in \check{T} \times Ł \mid (u, t, \upsilon) \in \check{N}\}$.

2.1 Recommendation Based on the Semantic Model

In our social recommender system, we first look into the set of similar items that the target user has tagged and then compute how semantically similar they are to the target item, called a *semantic item-item similarity*. Based on the semantically similar items, we recommend relevant items to the target user through capturing how he/she annotated the similar items. We define semantically similar items as a group of items that tagged categories of IEML close to those of the target item. Note that a *USL* can have at most seven distinct layers. Therefore, semantic similarity between two items, i and j, can be computed by the weighted sum of layer similarities from layer 0 to layer 6. Formally, the semantic item similarity measure is defined as:

$$semISim(i, j) = \omega \cdot \sum_{l=0}^{6} \frac{(l+1)}{7} \times \frac{|USL_*^i(l) \cap USL_*^j(l)|}{|USL_*^i(l) \cup USL_*^j(l)|} \tag{1}$$

where ω is a normalizing factor such that the layer weights sum to unity. $USL_*^i(l)$ and $USL_*^j(l)$ refer to the union of *USL*s for item i and j at layer l, $0 \leq l \leq 6$, respectively. The layer similarity between two *USL* sets is defined as the weighted *Jaccard coefficient* of two *USL* sets. Here we give more layer weights at higher layer when computing the semantic item similarity. That is, the intersections of higher layers present more contribution than intersections of lower layers.

Once we have identified a group of semantically similar items, the final step is a prediction, this is, attempting to speculate upon how a certain user would prefer unseen items. In our study, the basic idea of discovering relevant items starts from assuming that a target user is likely to prefer items which are semantically similar to items that he/she has tagged before. Formally, the prediction value of the target user *u* for the target item *i*, denoted as *P*(*u*, *i*), is obtained as follows:

$$P(u,i) = \sum_{j \in SSI_k(i)} \frac{|USL_*^i \cap USL_u^j|}{|USL_u^j|} \times semISim\,(i,j) \tag{2}$$

where $SSI_k(i)$ is a set of *k most similar items* of item *i* grouped by the semantic item similarity and USL_u^j is the union of *USLs* connected to tags that user *u* has annotated item *j*. *semISim*(*i*, *j*) denotes the semantic similarity between item *i* and item *j*. Finally, a set of top-*N* ranked items that have obtained the higher scores are identified for user *u*, and then, those items are recommended to user *u*.

3 Experiment Result

In this section, we compare the performance of our method (*SM*) against that of the benchmark algorithms: a user-based collaborative filtering (*UCF*), an item-based collaborative filtering (*ICF*), and a most popular tags approach (*MPT*) [2]. The dataset used in this study is the *p*-core at level 5 from *BibSonomy* (http://bibsonomy.org) [2]. To evaluate the performance of the recommendations, we randomly divided the dataset into *a training set* (80%) and *a test set* (20%) for each user. We used a 5-fold cross validation scheme. Therefore, the result values reported in the experiment section are the averages over all five runs. We adopted *precision* and *recall* to measure the performance of the recommendations [1].

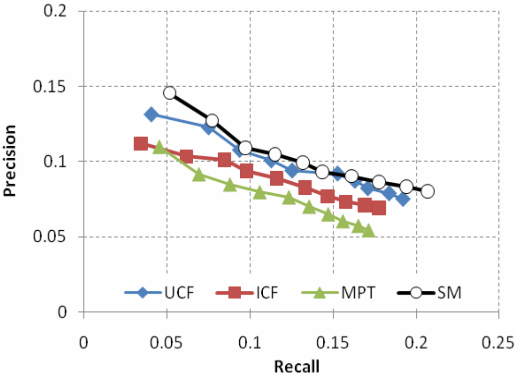


Fig. 1. *Recall* and *precision* as the value of the number of recommended items *N* increases

Prior to the experiment, we measured the performance of *UCF*, *ICF* and SM according to the variation of the neighborhood size. As a result, the best values of the neighborhood size were 20 for *UCF*, 40 for *ICF*, and 30 for SM, respectively.

For evaluation the top *N* recommendation, we measured *precision* and *recall* obtained by *UCF*, *ICF*, *MPT*, and *SM* according to the variation of *N* value from 1 to 10.

Fig. 1 depicts the precision-recall plot, showing how *precisions* and *recalls* of four methods changes as *N* value increases. Data points on the graph curves refer to the number of recommended items. Comparing the results achieved by *SM* and the benchmark methods, both *recall* and *precision* of the former was found to be superior to that of the benchmark methods in all cases. Only *UCF* achieves comparable results on some occasions. An important observation is that *SM* significantly outperforms the other methods when a relatively small number of items were recommended (e.g., top-1, top-2). For example, when *N* is 1 (starting points on the left of the curves), with respect to *recall*, *SM* obtains 1.12%, 1.73%, and 0.64% improvement compared to *UCF*, *ICF*, and *MPT*, respectively. With respect to *precision*, similar results are demonstrated. SM outperforms *UCF*, *ICF*, and MPT by 1.43%, 3.37%, and 3.59%, respectively. That is, *SM* provides more suitable items with a higher rank in the recommended item set, and thus can provide better quality of items for the target user than the other methods. We conclude from the comparison results that our semantic model can provide better performance of recommendations than other methods.

4 Conclusions and Future Work

In this paper, we have presented a semantic model and a method of applying the model to social recommender systems. As noted in our experimental results, our model can successfully enhance the performance of item recommendations. Moreover, we also observed that our approach can provide more suitable items for user interests, even when the number of recommended is small. For the future work, we intend to explore semantic relations of IEML and apply the relations to the semantic item similarity and item recommendations.

Acknowledgment. The work was mainly funded since 2009 by the Canada Research Chair in Collective Intelligence at University of Ottawa.

References

1. Herlocker, J.L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating Collaborative Filtering Recommender Systems. ACM Transactions on Information Systems 22(1), 5–53 (2004)
2. Jäschke, R., Marinho, L., Hotho, A., Schmidt-Thieme, L., Stumme, G.: Tag Recommendations in Social Bookmarking Systems. AI Communications 21(4), 231–247 (2008)
3. Kim, H.-N., Ji, A.-T., Ha, I., Jo, G.-S.: Collaborative Filtering based on Collaborative Tagging for Enhancing the Quality of Recommendation. Electronic Commerce Research and Applications 9(1), 73–83 (2010)
4. Lévy, P.: From Social Computing to Reflexive Collective Intelligence: The IEML Research Program. Information Sciences 180(1), 71–94 (2010)

Winner Determination Based on Preference Elicitation Methods

Farnaz Ghavamifar, Samira Sadaoui, and Malek Mouhoub

Department of Computer Science
University of Regina, Canada
{sadaouis,mouhoubm}@uregina.ca

Abstract. Multi-Attribute Reverse Auction (MARA) is widely used in modern electronic procurement systems. However, eliciting the buyer's preferences and determining the winner are challenging tasks for MARA systems. In this paper, we study and extend two methods, the Multi-Attribute Utility Theory and constrained CP-nets, for solving the winner determination problem in MARA. Thanks to the proposed extensions, the buyer can now express conditional preferences over the product attributes. In addition, he can submit his preferences qualitatively or quantitatively. In this way, we provide the buyers with more flexibility in the specification of their purchasing requests.

Keywords: Winner Determination Problem, Multi-Attribute Utility Theory, Multi-Attribute Auctions, CP-nets.

1 Introduction

In reverse auctions, the sellers compete to win the auction based on the attributes of the item the buyer has announced. Winner determination refers to choosing the best seller who would be awarded the contract for supplying the required item [2]. In Multi-Attribute Reverse Auctions (MARAs), sellers should be ranked based on the buyer's preferences over the item attributes [6]. The winner determination in these auctions is a computationally hard problem which restrains the widespread use of these auction models. Relatively little research has been done about winner determination in MARAs. In Che's auction mechanism bidders place bids based on their own scoring function and the bidder with the highest score according to the auctioneer's scoring function is the winner [5]. Bichler was the first one who proposed multi-attribute auctions based on multi-attribute utility theory (MAUT) [2]. In the iterative auction mechanism defined in [1], the buyer announces an additive scoring function in each round. Through inverse optimization techniques, the buyer learns from the bids of the sellers and updates his scoring function at the end of each round. In all of these mechanisms, a single numerical value is assigned to each bid and then bids are ranked based to these values.

In this paper, we consider a MARA system in which the buyer can specify constraints and conditional preferences over the product attributes. Our contributions to winner determination problem are as follows. First we enable the winner determination in case of conditional preferences. The buyers may have conditional preferences

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 332–335, 2010.

over attributes, e.g. when the buyer's choice in the brand of a laptop is depended on its price. Second we allow the buyers to choose between quantitative and qualitative preference expression.

The rest of this paper is structured as follows. In Section 2 we describe MAUT and the constrained CP-net and subsequently introduce the improved algorithms, MAUT* and CP-net*. In Section 3, we present the implementation of our auction system based on MAUT* and CP-net*. We also explore experimentally how each method chooses the winner. Finally, concluding remarks and future works are reported in Section 4.

2 Improving MAUT and CP-Net

The most widely used technique for multi attribute decision making is the Multi Attribute Utility Theory (MAUT) [2]. When we apply MAUT to MARA, qualitative and conditional preferences and constraints cannot be shown. For winner determination it is necessary to first check the constraint consistency of the bids. In the next step from the set of feasible bids, the best bid is selected based on the buyer's preferences. To have conditional preferences, the buyer can specify the utility function of an attribute which has conditional preferences based on the value assigned to the other attributes it depends on. Suppose that the preference of the buyer for attribute j is conditional and depends on the value of attribute k. Then for evaluating attribute j, the utility function $Uj(v_{ij})$ is used and has different values based on v_{ik} (value of attribute k in bid i). We add these features to MAUT and call the new method MAUT*.

Constrained CP-nets is another method which can be used to solve multi-attribute decision problems. Boutilier et al. proposed the CP-net, a graphical representation of conditional and qualitative preference relations [3, 4]. Sometimes, there may be some soft and hard constraints that we wish to add to our CP-net. For this reason, constrained CP-net with some additional constraints on assignments is introduced [3]. To find the best bid in MARA, a total ordering of bids is needed. However CP-net does not have full ordering over outcomes. If two outcomes do not have paths in the preference graph, they cannot be compared. To solve this problem, the number of buyer's preferences which are violated can be used as a good clue to compare the outcomes which do not have any path in the preference graph. This means that if the value assigned to an attribute is not what is preferred, this assignment is not desired by the buyer and it is a preference violation. In the case that the number of violations of two outcomes is the same, we can ask the buyer to complete his preferences. We add these two ways of outcome comparison to the CP-net and call the new method CP-net*.

3 Implementation and Experimentation

We implemented our tool to assist the buyers when submitting their preferences and constraints and the sellers when biding. Our tool has been fully implemented in Java (J2E version 1.6) with a total of 3000 lines of code and ten classes. The user interface is implemented using powerful features and reliable development environment of the Netbeans IDE version 6.0. Our tool provides the following functions:

1. Submit the Request of Purchase. After the registration, the buyer should submit the request for purchase. In our experimentation, the requested good is a laptop and the buyer should specify the technical information of this product.

2. Invite the Sellers to the Auction. In this step, the sellers are invited to register and they need to provide their capabilities in producing the item.

3. Submit Preferences and Constraints. Among all the provided attributes, the buyer selects Price, RAM, Weight and Brand. Based on the available seller's offers, the domains of the attributes are generated as follows:

- D_{Brand} = {Dell, Sony, Toshiba}, D_{Weight} = [3,4] lb, D_{RAM} = {1, 2, 4} GB, D_{Price} = [500, 1100] Dollars

Then the buyer submits his preferences and constraints as follows:

- *p1:* RAM and Weight are more important to the buyer than Price and Brand.
- *p2*: Price is more important than Brand.
- *p3*: The buyer prefers the highest RAM size.
- *p4*: The buyer prefers the lightest laptop.
- *p5*: The buyer prefers to pay more than $1000 if RAM is at least 4GB and Weight is less than 3lb. Otherwise, he prefers the cheapest laptop.
- *p6*: the buyer prefers Dell more than Sony and Sony more than Toshiba if price is more than $900. Otherwise, he prefers Sony more than Toshiba and Toshiba more than Dell.

- *c1*: If Weight is equal or more than 4lb, the buyer does not buy Sony.
- *c2*: If Price is more than $800, the buyer wants to buy a laptop with higher than 1 GB of RAM.

Table 1. Sellers' Bids

	Brand	Weight	RAM	Price
Seller1	Toshiba	3.5lb	2GB	$890
Seller2	Sony	4lb	1GB	$680
Seller3	Dell	2.2lb	2GB	$1100
Seller4	Sony	3.5lb	4GB	$750
Seller5	Toshiba	4lb	1GB	$810

4. Submit Bids. In this step, sellers enter their laptop configuration as it is shown in Table 1.

5. Determine the Winner. According to the buyer choice one of the following two methods is used to determine the winner:

- MAUT*: For determining the winner in this method, first consistency of all the bids with the buyer's constraints is checked. If a bid violates any of these constraints, the bid is not feasible and is deleted. In the next step, the overall utility of each remaining bids is calculated and finally the bid with the highest utility is the best bid.

- CP-net*: Similar to the MAUT* the consistency of bids is checked. After that each of the two bids are compared respectively based on the path in the induced graph and number of preference violations. If still the best bid cannot be found, we need to ask the buyer to complete his preferences.

As it is shown in Table 2, MAUT and CP-net cannot determine the winner, however MAUT* chooses Seller4 as the winner. When CP-net* is used, the buyer has to complete his preference model. If the buyer prefers to have a lower Weight rather than a larger RAM, Seller3 is the winner, otherwise Seller4 wins the auction.

Table 2. Winner Determination

	MAUT	CP-net	MAUT*	CP-net*
Winner	×	×	Seller4	Seller3: Weight has priority. Seller4: RAM has priority.

×: cannot be determined.

4 Conclusion and Future Work

In this paper, we first analyzed two methods, MAUT and constrained CP-net, to determine the winner when preferences are respectively quantitative and qualitative. We showed that none of these two techniques is ideally suitable for solving the winner determination problem in MARA systems. Subsequently, we extended these two methods with some solutions and implemented our auction system based on these methods. We then explored experimentally how each method chooses the winner.

Our future work consists of improving CP-net* by incorporating quantitative preferences which will enable the buyers to express their preferences over some attributes quantitatively and over others qualitatively.

References

1. Beil, D.R., Wein, L.M.: An inverse-optimization-based auction mechanism to support a multiattribute RFQ process. Management Science 49(11), 1529–1545 (2003)
2. Bichler, M., Kalagnanam, J.: Configurable offers and winner determination in multi-attribute auctions. European Journal of Operational Research 160, 380–394 (2005)
3. Boutilier, C., Brafman, R.I., Domshlak, C., Hoos, H.H., Poole, D.: Preference-based Constrained Optimization with CP-nets. Computational Intelligence 20(2), 137–157 (2004)
4. Boutilier, C., Brafman, R.I., Domshlak, C., Hoos, H.H., Poole, D.: CP-nets: A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements. Journal of Artificial Intelligence Research 21, 135–191 (2004)
5. Che, Y.K.: Design competition through multidimensional auctions. RAND Journal of Economics 24, 668–680 (1993)
6. Zhang, J., Pu, P.: Survey of Solving Multi-Attribute Decision Problems. EPFL Technical Report No: IC/2004/54, June 17 (2004)

A Domain Ontology Model for Mould Design Automation

Ziad Kobti[1], Ding Chen[1], and Alan Baljeu[2]

[1] Department of Computer Science, University of Windsor, 401 Sunset Avenue,
Windsor, ON, Canada N9B-3P4
{kobti,chen111t}@uwindsor.ca
[2] Cornerstone Intelligent Software Corp., 2245 Farser Ave.,
Windsor, ON, Canada N8X-3Z6
alanb@corintsoft.com

Abstract. An ontology-based search model with semantic distance measures is proposed to improve the traditional keyword-based search for the mould design domain. The model has three components. First an NLP component is used to extract independent concepts from text with keywords extracted from sentences. Next, the ontology layer is built to process concepts with minimal total semantic distance to all these keywords found with a ranking algorithm. Finally, the concepts in the ontology are mapped to the concepts in a proprietary database to implement the matching process from sentences to database concepts; enabling integration with existing mould design software. The ontological search is compared against traditional keyword based search in the mould design domain and showed more fault tolerance and flexibility in maximizing the accuracy and number of detected matches.

Keywords: Semantic Search, Knowledge Retrieval, Ontological Modeling, Domain Knowledge, Natural Language Processing, Mould design automation.

1 Introduction

This work is motivated by a collaborative research project between researchers at the University of Windsor and Cornerstone Intelligent Software Corp. The latter is a leading company in mould design automation software development and research. Mould design involves an extensive manual process that starts with libraries of standards of user specific documentations describing various part details to be generated by the mould being designed. It is then up to the mould designers with extensive industrial experience in the domain to comb through the specification documents and translate the client requirements into meaningful design for a mould that matches the desired specifications. It is typical for domain experts to spend a significant amount of time in manual labor going from specification to a digital design model ready for manufacturing in tool and die shops.

The aim of this project is to build smart tools that can help mould designers to save time and cost in the process of converting mould specification to a computer design model. This paper reports the initial phase of this project in presenting the tool and underlying algorithms for knowledge extraction. The goal is to find the maximum

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 336–339, 2010.

correct matches extracted from general English specification documents into a proprietary design database. The initial research is being conducted in the domain of mould design and manufacturing specifications, but we expect to be able to adapt this technology to acquire and manage specifications for other industry.

The use of ontology to overcome the limitations of keyword-based search has been put forward as one of the motivations of the Semantic Web since its emergence in the late 90's [1]. While some approaches such as word similarity calculation [2], keywords expansion [3] and spread activation [4] have been proposed in the last few years, most achievements so far cannot meet our special requirements for practical use. Thus instead, we propose an ontology-based search model to extract the concepts from specifications written in natural language and help users to find the matching concepts in small data repositories.

2 Related Work and Semantic Search

With the development of the Semantic Web, there have been contributions to improve document keyword search. TAP [5] presents a view of the Semantic Web where documents and concepts are nodes in a semantic graph. Besides searching the keywords in the document context, some additional metadata attached to the document are also searched to strengthen the searching ability. In [4] a search architecture that combines traditional search techniques with spread activation techniques shows that more possible results can be found. However, these techniques are mainly developed to exploit large document repositories, usually Web pages. For small repositories such as the database of a company, the search targets may not be documents. Single words or phrases are all possible input to be the search targets. In this case, these techniques are not fit because their effectiveness relies on the information contained in search targets, which single word or phrase cannot provide. Moreover, the lack of a ranking method for the search results also weakens the search precision.

Instead of enhancing the data repositories, another widely used method to improve search effectiveness is query expansion. The main aim of query expansion (also known as query augmentation) is to add new meaningful terms to the initial query [3]. In [6] relevance feedback as ranking method is used, and in [7] we learn that good quality expansion terms can only be generated if the original document collection contains a large number of relevant documents. WordNet is adapted in [8] as a knowledge base to expand a query and argues that query expansion is suitable for short queries. Thus, in solving the problem of small repositories and long queries, query expansion has many inconveniencing drawbacks in practice. Richardon and Smeaton [2] tackle the problems posed by the richness of natural language by calculating semantic similarity and computing the semantic distance between concepts and words. WordNet however is not suitable in a specific mould domain.

3 Proposed Ontology Based Search Model

The proposed ontology-based search evolves the search from a heuristic using traditional keyword-based search methods, and then moves towards a more semantic contextual search supported by a domain specific ontology. The overall architecture is based on three functional components: the natural language processing module (NLP), the knowledge base maintenance, and the search engine. The goal is to identify the concepts in a given document written in natural language and find the same or corresponding concepts in data repository by using the Ontology.

The search engine is composed of two functions: searching and ranking. The concept graph parsed from the ontology and the query generated by NLP component will be the inputs for the search function. For a given query, the search engine matches the nouns in the query with the node names in the concept graph. After this step, all the following steps are operated on the concept graph. Basically, the aim of the search is to find the concept with the minimum total semantic distance to all the concepts found by matching the nouns in the query. There are four steps to implement the search process for a given concept graph G and a query containing nouns group {k1, k2, k3….kk}. First of all, all the nodes in G whose node name is equal to any one in nouns group are selected as a concept group {c1, c2, c3 …. cm}. In the next step, the semantic distances from c1 to all the other nodes in G are calculated by using Dijkstra's algorithm and recorded. The same step is conducted on c2….cm. After all the semantic distances are calculated, for each node in G, the total semantic distance between the node and {c1, c2, c3 …. cm} is calculated.

In the ranking function, we suppose the node with minimum total semantic distance is the most closely related node to the given query. Thus total semantic distances of the nodes in *G* are sorted by ascending order to generate search result candidate list. There are two kinds of nodes in *G*. Key concepts are nodes from database and they are our final target. So the nodes for related concepts in candidate list are filtered and the remaining part of the list is our final result list.

4 Results and Conclusions

The ejection process of mould engineering [9] is selected as our experiment domain because it is relatively specific and independent. A semantic graph is created for this process containing 92 concepts: 44 key concepts and 48 related concepts. A specification about ejection process provided by Cornerstone Intelligent Software Corp. is taken as the query document for the experiment. This query document contains 41 sentences and can be grouped into 14 independent concepts. The goal of our search is to identify the concepts in the specification and match them to the concepts in the proprietary data repository. We assume one unique match per query.

For a formal and ambiguity-free document, our ontology-based search approach is disappointedly not as efficient as traditional search. Although for more than 85% cases we can find the right result but not as the first candidate in the result list. This problem may result from the ranking algorithm. The usage of compound word recognition improves both ontology-based search and keyword-based search as we expected. However, it improves significantly on ontology-based search. For an ambiguous document, keyword-based search effectiveness is severely decreased while our ontology-based can still work well. In practical use, most of documents in the mould domain cannot be formal and ambiguity-free, so our flexible and fault tolerant search model will be more useful than traditional keyword-based search.

According to the results our approach improves the search results and is more flexible and fault tolerant than traditional keyword-based search. Consequently, this demonstrates that this approach can appropriately solve the ambiguity problem brought by the practical use. Due to the separation between the search and the domain ontology, this approach can potentially work with a different domain, making it portable to other domains. Further improvements can be made on several directions including improving the ontology and adapting more suitable searching or ranking algorithms.

Acknowledgements

This work was supported by a grant from the Ontario Centers of Excellence and NSERC Discovery, with matching funds from Cornerstone Intelligent Software Corp.

References

1. Vellet, D., Fernández, M., Castells, P.: An Ontology-Based Information Retrieval Model. In: Gómez-Pérez, A., Euzenat, J. (eds.) ESWC 2005. LNCS, vol. 3532, pp. 455–470. Springer, Heidelberg (2005)
2. Richardson, R., Smeaton, A.F.: Using WordNet in a Knowledge-Based Approach to Information Retrieval. Technical report ca-0395, Dublin City University, School of Computer Applications (1995)
3. Bhogal, J., Macfarlane, A., Smith, P.: A review of ontology based query expansion. Information Processing & Management 43, 866–886 (2007)
4. Rocha, C., Schwabe, D., Aragao, M.P.: A Hybrid Approach for Searching in the Semantic Web. In: 13th International Conference on World Wide Web, pp. 374–383. ACM, New York (2004)
5. Guha, R., McCool, R., Miller, E.: Semantic Search. In: 12th International Conference on World Wide Web, pp. 700–709. ACM, New York (2003)
6. Tombros, A., Sanderson, M.: Advantages of Query Biased Summaries in Information Retrieval. In: 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 2–10. ACM, New York (1998)
7. Ruthven, I., Tombros, A., Jose, J.M.: A Study on the Use of Summaries and Summary-based Query Expansion for a Question-answering Task. In: 23rd BCS European annual colloquium on IR research, ECIR 2001 (2001)
8. Navigli, R., Velardi, P.: An analysis of ontology-based query expansion strategies workshop on adaptive text extraction and mining. In: 14th European conference on machine learning, ECML 2003 (2003)
9. Rees, R.: Mould Engineering, 2nd edn. Distributed by Hanser Gardner Publications, Inc., Ohio (2002)

Dynamic Reasoning for Description Logic Terminologies

Stanislav Ustymenko[1] and Daniel G. Schwartz[2]

[1] Meritus University, School of Information Technology,
30 Knowledge Park Drive (Suite 301),
Fredericton, New Brunswick, Canada E3C 2R2
sustymenko@meritusu.ca

[2] Department of Computer Science, Florida State University,
Tallahassee, Florida, USA
schwartz@cs.fsu.edu

Abstract. The Semantic Web presents the challenge of designing agents capable of continuously updating their knowledge bases. Semantic Web ontologies are commonly represented using description logic knowledge bases. We demonstrate description logic reasoning using a Dynamic Reasoning System (DRS). This explicitly portrays reasoning as a process taking place in time and allows for manipulating inconsistent knowledge bases.

Keywords: Description logics, belief revision, dynamic reasoning systems.

1 Introduction

The Semantic Web (SW) is a common name of a family of technologies extending the Web with rich, machine-interpretable knowledge. It supports rich metadata annotation, including expressive ontology languages. Description Logics (DLs) [1] emerged as leading formalism for knowledge representation and reasoning on the Semantic Web. Once widely implemented, the Semantic Web agents will need a way to absorb new knowledge in a timely fashion, all the while protecting the consistency of their knowledge bases.

Dynamic Reasoning Systems (DRS) provide a formal framework for modeling the reasoning process of an agent that explicitly portrays reasoning as an activity that takes place in time [4]. It sidesteps the logical omniscience assumption of the classical AGM belief revision framework [3] and has means for working with inconsistent knowledge bases by keeping track of a *derivation path*.

A DRS can be defined for any language. DLs present a challenge in that they do not have explicit derivation rules. Instead, DLs rely on *inference algorithms* to accomplish common reasoning tasks. The goal of this paper is to present the DRS framework as a suitable formalism for Semantic Web reasoning. To this end, we give an instance of DRS capable of building a concept subsumption hierarchy for a well-known description logic.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 340–343, 2010.

2 Preliminary Definitions

Languages for any description logic contain *concept names, role names*, and *individual names*. We will use uppercase A and B for concept names, uppercase letters R, P for role names, and lowercase x, y, z for individual names. DL languages combine role and concept names into *concept definitions*. Concepts of a description logic $\mathcal{ALCN}$ [2] are defined recursively: atomic concept A, universal concept $\top$, ground concept $\bot$, concept negation $\neg C$, intersection $C \sqcap D$, union $C \sqcup D$, value restriction $\forall R.C$, limited existential quantification $\exists R.\top$, number restriction $\leq nR, \geq nR$ are concept definitions, where C, D denote concept definitions. Concepts are given a natural set-theoretic interpretation (see [1]).

Description Logic knowledge bases consist of two components: a TBox, a set of statements about concepts, and an ABox, a set of assertions about individuals. In general, a TBox T contains *general concept inclusion axioms* $C \sqsubseteq D$. The pair of axioms $C \sqsubseteq D, D \sqsubseteq C$ is abbreviated $C \equiv D$ (equality axiom). A *terminology* is a TBox that consists of equality axioms with atomic left hand side.

An ABox contains assertions regarding individual names. These include concept assertions $C(a)$ and role assertions $R(a, b)$.

The notion of a DRS is obtained from the conventional notion of formal logical system by lending special semantic status to the concept of a derivation path (i.e., a proof). Introduction of new knowledge or beliefs into the path occurs in two ways: either new propositions are added in the form of axioms, or some propositions are derived from earlier ones by means of an inference rule. In either case, the action is regarded as occurring in a discrete time step, and the new proposition is labeled with a time stamp (an integer) indicating the step at which this occurred. Moreover, for propositions entered into the path as a result of rule applications, the label additionally contains a record of which inference rule was used and which propositions were employed as premises. The core ideas were presented in [4] and have been revised and clarified in [5].

3 Dynamic Reasoning for DL $\mathcal{ALCN}$

A DRS comprises a framework for modeling the knowledge base and reasoning processes of artificial agent. We describe an agent that extracts ontological knowledge from the Web and uses it to support a user's browsing and querying activities. Thus, our DRS needs to support two DL reasoning tasks:

1. Check if a defined concept A is satisfiable
2. Deduce atomic subsumption, that is, a statement of the form $A \sqsubseteq B$

To construct the DRS, we first note that if A and B are defined by axioms $A \equiv C, B \equiv D$, where C, D are concept definitions, then $A \sqsubseteq B$ iff $C \sqsubseteq D$. Second, note that $C \sqsubseteq D$ iff $C \sqcap \neg D$ is unsatisfiable. Therefore, both our reasoning tasks would require checking satisfiability of concepts.

Now we can build our DRS. First, we define the language, L. The symbols of L are the same as the symbols of logic $\mathcal{ALCN}$. We use A, B for concept names occurring in the incoming statements and A', B' for the names introduced by the agents. The formulas of L include definitorial $\mathcal{ALCN}$ terminologies, atomic subsumption statements of the form $A \sqsubseteq B$ (arcs of the subsumption tree), ABox assertions $C(a), R(a, b)$ and explicit inequality assertions $a \neq b$. Wlog, we assume that all concepts are in negation normal form.

Then we define inference rules. Implicitly, every rule that modifies a concept definition also puts the result into negation normal form. The inference rules will be:

1. *Substitution*: from $A \equiv C$ and $B \equiv D$ infer $A \equiv E$, where E is C with all occurrences of B replaced by E.
2. *Subsumption test introduction*: from $A \equiv C, B \equiv D$ infer $A' \equiv C \sqcap \neg D$, where A' is a new agent-generated concept name.
3. From $A \equiv C, B \equiv D$ and $A' \equiv \bot$, provided that name A' was introduced using Rule 2 as above, derive $A \sqsubseteq B$.

The following rules 4-10 are added to enable tableau-based consistency checks. These are derived from the transformation rules listed in [1], p. 81.

4. From $A \equiv C_0$, infer $C_0(x_0)$
5. From $A \equiv C_0$ and $(C_1 \sqcap C_2)(x)$, infer $C_1(x)$ and $C_2(x)$
6. From $A \equiv C_0$ and $(C_1 \sqcup C_2)(x)$, infer $C_1(x)$ or $C_2(x)$
7. From $A \equiv C_0$ and $(\exists R.C)(x)$, infer $C(y)$ and $R(x, y)$, where y is a new generated name
8. From $A \equiv C_0$, $(\forall R.C)(x)$ and $R(x, y)$, infer $C(y)$
9. From $A \equiv C_0$ and $(\geq nR)(x)$, infer $R(x, y_1), \ldots, R(x, y_n)$ and $y_i \neq y_j$, and $R(x, y)$
10. From $A \equiv C_0$ and $(\leq nR)(x)$, if $R(x, y_1), \ldots, R(x, y_n)$ are in the derivation path and $y_i \neq y_j$ is not in the path for some $i \neq j$: replace all occurrences of y_i with y_j.

The following rules 11-13 detect inconsistency in ABoxes built using rules 4-10. As above, statements are derived from $A \equiv C_0$:

11. From $A \equiv C_0$ and $\bot(x)$, derive $A \equiv \bot$
12. From $A \equiv C_0$, $A_1(x)$ and $\neg A_1(x)$, derive $A \equiv \bot$
13. From $A \equiv C_0, (\leq nR)(x)$, set $\{R(x, y_i) | 1 \leq n + 1\}$ and set $\{y_i \neq y_j | 1 \leq i \leq j \leq n + 1\}$, derive $A \equiv \bot$

Finally, rule 14 derives a subsumption axiom, using reduction to unsatisfiability:

14. From $A \equiv C, B \equiv D, A_1 \equiv C \sqcup \neg D, A_1 \equiv \bot$, derive $A \sqsubseteq B$

A Dynamic Reasoning System based on language L and rules 1-14 is capable of supporting an agent that builds an explicit subsumption hierarchy. An agent starts with an empty derivation path and empty subsumption hierarchy. It will receive TBox definitions from the user. Before receiving the first axiom, the controller will enter a root concept, $R \equiv \top$, as a first formula in the path.

Upon entering a new axiom of the form $A \equiv C$, the controller will perform the following actions:

1. Derive an expanded definition of A by repeatedly employing Rule 1.
2. Test satisfiability of A using Rules 4-13. If it is unsatisfiable, flag it for a belief revision procedure.
3. Expand all (extended) definitions that depend on using Rule 1. Test the affected concepts' satisfiability, flagging for a belief revision process if unsatisfiable. Update the hierarchy of concepts affected by this step, testing subsumption by using Rules 2-14.
4. Place A into its appropriate place in the subsumption hierarchy, using Rules 2-14 to test subsumption.

To test satisfiability by employing Rules 3-13, an agent follows a tableau-based algorithm. Details of the appropriate algorithm, with discussion of termination and complexity, can be found in [2].

Rules 6 and 10 are *non-deterministic*: for a given ABox, they can be applied in finitely many different ways, leading to finitely many ABox'es. The controller may handle branches by setting the belief status of statements on inactive branches to *off*.

We did not specify the details of modifying subsumption hierarchy in steps 3 and 4. In principle, the controller may simply search the existing hierarchy starting at the root, testing the concept in question's subsumption with each node. Methods for assisting the user or for achieving this task without user interaction can be developed, based on research in ontology debugging and belief revision for description logics [eg., 6]. Developing such methods is a task left for future research.

References

1. Baader, F., Nutt, W.: Basic description logics. In: Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, p. (eds.) The Description Logic Handbook. Cambridge University Press, Cambridge (2003)
2. Baader, F., Sattler, S.: Expressive number restrictions in Description Logics. J. of Logic and Computation 9(3), 319–350 (1999)
3. Gärdenfors, P.: Knowledge in Flux: Modeling the Dynamics of Epistemic States. MIT Press/Bradford Books, Cambridge (1988)
4. Schwartz, D.G.: Dynamic reasoning with qualified syllogisms. Artificial Intelligence 93(1-2), 103–167 (1997)
5. Schwartz, D.G.: Formal specifications for a document management assistant. In: Proc. International Conference on Systems, Computing Sciences and Software Engineering (CISSE 2009), University of Bridgeport, CT, December 4-12 (2009)
6. Qi, G., Liu, W., Bell, D.A.: Knowledge Base Revision in Description Logics. In: Fisher, M., van der Hoek, W., Konev, B., Lisitsa, A. (eds.) JELIA 2006. LNCS (LNAI), vol. 4160, pp. 386–398. Springer, Heidelberg (2006)

Deep Distributed News: Ontologies to the Rescue of Journalism

Alexandre Cayla-Irigoyen and Esma Aïmeur

Department of Computer Science and Operations Research, University of Montreal,
Québec, Canada
Alexandre.Cayla-Irigoyen@Umontreal.ca,
Aimeur@Iro.Umontreal.ca

Abstract. New media such as the Internet has greatly expanded the informational horizons of news consumers. However, it has brought about its own set of problems such as information overload. In this paper, we suggest that the semantic web is the best means to help news consumers regain control of the news. We posit that by modeling the environment in which social activity takes place (and news events take place) it is possible to clarify the links between news items and the general context. Additionally, by comparing this model to the user's own conceptualization, news content could be adapted to better fit his or her needs. We call this theoretical model Deep Distributed News.

Keywords: Knowledge Representation, Ontologies, Journalism.

1 Introduction

New media such as the Internet has greatly expanded the informational horizons of news consumers. However, information abundance can sometimes be synonymous of information overload and while tools such as search engines help greatly in organizing the vast sea of information that is the Web, they still are insufficient solutions in regards to news and its meaning. High-speed news cycles, redundant information and the multiplication of sources of news (news outlets, governments, NGOs, etc.) have transformed what was once a manageable quantity of information into a torrent that news consumers struggle to contain. New services have arisen to alleviate these problems. However, they only solve part of the equation. Fox example, Google News aggregates closely related or identical news stories, Techmeme aggregates "conversations", Google's Living Stories tries to offer context and continuity to news stories. Often, they offer a "one size fits all" solutions without the possibility of personalization and a few sources, such government data, are usually left out. A real solution would aggregate and integrate news and information and enable news consumer to tailor the news consumption experience to their needs. The theoretical model for online news we present, *Deep Distributed News* (DDN), by relying on semantic web ontologies, should permit this. We posit that by placing news items in a general model of the social environment, it should be possible to tame the chaotic flow of information that characterizes the news ecosystem and personalize the news consumption

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 344–347, 2010.

experience. It is called *Deep* because it should provide news consumers with a deeper understanding of the news, and *distributed* because, rather than centralizing all the information in one place, our model integrates information that is published by a variety of sources across the web.

2 News Consumption and Representation

News (information about recent events) and information (e.g. context, facts, reactions, etc.) are the two sides of the same coin. They cannot and should not be seen as separate since, fundamentally, news denotes change and change must be measured *from* something. This point of reference is the social representation of reality. The objective of a news item is not to convey all the necessary information, but to provide cues to help the reader mentally reconstruct a model of the situation and update it [1]. When the reader cannot do this, contextual knowledge has to be provided. Luckily, even if human activity changes incessantly, it is still quite predictable (many domains of our lives are structured and follow specific rules) and thus, can be modeled easily. On top of providing general context for news items, by identifying this underlying structure and modeling it, it becomes possible to aggregate and integrate various bits and pieces of information originating from different sources into a coherent whole. Semantic web technologies, most notably ***web ontologies,*** are used to accomplish this.

An ontology defines a set of representational primitives with which to model a domain of knowledge or discourse. These representational primitives are typically classes (or sets), attributes (or properties), and relationships (or relations among class members). The definitions of these representational primitives include information about their meaning and constraints on their logically consistent application [2]. More interestingly, by making ontological commitments (agreements to use the shared vocabulary), agents sharing a vocabulary do not need to share a knowledge base to be consistent with one another as a commitment to a common ontology is a guarantee of consistency [3]. This implies that using an ontology for the online news ecosystem makes it possible to integrate a disparate body of knowledge as long as the different information providers commit to the same vocabulary.

3 Deep Distributed News

Our model, Deep Distributed News, has three parts: *a user module, an environmental module and a news module*. The most important part of our model is the representation of the boundaries within which social activity takes place (and thus news is generated); this is the *environment module* and it constitutes the starting point from which change is measured. The *news module* is responsible for “mapping” this change. Lastly, the *user module* serves essentially as a filter for deciding which information should be shown and can be used to personalize the news experience. They are illustrated in Figure 1.

Environment Module. The environment in which news occurs is, essentially, society and society is governed by rules, institutions, actors, etc. This society is composed of different *domains* in which different activities take place. The domains are composed

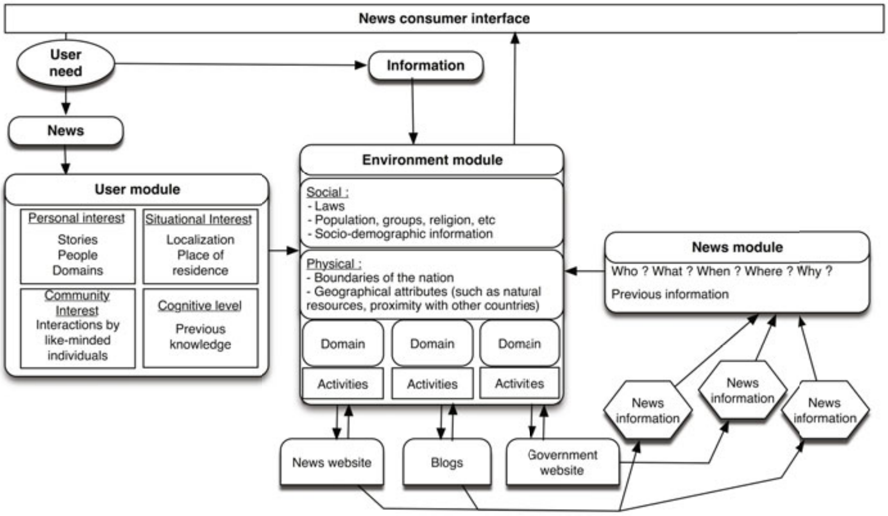


Fig. 1. Modeling Deep Distributed News

of different *actors* who take part in *activities*. For example, the domain for a redevelopment project is urban planning, its actors are political parties, citizens, governments, etc. and an activity can be the negotiation of a contentious part the project. Sometimes, usual activities are disrupted by important *events* (for example, a breakthrough in negotiations) sometimes they change slowly until we can say important *developments* have occurred. *These two last elements (events and developments) are what we generally consider to be news or newsworthy***.** The news module manages them.

News Module. The news module seeks the information instantiated by the environment module, compares it to the new information (who? what? when? why? and where?) and returns the information needed by the consumer to understand the new news item; if he had been following the story for quite some time, only the recent developments would be shown. If not, the system might propose some information about the context. A *news event* is the starting point of a deviance from this usual course of things it can be a single (implementation of a new policy) or a series of events (diplomatic trip with a series of meetings) as long as it has an impact (good or bad) on the field. *Developments* measure slower changes. For example, in the case of manufacturing, there are multiple measures indicators available: the number of jobs in the sector, the type of work the workers accomplish and the products they produce, etc. Developments are important because even if they do not have a direct impact on the domain at hand, they give us as clear a picture as we can get.

User Module. The last part of the model is the user model; it enables the personalization of the news consumption experience. Three types of personalization could be put in place based on *personal, situational and community interests.* These are based on information contained in a user profile. *Personal interests* are quite clear: they are the subjects and the domains the user wants explicitly to follow (e.g. technology or politics). In situational personalization, the user is not as active. Socio-economic and geographical information is extracted from a profile to determine what news items should be presented. An example would be information relating to their city, borough and general social and political news. *The keyword here is membership to a group or a place.* The last type of personalization is *community interests.* It is done by the community or rather, like minded individuals. It is akin to the, *"users who bought this book, also bought…"* functionality.

The interest of this method is that it considers the news item (rather than the news text) and its context as the fundamental block of a news website. A news text is finite whereas new information can always be instantiated in a domain model. It also makes it easier to connect information pieces through time. For instance, in the United States, healthcare reform is a major issue. In a traditional article, the number of uninsured would be mentioned (41 million, for example). However, if the government keeps a count and updates its numbers every month or two months, it could be a lot more telling to show the progression of the number of uninsured since the beginning of the process of healthcare reform. If, for example, the number has grown between 5% and 10% each month for the last seven months, there would be little need to explain the costs of inaction. The increase could also be correlated to other events.

4 Discussion and Limitations

In this paper, we outlined the basic problems of journalism in the current news ecosystem, asserted that using ontologies as a reference point provides the means to aggregate and integrate information originating from different actors and/or at different moments because information is individualized, but instantiated in a given structure that can be made visible and interacted with by the news consumer. The development of our proposed architecture is the first of many steps needed to achieve this goal. Modeling and creating an easy way for online sources to mark up their information is essential for the success of our proposal. Luckily, many domains of interest have already an implicit structure waiting to be modeled explicitly. This is as true for sports as it is for urban planning.

References

1. Yaros, R.: Is It the Medium or the Message? Structuring Complex News to Enhance Engagement and Situational Understanding by Nonexperts. Comm. Rrch. 33, 285–309 (2006)
2. Gruber, T.: Ontology. In: Liu, L., Tamer Özsu, M. (eds.) Encyclopedia of Database Systems. Springer, Heidelberg (2009)
3. Gruber, T.: A translation approach to portable ontology specifications. Knowledge Acquisition 5, 199–220 (1993)

Improving Bayesian Learning Using Public Knowledge

Farid Seifi[1], Chris Drummond[2], Nathalie Japkowicz[1], and Stan Matwin[1,3]

[1] School of Information Technology and Engineering
University of Ottawa
Ottawa Ontario Canada, K1N 6N5
fseif050@uottawa.ca, nat@site.uottawa.ca, stan@site.uottawa.ca
[2] Institute for Information Technology,
National Research Council Canada,
Ottawa, Ontario, Canada, K1A 0R6
Chris.Drummond@nrc-cnrc.gc.ca
[3] Institute for Computer Science,
Polish Academy of Sciences, Warsaw, Poland

Abstract. Both intensional and extensional background knowledge have previously been used in inductive problems to complement the training set used for a task. In this research, we propose to explore the usefulness, for inductive learning, of a new kind of intensional background knowledge: the inter-relationships or conditional probability distributions between subsets of attributes. Such information could be mined from publicly available knowledge sources but including only some of the attributes involved in the inductive task at hand. The purpose of our work is to show how this information can be useful in inductive tasks, and under what circumstances. We will consider injection of background knowledge into Bayesian Networks and explore its effectiveness on training sets of different sizes. We show that this additional knowledge not only improves the estimate of classification accuracy, it also reduces the variance in the accuracy of the model.

Keywords: Bayesian Networks, Public Knowledge, Classification.

1 Introduction

While standard machine learning acquires knowledge from instances of the learning problem, there has always been interest in a more cognitively plausible scenario in which learning - besides the training instances - also utilizes background knowledge. In many inductive problems, the training set, which is a set of labeled samples, could be complemented using intensional or extensional background knowledge in order to improve the learning performance [4,9]. In Inductive Logic Programming, intensional background knowledge is provided in the form of a theory expressed in logical form. In Semi-Supervised Learning, the extensional background knowledge is provided in the form of unlabeled data.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 348–351, 2010.

In this research, we propose to explore a different type of intensional background knowledge. In many domains, there exist publicly available very large, and related, datasets, for example from demographics and statistical surveys. This sort of information is ubiquitous: it is published by many national governments, international organizations, and private companies. Such datasets may not have exactly the same attributes as the dataset we are studying. However, using an intensionalising process [5], we can derive intensional background knowledge, in the form of distributions, from this extensional background knowledge, given as collections of instances. A question that we consider, here, is if it is possible to use such information to improve machine learning methods.

Let us consider a simple medical example. Suppose we are learning a model for the prediction of heart attacks in patients. The data used in the inductive learning of this model may include attributes describing sleep disturbance, as a disease outcome, and stress, as a disease, but does not include enough instances to relate these attributes in a statistically significant way. There exists, independently of the data used in model building, a large medical survey that describes quantitatively sleep disturbance in patients who experience cardiac problems or stress. We assume that this dataset could be used in learning a better predictive model, capturing the important relationship between sleep disturbance, stress, and a heart attack, if we can integrate the data from the medical study with the data we are using in learning the predictive model.

The big challenge in this research is how such background knowledge can be integrated with existing datasets. Bayesian learning is a natural candidate as it draws on distributional data for its assessment of the probabilities of an instance belonging to different classes of the concept. In Bayesian Networks the attribute inter-relationships are encoded into a network structure. We propose, here, to replace parts of this structure - some of the conditional probability distributions - with more accurate alternatives, which are available as background knowledge contained in large public datasets, e.g. statistical surveys. A more detailed version of our work is available in [8].

In a Bayesian network [7,6], there is a structure which encodes a set of conditional independence assumptions between attributes; a node is conditionally independent of its non-descendants given its parents. Also, there are conditional probability distributions capturing each attribute's dependency on others, typically represented by multi-dimensional tables. Together, these define the joint probability distribution of the attributes and class. With such a distribution, we can use Bayes rule to do inference. There exist many different ways of building Bayesian networks from training data. We used the software package *BN predictor* [2,3] to build the network and used a maximum likelihood estimator (frequency counts) to construct the tables.

Normally, we obtain the conditional probability distributions which we use in the Bayesian Network inference from the training set. If we do not have enough training data samples, our estimates of the true distribution will be poor and the result will not be an accurate classifier. These distributions are independent, so it could be possible to improve the performance even by replacement of a few of them with accurate alternatives, obtained from statistical surveys.

2 Experiments and Discussions

For finding out whether the replacement of a selected set of distributions makes a significantly better classifier or not we run a set of experiments. In each experiment a large dataset as well as training and testing sets are sampled from the huge dataset which represents the universe. Then the classifier is trained using the small training set. More specifically, potentially inaccurate conditional probability distributions are built from the training set. Instead of using statistical surveys to extract accurate distributions, we use the distributions which were obtained from the large dataset. Then we replace the selected set of distributions with accurate alternatives and compute the performance of the new modified classifier. We run several experiments with the same replacements and then we use the paired t-test to see whether these sets of replacements make a statistically significantly better classifier or not. Our experiments show that replacing more distributions results in a more accurate classifier. The Letter dataset from the UCI machine learning repository [1] is used as the real dataset. In addition, an artificial dataset from the heart attack domain is used in a second experiment.

We have tested the effect of replacement of different permutations of conditional probability distributions in the letter dataset. These results are shown in table 1. Our experiments on the letter dataset show that for all except one replacement of the conditional probability distributions, the modified model is statisticaly significantly better. The results are obtained using the paired t-test with 95% confidence interval. In all cases, the variance of the accuracies of the modified model is smaller than that of the unmodified model. This means that when we replace a conditional probability distribution in a Bayesian Network with an accurate alternative, the new model tends to be more robust when sampling new datasets for training and testing.

In the artificial heart attack dataset we know all the correct conditional independencies. Our experiments on this dataset show that the replacement of conditional probability distributions, which are in the form of tables, with accurate alternatives, makes statistically significantly better classifiers in all cases of

Table 1. Accuracy for different replacements in the Bayesian Network on the letter data set as well as the results of the t-test for 20 different experiments

experiment	**no**	**y-bar**	**xy2br**	**xegvy**	**y-bar**	**y-bar**	**Xy2br**	**y-bar**
	change				**xy2br**	**xegvy**	**xegvy**	**xy2br xegvy**
Average of the accuracy	61.7	66.1	72.3	61.4	72.9	64.3	71.4	72.7
$Varianceofaccuracy$		4.36	10.62	-0.338	11.163	2.583	9.663	11.015
$T-Testresult:$		**ESS**	**ESS**	**NSS**	**ESS**	**VSS**	**ESS**	**ESS**

* ESS- extremely statistically significant * VSS- Very statistically significant * NSS- Not statistically significant * SS- Statistically Significant.

replacement. The results of these experiments again show that the variance in the accuracy of the modified model is smaller than the variance in the accuracy of the unmodified model. It is also experienced that using incomplete or wrong dependencies for an attribute may lead to a not statistically significant classifier.

3 Conclusion

In this study we propose a practical method for improving Bayesian classifiers by using background knowledge from large, publicly available datasets existing independently of the training dataset. We present a method which manipulates the Bayesian Network's conditional probability distributions, given in the form of tables, based on background knowledge. The idea is tested on a real and an artificial dataset. The results show that such changes produce significantly better classifiers than normal Bayesian Network classifiers.

Acknowledgments. This research was supported by the Natural Sciences and Engineering research Council of Canada.

References

1. Blake, C., Merz, C.: UCI repository of machine learning databases. Univ. of California, Irvine, http://www.ics.uci.edu/~mlearn/MLRepository.html
2. Cheng, J.: BN powerpredictor, http://webdocs.cs.ualberta.ca/~jcheng/bnsoft.htm
3. Cheng, J., Greiner, R.: Learning bayesian belief network classifiers: Algorithms and system. In: Stroulia, E., Matwin, S. (eds.) Canadian AI 2001. LNCS (LNAI), vol. 2056, pp. 141–151. Springer, Heidelberg (2001)
4. Clark, P., Matwin, S.: Learning domain theories using abstract beckground knowledge. In: Brazdil, P.B. (ed.) ECML 1993. LNCS, vol. 667, pp. 360–365. Springer, Heidelberg (1993)
5. Flach, P.A.: From extensional to intensional knowledge: Inductive logic programming techniques and their application to deductive databases. In: Kifer, M., Voronkov, A., Freitag, B., Decker, H. (eds.) Dagstuhl Seminar 1997, DYNAMICS 1997, and ILPS-WS 1997. LNCS, vol. 1472, pp. 356–387. Springer, Heidelberg (1998)
6. Heckerman, D.: A tutorial on learning with bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research (1995)
7. Mitchell, T.M.: Machine learning. McGraw-Hill, New York (1997)
8. Seifi, F., Drummond, C., Japkowicz, N., Matwin, S.: Improving Bayesian Learning Using Public Knowledge, p. 12, http://www.archive.org/details/ImprovingBayesianLearningUsingPublicKnowledge
9. Wu, P., Dietterich, T.G.: Improving svm accuracy by training on auxiliary data sources. In: Brodley, C.E. (ed.) ICML. ACM International Conference Proceeding Series, vol. 69. ACM, New York (2004)

Potential AI Strategies to Solve the *Commons Game*: A Position Paper

Petro Verkhogliad[1] and B. John Oommen[2]

[1] School of Computer Science, Carleton University, Ottawa, Canada
pverkhog@scs.carleton.ca
[2] *Chancellor's Professor*; *Fellow: IEEE* and *Fellow: IAPR*.
School of Computer Science, Carleton University, Ottawa, Canada
Also *Adjunct Professor* with the University of Agder in Grimstad, Norway
oommen@scs.carleton.ca

Abstract. In this paper, we propose the use of hill climbing and particle swarm optimization to find strategies in order to play the *Commons* Game (CG). The game, which is a non-trivial N-person non-zero-sum game, presents a simple mechanism to formulate how different parties can use shared resources. If the parties cooperate, the resources are sustainable. However, the resources get depleted if used indiscriminately. We consider the case when a single player has to determine the "optimal" solution, and when the other $N-1$ players play the game by choosing the options with a fixed probability vector.

Keywords: Intelligent Game Playing, Resolving N-person Games, Commons game, AI-based Resource Management.

1 Introduction

Artificial intelligence (AI) techniques have been used for decades to aid in the analysis and solution of games. During this time, most research activity has revolved around popular 2-player games, such as *Chess* and *Checkers*. Considerable effort has also been applied to the study of 2-player social dilemma games [3,6], such as the *Prisoner's Dilemma* [8]. Unfortunately, significantly less attention has been focused on N-player games, partly due to the difficulty of applying previously successful techniques [10] to games of this class. In this work, we consider the *Commons* Game (CG) [9], which is a non-trivial N-person non-zero-sum game. In this position paper, we suggest methods by which we can apply two AI techniques, namely, hill climbing (HC) and particle swarm optimization (PSO), in order to find a solution to this complex N-player social dilemma type game.

2 *Commons Game* Description

The *Commons game* designed by Powers *et al.* can be played by groups of 6 to 50 players. At every turn, every player selects one of five available actions:

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 352–356, 2010.

selfish use, cooperative use, abstention, penalty to selfish players, and reward to cooperative players. Each of these actions are mapped onto five playing cards identified by their colors: green, red, yellow, black and orange respectively.

The green card symbolizes selfish behavior and returns the maximum number of points. The same player[1] may receive a score of -20 points if a black card is played during the same turn. The red card represents cooperative behavior and returns a reward of 40% of the value of the green card. Red card players receive an additional 10 points for every orange card played in the same round. The yellow card represents abstention. Each yellow card receives 6 points regardless of the state of the environment or the number of players in the game. The black card is used to penalize selfish players. The returned score is defined by $-N_p/N_b$, where N_b is the number of black cards played in that round, and N_p is the number of participants. The orange card is used to encourage cooperative players by increasing their score for that turn by 10 points, and the player receives $-N_p/N_o$ points, where N_p is as above and N_o is the number of orange cards played in the round.

The state of the environment determines the exact number of points scored. The states range from +8 to -8. At the start of the game the environment is at state 0. Table 1 shows the scoring table for states +8, +4, 0, -4 and -8.

The depletion and replenishment of the environment are modeled using a marker, m, which ranges between $[0, 180]$. At the end of every turn, the marker is updated using Eq. (1) below, where m_{t+1} is the marker value in the next turn, N_g is the number of green cards played in the current turn, S_t is the current state number, $I(S_t)$ is the replenishment value in the given state, and t is the current turn number.

$$m_{t+1} = \begin{cases} m_t - N_g + I(S_t) & \text{if } t \mod 6 = 0 \\ m_t - N_g & \text{if } t \mod 6 \neq 0. \end{cases} \tag{1}$$

The value of the marker is used to determine the state of the environment in the next turn as shown in the Eq. (2) below:

$$S_{t+1} = \begin{cases} 0 & \text{if } 80 \leq m_t \leq 100 \\ \dfrac{m_t - 90}{10} & \text{if } m_t < 80 \text{ or } m_t > 100. \end{cases} \tag{2}$$

In the interest of clarification, consider the following example of score calculation. In this example, the number of players, N_p, is 8, the current turn, t, is 0, the value of the marker, m, is 100, and the current state, S_0, is 0. Five players use the red card ($N_r = 5$), two players use the green card ($N_g = 2$), and one player uses the black card ($N_b = 1$). The rewards of the players are then: $R_r = 44, R_b = -8$ and $R_g = -20$ instead of 106, since a black card was played.

[1] Although we consider specific numeric score values as defined in the original manual [9], the principles presented here work even if one changes the values so as to preserve the main properties of the game.

3 Methods

The focus of this work is on finding a strategy vector which constitutes the respective probabilities for playing a given card, $P = \{p_{green}, p_{red}, p_{yellow}, p_{orange}, p_{black}\}$, such that $\sum_{p_i \in P} p_i = 1$. We propose to do this by the use of hill climbing and particle swarm optimization.

3.1 Hill Climbing

Hill climbing (HC) is one of the earliest and most well known optimization techniques. At the start of the game the scheme is initialized with the following parameters in the solution/search space:

P - a random solution
λ - the learning rate parameter
T - number of turns used for learning
$f(P_t)$ - fitness function, where P_t is the vector of card probabilities at turn t.

At every turn, HC attempts to increase its reward by updating the probability vector, P. We propose that *five* candidate vectors are created by choosing a card, p_i, and increasing its probability while decreasing the probabilities of the other cards. Eq. (3) shows the updating function of the non-target cards, and Eq. (4) shows the updating function of the target card, after the others have been updated. Here $p_i(t)$ is the current probability of selecting card i, $p_i(t+1)$ is the probability that will be used in the next turn, and $i, j \in \{green, red, yellow, orange, black\}$ such that $i \neq j$.

$$p_i(t+1) = \lambda p_i(t) \tag{3}$$

$$p_j(t+1) = p_j(t) + (1 - \sum_{i \in P} p_i(t+1)) \tag{4}$$

Each of the five candidate vectors are then tested by playing T turns of the game. The vector with highest fitness value, i.e., the sum of scores over the T

Table 1. Reward table for an 8-player group

State +8			State +4			State 0			State -4			State -8		
$\mathcal{N}_r$	$\mathcal{R}_r$	$\mathcal{R}_g$	$\mathcal{N}_r$	$\mathcal{R}_r$	$\mathcal{R}_g$	$\mathcal{N}_r$	$\mathcal{R}_r$	$\mathcal{R}_g$	$\mathcal{N}_r$	$\mathcal{R}_r$	$\mathcal{R}_g$	$\mathcal{N}_r$	$\mathcal{R}_r$	$\mathcal{R}_g$
0	-	200	0	-	186	0	-	100	0	-	14	0	-	0
1	90	202	1	83	188	1	40	102	1	-3	16	1	-10	2
2	90	202	2	83	188	2	40	102	2	-3	16	2	-10	2
3	90	202	3	83	188	3	40	102	3	-3	16	3	-10	2
4	92	204	4	85	190	4	42	104	4	-2	18	4	-8	4
5	94	206	6	87	192	5	44	106	5	1	20	5	-6	6
6	96	208	7	89	194	6	46	108	6	3	22	6	-4	8
7	98	210	8	91	196	7	48	110	7	5	24	7	-2	10
8	100	-	8	93	-	8	50	-	8	7	-	8	0	-

turns, is selected as the current best solution. This process is repeated for an epoch of J (say 1,000) turns.

3.2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) was originally developed and introduced by Kennedy and Eberhart [5]. The algorithm is based on the flocking behavior of fish/birds, and has previously been successfully applied to many optimization as well as some gaming problems [1,2,4,7].

Similar to HC, each particle uses a set of updating equations to direct itself and the rest of the swarm toward the optimum. The updating rule for $\overrightarrow{x_i}(t)$, the position of particle i at time 't', is shown in Eq. (5), where $\overrightarrow{v_i}(t)$ is the particle velocity at time 't'. The updating rule for the latter, is shown in Eq. (6), where g and $\hat{g}$ are, respectively, the particle's and the swarm's best solutions. Further, c_1 and c_2 are the acceleration coefficients, and ω is the inertia weight.

$$\overrightarrow{x_i}(t+1) = \overrightarrow{v_i}(t) + \overrightarrow{x_i}(t) \tag{5}$$

$$\begin{aligned} v_{ij}(t+1) = {} & \omega v_{ij}(t) + c_1 r_1 j(t)[g_{ij}(t) - x_{ij}(t)] \\ & + c_2 r_2 j(t)[\hat{g}_{ij}(t) - x_{ij}(t)]. \end{aligned} \tag{6}$$

In the above equations, x_{ij} and v_{ij} are the j^{th} components of $\overrightarrow{x_i}(t)$ and $\overrightarrow{v_i}(t)$ respectively, and r_1 and $r_2 \in U(0,1)^n$. It has been shown that under the following conditions:

$$\omega > \frac{1}{2}(c_1 + c_2) - 1; \quad \texttt{and} \quad 0 < \omega < 1,$$

convergence to a stable equilibrium point is guaranteed.

At the start of the game, each particle in the swarm is initialized with a random strategy. At every turn, the fitness of every particle is evaluated by using the particle's position to play T turns of the training instance of the game. The fitness value is the sum of the rewards received over the T turns. The fitness function does not incorporate any information about the game, except for the score that the player received in each of the T turns. If the new fitness value is higher than any previously seen value, the new strategy becomes the particle's "Best" solution. Each of these individual "Best" solutions are then compared to each other in order to find the global "Best" solution. This process is repeated for an epoch of J (say 1,000) turns.

4 Results and Conclusion

Early results from applying HC and PSO to the *Commons Game* are highly encouraging. For example, when the game is confined to the +8 state, both algorithms converge on the solution of playing red with probability 7/8 and playing green with probability 1/8, which have been suggested by Powers as the cooperative "game end" scenario [9]. More detailed results and comparisons are currently being compiled, but are omitted here as this is only a position paper.

In conclusion, in this work we have suggested the use hill climbing and particle swarm optimization to solve the *Commons game*, which is an N-player, imperfect-information, non-zero-sum game. Early results suggest that general purpose optimization techniques can be used to find good strategies for this complex N-person game.

References

1. Abdelbar, A.M., Ragab, S., Mitri, S.: Applying co-evolutionary particle swam optimization to the egyptian board game seega. In: Proceedings of The First Asian-Pacific Workshop on Genetic Programming, pp. 9–15 (2003)
2. Conradie, J., Engelbrecht, A.P.: Training bao game-playing agents using coevolutionary particle swarm optimization. In: Proceedings of the IEEE 2006 Symposium on Computational Intelligence and Games, pp. 67–74 (2006)
3. Dawes, R.M.: Social dilemmas. Annual Review of Psychology 31(1), 169–193 (1980)
4. Franken, N., Engelbrecht, A.P.: Particle swarm optimization approaches to coevolve strategies for the iterated prisoner's dilemma. IEEE Transactions on Evolutionary Computation 9(6), 562–579 (2005)
5. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)
6. Kollock, P.: Social dilemmas: The anatomy of cooperation. Annual Review of Sociology 24(1), 183–214 (1998)
7. Laskari, E.C., Parsopoulos, K.E., Vrahatis, M.N.: Particle swarm optimization for minimax problems. In: Proceedings of the IEEE 2002 Congress on Evolutionary Computation, pp. 1582–1587 (2002)
8. Poundstone, W.: Prisoner's Dilemma. Doubleday (1992)
9. Powers, R.B., Duss, R.E., Norton, R.S.: THE COMMONS GAME Manual (1977)
10. Sturtevant, N.: Current challenges in multi-player game search. In: van den Herik, H.J., Björnsson, Y., Netanyahu, N.S. (eds.) CG 2004. LNCS, vol. 3846, pp. 285–300. Springer, Heidelberg (2006)

Score Calibration for Optimal Biometric Identification

Dmitry O. Gorodnichy and Richard Hoshino

Science and Engineering Directorate, Canada Border Services Agency
14 Colonnade Road, Ottawa, Ontario, Canada, K2E 7M6

Abstract. We present a calibration algorithm that converts biometric matching scores into probability-based confidence scores. Using the context of iris biometrics, we show – theoretically and by experiments – that in addition to attaching a meaningful confidence measure to the output, this calibration technique yields the best possible detection error tradeoff (*DET*) curves, both at the score level and at the decision level, thus maximizing the overall performance of the biometric system.

1 Motivation, Definitions, Theoretical Proof of Optimality

Let X be an individual arriving at a biometric kiosk, and let $\{x_1, x_2, \ldots, x_n\}$ represent the gallery of enrolled passengers. By comparing X with enrollee x_i, a matching score s_i is generated, representing the degree of similarity between the biometric feature(s) of X and x_i. Provided one of these matching scores is above or below a certain pre-determined threshold, the decision is made to grant or deny access to an individual.

In making this decision, no consideration is given to the fact that there might be other individuals in the enrollment gallery having similar matching scores. As a result, the performance of the biometric system is sub-optimal, leading to false accepts and false rejects. This decreases the reliability of the system, which is especially problematic in security-sensitive environments, as exposed in [4].

Our objective is to attach a probabilistic measure to a biometric system, by converting the n-tuple of matching scores $S = (s_1, s_2, \ldots, s_n)$ into the n-tuple $C = (c_1, c_2, \ldots, c_n)$ of confidence scores, where each $c_i = P(\{X = x_i\} \mid S)$ represents the probability that the identity of X is x_i, given the n-tuple S. Our formula then yields a scoring algorithm that is both *normalized* (the sum of the c_i's is 1), and *calibrated* (e.g., the statement "I am 60% sure that this person is Alice" is correct exactly 60% of the time.)

In [5], this confidence measure is presented in the context of iris biometrics, where an iris image is converted into a binary string in which each bit is equally likely to be 0 or 1, for which the histograms of impostor and genuine matching scores, as measured by the Hamming Distance (HD), are known [1] to follow binomial distributions. Let G be the set of genuine matching scores, and I be the set of impostor matching scores. Let $G \sim Binom(\hat{m}, \hat{u})$ and $I \sim Binom(m, u)$,

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 357–361, 2010.

where $(\hat{m}, \hat{u})$ and (m, u) are the degrees-of-freedom and mean of the two distributions. The following *Score Calibration Function (SCF)* is proven in [5]:

$$c_i = \frac{p_i z_i}{\displaystyle\sum_{i=1}^{n} p_i z_i + q \cdot \frac{(1-u)^m}{(1-\hat{u})^{\hat{m}}}}, \quad \text{where } z_i = \frac{\binom{\hat{m}}{\hat{m}s_i}}{\binom{m}{ms_i}} \cdot \left(\frac{\hat{u}^{\hat{m}}(1-u)^m}{u^m(1-\hat{u})^{\hat{m}}} \right)^{s_i}, \tag{1}$$

where $p_i = P(X = x_i)$ is the a-priori probability that an individual arriving at the kiosk is person x_i, and q is the probability that the individual is unenrolled.

This SCF function replaces matching scores with meaningful confidence scores that are perfectly calibrated and normalized.

We now prove that this calibration algorithm also produces a convex *Detection Error Trade-off (DET)* curve with the minimum *Area Under the Curve (AUC)*, both at the score level and at the decision level. By definition, the DET curve at the *score level* graphs the false match rate (FMR) against the false non-match rate ($FNMR$) over all possible thresholds, which is done by examining the scores given to genuine and impostor comparisons. On the other hand, the DET curve at the *decision level* graphs the false accept rate (FAR) against the false reject rate (FRR) over all possible thresholds, which is done by comparing all n scores and seeing if the highest score lies above the threshold.

Theorem: *If G and I are both binomially distributed, then the algorithm whose scores and match decisions are based on the calibrated confidence function (Eq. 1), rather than on the matching scores, produces the biometric system's best possible DET curve both at the score level and at the decision level.*

Proof. By Eq. 1, each vector $S = (s_1, s_2, \ldots, s_n)$ of matching scores gives rise to a vector $C = (c_1, c_2, \ldots, c_n)$ of confidence scores. Since there are only finitely many values for each $s_i = HD(X, x_i)$, there are only finitely many n-tuples S that can arise. Assuming there are t possible matching score vectors S, there are at most tn confidence scores. Suppose there are k unique confidence scores, where $k < tn$. Rank these k scores from highest to lowest, labeling them $r_1, r_2, \ldots, r_k$.

Taken over all genuine and impostor comparisons, let f_i and t_i represent the number of false matches and true matches with score r_i. Letting a_i be the accuracy of matches with score r_i, we have $a_i = \frac{t_i}{f_i + t_i}$ for all $1 \leq i \leq k$. Let $F = \sum f_i$ be the total number of impostor comparisons, and $T = \sum t_i$ be the total number of genuine comparisons. For each possible threshold, we now determine the values of FMR and $FNMR$, and the corresponding $k+1$ points of the DET curve. This is shown in Table 1.

Each point on the DET curve is represented by $(FMR_j, FNMR_j)$, where $FMR_j = x_j = \left(\sum_{i=1}^{j} f_i\right)/F$ and $FNMR_j = y_j = \left(T - \sum_{i=1}^{j} t_i\right)/T$, and by definition, $(x_0, y_0) = (0, 1)$ and $(x_k, y_k) = (1, 0)$. These k classes should be ordered so that the resulting DET curve becomes convex, as this minimizes the AUC. This important observation has been cited in previous papers [2,3]. The DET curve of any scoring algorithm can be made convex by arranging the

Table 1. False Match and False Non-Match Rates for each threshold

[t] Threshold	Included Indices	False Match Rate	False Non-Match Rate
$r_1 + \epsilon$	None	0	$(t_1 + t_2 + \ldots + t_k)/T$
r_1	1	f_1/F	$(t_2 + \ldots + t_k)/T$
r_2	$1, 2$	$(f_1 + f_2)/F$	$(t_3 + \ldots + t_k)/T$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
r_{k-1}	$1, 2, \ldots, k-1$	$(f_1 + f_2 + \ldots + f_{k-1})/F$	t_k/T
r_k	$1, 2, \ldots, k$	$(f_1 + f_2 + \ldots + f_k)/F$	0

resulting classes in the proper order. As we will see, arranging the classes by calibrated confidence score achieves convexity.

For convexity, we require the slopes $g_j = \frac{y_j - y_{j-1}}{x_j - x_{j-1}}$ to increase as j increases. We have $g_j = \frac{-t_j/T}{f_j/F} = -\frac{F}{T} \cdot \frac{t_j}{f_j}$. Since F and T are constant, if g_j is an increasing function, we require $\frac{t_j}{f_j}$ to be a decreasing function. Since $\frac{1}{a_j} = \frac{f_j + t_j}{t_j} = 1 + \frac{f_j}{t_j}$, if g_j is increasing, this implies that a_j must be decreasing. By definition, $r_1 > r_2 > \cdots > r_k$, and by design the scores are perfectly calibrated. This implies that $a_j = r_j$, which shows that a_j is indeed a decreasing function. Thus, we have shown that by transforming matching scores into calibrated confidence scores, we ensure that the resulting DET curve becomes convex, thus implying optimality.

The exact same technique shows the optimality of the DET curve produced at the decision level, by replacing false matches with false accepts in the above proof, as well as false non-matches with false rejects. By the definition of calibration, a score of X is correct exactly $X\%$ of the time, both at the score level and at the decision level. This completes our proof. □

2 Practical Use, Experimental Proof, Conclusions

To apply the Score Calibration Function (Eq. 1), one would obtain $m, u, \hat{m}, \hat{u}$ values from the vendor or find them experimentally, where m is found from the standard deviation σ as $m = \frac{u(1-u)}{\sigma^2}$, as in a binomial distribution. Instead of applying the SCF to all n matching scores, one could take a smaller subset (e.g. the best 10 scores) and restrict the formula to this subset, since the remaining scores would almost certainly all have a confidence score close to 0. This would reduce the required computational costs and enable the real-time implementation of this calibration function as a post-processing filter to existing conventional biometric systems. Since q would normally be unknown, the formula can be applied for different values of q to obtain a range of possible outputs for the vector C. We now describe how the SCF was applied and tested with an actual iris biometric system and real iris data.

The data set consisted of 100 enrollee and 595 probe iris images (six for each enrollee minus five that failed to acquire), producing 595 genuine comparisons and 58,905 impostor comparisons. For simplicity, we assumed that every enrollee used

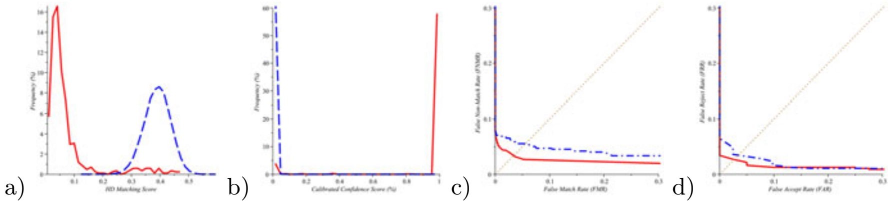


Fig. 1. Genuine and impostor distributions of the matching scores (a) and the calibrated scores (b), and their corresponding *DET* curves (dashed and solid lines, respectively) at the score level (c) and at the decision level (d)

the system equally often, and that unenrolled people did not use it. Thus, we set $p_1 = p_2 = \ldots = p_{100} = 0.01$ and $q = 0$. Having computed all matching scores, the mean and degrees-of-freedom of the genuine and impostor score distributions were then obtained: $\hat{u} = 0.074, \hat{m} = 9$ $(\sigma = 0.0456)$ and $u = 0.39, m = 114$ $(\hat{\sigma} = 0.088)$.

For each of the 595 people in the probe set, we determined the matching score vector $S = (s_1, s_2, \ldots, s_{100})$, and then applied Eq. 1 to transform S into the calibrated confidence score vector $C = (c_1, c_2, \ldots, c_{100})$, where $\sum c_i = 1$. Each c_i score was rounded to six decimal places.

Figure 1 presents the measured score distributions and the *DET* curves, showing that the *DET* curves of this calibration algorithm (solid lines) completely dominate the *DET* curves of the algorithm based on the matching scores (dashed lines), thus confirming our theoretical analysis.

To see how well the algorithm calibrates the scores, we tabulate the number of true matches and false matches for different threshold values. By the design, the calibrated confidence score c should be equal to the corresponding true match/accept rate. The experiments show this to be almost the case. – At the *score level*, of the 536 comparisons with the maximum score of $c = 1$, all 536 were true matches (100%), while of the 55832 comparisons with the minimum score of $c = 0$, only 16 were true matches (0.03%). At the *decision level*, there were 536 instances where the highest-scoring individual was given a confidence score of $c = 1$, and each was a true accept (100%), while the remaining 59 instances were insufficient to draw any statistically-significant conclusion.

Despite the small sample size, this analysis demonstrates the viability of the SCF in introducing meaningful confidence measures to the output of a biometric system. This approach may become particularly important in applications such as fully-automated Trusted Traveler identification, where the decision of the system is final and non-confident outputs may not be allowed.

References

1. Daugman, J.: How iris recognition works. IEEE Transactions on Circuits and Systems for Video Technology 14(1), 21–30 (2004)
2. Boström, H.: Maximizing the area under the ROC curve using incremental reduced error pruning. In: ICML 2005 Workshop on ROC Analysis in Machine Learning (2005)

3. Flach, P.A.: The geometry of ROC space: Understanding machine learning metrics through ROC isometrics. In: Intern. Conference on Machine Learning, pp. 194–201 (2003)
4. Gorodnichy, D.O.: Multi-order analysis framework for comprehensive biometric performance evaluation. In: SPIE Conference on Defense, Security, and Sensing (2010)
5. Gorodnichy, D.O., Hoshino, R.: Calibrated confidence scoring for biometric identification. In: NIST International Biometric Performance Conference, IBPC 2010 (2010)

Detection of Mine-Like Objects Using Restricted Boltzmann Machines

Warren A. Connors[1], Patrick C. Connor[2], and Thomas Trappenberg[3]

[1] Defence Research and Development Canada Atlantic
warren.connors@drdc-rddc.gc.ca
[2] Department of Computer Science, Dalhousie University
patrick.connor@dal.ca
[3] Department of Computer Science, Dalhousie University
tt@cs.dal.ca

Abstract. Automatic target recognition (ATR) of objects in side scan sonar imagery typically employs image processing techniques (e.g. segmentation, Fourier transform) to extract features describing the objects. The features are used to discriminate between sea floor clutter and targets (e.g. sea mines). These methods are typically developed for a specific sonar, and are computationally intensive. The present work[1] used the Restricted Boltzmann Machine (RBM) to discriminate between images of targets and clutter, achieving a 90% probability of detection and a 15% probability of false alarm, which is comparable to the performance of a Support Vector Machine (SVM) and other state-of-the-art methods on the data. The RBM method uses raw image pixels and thus avoids the issue of manually selecting good representations (features) of the data.

1 Introduction

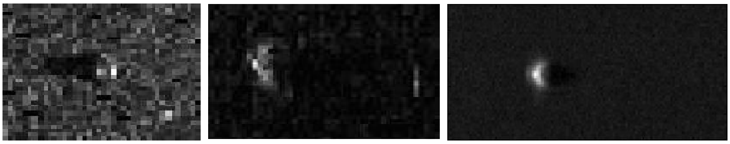
Naval mine detection is a resource intensive task. Recent research has focused on development of automated tools for detection and classification of sea floor objects. The detection and classification phases of the process have been implemented using a set of image processing or statistical methods (Z-test, matched filter), feature extraction, and a template-based classification [1,2,3]. These techniques are effective, but sensitive to the environment under test, the extracted features, and the tuning of algorithm parameters. Success with learning algorithms like Artificial Neural Networks has been limited partly because training sets must be statistically representative of the environment of the actual test data. Using an RBM avoids the explicit feature extraction step, using the raw image pixels. Although not explored here, RBMs can also be trained with unlabeled data which poses promise for future improvement.

Side scan sonar imagery (e.g. Figure 1a) depicts sea floor objects by a strong bright region (highlight) where it is insonified by sound waves, followed by a dark region (shadow) cast behind it. The dataset used here consists of 49x113

[1] This research was supported in part by a Walter C. Sumner Memorial Fellowship.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 362–365, 2010.

pixel images collected with a 455kHz Klein 5500 side scan sonar during the CITADEL trial, conducted in October, 2005 [5]. Acoustic and electronic noise in the system led to pixel-value scaling (Figure 1b). Also, as Figure 1c shows, RBMs can generate realistic imagery after training.



(a) Example cone (b) CITADEL cone (c) Generated cone

Fig. 1. Side scan images of mines showing (a) a typical image, (b) the data from the CITADEL trial, and (c) an image generated from a trained RBM

2 Employing the Restricted Boltzmann Machine

The Restricted Boltzmann Machine (RBM) [6,7] is a generative model that can learn to represent the distribution of training data. The lower the energy of an RBM, the more familiar it is with the associated input configuration. Hinton et al. [4] showed that RBMs can be stacked, creating a deep belief network (DBN). In this work a DBN of three RBMs was chosen. The first two layers of hidden nodes (H_1 and H_2) have the same number of nodes and the top layer (H_3) has twice as many nodes.

ATR results are quantified in receiver operator characteristic (ROC) curves, comparing the probability of detection (P(d)) with the probability of false alarms (P(fa)). The goal is to maximize P(d) while minimizing P(fa), both of which need to fall within a certain range for a system to be useful. The ROC curve is computed by sliding a decision boundary through a feature space, which shows the trade-off between increasing P(fa) with P(d). Here, the ROC curve is computed in terms of the free energy in the top RBM module. First, images are propagated up through the lower two RBM modules to provide input to the top RBM. Second, the free energy of the top RBM is computed, which is its energy minus its entropy or,

$$F(p, s, w) = -\sum_{ij} p_j s_i w_{ij} - \sum_j p_j b_j - \sum_i s_i c_i - S(p) \tag{1}$$

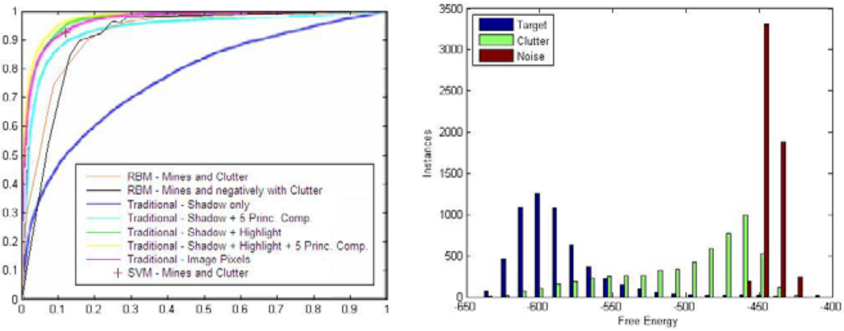
where $S(p)$ is the entropy of the system, expressed by

$$S(p) = -\sum_j (p_j log(p_j) + (1 - p_j) log(1 - p_j)), \tag{2}$$

where p, s, and w, represent the hidden unit probabilities, visible unit states, and connection weights respectively. Also, b and c are the biases for the hidden and visible layers respectively.

3 RBM Classification Results

The sliding RBM decision boundary was varied between $\mu + 2\sigma$ and $\mu + 10\sigma$, where μ is the mean and σ is the standard deviation of the free energies of the training data in the top RBM module. Results for evenly balanced training and test data are plotted in Figure 3a. The best RBM had 500 nodes in H_1 and H_2 and 1000 nodes in H_3 and was trained on both target and clutter images, achieving results of P(d) = 0.90 and P(fa) = 0.15 (Figure 3b shows its free energy distribution). New results from using SVM and previous results from previous ATR methods [5] are also included in Figure 3a. The SVM was trained on the raw image pixels giving P(d) = 0.93 and P(fa) = 0.12, performing slightly better than the RBM. The SVM maintained this result regardless of significant changes to its parameters and use of several different kernel functions, including a Gaussian (radial basis) kernel. The results show that the RBM can give comparable results while providing additional future possibilities such as the use of unlabeled training data.



(a) (b)

Fig. 2. (a) Classification results and (b) the free energy distribution of target (test set), clutter (test set), and noise imagery

4 Discussion

The RBM results are very encouraging. For an initial attempt at using RBMs for ATR, the best results are in the same league as the SVM and state-of-the-art traditional methods (which use features selected with problem-specific knowledge).

In the most effective scenario, the RBM is trained on targets and negatively on clutter (via weight and bias negation), which helps to separate the free energies more than training on targets alone. Looking at the free energy distributions in this scenario (Figure 3b) reveals what we expect: the RBM has lowest energies for targets, followed by clutter, followed by random input. In another effective scenario, the RBM is trained with targets and clutter together. We hypothesize that the additional training data (clutter) helps the RBM focus less on the background noise and focus more on modeling notable features of objects in the image, which target images consistently possess and clutter images do not.

Both lowering and raising the number of nodes in the RBMs worsens the results, presumably because either there are not enough nodes to model all of the important features of targets, or because there are too many nodes such that the noise gets modeled. Also, because the images are real-valued (instead of binary), the learning rate of the input RBM stage was set lower (by an order of magnitude) than for the other RBMs, to lead to convergence and good results.

After training, the RBM does not require considerable processing power or memory to be employed. This allows for the system to be implemented on low power computers, therefore making the system ideal to be embedded into a minehunting platform or autonomous underwater vehicle.

Which is more effective for ATR, RBMs or SVMs? Both use raw pixels as input and give similar results for this dataset. Both may achieve better results with more investigation. This could include a) augmenting the training data to give a more complete range of possible target orientations, b) varying the number of RBM modules, and c) using the RBM with unlabeled data to name a few.

References

1. Chapple, P.: Automated detection and classification in high-resolution sonar imagery for autonomous underwater vehicle operations. Technical report, Defence Science and Technology Organization (2008)
2. Fawcett, J., Crawford, A., Hopkin, D., Myers, V., Zerr, B.: Computer-aided detection of targets from the CITADEL trial Klein sonar data. Defence Research and Development Canada Atlantic TM 2006-115 (November 2006), pubs.drdc.gc.ca
3. Fawcett, J., Crawford, A., Hopkin, D., Couillard, M., Myers, V., Zerr, B.: Computer-aided classification of the Citadel Trial sidescan sonar images. Defence Research and Development Canada Atlantic TM 2007-162 (2007), pubs.drdc.gc.ca
4. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Comput. 18(7), 1527–1554 (2006)
5. Fawcett, J., Couillard, M., Hopkin, D., Crawford, A., Myers, V., Zerr, B.: Computer-aided detection and classification of sidescan sonar images from the citadel trial. In: Proceedings of the Institute of Acoustics (2007)
6. Smolensky, P.: Information processing in dynamical systems: foundations of harmony theory, pp. 194–281 (1986)
7. Hinton, G.E.: Training products of experts by minimizing contrastive divergence. Neural Comput. 14(8), 1771–1800 (2002)
8. Nair, V., Hinton, G.E.: Implicit mixtures of restricted boltzmann machines. In: NIPS, pp. 1145–1152 (2008)

Data Mining Techniques for Proactive Fault Diagnostics of Electronic Gaming Machines

Matthew Butler[1,2] and Vlado Kešelj[2]

[1] Faculty of Computer Science, Artificial Intelligence Group, University of York, UK
[2] Faculty of Computer Science, Dalhousie University, Canada
mbutler@cs.york.ac.uk, vlado@cs.dal.ca

Abstract. This paper details the preliminary research into modeling the behavior of Electronic Gaming Machines (EGM) for the task of proactive fault diagnostics. The EGMs operate within a state space and therefore their behavior was modeled, using supervised learning, as the frequency at which a given machine is operating in a particular state. The results indicated that EGMs did exhibit measurably different behavior when they were about to experience a fault and these relationships were modeled effectively by several algorithms.

Keywords: Decision Trees, Support Vector Machines, Proactive Fault Diagnosis, Pattern Recognition.

1 Introduction

The area of proactive fault diagnosis is focused on reducing downtime of machinery or equipment. In the area of electronic gaming machines it would be very beneficial to predict machine failure. An inoperable machine represents a loss of income, and the time between a failure of a machine and a technician's intervention does not only affect a business operation and income, but exhibits a threat to reputation of the manufacturer of the machine.

There have been attempts at prediction of a machine failure using data mining on sensor data [1,2], but we are not aware of any previous attempts to model the behaviour of Electronic Gaming Machines (EGM) for the task of proactive fault diagnostics. Our objective is to explore the problem of predicting a failure with sufficient time in advance to allow for adequate time for the maintenance provider to respond and send a technician. Additionally it was desirable to have a transparent model that easily transformed into business rules.

2 Data Description and Preprocessing

The raw unrefined data is comprised of two non-sequential time periods of event updates that are sent from the Electronic Gaming Machines (EGM) to a central system. A summary is provided in Table 1.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 366–369, 2010.

Table 1. Summary of event data files

Data Type	**EV1**	**EV2**
Time period	02/16/2009 to 02/23/2009	04/09/2009 to 04/17/2009
# of data entries	3,135,508	3,107,201
# of machines	7867	7660
# of potential states	104	104

The two time periods had 7,370 machine identifiers in common. The machine identifier was not used as one of the input attributes in classification to eliminate the algorithms remembering problematic machines.

The raw event data was processed for meaningful inputs for the algorithms using a developed Perl program. The program builds the files based on a user supplied time-horizon for making predictions and a time-window for collecting data for the input attributes. The actual inputs are a count of how many times a given machine was in each state during the time-window. The assumption is that machines will visit certain states when they are beginning to experience problems.

2.1 Class Information

The analysis was directed at identifying faults which would result in malfunctions in EGMs causing downtime. The experiments are setup as a classification problem where the classifiers are trained to identify behavior that predicates faults of any kind. As such, the problem is binary in nature where we have normal and abnormal behavior of machines. To facilitate quicker building times of input data sets a sampling method was used to randomly sample machines from classes which were over-represented. As a result, the initial classification is treated as a 4-class problem where certain behaviors are included which were predetermined to be potential confounding variables or false alarms. Table 2 details the 4-class problem that was initially used.

3 Experiment Setup and Results

The classifiers were trained on a data set created from one of the time-periods with 10-fold cross validation. After training was completed the constructed model

Table 2. Classification details for event summary data

Class	Label	Description
Normal	0	EGMs which did not experience any faults.
Abnormal	1	EGMs which experienced a fault which took the machine offline.
General False Alarm	98	EGMs which reported a false alarm for any fault, other than bill acceptor.
Specific False Alarm	99	EGMs which reported a bill acceptor false alarm.

would be tested with an out-of-sample data set from the other time-period which contained data that reflected the same time-horizon and window as the training set. Later the data sets were switched and the classifiers were re-trained and tested. For our analysis we are particularly interested in precision of class “1”. This is important because any EGM labeled as class “1” would be scheduled for maintenance and therefore would be a wasted resource if that EGM was labeled incorrectly. Reported testing results reflect training and testing on data sets which contain identical time-horizon and time window durations. For example, the data set D1.1 was constructed from EV1 (Event summary 1), for a time window of 1 day and a time-horizon of 7 days. When D1.1 is used for testing, the model being tested was created using D2.1 which was created from EV2 and reflects a time window of 1 day and a time-horizon of 7 days.

3.1 C4.5 Entropy-Based Decision Tree

The results based on the C4.5 classification method, are presented in Table 3.

Table 3. Detailed training and testing results for C4.5

Testing		Precision				Recall				F-measure			
Data Sets	**Accuracy**	**98**	**1**	**0**	**99**	**98**	**1**	**0**	**99**	**98**	**1**	**0**	**99**
D2.1	96.15	0.0	1.0	.96	.98	0.0	.87	1.0	.85	0.0	.93	.98	.91
D2.2	94.02	.60	.99	.95	.89	.10	.95	.98	.85	.17	.97	.97	.87
D2.3	93.58	.44	.99	.95	.95	.62	.97	1.0	.76	.51	.98	.97	.84
D1.1	96.06	0.0	.94	.97	.94	0.0	.94	1.0	.90	0.0	.94	.98	.92
D1.2	94.03	.37	.95	.96	.95	.37	.96	.99	.83	.37	.96	.98	.88
D1.3	68.53	.10	.95	.93	.94	.63	.98	.60	.75	.16	.96	.73	.83

The results in Table 3 are relatively consistent across all data sets and experiment setups, with the exception of the testing results for D2.3/D1.3, where the overall accuracy and several of the precision measures decreased significantly. However, in spite of these results the precision for fault detection was .95, which is consistent with the other results.

3.2 Support Vector Machine

The SVM had comparable fault detection precision rates to that of C4.5. Although the SVM model proved to be an effective classifier for the fault detection the models it created are more complex to interpret into simple business rules.

4 Discussion and Conclusions

Given the underlying assumption that class 1 faults are significant and the ability to predict such faults is desirable, it appears that the data does have predictive

Table 4. Detailed training and testing results for Support Vector Machines

Testing		**Precision**				**Recall**				**F-measure**			
Data Sets	**Accuracy**	**98**	**1**	**0**	**99**	**98**	**1**	**0**	**99**	**98**	**1**	**0**	**99**
D2.1	94.99	0.0	.72	.96	.92	0.0	.87	.98	.86	0.0	.79	.97	.89
D2.2	87.91	0.0	.91	.94	.72	0.0	.56	.92	.88	0.0	.69	.93	.80
D2.3	89.17	.94	.93	.96	.72	.52	.87	.91	.88	.67	.90	.93	.79
D1.1	92.76	0.0	.90	.95	.84	0.0	.41	.99	.89	0.0	.56	.97	.86
D1.2	90.77	.60	.96	.96	.78	.08	.71	.97	.90	.14	.82	.96	.83
D1.3	89.17	.94	.93	.96	.72	.52	.87	.91	.88	.67	.90	.93	.79

abilities and that there is a possibility to predict said faults with a reasonable amount of time to respond. The models developed in one time period tested well in another indicating that the patterns are somewhat static and that the machines exhibit similar behavior prior to experiencing a fault. There is a desire to create and implement business rules from the information extracted by the data mining algorithms, as such the decision tree models are the most appropriate. In general the effort put forth in this analysis is to establish if proactive fault diagnosis of electronic gaming machines is possible with data mining techniques. The results are promising given that the models maintained their high performance over both data sets and the algorithms were equally as effective regardless of which data set was used for training and for testing.

Acknowledgements

This research was sponsored by MITACS ACCELERATOR program and our industry partner SPIELO Manufacturing ULC.

References

1. Malkoff, D.: A framework for real-time fault detection and diagnosis using temporal data. International Journal for Artificial Intelligence in Engineering 2(2), 97–111 (1987)
2. Yairi, T., Kato, Y., Hori, K.: Fault detection by mining association rules from house-keeping data. In: Proc. of International Symposium on Artificial Intelligence, Robotics and Automation in Space, Citeseer (2001)
3. Sylvain, L., Fazel, F., Stan, M.: Data mining to predict aircraft component replacement. IEEE Intell. Syst. 14(6), 59–65 (1999)

Multivariate Decision Trees Using Different Splitting Attribute Subsets for Large Datasets

Anilu Franco-Arcega[1], José Ariel Carrasco-Ochoa[1], Guillermo Sánchez-Díaz[2], and José Fco. Martínez-Trinidad[1]

[1] Computer Science Department
National Institute of Astrophysics, Optics and Electronics
Luis Enrique Erro # 1, Santa Maria Tonantzintla, Puebla, Mexico, C.P.72840
{anifranco6,ariel,fmartine}@inaoep.mx
[2] Computational Science and Technology Department
University of Guadalajara, CUValles
Carretera Guadalajara - Ameca Km. 45.5, Ameca, Jalisco, Mexico, C.P. 46600
guillermo.sanchez@profesores.valles.udg.mx

Abstract. In this paper, we introduce an incremental induction of multivariate decision tree algorithm, called IIMDTS, which allows choosing a different splitting attribute subset in each internal node of the decision tree and it processes large datasets. IIMDTS uses all instances of the training set for building the decision tree without storing the whole training set in memory. Experimental results show that our algorithm is faster than three of the most recent algorithms for building decision trees for large datasets.

Keywords: Decision tree(DT), large datasets, supervised classification.

1 Introduction

Nowadays, several algorithms have been proposed to build DTs for large datasets. The algorithms SLIQ [1], SPRINT [2], CLOUDS [3], RainForest [4] and BOAI [5] use lists for representing the training set, however these lists need a lot of space for large datasets. The algorithms BOAT [6] and ICE [7] are incremental algorithms that build DTs based only on a small subset of the training instances, but for determining this subset these algorithms spent a lot of time. The algorithm VFDT [8] builds DTs using the Hoeffding bound, and it uses several parameters, which could be very difficult to determine by the user. The algorithm IIMDT [9] builds multivariate DTs using all the attributes for characterizing each internal node, but if the problem is described by a big amount of attributes, the expansion and the traversal processes would be too expensive.

This work introduces an extension of IIMDT [9] called IIMDTS for building multivariate DTs for large datasets. Our algorithm, unlike IIMDT, allows choosing a different splitting attribute subset in each node to be expanded. Besides, IIMDTS uses only a few instances for selecting the splitting attributes, which allows processing large datasets.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 370–373, 2010.

2 Proposed Algorithm

A DT built by IIMDTS (**I**ncremental **I**nduction of **M**ultivariate **D**ecision **T**rees with **S**ubset selection) has in each internal node a splitting attribute subset, each edge corresponds to a possible outcome of the splitting attribute subset (that is, a combination of values for the splitting attributes in the node) and each leaf has associated a class label.

For avoiding to store the whole training set in main memory, IIMDTS processes the training instances one by one, in an incremental way, traversing the DT with each one, until it reaches a leaf, where the instance will be stored. Besides, in order to avoid storing all the training instances into the tree, a leaf stores at most s instances (s is a parameter of IIMDTS), discarding those instances once the node has been expanded or updated.

In multivariate DTs, different methods have been used for selecting a splitting attribute subset, for example sequential selection [10] or linear machines [11]. However these methods evaluate a big amount of attribute subsets for finding the best, which could be too expensive for large datasets. Therefore, in order to make a fast selection, we propose to use the Gain Ratio Criterion [12], but considering only the s instances stored in the leaf to be expanded. IIMDTS chooses, as splitting attribute subset, the n (another parameter of our algorithm) best attributes, according to the Gain Ratio measure.

At the beginning, IIMDTS starts with an empty root node (a leaf). Each instance of the training set traverses the DT until it reaches a leaf, where the instance will be stored. Since IIMDTS processes the training instances in an incremental way, processing a long sequence of instances of the same class would lead to build a biased DT. In order to avoid this situation, IIMDTS, in a preprocessing step, reorganizes the training set alternating instances from each class, i.e., the first instance from the first class, the second instance from the second class and so on, if there are r classes, the instance in the position $r + 1$ will be from class 1 and so on.

When a leaf reaches s instances from two or more classes, IIMDTS replaces the leaf by an internal node. To expand a node, IIMDTS chooses as splitting attribute subset the first n attributes according to the Gain Ratio Criterion, but taking into account only the s instances stored in the node for computing it. Then, for each class of instances in the node, IIMDTS creates an edge connected to a new empty leaf and, using the obtained splitting attributes, IIMDTS computes a combination of values for each edge. For obtaining the combination of values for an edge, IIMDTS computes for each splitting attribute the mean among the values of the instances in the expanded node belonging to the class associated to the edge. The splitting attributes and their combination of values will be used by IIMDTS for traversing the DT. Finally, IIMDTS deletes the instances used for expanding the node.

On the other hand, when a leaf has s instances from a single class, IIMDTS does not expand the node, it keeps the node as a leaf, and updates the combination of values associated to the splitting attributes of the input edge. For this update, IIMDTS computes a combination of values for the splitting attributes from the instances stored in the leaf (as we explained before), and IIMDTS computes for each splitting attribute the average between the value associated to the input edge and the value computed from the instances in the leaf. Finally, IIMDTS deletes the instances stored in the leaf.

IIMDTS finishes when all the instances of the training set have been processed. Finally, our algorithm assigns to each leaf the label of the majority class of the instances stored in it or the label of the class associated to the input edge, if the leaf is empty. IIMDTS traverses the DT and classifies new instances in a similar way as a conventional DT algorithm.

3 Experimental Results

We show a comparison of IIMDTS against the performance of VFDT, BOAI and IIMDT (the most recent algorithms for building DTs for large datasets). The dataset used in our experiments was GalStar [13]. We used 10-fold cross validation reporting the average of the ten tests and the 95% confidence interval. The results show the processing time (including induction and classification time, and for IIMDTS also the time for the preprocessing step) and the accuracy rate. Our experiments were performed on a PC with a Pentium 4 at 3.06 GHz, with 2 GB of RAM running Kubuntu 7.10.

For GalStar, we created training sets from 500,000 to 4,000,000 with increments of 500,000. Fig. 1 presents a comparison among IIMDTS, VFDT and IIMDT. BOAI does not appear in this figure because it could not process training sets with more than 300,000 instances. As it can be noticed all algorithms obtained similar accuracies, but IIMDTS was about 4.25 and 3.75 times faster than VFDT and IIMDT, respectively.

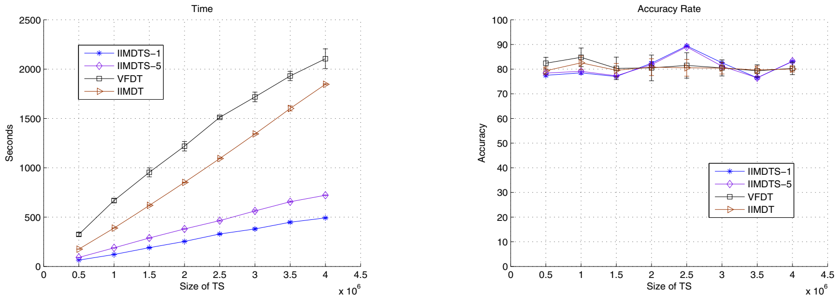


Fig. 1. Processing time and accuracy rate for GalStar

4 Conclusion

Based on the experiments, we concluded that our algorithm is faster than three of the most recent algorithms for building DT for large datasets, VFDT, BOAI and IIMDT, while IIMDTS maintains competitive accuracy. As future work, we will study different ways to select the splitting values, in order to build more accurate DTs.

Acknowledgments

This work is partly supported by the National Council of Science and Technology of Mexico under the project CB-2008-01-106443 and grant 165151.

References

1. Mehta, M., Agrawal, R., Rissanen, J.: SLIQ: A fast scalable classifier for data mining. In: Apers, P.M.G., Bouzeghoub, M., Gardarin, G. (eds.) EDBT 1996. LNCS, vol. 1057, pp. 18–32. Springer, Heidelberg (1996)
2. Shafer, J.C., Agrawal, R., Mehta, M.: SPRINT: A scalable parallel classifier for data mining. In: Proc. 22nd International Conference Very Large Databases, pp. 544–555 (1996)
3. Alsabti, K., Ranka, S., Singh, V.: CLOUDS: A decision tree classifier for large datasets. In: KDD, pp. 2–8 (1998)
4. Gehrke, J., Ramakrishnan, R., Ganti, V.: Rainforest - A frame- work for fast decision tree construction of large datasets. Data Mining and Knowledge Discovery 4, 127–162 (2000)
5. Yang, B., Wang, T., Yang, D., Chang, L.: BOAI: Fast Alternating Decision Tree Induction Based on Bottom-Up Evaluation. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 405–416. Springer, Heidelberg (2008)
6. Gehrke, J., Ganti, V., Ramakrishnan, R., Loh, W.: BOAT - optimistic decision tree construction. In: Proc. of the ACM SIGMOD Conference on Management of Data, pp. 169–180 (1999)
7. Yoon, H., Alsabti, K., Ranka, S.: Tree-based incremental classification for large datasets. Technical Report TR-99-013, CISE Department, University of Florida, Gainesville, FL. 32611 (1999)
8. Domingos, P., Hulten, G.: Mining high-speed data streams. In: Proc. of Six Int. Conference on Knowledge Discovery and Data Mining, pp. 71–80 (2000)
9. Franco-Arcega, A., Carrasco-Ochoa, J.A., Sánchez-Díaz, G., Martínez-Trinidad, J.F.: A new incremental algorithm for induction of multivariate decision trees for large datasets. In: Fyfe, C., Kim, D., Lee, S.-Y., Yin, H. (eds.) IDEAL 2008. LNCS, vol. 5326, pp. 282–289. Springer, Heidelberg (2008)
10. Brodley, C.E., Utgoff, P.E.: Multivariate decision trees. Machine Learning 19(1), 45–77 (1995)
11. Li, X.B., Sweigart, J.R., Teng, J.T., Donohue, J.M., Thombs, L.A., Wang, S.M.: Multivariate decision trees using linear discriminants and tabu search. IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans 33(2), 194–205 (2003)
12. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo (1993)
13. Adelman-McCarthy, J., Agueros, M.A., Allam, S.S.: Data Release 6. ApJS, 175 (in press, 2008)

Using Non Boolean Similarity Functions for Frequent Similar Pattern Mining

Ansel Y. Rodríguez-González[1,2], José Fco. Martínez-Trinidad[2], Jesús Ariel Carrasco-Ochoa[2], and José Ruiz-Shulcloper[1]

[1] Advanced Technologies Applications Center, Havana, Cuba
{arodriguez,jshulcloper}@cenatav.co.cu
[2] National Institute of Astrophysics, Optics and Electronics, Puebla, México
{ansel,fmartine,ariel}@inaoep.mx

Abstract. In this paper, we focus on frequent pattern mining using non Boolean similarity functions. Several properties and propositions that allow pruning the search space of frequent similar patterns, are proposed. Based on these properties, an algorithm for mining frequent similar patterns using non Boolean similarity functions is also introduced. We evaluate the quality of the frequent similar patterns computed by our algorithm by means of a supervised classifier based on frequent patterns.

1 Introduction

Frequent pattern mining has become a key task in data mining[1]. In the classical approach on frequent pattern mining[2], datasets are described exclusively by Boolean features. However, in soft sciences there are datasets containing objects described simultaneously by numerical and non numerical features (mixed data) [3]. Besides, in this context, two real life objects are hardly ever exactly equal, thus similarity functions different from the equality, are commonly used to compare object descriptions.

For mining frequent patterns using Boolean similarity functions different from the equality, some algorithms (*ObjectMiner*[4]; and *STreeDC-Miner*[5] and *STree NDC-Miner*[5]) have been developed. However, as we have pointed out, there are problems where the similarity function is not Boolean. For these cases, there is no algorithm for mining frequent patterns. Therefore, in this context, the similarity function must be Booleanized, which could lead to losing information. For this reason, in this paper, we focus on mining frequent pattern using non Boolean similarity functions.

2 Basic Concepts and Proposed Algorithm

Let $\Omega = \{O_1, \ldots, O_n\}$ be a dataset. Each object is described through a set of features $R = \{r_1, \ldots, r_m\}$ and represented as a tuple $(v_1, \ldots, v_m)$. A *subdescription* of an object O for a subset of features $S \subseteq R$ denoted by $I_S(O)$, is the description of O only in terms of the features in S. Each subset of features $\emptyset \neq S \subseteq R$,

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 374–378, 2010.

has associated a *similarity function*[6] $f_S \in [0,1]$ between subdescriptions. An example of a similarity function is:

$$f_S(O,O') = \prod_{r \in S} C_r(I_{\{r\}}(O), I_{\{r\}}(O')) \tag{1}$$

where $C_r : D_r \times D_r \rightarrow [0,1]$ is a comparison function between values of feature r; D_r is the domain of r. Two examples of comparison functions are:

$$C_r(x,y) = 1 - \frac{x-y}{Max_r - Min_r} \quad (2) \qquad C_r(x,y) = \begin{cases} 1 & \text{if } 1 - \frac{x-y}{Max_r - Min_r} \geq \alpha \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where $Max_r = \max_{O \in \Omega} I_{\{r\}}(O)$, $Min_r = \min_{O \in \Omega} I_{\{r\}}(O)$ and $\alpha \in [0,1]$.

Let f_S be a non Boolean similarity function. We define the *frequency* of a subdescription $I_S(O)$ in Ω for f_S as $f_S\text{-}freq(O) = \frac{\sum_{O' \in \Omega} f_S(O,O')}{|\Omega|}$. A subdescription $I_S(O)$ is a *frequent similar pattern* (*FSP*) in Ω, if its frequency is not less than a frequency threshold $minFreq$. Given a frequency threshold $minFreq$, the frequent similar pattern mining problem consists in finding all *FSPs* in Ω.

A similarity function f_S is *non increasing monotonic* iff $\forall O, O', S_1, S_2; O, O' \in \Omega\ [\emptyset \neq S_1 \subseteq S_2 \subseteq R] \Rightarrow [f_{S_1}(O,O') \geq f_{S_2}(O,O')]$.

Property 1 (*Monotony of the frequency*). $\forall O, S_1, S_2;\ O \in \Omega\ [\emptyset \neq S_1 \subseteq S_2 \subseteq R] \Rightarrow [f_{S_1}\text{-}freq(O) \geq f_{S_2}\text{-}freq(O)]$.

Property 2 (*f_S-Downward Closure*). $\forall O, S_1, S_2;\ O \in \Omega;\ \emptyset \neq S_1 \subseteq S_2 \subseteq R\ [f_{S_1}\text{-}freq(O) < minFreq] \Rightarrow [f_{S_2}\text{-}freq(O) < minFreq]$.

The f_S-Downward Closure property, unlike the Downward Closure property for frequent itemset mining, is not always true, because its fulfillment depends on the monotony of the frequency and the monotony of the similarity function. These dependencies are expressed as follows:

Proposition 1. *If f_S is a non increasing monotonic similarity function, then f_S fulfills the monotony of the frequency.*

Proposition 2. *If f_S fulfills the monotony of the frequency, then f_S fulfills the f_S-Downward Closure.*

A subdescription $I_S(O)$ is a *f_S-interesting pattern* if $I_S(O)$ is a *FSP* or it contributes to the frequency of a *FSP* $I_S(O')$ (i.e., $f_S(O',O) \neq 0$); $I_S(O') \neq I_S(O)$. In contraposition, a subdescription $I_S(O)$ is a *non f_S-interesting pattern* if $f_S\text{-}freq(O) < minFreq$ and $\forall O'; O' \in \Omega; I_S(O') \neq I_S(O)\ [f_S\text{-}freq(O') \geq minFreq] \Rightarrow [f_S(O',O) = 0]$.

Proposition 3. *If f_S is a non increasing monotonic similarity function and a subdescription $I_S(O)$ is a non f_S-interesting pattern, then $\forall S'$; $S \subset S'$, $I_{S'}(O)$ is a non f_S-interesting pattern.*

Following these ideas, we propose the *DC-SPMiner* algorithm, which uses the previous propositions to prune the search space of *FSPs*. Let $\prec$ be a linear order in R. *DC-SPMiner* starts the search from $S = \emptyset$. Recursively, if the number of *FSPs* with respect to S is greater than zero or $S = \emptyset$ then S is expanded to each $\hat{S} = S \cup \{r\}$ such that $r \in R - S$, $\forall r' \in S, r' \prec r$, until S can not be expanded.

In order to compute the *FSPs* of an expansion $\hat{S}$, for each $\hat{S}$ we build a structure[1] called $STree^{+}_{\hat{S}}$. This structure is a tree where each path from the root to a leaf represents a subdescription $I_{\hat{S}(O)}$. Each leaf contains: the list of objects in Ω having a subdescription equal to $I_{\hat{S}}(O)$ ($I_{\hat{S}}(O).objs$), the list of pairs $(I_{\hat{S}}(O), f_{\hat{S}}(O', O))$, such that, $O' \in \Omega$, $I_{\hat{S}}(O) \neq I_{\hat{S}}(O')$, $0 < f_{\hat{S}}(O', O)$ ($I_{\hat{S}}(O).similars$) and the amount of partial occurrences of the subdescription $I_{\hat{S}}(O)$ in Ω ($I_{\hat{S}}(O).c_{\approx}$). If $f_{\hat{S}}(O, O') > 0$, then $I_{\hat{S}}(O')$ is a partial occurrence of $I_{\hat{S}}(O)$ with value $f_{\hat{S}}(O, O')$.

We distinguish two cases to build $STree^{+}_{\hat{S}}$: **I**) $|\hat{S}| = 1$. All the objects in Ω are added to $STree^{+}_{\hat{S}}$. After that, for each subdescription P' in $STree^{+}_{\hat{S}}$, the list $P'.similars$ is updated, joining to it only the subdescriptions P in $STree^{+}_{\hat{S}}$, such that $f_{\hat{S}}(P, P') > 0$. **II**) $|\hat{S}| > 1$. For each f_S*-interesting pattern* P in $STree^{+}_{S}$, the objects in $P.objs$ are added to $STree^{+}_{\hat{S}}$. After that, for each subdescription P', the list $P'.similars$ is updated, joining to it only the subdescriptions P, such that $I|_S(P)$ is a *FSP* and $f_{\hat{S}}(P, P') > 0$.

Finally, in both cases, for each subdescription P in $STree^{+}_{\hat{S}}$, $P.c_{\approx}$ is computed and the *FSPs* in $STree^{+}_{\hat{S}}$ are also computed.

3 Experimental Results

In this section, we compare the quality of the set of patterns obtained by *DC-SPMiner*, using the similarity function (1) jointly together with (2) for numerical features against the set of patterns obtained by *ObjectMiner*, *STreeDC-Miner*, and *STreeNDC-Miner* algorithms using the similarity function (1) jointly together with (3) with $\alpha = 0.9$ for numerical features; in both cases, for the non numerical features we used the equality as comparison function. Notice that (1) jointly together with (3), for numerical features, and the equality, for non numerical features, as comparison functions becomes a Booleanization of the non Boolean similarity function used for *DC-SPMiner*. Also, we included the results of *STreeDC-Miner* using the equality as similarity function (*EQ-STreeDC-Miner*).

For the experiments, we used a simple classifier based on *FSPs*. The tested datasets were *Diabetes*, *Liver Disorders*, *Iris* and *Page Blocks*[2]. For each dataset and each value of $minFreq \in \{0.1, 0.2, \ldots, 0.9\}$ we repeated the experiment 10 times, randomly selecting 50% of the dataset for training and using the remaining objects for testing.

[1] This is an extension of the *STree* structure proposed in [5].

[2] http://archive.ics.uci.edu/ml/datasets.html

Table 1 shows that the classification accuracies obtained using the *FSPs* obtained by *DC-SPMiner* are better than those obtained using the *FSPs* obtained by the other 3 algorithms[3]. These results confirm the negative effect of transforming a non Boolean similarity function into a Boolean similarity function.

Table 1. Classification accuracies for *Diabetes*, *Liver Disorders*, *Iris* and *Page Blocks*

		$minFreq$								
Dataset	Algorithm	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Diabetes	*STreeDC-Miner*	71.9	73.4	73.2	72.2	70.5	69.7	62.4	25.1	0.0
	EQ-STreeDC-Miner	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	DC-SPMiner	74.0	74.1	74.1	74.0	74.1	73.8	73.4	71.3	55, 5
Liver Disorders	*STreeDC-Miner*	53.4	53.2	50.0	49.5	43.6	33.2	14.3	0.0	0.0
	EQ-STreeDC-Miner	32.0	20.8	13.3	2.3	0.0	0.0	0.0	0.0	0.0
	DC-SPMiner	55.3	55.2	55.3	54.9	54.1	53.5	55.4	57.1	6.3
Iris	*STreeDC-Miner*	88.3	84.7	74.0	60.1	30.1	15.6	3.1	0.0	0.0
	EQ-STreeDC-Miner	64.9	42.7	23.9	17.9	15.1	7.5	1.3	0.0	0.0
	DC-SPMiner	92.3	92.3	92.0	92.3	92.3	90.8	85.1	55.9	0.0
Page Blocks	*STreeDC-Miner*	38.1	26.6	22.9	13.9	8.4	5.6	3.3	0.8	0.7
	EQ-STreeDC-Miner	29.7	26.6	9.2	4.9	3.5	2.5	1.8	1.1	0.7
	DC-SPMiner	42.6	41.1	41.0	40.3	38.0	28.3	19.0	10.0	0.7

It can also be noticed that using the equality function, like in the classical approach, the classification accuracies reached by the set of frequent patterns are lower than those reached by the set of *FSPs* obtained using both Boolean or non Boolean similarity functions.

4 Conclusions

From the experiments, we can conclude that, for problems where the similarity function is non Boolean, the quality of the frequent similar patterns computed by our algorithm is higher than the quality of those obtained by *STreeDC-Miner*, *ObjectMiner* and *STreeNDC-Miner* which need to Booleanize the similarity function. Moreover, the quality of the frequent similar patterns computed by our algorithm is also higher than the quality of the frequent patterns obtained using the classical approach, which uses the equality as similarity function.

References

[1] Han, J., Cheng, H., Xin, D., Yan, X.: Frequent Pattern Mining: Current Status and Future Directions. Data Mining and Knowledge Discovery 15(1), 55–86 (2007)

[2] Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules between Sets of Items in Large Databases. In: 1993 ACM SIGMOD International Conference on Management of Data, Washington, USA, pp. 207–216 (1993)

[3] Ruiz-Shulcloper, J.: Pattern Recognition with Mixed and Incomplete Data. Journal Pattern Recognition and Image Analysis 18(4), 563–576 (2008)

[3] *ObjectMiner*, *STreeDC-Miner*, and *STreeNDC-Miner* obtain the same set of *FSPs*, therefore we only show the accuracies of *STreeDC-Miner*.

[4] This work is partly supported by the National Council of Science and Technology of Mexico under the project CB2008-106443 and grant 32086.

[4] Dánger, R., Ruiz-Shulcloper, J., Berlanga, R.: Objectminer: A New Approach for Mining Complex Objects. In: Sixth International Conference on Enterprise Information Systems, Oporto, Portugal, pp. 42–47 (2004)

[5] Rodríguez-González, A.Y., Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A., Ruiz-Shulcloper, J.: Mining Frequent Similar Patterns on Mixed Data. In: Ruiz-Shulcloper, J., Kropatsch, W.G. (eds.) CIARP 2008. LNCS, vol. 5197, pp. 136–144. Springer, Heidelberg (2008)

[6] Martínez-Trinidad, J.F., Ruiz-Shulcloper, J., Lazo-Cortés, M.S.: Structuralization of Universes. Fuzzy Sets and Systems 112(3), 485–500 (2000)

Evaluation of Rare Event Detection

Marina Sokolova[1], Khaled El Emam[1,2], Sadrul Chowdhury[1], Emilio Neri[1],
Sean Rose[1], and Elizabeth Jonker[1]

[1] Children's Hospital of Eastern Ontario
{msokolova,schowdhury,eneri,srose,ejonker}@ehealthinformation.ca
[2] University of Ottawa
kelemam@uottawa.ca

Abstract. This study analyzes evaluation measures for rare event detection. We introduce a procedure which is built upon the characteristics of rare events. We propose properties for evaluation measures which assess the measure applicability to classification of rare events. Prevention of leaks of personal health information supports the empirical evidence.

Keywords: Evaluation measures, classification, rare events.

1 Motivations

Performance evaluation is an integral part of Data Mining (DM) and Machine Learning (ML). Researchers have intensively studied performance measures for association and classification rules. Several measure properties have been proposed, such as symmetry under variable permutation, and an increasing function of support (given the fixed margins in the contingency table)[1]. Performance measures have been assessed through their ability to reflect on changes in training and test data, class distribution and reliability of data labels [2].

We consider performance evaluation of systems that detect *rare* events. By rare events, we mean extremely infrequent events whose characteristics make them or their consequences highly valuable. Such events appear with extreme scarcity and are hard to predict, although they are expected eventually.[1] We propose a multi-step process of rare event classification. The process is based on both positive and negative evidence about happening of the event [3]. We, then, analyze applicability of performance measures which are used in binary classification. Only the effectiveness of class identification is considered at this stage, leaving computational costs and time efficiency for future work.

Our procedure can be used in personal health information (PHI) leak detection over peer-to-peer networks [4].[2] Leakage of patient's PHI can be costly and damage the reputation of data custodians. It has been shown that in some domains PHI leaks may constitute approx. 0.5% in terms of the number of files. To find those files, we want to use systems which effectively detect PHI among

[1] http://www.fas.org/irp/agency/dod/jason/rare.pdf

[2] A PHI file contains a person's ID, a geographic pointer and health information.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 379–383, 2010.

Table 1. PHI detection: the conviction categories and text examples

Negative conviction		Positive conviction	
impossible	fiction, music files	*possible*	*uncertain* texts with location pointers,
improbable	assignments, manuals	*probable*	the *possible* texts with person names,
uncertain	other text files	*certain*	the *probable* texts with health information

vast volumes of unrelated data, e.g.,one page medical authorization notes among Twilight and Harry Potter books. We show that the measure's *convergence* helps to make an appropriate choice among the classification systems.

2 Rare Event Detection

Rare events. So far, DM and ML do not have a unified definition of what constitutes a rare event: some authors impose conditions on prior distributions $p_{non-rare} >> p_{rare}$ [5], others constrain its proportion in data from 0.1% to $<$ 10%[3]. Those criteria address only occurrence of the events in data. Taken in isolation, this condition does not indicate that the less frequent event has a special significance as the classes may have approximately the same importance with respect to a given task. In contrast, we consider that rare events are significantly more important than other, non-rare events. Furthermore, humans can reliably describe the rare events within the applications and differentiate them from other, non-rare events. Based on these, we list the additional properties which are critical in defining whether an extremely infrequent event indeed forms a rare event: (**i**) rare events exhibit specific, outstanding characteristics; (**ii**) those characteristics make the events, or their consequences, highly valuable.

We start with the identification of the characteristics that define an example as a rare event. We, then, assign examples with the following labels: ***negative conviction***, i.e. *impossible, improbable, uncertain*, based on the negative evidence that the example *does not represent* the rare event; ***positive conviction***, i.e. *certain, probable, possible*, based on the positive evidence that the example *represents* the rare event. Our procedure begins by filtering out examples marked with "negative" evidence and, then, concentrates on processing of examples marked with "positive" evidence. Table 1 shows texts in these conviction categories when we have applied this system in PHI leak detection over peer-to-peer networks.

Performance evaluation. We compute correctly recognized rare events (tp), correctly recognized non-rare events (tn), and examples that were either incorrectly assigned to the rare events (fp) or not recognized as the rare event examples (fn). Measures can assess performance on the positive conviction labels if they depend on tp. A measure's dependence on tn shows that it can assess performance on the negative conviction labels. We assume that all the rare events

[3] http://www-users.cs.umn.edu/~aleks/pakdd04_tutorial.pdf

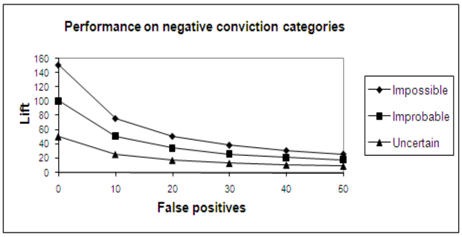


Fig. 1. A converging measure

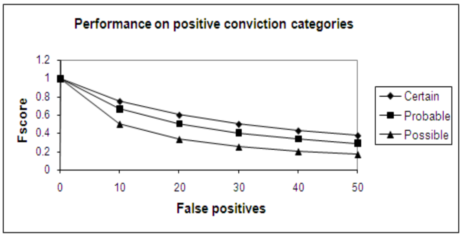


Fig. 2. A diverging measure

are identified correctly, i.e. *fn*=0. As a result, *fp* becomes the main indicator of the faulty event identification. We, thus, are interested in how measures reflect changes in *fp*. To incorporate the strength of conviction, we substitute *tp* and *tn* by $\alpha_i tp$ and $\beta_j tn$, where $i \in \{certain, probable, possible\}, \alpha_{certain} > \alpha_{probable} > \alpha_{possible}$, $j \in \{uncertain, improbable, impossible\}, \beta_{uncertain} < \beta_{improbable} < \beta_{impossible}$. The coefficient values can be determined empirically.

Evaluation measures. We propose that *convergence* of measures is important when the measures assess the system performance on examples belonging to different conviction categories. Convergence is estimated for positive (negative) conviction categories. We use *tp*=10, *fp*=0,..., 50, *tn*= 1000, *fn*= 0, $\alpha_{certain} = 1.5, \alpha_{probable} = 1.0, \alpha_{possible} = 0.5$, $\beta_{uncertain} = 0.5, \beta_{improbable} = 1.0, \beta_{impossible} = 1.5$. The measure may not be able to distinguish among the categories if its values converge (Figure 1); in contrast, the diverging measures are able to distinguish among the categories (Figure 2). Table 2 shows the measures, their formulas, and the convergence properties (Cvrg).

3 Summary and Future Work

Our study has introduced a framework for measure assessment when those measures evaluate a rare event classification system, and have derived formulas for those measures. Mining for rare events is often application-driven (*e.g.* intrusion protection and anomaly detection [5]). In [6], the authors analyze how classification results are affected by data set sizes and class distribution. Our work, in contrast, addresses measure properties, such as convergence. In [7], discussing rare class mining, the authors consider *Precision*, *Recall*, and *Fscore* to be preferred metrics in comparison with *Accuracy*. We, instead, propose that measures should be compared based on tolerance to false positive rates. In [8], the author suggests measure appreciation based only on dependency on the recall and precision of the rare class. Our use of positive and negative conviction categories allows comparison of performance on both rare and non-rare examples.

We have proposed a new approach which is based on characteristics of rare events. The measures, then, are considered with respect to their ability to comply with requirements for rare event detection. We have shown that certain measures

Table 2. Performance measures; $\surd$ means convergence, - shows the opposite, fn=0

Measure	Positive conviction		Negative conviction	
	Formula	Cvrg	Formula	Cvrg
Accuracy	$\frac{\alpha_i tp + tn}{\alpha_i tp + fp + tn}$	-	$\frac{tp + \beta_j tn}{tp + fp + \beta_j tn}$	-
AUC	$\frac{1}{2}(\frac{\alpha_i tp}{\alpha_i tp} + \frac{tn}{tn + fp})$	$\surd$	$\frac{1}{2}(\frac{tp}{tp} + \frac{\beta_j tn}{\beta_j tn + fp})$	-
Certainty factor	$\frac{\alpha_i tp(\alpha_i tp + tn + fp)}{(\alpha_i tp + fp)(fp)}$ - $\frac{\alpha_i tp}{tn + fp}$	-	$\frac{tp(tp + \beta_j tn + fp)}{(tp + fp)(\beta_j tn + fp)}$ - $\frac{tp}{\beta_j tn + fp}$	-
Conviction	$\frac{(\alpha_i tp + fp)(tn + fp)}{(\alpha_i tp + tn + fp)fp}$	$\surd$	$\frac{(tp + fp)(\beta_j tn + fp)}{(tp + \beta_j tn + fp)fp}$	-
Fscore	$\frac{(\gamma^2 + 1)\alpha_i tp}{(\gamma^2 + 1)\alpha_i tp + fp}$	-		
Information Gain	$\log \frac{\alpha_i tp(\alpha_i tp + fp + tn)}{(\alpha_i tp + fp)(\alpha_i tp)}$	$\surd$	$\log \frac{tp(tp + fp + \beta_j tn)}{(tp + fp)tp}$	$\surd$
Leverage	$\frac{\alpha_i tp}{\alpha_i tp + fp}$ - $\frac{(\alpha_i tp)(\alpha_i tp + fp)}{(\alpha_i tp + fp + tn)^2}$	-	$\frac{tp}{tp + fp}$ - $\frac{tp(tp + fp)}{(tp + fp + \beta_j tn)^2}$	$\surd$
Least contradiction	$\frac{\alpha_i tp - fp}{tp}$	-		
Lift	$\frac{\alpha_i tp(\alpha_i tp + tn + fp)}{(\alpha_i tp + fp)(\alpha_i tp)}$	$\surd$	$\frac{tp(tp + \beta_j tn + fp)}{(tp + fp)tp}$	$\surd$
Loevinger	$\frac{-\alpha_i tp tn}{fp(\alpha_i tp + fp + tn)}$	$\surd$	$\frac{-tp \beta_j tn}{fp(tp + fp + \beta_j tn)}$	-
Odd multiplier	$\frac{\alpha_i tp(fp + tn)}{(\alpha_i tp)fp}$	$\surd$	$\frac{tp(fp + \beta_j tn)}{tp fp}$	$\surd$
Precision	$\frac{\alpha_i tp}{\alpha_i tp + fp}$	-		
Recall (Sensitivity)	$\frac{\alpha_i tp}{\alpha_i tp}$	-		
Sebag – Schoenauer	$\frac{\alpha_i tp}{fp}$	$\surd$		
Specificity			$\frac{\beta_j tn}{fp + \beta_j tn}$	-
Support change	$\frac{\alpha_i tp}{\alpha_i tp + fp}$ - $\frac{\alpha_i tp}{\alpha_i tp + fp + tn}$	-	$\frac{tp}{tp + fp}$ - $\frac{tp}{tp + fp + \beta_j tn}$	-

are better suited to evaluate system components that filter examples out (the right part of Table 2), whereas some measures are useful for evaluation of components which filter the examples in (the left part of Table 2). For the case study, we considered prevention of leaks of Personal Health Information [4]. For future work, we plan to gather more empirical evidence of PHI leak detection. From the text analysis perspective, we want to improve classification of documents marked as *probable*. This step should reduce the number of documents processed during the last, most computationally expensive step.

Acknowledgements. The authors thank CIHR and NSERC for the research funding and the anonymous reviewers - for helpful comments.

References

1. Geng, L., Hamilton, H.: Interestingness measures for data mining: A survey. ACM Computing Surveys 38(3), 1–32 (2006)
2. Sokolova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. Information Processing & Management 45(14), 427–437 (2009)
3. Horn, L.: A Natural History of Negation. The University of Chicago Press (1989)
4. El Emam, K., Neri, E., Jonker, E., Sokolova, M., Peyton, L., Neisa, A., Scasa, T.: The inadvertent disclosure of personal health information through peer-to-peer file sharing programs. JAMIA 17, 148–158 (2010)

5. Kumar, V., Srivastava, J., Lazarevic, A.: Managing Cyber Threats: Issues, Approaches and Challenges. Springer, Heidelberg (2005)
6. Seiffert, C., Khoshgoftaar, T., Hulse, J.V., Napolitano, A.: Mining data with rare events. In: Proceedings of the ICTAI 2007, vol. 2, pp. 132–139 (2007)
7. Han, S., Yuan, B., Liu, W.: Rare class mining: Progress and prospect. In: Proceedings of the 2009 Chinese Conference on Pattern Recognition, pp. 137–141 (2009)
8. Joshi, M.: On evaluating performance of classifers for rare classes. In: Proceedings of the ICDM 2002, pp. 641–644 (2002)

On Multi-robot Area Coverage

Pooyan Fazli

Department of Computer Science
University of British Columbia
Vancouver, B.C., Canada, V6T 1Z4
pooyanf@cs.ubc.ca

Abstract. Area coverage is one of the emerging problems in multi-robot coordination. In this task a team of robots is cooperatively trying to observe or sweep an entire area, possibly containing obstacles, with their sensors or actuators. The goal is to build an efficient path for each robot which jointly ensure that every single point in the environment can be seen or swept by at least one of the robots while performing the task.

Keywords: Multi-Robot Systems, Teamwork, Coordination, Area Coverage, Offline Coverage, Heterogeneity, Open Systems, Online Coverage, Communication.

1 Introduction

Multi-robot area coverage is receiving considerable attention due to its applicability in different scenarios such as search and rescue operations, planetary exploration, intruder detection, environment monitoring, floor cleaning and the like. Several optimization metrics can be considered in building the coverage paths for the robots including time, length of the longest path, sum of the path lengths, initial location of the robots and so on.

There is confusion in the literature regarding the terms *Coverage* and *Exploration*. To clarify the problem definition, we note that in exploration, we have an unknown environment and a team of robots is trying to build a map of the area together [14,3]. In a coverage problem, the map of the environment may be known or unknown and the goal of the team is to jointly observe/sweep the whole area with their sensors or physical actuators. Building a map of the environment is not the ultimate aim of the coverage mission.

Another similar class of problems is *Boundary Coverage* in which, the purpose of coverage is slightly different from the original coverage problem we have defined in this paper. In boundary coverage the aim is to inspect all of the obstacles' boundaries by a team of robots instead of complete coverage of the area [13].

Several research communities including robotics/agents [5], sensor networks [2,8] and computational geometry [4] work on this class of area coverage problems. In computational geometry, this problem stems from the *Art Gallery* problem [9] and its variation for mobile guards called the *Watchman Route* problem [10]. In the *Art Gallery* problem, the goal is to find a minimum number of static guards (control points) which jointly can cover a work space under different restrictions. On the other hand, in the *Watchman Route* problem the objective is

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 384–387, 2010.

to compute routes watchmen should take to guard an entire area given only the map of the environment. Most research done on the above problem definitions deal with simple polygons without obstacles, unconstrained guard visibility, and single watchman scenarios.

From a robotics point of view, in a taxonomy presented by Choset [5], the proposed approaches for area coverage are divided into *offline* methods, in which the map of the environment is known, and *online* methods, in which the map of the environment is unknown. Choset [5] further divides the approaches for area coverage based on the methods they employ for decomposing the area: *Exact Cellular Decomposition*, and *Approximate Cellular Decomposition.*

Previous research on multi-robot area coverage is mostly focused on approaches using the *Approximate Cellular Decomposition* (e.g. grid-based methods) [1,15]. These methods have limitations since they do not consider the structure of the environment and have difficulties handling partially occluded cells or covering areas close to the boundaries in continuous spaces. In contrast, methods based on the *Exact Cellular Decomposition* (e.g. graph-based methods) which employ structures such as the *visibility graph* for environment representation do not suffer those restrictions [11,12]. However, while traversing a *visibility graph* guarantees covering the whole environment in continuous spaces, it might include many redundant movements.

2 Health Care Facilities: A Sample Task Domain

Hospitals, nursing homes and eldercare facilities are suitable environments where robots can help a lot when human care is limited for various reasons such as demographic issues, emergencies, conflicts, lack of funding and so on.

This thesis is motivated by the situations could occur in future health care facilities. Specifically, we are interested in scenarios in which a robot team together monitors the environment (e.g. for fire detection or emergency handling) and the persons living in it (e.g. for tracking patients), and also provide the functionalities necessary in everyday life within those health care facilities (e.g. food delivery). For these purposes, the robots need to repeatedly cover the whole target area over time.

3 Thesis Contributions

The following are the expected contributions of the proposed work.

3.1 Offline Coverage

We present a new algorithm for covering a known polygonal environment cluttered with static obstacles by a team of mobile robots [6,7]. To this end, two environment representation methods called *Reduced-CDT* and *Reduced-Vis* based on the *Constrained Delaunay Triangulation* and the *Visibility Graph* are introduced so as to model the structure of the target area more efficiently. Also, due to the distributed characteristic of the coverage problem, another algorithm called *Multi-Prim's* is applied to decompose the graph representing the environment into a forest of *partial spanning trees*. Each tree is modified through a mechanism called *Constrained Spanning Tour* to build a cycle which is then assigned to a covering robot. Subsequently, robots start navigating the cycles and consequently cover the whole area. This method has the benefit that it returns the robots to their initial locations, facilitating tasks like collection and storage.

The proposed approach is complete, meaning that every accessible point in the environment will be visited in a finite time. Furthermore, it supports robustness by handling individual robot failure. Our approach has been designed to overcome the restrictive constraint imposed by the robots' limited visibility range as well.

Experiments demonstrate the ability of the robot team to cover the target area based on metrics such as the number of robots in the team, visibility range of the robots, and frequency of robot failure during the coverage mission. We are also going to compare the performance of the proposed approach based on the two methods being used for area representation, i.e. *Visibility Graph* and the *Constrained Delaunay Triangulation*.

3.2 Heterogeneity

Dealing with robots' heterogeneity is an interesting challenge in this domain. Heterogeneity can be defined in different aspects and contexts, such as different movement capabilities (movement model or velocity constraints) or different task handling abilities. However, the focus of this thesis is to investigate different sensing capabilities in a team of robots. In this variation, the aim is to cover and monitor an environment by using robots with differing visibility ranges; say, one robot could see the objects located in $r1$-distance while the other robot could recognize the objects located in $r2$-distance.

3.3 Open Systems

We will investigate the situations in which new robots may join the team in the course of the coverage mission. To this end, we propose an algorithm to handle the situation dynamically by efficient task re-allocation among the robots. The goal is to distribute the workload among the team by taking advantage of the new resources (robots) available in the system.

3.4 Online Coverage

We will investigate the area coverage task in partially or completely unknown environments, where just local knowledge of the area is provided to the robots. The goal is to present an online, decentralized algorithm for multi-robot area coverage instead of the offline, centralized method discussed earlier.

3.5 Communication

We will explore the development of new algorithms to handle communication failure or message loss during the coverage mission. The proposed approaches should also support the case when the robots have a limited range of communication, meaning that a message sent by a robot is transmitted only to robots within a certain distance from that robot.

Acknowledgment. This work is being supervised by Prof. Alan K. Mackworth at the Computer Science Department of the University of British Columbia.

References

1. Agmon, N., Hazon, N., Kaminka, G.A.: The giving tree: constructing trees for efficient offline and online multi-robot coverage. Annals of Mathematics and Artificial Intelligence 52(2-4), 143–168 (2008)
2. Akyildiz, I.F., Weilian, S., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. IEEE Communications Magazine 40(8), 102–114 (2002)
3. Burgard, W., Moors, M., Fox, D., Simmons, R., Thrun, S.: Collaborative multi-robot exploration. In: Proceedings. of the IEEE International Conference on Robotics and Automation, ICRA 2000, vol. 1, pp. 476–481 (2000)
4. Carlsson, S., Nilsson, B.J., Ntafos, S.C.: Optimum guard covers and m-watchmen routes for restricted polygons. International Journal of Computational Geometry and Applications 3(1), 85–105 (1993)
5. Choset, H.: Coverage for robotics – a survey of recent results. Annals of Mathematics and Artificial Intelligence 31(1-4), 113–126 (2001)
6. Davoodi, A., Fazli, P., Pasquier, P., Mackworth, A.K.: On multi-robot area coverage. In: Proceedings of the 7th Japan Conference on Computational Geometry and Graphs, JCCGG 2009, pp. 75–76 (2009)
7. Fazli, P., Davoodi, A., Pasquier, P., Mackworth, A.K.: Multi-robot area coverage with limited visibility. In: Proceedings of The 9th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2010 (2010)
8. Krause, A., Guestrin, C., Gupta, A., Kleinberg, J.: Near-optimal sensor placements: maximizing information while minimizing communication cost. In: Proceedings of the 5th international conference on Information processing in sensor networks, IPSN 2006, pp. 2–10 (2006)
9. O'Rourke, J.: Art gallery theorems and algorithms. Oxford University Press, New York (1987)
10. Packer, E.: Computing multiple watchman routes. In: McGeoch, C.C. (ed.) WEA 2008. LNCS, vol. 5038, pp. 114–128. Springer, Heidelberg (2008)
11. Rekleitis, I.M., Dudek, G., Milios, E.E.: Multi-robot exploration of an unknown environment, efficiently reducing the odometry error. In: Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI 1997, pp. 1340–1345 (1997)
12. Rekleitis, I.M., Lee-Shue, V., Peng New, A., Choset, H.: Limited communication, multi-robot team based coverage. In: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2004, vol. 4, pp. 3462–3468 (2004)
13. Williams, K., Burdick, J.: Multi-robot boundary coverage with plan revision. In: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2006, pp. 1716–1723 (2006)
14. Yamauchi, B.: Frontier-based exploration using multiple robots. In: Proceedings of the second international conference on Autonomous agents, AGENTS 1998, pp. 47–53 (1998)
15. Zheng, X., Jain, S., Koenig, S., Kempe, D.: Multi-robot forest coverage. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2005, pp. 3852–3857 (2005)

Exploiting Rhetorical Relations in Blog Summarization

Shamima Mithun

Concordia University
Department of Computer Science and Software Engineering
Montreal, Quebec, Canada
s_mithun@encs.concordia.ca

Abstract. With the goal of developing an efficient query-based opinion summarization approach, we have targeted to resolve Question Irrelevancy and Discourse Incoherency problems which have been found to be the most frequently occurring problems for opinion summarization. To address these problems, we have utilized rhetorical relations of texts with the help of text schema and Rhetorical Structure Theory (RST).

1 Introduction

Due to the rapid growth of the Social Web, huge amount of informal opinionated texts are created after an event. Natural language tools to help analyze these texts in order to facilitate decisions making at every level (e.g. individuals are interested to know others' opinions when they are intended to purchase some products or services) have become a necessary. Query-based opinion summarizers from opinionated documents, as introduced in 2008 at the Text Analysis Conference (TAC), can address this need. Query-based opinion summarizers present what people think or feel on a given topic in a condensed manner to analyze others' opinions regarding a specific question. This research interest motivated us to develop an effective query-based multi-document opinion summarization approach for blogs.

The TAC 2008 summarization results [6] show that blog summarizers typically perform worse than news summarizers. To analyze this in greater detail, we first tried to identify and categorize problems which typically occur in opinion summarization through an error analysis of the current blog summarizers. The goal of our research is to develop a blog summarization approach that addresses these most frequently occurring problems. For this error analysis, we used summaries from participating systems of the TAC 2008 summarization track. Our study [6] shows that *Question Irrelevancy*, *Topic Irrelevancy*, *Discourse Incoherency*, and *Irrelevant Information* are the most frequently occurring problems for blog summarization and the first three problems occur more frequently in blog summarization compared to news summarization. Here is a sample summary (taken from the TAC 2008 opinion summarization track) that contains *Question Irrelevancy* and *Discourse Incoherency* problems.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 388–392, 2010.

Topic: *Carmax*
Question: *What motivated positive opinions of Carmax from car buyers?*
Summary: *At Carmax, the price is the price and when you want a car you go get one. Arthur Smith, 36, has been living in a van outside the Carmax lot, 24 hours a day, for more than a month. [...]*

The second sentence of the sample summary is not relevant to the question; it exhibits a *Question Irrelevancy* problem. Moreover, in the summary, sentences are not interlinked; as a result, they create a *Discourse Incoherency* problem. In our work, we target to deal with *Question Irrelevancy* and *Discourse Incoherency* and we believe our content selection approach may also reduce *Topic Irrelevancy.*

To handle *Question Irrelevancy* and *Discourse Incoherency* problems, we have utilized rhetorical relations because rhetorical relations have been found useful in news summarization (e.g. [1]), in natural language generation (e.g. [5]), and in other areas of NLP. However, to the best of our knowledge, rhetorical relations have never been used for blog summarization. In order to utilize rhetorical relations for blog summarization, we have employed text schema [5] and Rhetorical Structure Theory (RST) [3] which are two standard text organization approaches based on rhetorical relations.

2 Rhetorical Relations in Blog Summarization

With the goal of reducing *Question Irrelevancy* and *Discourse Incoherency* of blog summarization to utilize rhetorical relations, we have adopted McKeown's text schema [5]. The text schema approach shows that by using an organizational framework (a combination of rhetorical predicates called schemata) for the text, a system can generate coherent multi-sentential texts given a communicative goal where rhetorical predicates delineate the structural (rhetorical) relations between propositions in a text. In a text-schema based approach, one can design and associate appropriate schemata (e.g. *compare and contrast*) to generate a summary that answers specific types of questions (e.g. *comparative*, *suggestion*). In the schema design, one can define constraints on the types of predicates (e.g. *analogy*, *condition*) and the order in which they should appear in the output summary for a particular question type. One can also specify constraints for each predicate of a schema to fulfill the communicative goal. These characteristics of the text schema-based approach should help to filter question irrelevant sentences and improve the coherence of the output summaries.

In the text schema-based approach, the most challenging task is to identify which rhetorical predicate (e.g. *analogy*, *comparison*) is communicated by a candidate sentence in order to figure out if it should be included in the summary and where. In previous schema-based systems (e.g.[5]), the application domain is typically represented as a knowledge base and the structure of the knowledge base is used to identify predicates. In certain sub-languages, predicates are often identified by means of key words and other clues (e.g. *because*, *if*, *then*). To the best of our knowledge, there does not exist an approach to identify rhetorical predicates which is domain and genre independent.

Rhetorical relations modelled by the Rhetorical Structure Theory (RST) [3] express discourse relations characterized by rhetorical predicates with constraints on nucleus and satellites and effects on readers. Various rhetorical and discourse relations listed in different rhetorical predicates are comparable to those in RST. As rhetorical predicates and rhetorical relations in RST are comparable and automatic approaches to identify RST-based rhetorical relations are available, we propose a new way of identifying rhetorical predicates for any domain by making use of rhetorical relations in RST. We are using the RST-based discourse parser SPADE [7], which can automatically identify RST-based rhetorical relations within a sentence, in combination with dependency relations from the Stanford parser [4] and a comparative relations classifier [2] to identify rhetorical relations within a sentence. We have built a system called BlogSum to test our proposed summarization approach and currently, we are evaluating it using the TAC 2008 opinion summarization track data.

3 BlogSum Description

Given an initial query and a set of related blogs, our summarization approach performs two main tasks: *content selection* and *content organization* but the novelty of our work is on content organization.

3.1 Content Selection

Content selection performs the following tasks: blog pre-processing, question polarity identification, and sentence ranking to generate a ranked list of candidate sentences. To rank sentences, BlogSum mainly ranks sentences based on their similarity with the question and the topic and the subjectivity scores. To calculate the question (or topic) similarity, we used the cosine similarity between the sentence and the question (or topic). Sentences and questions (or topic) are represented as a weighted word vector based on *tf.idf* (for sentences) and tf (for questions or topic). BlogSum uses the MPQA subjectivity lexicon[1] to find the subjectivity score of a sentence. Redundant sentences are also removed based on the cosine similarity measure from the candidate sentence list.

3.2 Content Organization

The role of content organization is to select a few candidate sentences and order them so as to produce a coherent and a query relevant summary. For content organization, BlogSum performs the following main tasks: question categorization, schema selection, predicate identification, and summary generation. By analyzing the TAC 2008 opinion summarization track questions manually, we have categorized them into 3 categories based on their communicative goals : *1. comparative* - e.g. Why do people like Starbucks better than Dunkin Donuts?, *2.*

[1] Available at `http://www.cs.pitt.edu/mpqa/`

suggestion - e.g. What do Canadian political parties want to happen regarding NAFTA?, *3. reason* - e.g. Why do people like Mythbusters? We have also designed three schemata, one for each question type; 1) *comparative*, 2) *suggestion*, and 3) *reason*. BlogSum selects the schema based on the question categories. Then to fill in the selected schema for a particular question type using candidate sentences to generate the output summary, BlogSum needs to classify each sentence into a predefined set of rhetorical predicates; we called this process, predicate identification. For predicate identification, we first defined a set of rhetorical predicates (e.g. *comparison*, *contingency*) to be used; then candidate sentences are classified into these predicates. We are using the RST-based discourse parser SPADE [7] in combination with dependency relations from the Stanford parser [4] and a comparative relations classifier [2] to identify rhetorical relations within a sentence. The SPADE parser identifies discourse relations within a sentence by first identifying elementary discourse units (EDU)s, then identifying rhetorical relations between two EDUs (clauses) by following the RST theory. In this process, each sentence processed by the SPADE parser is labelled with its rhetorical relations. BlogSum uses these relations to classify the sentence into the corresponding rhetorical predicate. If the SPADE parser fails to identify any relation for a given sentence, we use other approaches for this purpose.

4 Conclusion and Future Work

In order to develop an efficient query-based opinion summarization approach for blogs, we have utilized the rhetorical relations by combining text schema and Rhetorical Structure Theory (RST) with the goal of reducing question irrelevance and discourse incoherence of the summaries. Currently, we are evaluating our summarization approach for content which will give an indication of question relevance and linguistic quality especially coherence and overall readability. In future, we plan to incorporate "sentence ordering" in our current summarization approach to improve the coherence of BlogSum-generated summaries as text schema provides a partial order of texts. Future work will also be focus on further improvements of our approach based on the evaluation results.

References

[1] Bosma, W.: Query-Based Summarization using Rhetorical Structure Theory. In: 15th Meeting of Computational Linguistics in the Netherlands CLIN, Leiden, Netherlands, pp. 29–44 (2004)

[2] Jindal, N., Liu, B.: Identifying Comparative Sentences in Text Documents. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, USA, pp. 244–251 (2006)

[3] Mann, W.C., Thompson, S.A.: Rhetorical Structure Theory: Toward a Functional Theory of Text Organisation. J. Text 3(8), 234–281 (1988)

[4] de Marneffe, M.C., Manning, C.D.: The Stanford Typed Dependencies Representation. In: CrossParser 2008: Coling 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation, Manchester, UK, pp. 1–8 (2008)

[5] McKeown, K.R.: Discourse Strategies for Generating Natural-Language Text. J. Artificial Intelligence 27(1), 1–41 (1985)

[6] Mithun, S., Kosseim, L.: Summarizing Blog Entries versus NewsTexts. In: Proceedings of Events in Emerging Text Types (eETTS), A Workshop of Recent Advances in Natural Language Processing RANLP, Bulgaria, pp. 35–42 (2009)

[7] Soricut, R., Marcu, D.: Sentence Level Discourse Parsing using Syntactic and Lexical Information. In: NAACL 2003: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, Edmonton, Canada, pp. 149–156 (2003)

Peer-Based Intelligent Tutoring Systems: A Corpus-Oriented Approach

John Champaign

David R. Cheriton, School of Computer Science
University of Waterloo, Waterloo, ON, Canada
jchampai@uwaterloo.ca

Abstract. Our work takes as a starting point McCalla's proposed ecological approach for the design of peer-based intelligent tutoring systems and proposes three distinct directions for research. The first is to develop an algorithm for selecting appropriate content (learning objects) to present to a student, based on previous learning experiences of like-minded students. The second is to build on this research by also having students leaving explicit annotations on learning objects to convey refinements of their understanding to subsequent students; the challenge is to intelligently match students to those annotations that will be most beneficial for their tutoring. The third is to develop methods for intelligently extracting learning objects from a repository of knowledge, in a manner that may be customized to the needs of specific students. In order to develop our research we are exploring the specific application of assisting health care workers via peer-based intelligent tutoring.

1 Content Sequencing

Two central challenges in the design of intelligent tutoring systems are compiling the material for the lessons and determining the best methods to use, for the actual teaching of those lessons. We observe in particular that it is desirable to provide a framework for determining the material to be taught that does not rely on experts hand-coding all the lessons. Indeed, that particular approach presents considerable challenges in time and effort.

We are interested in techniques for bootstrapping the system in order to initiate peer-based learning and in developing robust methods for validating the models that are presented (including the technique of employing simulated students). Once the content is in place, our efforts will be aimed at refining our model in order to enable students to benefit the most from the learning that their peers are undergoing.

We have currently developed an algorithm for reasoning about the sequencing of content for students in a peer-based intelligent tutoring system inspired by McCalla's ecological approach[1]. We record with each learning object those students who experienced the object, together with their initial and final states of knowledge, and then use these interactions to reason about the most effective

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 393–396, 2010.

lessons to show future students based on their similarity to previous students[1]. As a result we are proposing a novel approach for peer-to-peer intelligent tutoring from repositories of learning objects.

We used simulated students to validate our content sequencing approach. Our motivation for performing this simulation was to validate that, in the experimental context, our approach leads to a higher average learning by the group of students than competing approaches. We added a modeling of the knowledge that each object is aimed at addressing (for example, an object in a first year computer science course may be aimed at addressing the knowledge of recursion). By abstracting all details from the intelligent tutoring system and the student, we defined a formula to simulate learning (Equation 1).

$$\Delta UK[j,k] = \frac{I[l,k]}{1 + (UK[j,k] - LOK[l,k])^2} \tag{1}$$

where UK is the user j's understanding of knowledge k, I is the educational benefit (how much it increases or decreases a student's knowledge) of learning object l on knowledge k and LOK is the learning object l's target level of instruction for knowledge k.

When running our algorithm in the simulation, each student would be presented with the learning object that was expected to bring the greatest increase in learning, determined by extracting those learning objects that had resulted in the greatest benefit for previous students considered to be at a similar level of understanding as the current students.[2]

Simulated students and content allowed us to avoid the expense of implementing and experimenting with an ITS and human students to see the impact of our approach in contrast with alternative approaches. In particular we contrasted our method with a baseline of randomly assigning students to learning objects and to a "look ahead" greedy approach where the learning was precalculated and used to make the best possible match. One variant we considered was a "simulated annealing" inspired approach, where greater randomness was used during the initial, exploratory phase of the algorithm, then less randomness was used once more information about learning objects had been obtained. We discovered that our approach showed a clear improvement over competing approaches and approached the ideal.

2 Annotation

To extend the basic evolutionary approach, we are particularly interested in exploring the use of student annotations, which would fit naturally with our proposed corpus-based design for the lesson base. Student annotations on learning objects would involve allowing students to leave short comments on lessons

[1] This is motivated by techniques from collaborative filtering recommenders.

[2] Each student in the simulation is modeled to have a current level of understanding for each possible knowledge area, a value from [0,1] reflecting an overall grade from 0 to 100.

they are interacting with (more than tags, this could be a question or commentary about what they're learning). Subsequent students would identify which annotations they found useful, which would then be intelligently shown to similar students. The idea behind this is allowing students to "collaborate" with one another but not in real time (or, at least, to allow the interactions of the student in the past to inform the interaction with the current student, which honours the ecological approach[1]). There will be a decision theoretic reasoning element to this, when "low quality" annotations should be shown as part of a dialogue involving high quality annotations, and some trust modeling in addition to student modeling similar to what we are advocating for Content Sequencing.

To date we have only developed preliminary steps towards an overall algorithm for reasoning about annotations. We have not yet explored how best to validate our approach.

3 Corpus-Based

We are interested in exploring the construction of the lesson base that forms the centrepiece of a peer-based intelligent tutoring system, and are concerned with facilitating the authoring of such a lesson base, through the mining of existing repositories of information. This stands in contrast to McCalla's ecological work[1], which assumes that learning objects are already created and available to the system. This would be especially useful for applications where large repositories of information already exist, possibly employing varied forms of media, that could be leveraged for the creation of an ITS. Towards this end we have been exploring scenarios applicable to peer-based home healthcare assistance for caregivers or patients (e.g. facilitating a healthcare decision by presenting a learning object that has been useful to others in the past). Working in conjunction with health care workers affiliated with our hSITE (Healthcare Support through Information Technology Enhancements) project (an NSERC Strategic Research Network), our aim is to design a system that will be of some benefit in actual healthcare environments.

For future work, we are interested in refining learning objects created for a variety of purposes (e.g. book chapters, instructional videos or research papers) by combining the original learning object with a student model. The aim is to separate the most important information in the object, potentially breaking it into multiple, more targeted learning objects based on what the student would find most relevant. A second path for future work is to identify learning objects currently missing from an existing ITS which could be deployed for pedagogical benefit.

4 Plan for Research

Our plan for research is to first of all explore in greater depth the problem of presenting annotations of learning objects to new students. The primary challenge is determining which annotations should be presented to each student. A reasonable first step would be to make use of our existing algorithm for content sequencing which proposes specific learning objects to students based on the benefits derived from these learning objects by students at a similar level of knowledge. We could then troubleshoot this algorithm, to see where the requirements for delivering effective annotations to students differ from simply presenting the most appropriate learning objects. One element that would be missing is an interpretation of the value of the annotations, recorded by previous students. As such, experiences of students who were not as similar to the current student would still be of use; likewise, annotations offered by similar students that were considered to be less valuable need not be included. Our aim would be to develop a rich user modeling framework to result in the most effective system for presenting annotations.

Once we have made some progress on the topic of annotations, we would return to our current framework for content sequencing, to determine whether additional improvements could be introduced to this model. In particular, we may have some insights into how to model the similarity between students in a somewhat richer manner.

The third problem of extracting appropriate learning objects from a corpus would be our next focus of study. We expect to receive valuable feedback from the health researchers with whom we have been corresponding, in order to work with a series of motivating examples, towards algorithms for delivering a set of refined learning objects from an existing repository of knowledge. In so doing, we will have made some modest first steps as well for providing an intelligent tutoring system of use as a health informatics application.

A final thread for our research will be to investigate additional methods for verifying the value of our models. One direction will be to develop more effective methods for simulating students. Another will be to consider user studies with real users; we would explore in particular the value of qualitative approaches to experimentation. We expect this research to be an important contribution to the subfield of peer-based intelligent tutoring systems (e.g. [3]). We would be providing a concrete realization of a framework for intelligent tutoring that honours McCalla's ecological approach[1]. We would also offer some contributions to other researchers interested in exploring the use of simulated students as a replacement for real students (e.g. [2]) as a method for verifying the value of an ITS.

References

1. McCalla, G.: The Ecological Approach to the Design of E-Learning Environments: Purpose-based Capture and Use of Information About Learners. Journal of Interactive Media in Education: Special Issue on the Educational Semantic Web 7, 1–23 (2004)
2. VanLehn, K., Ohlsson, S., Nason, R.: Applications of simulated students: An exploration. Journal of Artificial Intelligence in Education 5, 135–175 (1996)
3. Vassileva, J.: Toward social learning environments. IEEE Transactions on Learning Technology 1(4), 199–214 (2008)

On Sketch Based Anonymization That Satisfies Differential Privacy Model

Jennifer Lee[1,2]

[1] School of Information Technology and Engineering (SITE)
University of Ottawa, Ontario, Canada K1N 6N5
[2] School of Computer Science (SCS), Carleton University, Ottawa, K1S 5B6
jennifer_lee@carleton.ca

Abstract. We consider the problem of developing a user-centric toolkit for anonymizing medical data that uses ϵ-differential privacy to measure disclosure risk. Our work will use a randomized algorithm, in particular, the application of sketches to achieve differential privacy. Sketch based randomization is a form of multiplicative perturbation that has been proven to work effectively on sparse, high dimensional data. However, a differential privacy model has yet to be defined in order to work with sketches. The goal is to study whether this approach will yield any improvement over previous results in preserving the privacy of data. How much the anonymized data utility is retained will subsequently be evaluated by the usefulness of the published synthetic data for a number of common statistical learning algorithms.

Keywords: Differential privacy, sketches, output perturbation, non-interactive setting, randomization, data anonymization.

1 Introduction

Inappropriate use of information from sensitive data, such as electronic medical records can cause devastating damage. For example, if a sensitive value occurs frequently together with some quasi-identifier attributes, an individual can infer sensitive information from such attributes even though the exact record of an individual cannot be identified. This attack is known as attribute linkage. Other privacy threats include record linkage, composition (by combining independent releases) [8], and reconstruction attack. Thus, it is necessary for each institution to de-identify the data before releasing it to researchers for data mining purposes.

Record linkages have led to a number of high-profile privacy breaches. This is due to the availability of cheap collection of personal data across different sources, that allows datasets to be joint easily. In the earliest case, Latanya Sweeney [1] discovered that the combination of ZIP code, sex and birth date alone can be used to uniquely identify an individual. The last two cases were the result of the AOL search log release [2], and the release of Netflix movie database to support collaborative filtering research [3]. All three re-identification cases also suggest the difficulty in drawing the line between "public" and "private" information.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 397–400, 2010.

To overcome privacy threats in the context of relational databases, privacy models like k-anonymity [1] and ℓ-diversity [4] were proposed. The granularity of data representation is reduced with the use of techniques such as generalization and suppression. In k-anonymity, the data is transformed such that a given record cannot be distinguished from at least k-1 records in the data. The ℓ-diversity model was defined with the aim of solving the attribute disclosure problem that arises with k-anonymity. It is further enhanced by the t-closeness model [5], which uses the property that the distance between the distribution of the sensitive attribute within an anonymized group and its global distribution is no more than the threshold t.

Attribute values may be very skewed in the real data sets, which makes it very difficult to create feasible ℓ-diverse representations since they treat all values of a given attribute in a similar way irrespective of its distribution in the data. This may enable an adversary to learn about the sensitive values using background knowledge from the global distribution. Furthermore, not all values of an attribute are equally sensitive. The t-closeness approach was intended to address such limitations of the ℓ-diversity model. However, t-closeness tends to be more effective for the case of numeric attributes.

Other methods for privacy preserving data mining include the randomization or perturbation method, which adds noise to the data in order to mask the attribute values of records. Data mining techniques can be developed to derive aggregate distributions from the perturbed records. Unlike k-anonymity and its variance, the perturbation method does not depend on the behavior of the other records. Although this method can be performed at the data collection time, it treats all records equally irrespective of their local density. Therefore, outlier records are more susceptible to adversarial attacks as compared to records in more dense regions in the data.

With a very small probability, randomized algorithms can return anonymized datasets that are totally unrepresentative of the input. Regardless, together with the ϵ-differential privacy measure, we hope to overcome the above mentioned outlier records problem.

2 The Curse of Dimensionality Problem

Many text and market data are inherently high dimensional. Increasing number of attributes imply the distance between any two points in a fixed-size data set is very large. Thus for example, when grouping together data points for k-anonymity using local or global recoding, we will generally be putting very dissimilar points in the same partition, which would require many generalizations to be performed, and a large number of attributes need to be suppressed causing much information loss.

In the method of perturbation, the computation of maximum likelihood estimates for a record matching a database becomes increasingly accurate as the dimensionality increases, resulting in privacy loss. However, these high dimensional datasets have the special property that they are extremely sparse; with only a small percentage of the attributes have non-zero values. Our approach will take advantage of this sparsity, in particular, using a sketch based method to construct anonymized representations of the data.

3 Differential Privacy

Differential privacy [7] is a cryptographically motivated definition of privacy for statistical databases that is not domain specific. It requires that the output distribution of an anonymization procedure not be too sensitive to small changes in the data set. For example, if the database were to be consulted an insurance provider before deciding whether or not to insure a given individual, then the presence or absence of that individual's data in the database will not significantly affect her chance of receiving coverage. This implies that differential privacy does not make arbitrary assumptions about the adversary. It is the first approach to database privacy that provides such a universal guarantee.

The general mechanisms to achieve differential privacy are through uncertainty provided by randomizing. In general, let $\mathcal{A}$ be a randomized algorithm and $\mathcal{S}$ be the set of possible outputs of the algorithm, and let $\epsilon > 1$ (where ϵ is a tunable privacy parameter, smaller ϵ means better privacy). The algorithm $\mathcal{A}$ satisfies ϵ-differential privacy if for all pairs of data sets (D_1, D_2) that differ in exactly one row, and for $\forall S \in \mathcal{S}$:

$$Pr[\mathcal{A}(D_1) = S] \leq exp(\epsilon) \times Pr[\mathcal{A}(D_2) = S]$$

4 Sketch Based Anonymization

Sketches are special case of randomization to add noise to the aggregated data distribution. It is primarily proposed to perturb high-dimensional sparse data [6]. A sketch of the original record $x = (x_1, ..., x_d)$ is defined by an r dimensional vector $s = (s_1, ..., s_r), r \ll d$, where $s_j = \sum_{i=1}^{d} x_i \ \cdot \ r_{ij}$. The random variable r_{ij} is drawn from {-1,+1} with a mean of 0, and is generated from a pseudo-random number generator. Using the BCH error-correcting codes scheme [9], we produced 4-wise independent values for the variable r_{ij}.

The sketch based approach enables approximate calculation of dot product of the orginal data records with their sketches. Attackers can also estimate the original value x_k in the vector x. The precision of which is determined by its variance $\sum_{i=1}^{d} x_i^2 - x_k^2 \ / \ r, \ \ k = 1....d$, dependent only on the non-null attributes in the data. The larger the variance is, the better the reconstruction, however, at the cost of lower level of privacy guarantee.

The methods of sketch based anonymization were previously presented for δ-anonymity algorithm that relies on absolute level of perturbation, and the k-variance anonymity that uses other records to perform the transformation. We propose another method for ϵ-differential privacy.

5 Conclusion

Data are represented as feature vector for our purpose. At this stage, we have just completed the implementation of sketch based data representations. We are now working on the formal definition of the algorithm that connects both sketches and differential privacy. This will be determined by the minimum number of sketches for each record, that is needed to satisfy the ϵ-differential privacy criteria.

The privacy protection achieved with our technique will be measured by how closely the original values of a masked attribute can be estimated. As for the utility of the resulting data anonymization, we consider the Kullback-Leibler (KL) divergence metric to measure information loss on the perturbed data. The original table is treated as a probability distribution p_1 as follows. Let $p_1(t)$ be the fraction of tuples equal to t. The sanitized data are also converted to a probability distribution p_2 . The KL-divergence between the two is $\sum_t \; p_1 \; log \; \frac{p_1(t)}{p_2(t)}$. The larger this number is, the greater the information loss.

The initial experiments were run on the Adult dataset [10] from the UCI machine learning repository. There is a plan to work on real medical datasets from a local hospital. More experiments are required before we can reach conclusive results and compare them with notable previous work, including the work of Aggarwal and Yu [6]. These experiments will be performed over several data mining algorithms such as classification, clustering, and reconstruction.

References

1. Sweeney, L.: *k*-Anonymity: A Model for Protecting Privacy. International Journal on Uncertainty, Fuzziness an Knowledge-based System 10(5), 557–570 (2002)
2. A face is exposed for AOL searcher no. 4417749, http://www.nytimes.com/2006/08/09/technology/09aol.html
3. Narayanan, A., Shmatikov, V.: Robust de-anonymization of large sparse datasets. In: Proceedings of the IEEE Symposium on Security and Privacy, Oakland, California, pp. 111–125 (2008)
4. Machanavajjhala, A., Kiefer, D., Gehrke, J., Venkitasubramanian, M.: l-Diversity: Privacy beyond k-anonymity. In: IEEE International Conference on Data Engineering (2006)
5. Domingo, J.F., Torra, V.: A Critique of *k*-Anonimity and Some of Its Enhancements. In: Proceedings of the 3rd International Conference on Availability, Reliability and Security, Barcelona, Spain, pp. 990–993 (2008)
6. Aggarwal, C.C., Yu, P.S.: On Privacy-Preservation of Text and Sparse Binary Data with Sketches. In: SIAM International Conference on Data Mining (2007)
7. Dwork, C., Smith, A.: Differential Privacy for Statistics: What we Know and What we Want to Learn. In: CDC Data Confidentiality Workshop (2008)
8. Ganta, S.R., Kasiviswanathan, S.P., Smith, A.: Composition Attacks and Auxiliary Information in Data Privacy. In: Proceeding of the 14th ACM SIGKDD International Conference, Las Vegas, Nevada, pp. 265–273 (2008)
9. Rusu, F., Dobra, A.: Pseudo-Random Number Generation for Sketch-Based Estimations. ACM Transactions on Database Systems 32(2) (2007)
10. UCI Machine Learning Repository: Adult Data Set, http://archive.ics.uci.edu/ml/datasets/Adult

Predicting Optimal Constraint Satisfaction Methods

Craig D.S. Thompson

Department of Computer Science, 176 Thorvaldson Bldg. University of Saskatchewan
110 Science Place, Saskatoon, SK. S7N 5C9 Canada
cdt830@mail.usask.ca

Abstract. Given the breadth of constraint satisfaction problems (CSP) and the wide variety of CSP solvers, it is often very difficult to determine *a priori* which solving method is best suited to a problem. This work explores the use of machine learning to predict which solving method will be most effective for a given problem.

1 Introduction

Constraint satisfaction problems (CSP) represent a diverse range of problems in the planning, scheduling, circuit design, and software verification domains. It is difficult to determine *a priori* which algorithm will work best for a given problem and therefore it is usually left to an expert to decide which algorithm to apply, with poor choices resulting in wasted resources by applying an inefficient solver. Many problems require specialized solvers to take advantage of their structure. However, designing a problem-specific solver becomes increasingly difficult as increasingly complex problems are considered. To address this problem, we use machine learning to predict which of the many existing CSP solving methods will be most efficient for solving a given CSP. In this paper we present our technique and preliminary findings, as well as our active directions for further exploration.

Given a set of CSP solvers S, an oracle, S_{oracle} will always apply the optimal solver in S for a given problem. There exists a solver, S_{opt}, in S whose average performance over a range of problems is better than the average performance of any other solver in S. We know that there are specific problem types that favour one solver over another. Therefore, there is no single solver that will dominate over the entire range of CSP problems and we can conclude that $S_{oracle} \neq S_{opt}$. When attempting to predict which solver will be most effective, we aim to approximate S_{oracle}, and performance equal to S_{opt} can be considered a baseline for usability. This research investigates whether machine learning can be used to assess a CSP problem and, based on past experience, select a solving method that is believed to be effective. We collected a database of CSP solving experience by using many CSP solvers to attempt a wide range of problems and documented both the attributes of the problem and performance statistics about the solver. We then applied a learning algorithm to the data in order to predict which solving method would perform best on a given problem. Related works on learning to

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 401–404, 2010.

solve CSPs have studied the use of reinforcement learning to learn variable and value orderings [7] and propagation policies [3] for problems known to be drawn from the same class as the training problems. Also, positive results have been found by selecting solver algorithms for SAT based on problem attributes [9].

2 Preliminary Experimentation

For brevity, this paper will only consider two CSP solvers: forward checking, with dom+deg variable ordering and mc value ordering, and arc consistency, with preprocessing, dom+deg variable ordering and mc value ordering, which we will denote as fc and ac, respectively. Other solving techniques we are currently pursuing consist of all combinations of: two propagation techniques (forward checking and arc consistency), optional preprocessing, four variable selection heuristics (dom [6], deg [1], dom+deg [4], and lexicographical ordering), and two value ordering heuristics (mc [4] and lexicographical ordering) for a total selection of 32 solvers.

The two solvers described were applied to problems from $25 \leq n \leq 125$, $3 \leq m \leq 50$, $0 \leq p1$, and $0 \leq p2 \leq 1$, where n is the number of variables, m is the domain size of each variable, $p1$ is the constraint density and $p2$ is the constraint tightness. We have constrained the problems to the "hard" region, with κ [5] between 0.5 and 1.5. Each solver was applied to each problem 10 times, and the fastest time was recorded. As none of the solvers we considered had randomized attributes, the fastest time has the least noise from other active processes on the computer. The solving time allowed to any algorithm was limited to 2 seconds. Any problem that was not solved by any solver within 2 seconds was deemed "hard" and rejected, as we cannot learn which solving method is effective if all solvers reach the time limit without finding a solution. In truth, it is the "hard" problems that we are most interested in, however, for the purposes of gathering a sufficient number of data points we limited the solving time. Any problem solved by all solvers in less than 5 ms was also removed, as the choice of solver was not considered important when the time to solve using any solver is this small.

Table 1. Comparing fc,ac, and S_{oracle}

	avg time	unsolved	fastest	avg time (fastest)
fc	345 ms	88	603/917	119 ms
ac	182 ms	17	314/917	163 ms
S_{oracle}	134 ms	0	917/917	

In table 1, we present the average time taken per problem and the number of unsolved problems. The average time is a low estimate, as unsolved problems are treated as being "solved" in the maximum allowed time of 2 seconds. We also present the fraction of problems that are solved fastest by each solver, and the average solving times on these problems. We found that from the set consisting of fc and ac, S_{opt} was ac, and S_{oracle} was significantly better than either solver.

We used a C4.5 [8] decision tree to classify problems as being either solved faster using fc or ac using the attributes n,m,$p1$,$p2$, and κ as inputs. We also trained a cost sensitive C4.5 decision tree by reweighting the instances of the training set according to their misclassification cost [2]. The misclassification costs, and misclassification rates of the two solvers using 10-fold cross-validation are presented in table 2.

Table 2. Misclassification costs and rates

	misclassification cost	misclassification rate: C4.5	misclassification rate: cost sensitive C4.5
average	260 ms	95/917	149/917
fc	73 ms	60/603	133/603
ac	619 ms	35/314	16/314
total cost		26045	19613

$$\begin{aligned} E = {} & P(ac) * pred(ac) * E[ac] + P(ac) * (1 - pred(ac)) * (E[ac] + M[ac]) \\ & + P(fc) * pred(fc) * E[fc] + P(fc) * (1 - pred(fc)) * (E[fc] + M[fc]) \end{aligned} \quad (1)$$

We can estimate the expected value of a solver that consults our decision tree before choosing a CSP solver by using equation 1 and the misclassification costs and rates in table 2. $P(ac)$ is the probability that a given problem is best solved with arc consistency, in our case 314/917. $P(fc) = 1 - P(ac)$. $E[ac]$ and $E[fc]$ are the expected times to solve a problem suited to ac using ac, and the time to solve a problem suited to fc, respectively. $M[ac]$ and $M[fc]$ are the misclassification costs; that is, the penalty incurred by applying fc when ac is preferred, or when applying ac when fc is preferred, respectively. Finally, $pred(ac)$ and $pred(fc)$ represent the accuracy with which we can correctly identify ac instances, and fc instances. The expected cost of using fc,ac, or S_{oracle} according to equation 1 is consistent with our empirical findings in table 1. Additionally, the expected costs of consulting the C4.5 decision tree and the cost sensitive C4.5 decision tree are 162 ms and 155 ms respectively. The expected costs of both our learned models fall between S_{opt} and S_{oracle}, making them both candidates for possible use. However, we are continuing to work on determining the practicality of building a decision tree, and calculating the problem attributes used for classification, as these steps are a computational overhead not encountered by ac or fc.

3 Conclusion

We will continue to expand S, the set of candidate solvers to be considered, by implementing a variety of backtracking techniques, adding additional variable and value heuristics, and additional levels of consistency. In addition, we will extend our approach to other parameterized CSPs with non-random constraints and non-random constraint graphs, as well as CSPs drawn from the real world.

Lastly, in order to improve the prediction accuracy of the learned model we will record more attributes of each CSP, which will provide a richer set of data to analyze for patterns.

In this paper we have presented a classifier that chooses between two classes; however, in practice, we may have hundreds, or thousands, of parameterizations. Thus, another aim of this work is to find more effective learning methods for addressing this general classification problem. Also, misclassification costs should be considered on a per instance basis, as for some instances a misclassification may be more or less significant, and as such our estimated quality of a learning solver may not be accurate. We will continue to pursue learning methods with per-instance misclassification costs, rather than per-class costs. Finally, looking beyond approximating S_{oracle}, we expect that there may be sub-problems within a CSP that are better suited to one solving method over another; as such, we are interested in changing solvers mid-problem, which may enable us to produce a solver better than S_{oracle}.

Acknowledgements. The author would like to acknowledge the support of his supervisor, Dr. Michael Horsch, who is supported by NSERC.

References

1. Dechter, R., Meiri, I.: Experimental evaluation of preprocessing techniques in constraint satisfaction problems. In: Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, vol. 1, pp. 271–277 (1989)
2. Elkan, C.: The foundations of cost-sensitive learning. In: International Joint Conference on Artificial Intelligence, vol. 17, pp. 973–978 (2001)
3. Epstein, S., Wallace, R., Freuder, E., Li, X.: Learning Propagation Policies. In: Proceedings of the Second International Workshop on Constraint Propagation And Implementation, pp. 1–15 (2005)
4. Frost, D., Dechter, R.: Look-ahead value ordering for constraint satisfaction problems. In: International Joint Conference on Artificial Intelligence, vol. 14, pp. 572–578 (1995)
5. Gent, I., MacIntyre, E., Prosser, P., Walsh, T.: The constrainedness of search. In: Proceedings of the National Conference on Artificial Intelligence, pp. 246–252 (1996)
6. Haralick, R., Elliott, G.: Increasing tree search efficiency for constraint satisfaction problems. Artificial intelligence 14(3), 263–313 (1980)
7. Petrovic, S., Epstein, S.: Random Subsets Support Learning a Mixture of Heuristics. International Journal on Artificial Intelligence Tools 17(3), 501–520 (2008)
8. Quinlan, J.: C4. 5: programs for machine learning. Morgan Kaufmann, San Francisco (1993)
9. Xu, L., Hutter, F., Hoos, H., Leyton-Brown, K.: SATzilla: portfolio-based algorithm selection for SAT. Journal of Artificial Intelligence Research 32(1), 565–606 (2008)

A Computer Approach for Face Aging Problems

Khoa Luu

Centre for Pattern Recognition and Machine Intelligence,
Department of Computer Science and Software Engineering, Concordia University, Canada
kh_lu@cenparmi.concordia.ca

Abstract. This paper first presents a novel age-estimation approach combining Active Appearance Models (AAMs) and Support Vector Regression (SVR) to yield the highest accuracy of age recognition rates of all comparable published results both in overall Mean Absolute Error (MAE) and Mean Absolute Error per decade of life (MAEd). The combination of AAMs and AVR is used again for a newly proposed face age-progression method. The familial information of siblings is also collected so that the system can predict the future faces of an individual based on parental and sibling facial traits. Especially, a new longitudinal familial face database is presented. Compared to other databases, this database is unique in that it contains family-based longitudinal images. It contains not only frontal faces but also the corresponding profiles. It has the largest number of pre-adult face images per subject on average.

Keywords: Face aging, age-estimation, age-progression, active appearance models, support vector regression.

1 Introduction

Face aging can find its origins from missing children when police require age progressed pictures. This problem is also applicable in cases of wanted fugitives where face age-progression is also required. The predominant approach to aging pictures involves the use of forensic artists. Although forensic artists are trained in the anatomy and geometry of faces, their approach is still more art than science. In addition, an age-progressed image can differ significantly from one forensic artist to the next. Manual age progression usually takes lots of time and requires the work of numerous clever forensic artists. Moreover, these forensic artists are not inspired and enduring enough to give good results when they work under pressure. Therefore, automatic and computerized age-progression systems are important. Their applications range from very sensitive national security problems to tobacco or alcohol stores/bars to control the patron's age and cosmetic studies against aging.

In general, facial aging technologies address two areas: *facial age-estimation* and *face age-progression*. Facial age-estimation problem is to build computer software having ability to recognize ages of individuals in input photos. Face age-progression has a more powerful ability to predict future faces of individual in input photos.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 405–409, 2010.

Research in age-estimation is a relatively young field compared to other biometrics research. Most important works appeared only recently. The methods are divided into two categories: *local* and *holistic* approaches. Yan et al. [1] had the best experimental MAEs (Mean Absolute Age Errors) results before our new method [2] was proposed. They reached 4.95 on FG-NET database [6], 4.94 for males and 4.38 for females on the YAMAHA database. Meanwhile, face age-progression can be classified into two approaches. The first is based on face anthropometry theories to model facial aging [7]. The other uses computer vision methods to interpret face images and learn the aging rules using facial aging databases. The latter is further broken down into three sub-categories: grammatical face model [8], 3D face aging with Face Recognition applications [9] and age-progression tuned model [6].

Physical age-progression can be divided into two categories: *growth and development* (childhood to pre-adulthood), which primarily affects the physical structure of the craniofacial complex in the form of lengthening and widening of bony-tissue; and *adult aging* that spans from the end of pre-adulthood to senescence, where the primary impacts occur in the soft-tissue, in the formation of wrinkles, lines, ptosis, and soft tissue morphology. Since the two aging periods have fundamentally different aging mechanisms, there are two specific "aging functions" to be constructed: *the growth and development function* and *the adult aging function*.

2 Work Completed

Here, both problems are studied in parallel. In age-estimation, we introduce a novel technique [3] that combines AAMs and SVR, which dramatically improves the accuracy of age-estimation over the current state-of-the-art techniques [1, 6]. The combination of AAMs and AVR is used again for a newly proposed face age-progression method [4]. Familial information of siblings is also collected so that the system can predict the future faces based on parental and sibling facial traits.

2.1 The Novel Age Estimation Approach

In the training steps, AAMs feature vectors x are extracted from faces I and work as the inputs of the age classification module. There are two main steps in the classification module. First, a classifier $f(x)$ is built from SVMs to distinguish between youths (aging from 0 to 20) and adults (aging from 21 to 69). Then, the two aging functions, the *growth and development function* $f_1(x)$ and the *adult aging function* $f_2(x)$, are constructed from SVR.

In the recognition steps, an AAM feature vector is first extracted from a given input face and this vector works as the input for the youth/adult classifier. Given an output from that youth/adult classifier, an appropriate aging function is used to determine the age of the face. Compared to published results, this method yields the highest accuracy recognition rates both in MAEd (Table 1) and the overall MAE (Table 2). We use 802 faces with ages ranging from 0 to 69 from the FG-NET database to train the system. In the testing phase, we use a hold out set of 200 faces (not included in the training set). The overall MAE of our method is 4.37 years.

Table 1. Comparison of MAEds between RPK and our proposed method on FG-NET database [2]

Age Ranges	RPK [1]	Our method
0 – 9	2.30	1.39
10 – 19	4.86	2.10
20 – 29	4.02	3.00
30 – 39	7.32	2.45
40 – 49	15.24	2.40
50 – 59	22.20	2.00
60 – 69	33.15	5.33
0 – 69	4.95	4.37

Table 2. Comparison of estimation results on FG-NET database with ages from 0 to 69 [2, 3]

Methods	MAEs
Our method	4.37
RFK [1]	4.95
AGES	6.77
BM	5.33
WAS [6]	8.06

2.2 Face Progression Using Familial Information

In age-progression, we present a novel face aging method [4] by also combining AAMs and SVR to interpret faces and learn aging rules. The heredity information of faces of siblings is then used as the input to predict the future faces of an individual. This study is done effectively with the support of the familial longitudinal face database [5]. Comparing to existing work, this method gives the acceptable future faces of individuals with supplementary heritability information. Fig. 1 shows an example of generated results[1]. The images in first and third rows are of *A* and of a sibling of *A* corresponding at 8, 10, 12, 14 and 16 years of age. Given only an image of *A* at 8 years old (the first image in the first row), her future faces at ages of 10, 12, 14 and 16 are predicted. With input familial images of the sibling of *A* at ages when we need to predict, the simulated faces are generated as shown in the second row.

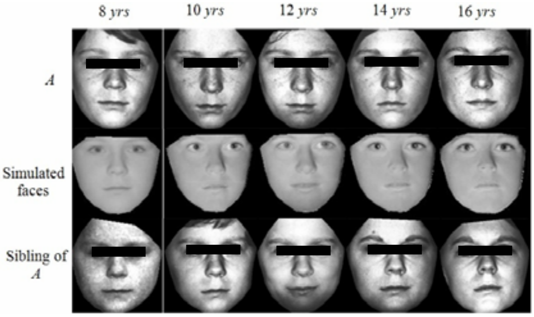


Fig. 1. An example of predicted faces [4]

2.3 The Familial Aging Face Database

We developed the novel longitudinal face database collected to study the problems associated with age-progression during growth and development. It is very difficult to collect this kind of databases because of chronometrical image series of an individual. As we know, there are only two public face aging databases at present: FG-NET [6] and MORPH [10]. Compared to other databases, ours is unique in that it contains family-based longitudinal images. Additionally, it has the largest number of pre-adult face images per subject (7.25). Statistical details of the database can be found in [5].

[1] The face images are masked as the requests in the contract between CENPARMI and Burlington Growth Centre, Faculty of Dentistry, University of Toronto, Canada.

3 Future Work

Based on our accomplished results, we continue to develop new methods to improve the quality of synthesis faces as well as the accuracy of age-estimation. Particularly, we are focusing on the following aspects:

- Integrated With Anthropometry Knowledge: there are many valuable anthropometry studies related to human face aging that can be applied to Computer Vision. We will continue to conduct further studies on extracting facial heredity features from family members. Furthermore, this knowledge could be integrated into a Face Aging Expert System.
- Texture Modification and 3-D Face Modeling: we propose to use 3-D Morphable Models instead of 2D AAM's to model future faces and age-progression. Additionally, we will experiment our proposed methods on larger databases with more than 20,000 photos instead of experimenting on the 1002 photos of FG-NET, and improve the results with smaller MAE values, by updating the configuration of learning parameters.

Acknowledgments. This work is being studied under the supervision of Prof. Tien D. Bui and Prof. Ching Y. Suen at CENPARMI. I also give my thanks to Prof. Karl Ricanek and Face Aging Group at Univesity of North Carolina at Wilmington, U.S. for their strong supports. This study was made possible by use of material from the Burlington Growth Centre, Faculty of Dentistry, University of Toronto, which was supported by funds provided by Grant (1) (No. 605-7-299) National Health Grant (Canada), (data collection); (2) Province of Ontario Grant PR 33 (duplicating) and (3) the Varsity Fund (for housing and collection). This work was also funded by the Natural Sciences and Engineering Research Council of Canada (NSERC), and the Project 322, Ministry of Education and Training, Vietnam Government.

References

1. Yan, S., Zhou, X., Liu, M., Johnson, M., Huang, T.: Regression from Patch-Kernel. In: ICPR 2008 (2008)
2. Luu, K.: A Computer Vision Approach for Face Aging Problems, Master Thesis of Computer Science, CENPARMI, Concordia University, Montreal, Canada (2009)
3. Luu, K., Bui, T.D., Ricanek Jr., K., Suen, C.Y.: Age Estimation using Active Appearance Models and Support Vector Machine Regression. BTAS Washington, DC, U.S. (September 2009)
4. Luu, K., Suen, C.Y., Bui, T.D., Ricanek Jr., K.: Automatic Child-Face Age-Progression Based on Heritability Factors of Familial Faces. BiDS, Florida, U.S. (September 2009)
5. Luu, K., Ricanek, K., Bui, T.D., Suen, C.Y.: The Familial Face Database: A Longitudinal Study of Family-based Growth and Development on Face Recognition. In: ROBUST 2008 (2008)
6. Lanitis, A., Taylor, C., Cootes, T.: Modeling the process of ageing in face images. In: ICCV 1999 (1999)
7. Pitanguy, I., Pamplona, D., Weber, H., Leta, F., Salgado, F., Radwanski, H.: Numerical modeling of facial aging. Plastic and Reconstructive Surgery 102, 200–204 (1998)

8. Suo, J., Min, F., Zhu, S., Shan, S.G., Chen, X.: A Multi-Resolution Dynamic Model for Face Aging Simulation. In: CVPR 2007 (2007)
9. Scherbaum, K., Sunkel, M., Seidel, H.P., Blanz, V.: Prediction of Individual Non-Linear Aging Trajectories of Faces. In: EUROGRAPHICS 2007 (2007)
10. Ricanek, K., Tesafaye, T.: MORPH: A longitudinal Image Database of Normal Adult Age-Progression. In: FG 2006 (2006)

Automatically Expanding the Lexicon of *Roget's Thesaurus*

Alistair Kennedy

School of Information Technology and Engineering
University of Ottawa
Ottawa, Ontario, Canada
akennedy@site.uottawa.ca

Abstract. In recent years much research has been conducted on building Thesauri and enhancing them with new terms and relationships. I propose to build and evaluate a system for automatically updating the lexicon of *Roget's Thesaurus*. *Roget's* has been shown to lend itself well to many Natural Language Processing tasks. One of the factors limiting *Roget's* use is that the only publicly available version of *Roget's* is from 1911 and is sorely in need of an updated lexicon.

1 Introduction

Thesauri are valued resources for the Natural Language Processing (NLP) community, and have played a role in many applications including building lexical chains and text summarization. *WordNet* [1] has become the default thesaurus that NLP researchers turns to. It is important for NLP researchers to remember that *WordNet* represents just one of many ways of organizing the English lexicon and is not necessarily the best system available for a given NLP task. The 1911 version of *Roget's Thesaurus* (available through Project Gutenberg[1]) was recently released in a Java package called Open *Roget's Thesaurus*[2]. The goal of my thesis is to create an accurate system for automatically updating *Roget's Thesaurus* with new words.

Roget's is a hierarchical thesaurus consisting of nine levels from top to bottom: *Class* → *Section*→ *Subsection*→ *Head Group*→ *Head*→ *Part of Speech (POS)*→ *Paragraph*→ *Semicolon Group (SG)*→ *Words*. Words always appear in the lowest, 9th level, of the hierarchy. I will denote the set of words contained within one of these levels as a *Roget's-grouping*. SGs contain the closest thing to synonyms, though their grouping tends to be looser than synsets in *WordNet*.

2 The Methodology

This project is planned in three stages. The first is to identify pairs of closely related words using corpus based measures of semantic relatedness, such as Lin [2].

[1] www.gutenberg.org/ebooks/22

[2] rogets.site.uottawa.ca/

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 410–411, 2010.

Using a variety of these measures as features for a Machine Learning classifier I will determine which pairs of words are likely to appear in the same *Roget's-grouping* (specifically the same POS, Paragraph or SG).

The second step is to use these pairs of related words to determine the correct location in the *Thesaurus* to place a new word. Probabilities that pairs of words belong in the same *Roget's-grouping* can be used to determine the probability that a new word should be placed into a particular *Roget's-grouping*.

The last step, evaluation, can be done both manually and automatically. For manual evaluation an annotator could be given a *Roget's-grouping* and asked to identify the new words. If humans have difficulty in identifying which words are new additions then I can deem the additions to be as good as human additions. For automatic evaluation there are a number of applications that can be used to compare the original and updated *Roget's*. These tasks include measuring semantic distance between word or sentences [3] and ranking sentences as a component of a text summarization application [4].

3 Progress so Far

At this stage I have implemented a prototype system, of the first two steps, to place words into a *Roget's-grouping*. I performed evaluation of this prototype by removing a set of words from *Roget's* and attempted to place the words back into the *Thesaurus*. The early results show a relatively good precision for adding new terms at the Paragraph level, however the results are lower at the SG level. I hope to improve these results by experimenting with more semantic distance measures and Machine Learning classifiers. The above mentioned applications for automatic evaluation of *Roget's* have already been implemented [3,4]. As I have not yet produced an updated version of *Roget's* no manual or automatic evaluation has yet been carried out.

Acknowledgments

My research is supported by the Natural Sciences and Engineering Research Council of Canada and the University of Ottawa. I would also like to thank my thesis supervisor Dr. Stan Szpakowicz for his guidance.

References

1. Fellbaum, C. (ed.): WordNet: an electronic lexical database. MIT Press, Cambridge (1998)
2. Lin, D.: Automatic retrieval and clustering of similar words. In: Proceedings of the 17th international conference on Computational linguistics, Morristown, NJ, USA, pp. 768–774. Association for Computational Linguistics (1998)
3. Kennedy, A., Szpakowicz, S.: Evaluating Roget's thesauri. In: Proceedings of ACL 2008: HLT, pp. 416–424. Association for Computational Linguistics (2008)
4. Copeck, T., Kennedy, A., Scaiano, M., Inkpen, D., Szpakowicz, S.: Summarizing with Roget's and with FrameNet. In: The Second Text Analysis Conference (2009)

Privacy Preserving Publication of Longitudinal Health Data*

Morvarid Sehatkar[1,2]

[1] School of Information Technology and Engineering,
University of Ottawa
Ottawa Ontario Canada, K1N 6N5
msehatkar@uottawa.ca
[2] CHEO Research Institute,
Ottawa Ontario Canada, K1H 8L1

With the development of Electronic Medical Records (EMRs) and Electronic Health Records(EHRs) huge amounts of Personal Health Information (PHI) are now being available and consequently demands for accessing and secondary use of such PHI[1] are increasing. Despite its benefits, the use of PHI for secondary purposes has increased privacy concerns among public due to potential privacy risks arising from improper release and usage of person-specific health data [1]. To address these concerns, governments and ethics boards regulated a set of privacy policies for disclosing (identifiable) personal health data which requires that either consent of patients to be obtained or data to be de-identified before publication[2]. However, as obtaining consent is often not practical in secondary use contexts, data de-identification becomes a better –and sometimes the only– practical approach.

Data de-identification generally resides in the context of privacy preserving data publishing(PPDP) [3], but there are some specific requirements[4] for de-identifying health data which makes it more challenging and most of the available approaches in PPDP will not be practical for use with health data. Moreover, health data, particularly the clinical data found in EMRs, are often longitudinal by nature and, to the best of our knowledge, no methods have been proposed so far for de-identification of such longitudinal clinical data. In this work, we are developing methods for publishing longitudinal clinical health data so that the disclosed data remain practically useful in secondary use contexts while the privacy of individuals are preserved. We are particularly interested in privacy models which thwart *attribute disclosure attacks* [5].

Our motivating scenario is sharing the longitudinal microdata from a hospital containing information of multiple patients' visits over a period of time. Each patient has a set of basic quasi-identifiers (QIs) which are independent of visits such as *date of birth* and *gender* as well as a set of visits each of which consists some visit-dependent QIs, e.g. *visit date* and *postal code*, and a sensitive attribute, *diagnosis*. It can be shown that representing such data either as a table with

* This work is part of a research project at the Electronic Health Information Lab, Children's Hospital of Eastern Ontario Research Institute, Ottawa, Ontario, Canada.

[1] Secondary use of PHI refers to the application of health data for the purposes other than providing care to the patient such as research, public health, and marketing.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 412–413, 2010.

multiple records for each patient corresponding to her multiple visits or in the form of a transactional dataset will not work and current techniques [3,5] will fail to effectively de-identify such longitudinal dataset.

In this work we introduce a new technique to effectively anonymize this longitudinal data. We propose to represent such longitudinal data in three levels. Level 1 contains basic QIs of patients, level 2 represents the set of corresponding visits for each patient and in level 3 we insert the values of sensitive attribute within each visit, i.e. *diagnosis*. Level 2 can be represented as transactional data in which each unique visit will be an item. However, each item (visit) will have several quasi identifiers. Also, we assume that the adversary's knowledge about a patient's visits is limited. In other words, we assume that, besides all QIs in level 1, the adversary has the knowledge of at most p visits of a patient. This is a realistic assumption, since in real life it is infeasible, or at least too difficult, that an adversary to be able to obtain all information of a patient's visits and it is more likely that he has partial background knowledge. Since we assume that the adversary knows all QI's in level 1, we will work in one equivalence class of level 1 at a time. Having this dataset, the adversary can violate the privacy of patients through an attribute disclosure attack, if the values of sensitive attribute within at least one of the possible combinations of p visits do not have adequate diversity. For example, if the adversary knows about 3 visits and in one of the combinations of 3 visits, all patients are diagnosed to have HIV, the adversary can infer sensitive value HIV with high confidence and, therefore, the privacy of all patients who belong to that group will be violated. Our goal is to anonymize the data in a way that for every combination of p visits, no attribute disclosure can occur. We evaluate the performance of our approach with respect to the time it takes to anonymize the data and the information loss. One possible measure of information loss can be the extent of suppression since practical end-users care about missingness and the amount of suppression is an important indicator of data quality for them. We also use the non-uniform entropy information loss metric suggested in [4].

References

1. Kulynych, J., Korn, D.: The effect of the new federal medical privacy rule on research. The New England Journal of Medicine 346(3), 201–204 (2002)
2. Willison, D., Emerson, C., et al.: Access to medical records for research purposes: Varying perceptions across research ethics boards. Journal of Medical Ethics 34, 308–314 (2008)
3. Fung, B.C.M., Wang, K., Chen, R., Yu, P.S.: Privacy-preserving data publishing: A survey on recent developments. ACM Computing Surveys 42(4) (December 2010) (impact factor 9.92 (2009))
4. El Emam, K., Dankar, F.K., et al.: A globally optimal k-anonymity method for the de-identification of health data. JAMIA 16, 670–682 (2009)
5. Machanavajjhala, A., Kifer, D., Gehrke, J., Venkitasubramaniam, M.: l-diversity: Privacy beyond k-anonymity. ACM Transactions on Knowledge Discovery from Data 1(1) (2007)

Reasoning about Interaction for Hospital Decision Making

Hyunggu Jung

Cheriton School of Computer Science, University of Waterloo
h3jung@uwaterloo.ca

1 The Problem, Plan of Research and Current Work

For our research, we present a model for reasoning about interaction in dynamic, time critical scenarios (such as decision making for hospital emergency room scenarios). In particular, we are concerned with how to incorporate a model of the possible bother generated when asking a user, to weigh this in as a cost of interaction, compared to the benefit derived from asking the user with the highest expected quality of decision. A detailed method for modeling user bother is presented in Cheng [1], which includes reasoning about interaction (partial transfers of control or PTOCs) as well as about full transfers of control of the decision making (FTOCs) to another entity. Distinct from Cheng's original model, attempts at FTOCs are in framed as PTOCs with the question Q: "Can you take over the decision making?". This then enables either a "yes" response, which results in an FTOC[1] or a "no" response or silence.

In Figure 1, one *world* consists of one PTOC, one FTOC, and one SG (strategy regeneration) node and includes all the parameters currently used to calculate benefits and costs to reason about interaction with entities. Therefore, when the current *world* is moved to the next step, our system asks a new entity. The number of worlds is equivalent to the number of entities that will be asked.

There are n FTOC nodes, n PTOC nodes, n SG nodes, and one virtual node in the overall framework with n worlds. We obtain the overall EU of strategy s by summing up n EU values for FTOC nodes, n EU values for SG nodes and one EU value for the virtual node as follows:

$$EU(s) = EU_n(dfl) + \sum_{j=1}^{n}(EU_j(fn_l) + EU_j(sg)) \tag{1}$$

where dfl[2] reflects a virtual node, n denotes the number of worlds, $EU(fn_l)$ reflects the utility of ending in a FTOC, and $EU(sg)$ reflects the utility of ending in SG node.

We are also interested in determining an appropriate reasoning strategy to find the right person, at the right time, to assist with the care of patients who are arriving at a hospital emergency room. Typically in these settings, patients who

[1] As a simplication, we assume that a "Yes" response results in the user successfully assuming control of the decision.

[2] The leaf node for the silence response is set to sg.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 414–415, 2010.

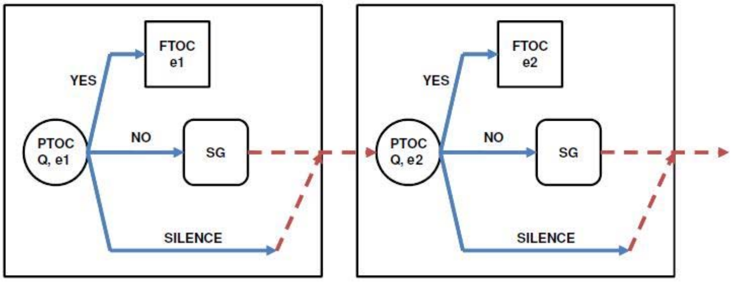


Fig. 1. Visual representation of strategy with the FTOCs and PTOCs; each world occupies one square

appear to require further assistance than can be immediately provided (what we could call "a decision") require soliciting aid form a particular specialist.

In order for the human first clinical assistants (FCAs) to make the best decisions about which specialists to bring in to assist the patients that are arriving, the proposal is to have our multiagent reasoning system running in the background, operating with current parameter values to suggest to the medical professionals who exactly they should contact to assist the current patient. These experts are then the entities $\{e_1,\ e_2, \ldots,\ e_n\}$ that are considered in our reasoning about interaction, with the FCA contacting the experts according to the optimal strategy our model generates (who first, waiting for how long, before contacting who next, etc.)

We model the cost of bothering users in detail, as in Cheng [1]. We propose the addition of one new parameter as part of the user modeling for the bother cost, a $Lack\_of\_ExpertiseFactor$. This parameter is used to help to record the general level of expertise of each doctor (i.e. medical specialist), with respect to the kind of medical problem that the patient is exhibiting. We introduce another new parameter, *task criticality (TC)*, to affect the reasoning about interaction. TC is used to enable the expected quality of a decision to be weighted more heavily in the overall calculation of expected utility (compared to bother cost), when the case at hand is very critical. This parameter may also be adjusted, dynamically. When a patient has high task criticality, strong expertise is required because the patient's problem may become much more serious if not treated intensively.

Our detailed bother modeling for time critical environments is an advance on other bother models [2]. We have validated our approach in comparison with the case where bother is not modeled, simulating hospital emergency decision making. Our current results demonstrate valuable improvements with our model.

References

1. Cheng, M., Cohen, R.: A hybrid transfer of control model for adjustable autonomy multiagent systems. In: Proceedings of AAMAS 2005 (2005)
2. Horvitz, E., Apacible, J.: Learning and reasoning about interruption. In: Proceedings of the 5th International Conference on Multimodal Interfaces (ICMI 2003), pp. 20–27 (2003)

Semantic Annotation Using Ontology and Bayesian Networks

Quratulain Rajput

Faculty of Computer Science, Institute of Business Administration,
Karachi, Pakistan
qrajput@iba.edu.pk

Abstract. The research presents a semantic annotation framework, named BNOSA. The framework uses ontology to conceptualize a problem domain and uses Bayesian networks to resolve conflicts and to predict missing data. Experiments have been conducted to analyze the performance of the presented semantic annotation framework on different problem domains. The sets of corpuses used in the experiment belong to selling-purchasing websites where product information is entered by ordinary web users.

Keywords: Ontology, Bayesian Network, Semantic Annotation.

1 Introduction

A large amount of useful information over the web is available in unstructured or semi-structured format. This includes reports, scientific papers, reviews, product advertisements, news, emails, Wikipedia, etc. Among this class of information sources, a significant percentage contains ungrammatical and incoherent contents where information is presented as a collection of words without following any grammatical rules. Several efforts have been made to extract relevant information from such contents [1-4].

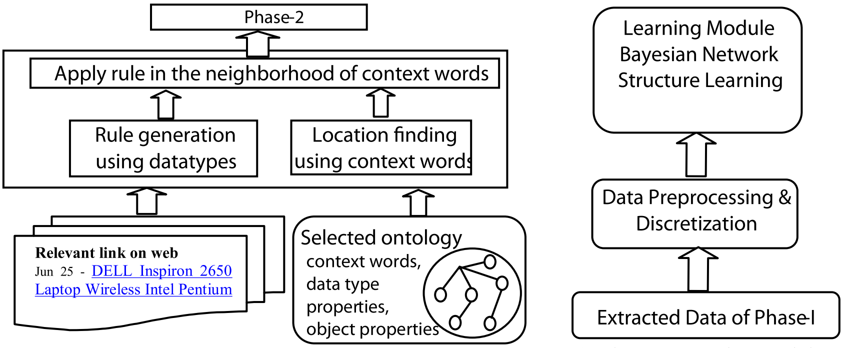
2 BNOSA Framework and Results

The proposed BNOSA (Bayesian Network and Ontology based Semantic Annotation) framework is capable of dynamically linking a domain-specific ontology and the corresponding BN (learnt separately) to annotate information extracted from the web. This dynamic linking capability makes it highly scalable and applicable to any problem domain. The extraction of data is performed in two phases which is also depicted in Fig. 1.

Phase-I: To extract the information two issues need to be addressed: (a) finding the location of relevant data on a web page and (b) defining patterns for extracting such data. To solve the location problem, lists of context words are defined for each attribute of the extraction ontology. If a match is found, this suggests that the corresponding value should also be available in the neighborhood of this word. The rules are generated on the basis on the attributes' data-types.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 416–418, 2010.

Phase-II: To extract information from unstructured and incoherent data sources, one has to deal with variable size of information available at different website within a single domain. In some cases, context words are same for more than one attributes and the situation becomes more complicated when the relevant context words are not available in the text. BNOSA applies probabilistic reasoning techniques, commonly known as Bayesian Networks, to address these problems.


Phase-2
Apply rule in the neighborhood of context words
Rule generation using datatypes
Location finding using context words
Relevant link on web
Jun 25 - DELL Inspiron 2650 Laptop Wireless Intel Pentium
Selected ontology
context words, data type properties, object properties
Learning Module Bayesian Network Structure Learning
Data Preprocessing & Discretization
Extracted Data of Phase-I

(a) Phase I - Information extraction using ontology **(b)** Phase-II Learning module

Fig. 1. Graphical representation of BNOSA Framework

To test the performance of the BNOSA framework, three case studies based on the selling/purchasing ads of used cars, laptops and cell phones were selected. Table 1 presents the precision and recall values at the end of Phase-I. The prediction accuracy of the missing values as a result of Phase-II is shown in Fig. 2.

Table 1. Performance of BNOSA using extraction ontology after Phase-I

	Laptop Ads						Cell Phone Ads				Car Ads						
	Hard Disk	Display Size	Price	Brand Name	Speed	Ram	Memory	Color	Price	Model	Mileage	Transmission	Color	Make	Model	Price	Year
Precision%v	100	100	100	100	97	100	98.1	100	100	100	23.4	100	61.1	91.2	100	98	94
Recall% v	98	87	100	87	94	97.8	85.2	97.8	100	96.2	21.7	98.4	66.7	52.5	37.9	98	94
Precision%m	92	82	100	45	89	100	68	97	100	100	73.9	97.5	100	2.33	7.81	100	100
Recall%m	100	100	100	100	89	100	94.4	100	100	100	100	100	82.4	100	100	100	100

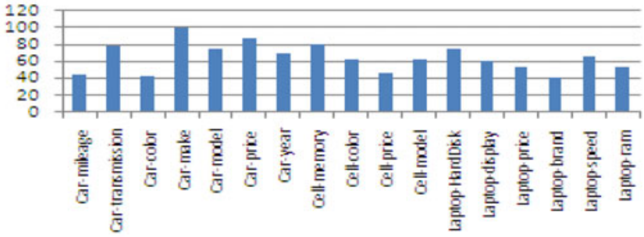


Fig. 2. Phase-II prediction results of attributes in three domains

References

1. Michelson, M., Knoblock, C.A.: Semantic annotation of unstructured and ungrammatical text. In: Proceedings of 19th International Joint Conference on Artificial Intelligence, pp. 1091–1098 (2005)
2. Ding, Y., Embley, D., Liddle, S.: Automatic Creation and Simplified Querying of Semantic Web Content: An Approach Based on Information-Extraction Ontologies. In: Mizoguchi, R., Shi, Z.-Z., Giunchiglia, F. (eds.) ASWC 2006. LNCS, vol. 4185, pp. 400–414. Springer, Heidelberg (2006)
3. Yildiz, B., Miksch, S.: OntoX - a method for ontology-driven information extraction. In: Gervasi, O., Gavrilova, M.L. (eds.) ICCSA 2007, Part III. LNCS, vol. 4707, pp. 660–673. Springer, Heidelberg (2007)
4. Rajput, Q.N., Haider, S.: Use of Bayesian Network in Information Extraction from Unstructured Data Sources. International Journal of Information Technology 5(4), 207–213 (2009)

Optimizing Advisor Network Size in a Personalized Trust-Modelling Framework for Multi-agent Systems

Joshua Gorner[*]

David R. Cheriton School of Computer Science, University of Waterloo

1 Problem

Zhang [1] has recently proposed a novel trust-based framework for systems including electronic commerce. This system relies on a model of the trustworthiness of advisors (other buyers offering advice to the current buyer) which incorporates estimates of each advisor's private and public reputations. Users create a social network of trusted advisors, and sellers will offer better rewards to satisfy trusted advisors and thus build their own reputations.

Specifically, Zhang's model incorporates a "personalized" approach for determining the trustworthiness of advisors from the perspective of a given buyer. The trustworthiness of each advisor is calculated using, in part, a private reputation value; that is, a comparison of the buyer's own ratings of sellers to that advisor's ratings among sellers that both users have had experience with. This is then combined with a public reputation value which remains consistent for all users, reflecting how consistent each advisor's ratings are with the community as a whole.

In this research, we look at one of the open questions stemming from Zhang's proposal, namely the determination of the optimal size of a user's advisor network — that is, the number of advisors on which the user relies. In our progress to date, we have identified three potential methods which may allow us to optimize the advisor network size.

We are mindful of the impact this work could have with trust modelling more generally, as well as the broader area of multiagent systems. For example, this model could be adapted to other applications, including information sharing among peers for home healthcare tasks, where the ability to trust others, and indeed the opinions of others with regards to one's health and safety, is paramount.

2 Progress to Date

2.1 Trustworthiness Thresholding

In our first method, we first note that for each advisor a, a trustworthiness value $Tr(a)$ is calculated by each buyer b, with the said value falling in the range $(0, 1)$.

[*] Supported by funding from the Natural Sciences and Engineering Research Council of Canada.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 419–421, 2010.

We then define some threshold L ($0 \leq L \leq 1$) representing the minimum value of $Tr(a)$ at which we allow an agent to be included in b's advisor network. A buyer b will then only use those advisors where $Tr(a) \geq L$ in determining the public trustworthiness of any sellers of interest.

2.2 Maximum Number of Advisors

In the second method, we set a maximum number of advisors $max\_nbors \leq n$, where n is the total number of advisors in the system, for the advisor network of each buyer. More specifically we sort n advisors according to their trustworthiness value $Tr(a)$, in order from greatest to least, then truncate this set to the first max_nbors advisors. Similar to the previous method, the buyer b will then only make use of these max_nbors advisors in his or her public trustworthiness calculations for sellers.

2.3 Advisor Referrals

Finally, we implement a version of the advisor referral scheme described in [2], for use in combination with either of the methods described above. We posit that by allowing a buyer to indirectly access other advisors with pertinent information outside its advisor network, we can further optimize the network size. However, we must limit the number of advisors accessed through such referrals, in order to ensure some degree of computational efficiency.

For each advisor a_j in the advisor network of a buyer b, that is, the set $A_b = \{a_1, a_2, \ldots, a_k\}$, b checks whether a_j is an acceptable advisor for the seller s. This will be the case if $N_{all}^{a_j} \geq N_{min}$, where $N_{all}^{a_j}$ is the number of ratings provided by an advisor a_j for s, and N_{min} is some minimum number of ratings that may be calculated using formulae provided in Zhang's model.

If a_j is not an acceptable advisor (that is, if $N_{all}^{a_j} < N_{min}$), the algorithm will query a_j's advisor network, sorted from most trustworthy to least trustworthy from the perspective of a_j, to determine in a similar fashion which (if any) of these advisors meet the criteria to be a suitable advisor for s. The first such advisor encountered that is itself not either (a) already in the set of acceptable advisors; or (b) in A_b — since this would imply that the recommended advisor may be added in any event at a later stage — will be accepted.

If none of the advisors of a_j meet the above criteria, this step would be repeated at each subsequent level of the network — that is, the advisors of each member of the set of advisors just considered — until an acceptable, unduplicated advisor was identified. The recursion may be repeated up to $\lceil log_k(|B|) \rceil$ network "levels", where B is the set of all buyers (advisors) in the system. We may need to set a lower maximum for the number of levels for computational efficiency.

This would of course be repeated for each advisor until a full set of k advisors that have each had at least N_{min} interactions with s is found, or a smaller set if the recursion limit has been exceeded on one or more occasions.

References

1. Zhang, J.: Promoting Honesty in E-Marketplaces: Combining Trust Modeling and Incentive Mechanism Design. PhD thesis, University of Waterloo (2009)
2. Yu, B., Singh, M.P.: A social mechanism of reputation management in electronic communities. In: Klusch, M., Kerschberg, L. (eds.) CIA 2000. LNCS (LNAI), vol. 1860, pp. 154–165. Springer, Heidelberg (2000)

An Adaptive Trust and Reputation System for Open Dynamic Environment

Zeinab Noorian

University of New Brunswick,Fredericton,Canada
Faculty of Computer Science
z.noorian@unb.ca

Abstract. The goal of my ongoing is to design an adaptive trust and reputation model suitable for open dynamic environment which is able to merge the cognitive view of trust with the probabilistic view of trust.

1 Problem Statement

Overcoming the inherent uncertainties and risks of the open electronic marketplace and online collaboration systems requires the establishment of mutual trust between service providers and service consumers.Trust and Reputation (T&R) systems are developed to evaluate the credibility of the participants in order to predict their trustworthiness in future actions. The aim of my thesis is to design an adaptive trust and reputation model for open dynamic environments. I intend to make a trust concept tangible in the virtual community by imitating dynamicity and fuzziness qualities of real-life in the virtual world. In particular, I aim to include the constituent elements of the cognitive view of trust such as competence, persistence and motivation beliefs into my trustworthiness evaluation metric. Currently, several T&R systems such as FIRE [1] and PeerTrust [2] have been proposed from distinctive perspectives. However, some important problems remain open in these works. For instance, none of them effectively distinguishes between dishonest and incompetent witnesses. Moreover, context compatibility is mostly neglected and also discriminatory attitudes are not detected thoroughly.I intend to shape this trust model in a way that it offers three main contributions. First, an introduction to decentralized adaptive model with an optimistic approach which minimizes the exclusion of participants by providing suitable mechanisms for differentiating between incompetence, mislead, victims of discrimination and dishonest participants. Notably, the model will consider progressively adjusting itself according to environmental and social dynamics such as volatility in peers' behavior and performances, collusive activities of certain groups of the community members with malicious intentions and the problem of newcomers. Second, through the notion of negotiation, individuals would execute context diversity checking to elicit the most similar experiences which have tremendous impact in the trustworthiness evaluation process. Third, an adaptive trust metric with different weighting strategies of parameters will be proposed which will aim to deduce high-quality judgements under various circumstances.

A. Farzindar and V. Keselj (Eds.): Canadian AI 2010, LNAI 6085, pp. 422–423, 2010.

2 Research Status

1. I have proposed a comparison framework[1] for T&R systems which encompasses most of the possible hard features and soft features inspired by real-life trust experience. The proposed framework can serve as a basis for understanding the current state of the art in the area of computational trust and reputation. This is the stepping stone of my thesis work.
2. Through the proposed T&R model, I maintain separate behavioral models for each participant based on the roles that they play: service provider and witness. In fact, I aim to restrict the good reputation of individuals as service providers to cascade to their reputation as witnesses. Moreover, by analyzing the testimonies of witnesses, we estimate the trend and discriminative attitude parameters of the service providers in certain time intervals.
3. I have developed some initial ideas to provide the service consumer with the ability to elicit the preferences of criterions in witnesses' perspectives and check their compatibility with its own viewpoint in order to obtain approximate predictions of the future service quality of the certain service provider.
4. The next ongoing work will be proposing an adaptive composite trust metric. More explicitly, this trust metric consists of credibility factor, context & criteria similarity rate, transitivity rate, trend and discriminative attitude of service provider and time parameters. Moreover, to deal with the dynamicity of open environments, I intend to define an adaptive accuracy threshold for certain parameters to determine their influence degree, which further will be used to select the trustworthiness evaluation strategies. For instance, one strategy would be if the influence of similarity rate is high enough, the transitivity rate would be ignored. Besides, I will present the credibility measures of witnesses by considering a competency factor, degree of honesty and discrimination tendency parameters which will be evaluated by conducting reasoning on their behavioral aspects in certain time intervals.
5. The last step in my thesis is providing a concrete simulation to examine the applicability of my trust model compared with other available models particularly with FIRE [1]and PeerTrust [2].For instance, I would evaluate their efficiency on dealing with discrimination behavior and examine how they differentiate between victim agents who undergo preferential treatments and dishonest agents who deliberately disseminate spurious ratings.

References

1. Huynh, T.D., Jennings, N.R., Shadbolt, N.R.: An integrated trust and reputation model for open multi-agent systems. AAMAS 13(2), 119–154 (2006)
2. Xiong, L., Liu, L.: Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. IEEE Transactions on Knowledge and Data Engineering 16(7), 843–857 (2004)

[1] The paper is submitted to the journal of Theoretical and Applied Electronic Commerce Research (JTEAR), Feb2010.

Author Index