Proceedings of the Twenty-third Midwest Artificial Intelligence and Cognitive Science Conference

April 21 – 22, 2012
University of Cincinnati
Cincinnati, Ohio

Edited by
Sofia Visa, Atsushi Inoue, and Anca Ralescu

Omnipress – Madison, WISCONSIN

This page is intentionally left blank

Preface

Welcome to the twenty-third Midwest Artificial Intelligence and Cognitive Science Conference (MAICS2012). This is the fourth time that MAICS is held at the University of Cincinnati.

In keeping with the tradition of the previous conferences, this year too, graduate students and junior researchers were encouraged to submit their work to be presented along with that of more senior researchers. We wish to thank the program committee for the thorough reviews that so many members of the committee supplied, including extended comments and suggestions for the authors. Given the diversity of topics in the papers submitted to the conference, the reviewers' tasks were quite challenging.

This year's one and half day conference schedule is packed with thirty papers included in the program and the printed proceedings. Submissions were made from six countries despite that MAICS is usually advertised as a regional conference. We consider this as a global recognition of the conference quality and special appeal.

We are thankful for active contribution and participation of students. About a half of accepted papers are primarily authored by students and, as a result of competitive review scores, the Program Committee nominated five student papers for the best student paper award sponsored by Cognitive Science Society.

This year the conference features critical needs and trends as illustrated by the three invited plenary lecturers, Dr. Michelle Quirk, Dr. Traian Marius Truta and Dr. Kevin Gluck, who will present their ideas on Deterrence Analysis, K-Anonymity in Social Networking Services, and some recent development of Cognitive Models for the Air Force research respectively.

The success of the conference depends on many people who contributed to it. Thanks are due to York F. Choo, who again did an excellent job in designing the conference web page and the front cover of the proceedings, and to many of our students, who helped in many ways.

Finally, we thank our sponsors for their generosity on financial and human recourse contributions – *CINCO Credit Union*, *Cognitive Science Society*, *College of Wooster*, and *the University of Cincinnati, School of Computing Sciences and Informatics*.

We hope for a good conference with lively and interesting discussions.

Atsushi Inoue, General Chair
Anca Ralescu, General Chair
Sofia Visa, Program Chair

AInoue SVisa

MAICS-2012 Organization

Conference Chair

Atsushi Inoue, *Eastern Washington University*
Anca Ralescu, *University of Cincinnati*

Program Chair

Sofia Visa, *College of Wooster*

Program Committee

Hani Abu-Salem, USA
Razvan Andonie, USA
Yoshinori Arai, Japan
Jim Buckley, USA
Denise Byrnes, USA
Dale Courte, USA
Valerie Cross, USA
Paul A DePalma, USA
Susana Irene Daz, Spain
Simona Doboli, USA
Yasser EL-Manzalawy, USA
Martha Evens, USA
Michael Glass, USA
Lluis Godo, Spain
Isao Hayashi, Japan
Cory Henson, USA
Suranga Hettiarachchi, USA
Vasant Honavar. USA
Mircea Ionescu, USA
Atsushi Inoue, USA
Daisuke Katagami, Japan
Jung H. Kim, USA
Dan Li, USA
Richard Maclin, USA
Toshiyuki Maeda, Japan
Mihaela Malita, USA
Logan Mayfield, USA
Augustine Nsang, Nigeria
Masaki Ogino, Japan
Ago Quaye, Nigeria
Muhammad A. Rahman, USA
Anca Ralescu, USA
Dan Ralescu, USA
Mohammad Rawashdeh, USA
Tomasz Rutkowski, Japan
Apkar Salatian, Nigeria
Tom Sudkamp, USA
Christopher Thomas, USA
Kewei Tu, USA
Traian M. Truta, USA
Sofia Visa, USA

Special Design Advisor

York Fang Choo - Front and back cover designed by York Fang Choo.

Teleconference Coordinator

Mojtaba Kohram

Special thanks for administrative support

Mary Davis, School of Computing Sciences and Informatics, University of Cincinnati
Kelly Voght, CINCO Federal Credit Union, Cincinnati

Sponsoring Organizations

School of Computing Sciences and Informatics, University of Cincinnati
Mathematics and Computer Science Department, College of Wooster
Cognitive Science Society
CINCO

Contents

Recognition and Feature Extraction 71

Scientific Computing and Applications 89

Poster Session 1 103

This page is intentionally left blank

Plenary Lectures

Chair: Atsushi Inoue

Deterrence Analysis: AI and Cognitive Science As Necessary Ingredients

Dr. Michelle Quirk

Project Scientist
Basic and Applied Research Office InnoVision Directorate
National Geospatial-Intelligence Agency

Abstract

The concept of deterrence can be defined as the display of possible threats by one party to convince another party to refrain from initiating certain courses of action. The most common deterrent is that of a threat that convinces the adversary not to carry out intended actions because of costly consequences.

In a nutshell, deterrence analysis comprises these three factors:

- The benefits of a course of action
- The costs of a course of action
- The consequences of restraint (i.e., costs and benefits of not taking the course of action we seek to deter).

Deterrence analysis aims to create scores based on these three elements. Often these exercises are static in nature and are conducted for a special problem, without automation or sensible reuse of previous results. One preferred method was to create typical colored tables (HIGH, MEDIUM, LOW) of scores and risks. These tables were compiled mostly manually. A more elegant approach was game theory. In this talk we discuss briefly behavioral game theory and its suitability to deterrence analysis. Further, we propose a computational framework that that has an Artificial Intelligence foundation and employs cognitive sciences in the design and as means to achieve a permeating validation.

We will close the talk with a list of deterrence open questions and an example of their refinement, as a first step to create a true computational engine.

Biographical Sketch

Michelle Quirk is a project scientist in the Basic and Applied Research Office, InnoVision Directorate, National Geospatial-Intelligence Agency (NGA). She has a career that spans over 25 years in the areas of applied computer science and computational mathematics.

Michelle began her research activities in modeling rock mechanics and continued with numerical methods for solving partial differential equations, where she pioneered the work on infinite elements method for the Maxwell's Equations. As a scientist at Los Alamos National laboratory, Michelle developed extensions to JPEG-2000 Standard with applications to hyper-spectral imagery analysis and processing.

Prior to joining NGA, Michelle was a computational scientist at the Defense Threat Reduction Agency, where she worked on strategic deterrence assessments and decision analysis under uncertainty with non-probabilistic, soft metrics.

Michelle earned a M.S. in computational and applied mathematics (1994) and a Ph.D. in mathematics (2003), from the University of Texas at Austin.

K-Anonymity in Social Networks: A Clustering Approach

Traian Marius Truta

Associate Professor of Computer Science
Northen Kentucky University

Abstract

The proliferation of social networks, where individuals share private information, has caused, in the last few years, a growth in the volume of sensitive data being stored in these networks. As users subscribe to more services and connect more with their friends, families, and colleagues, the desire to use this information from the networks has increased. Online social interaction has become very popular around the globe and most sociologists agree that this will not fade away. Social network sites gather confidential information from their users (for instance, the social network site PacientsLikeMe collects confidential health information) and, as a result, social network data has begun to be analyzed from a different, specific privacy perspective. Since the individual entities in social networks, besides the attribute values that characterize them, also have relationships with other entities, the risk of disclosure increases. In this talk we present a greedy algorithm for anonymizing a social network and a measure that quantifies the information loss in the anonymization process due to edge generalization.

Biographical Sketch

Traian Marius Truta is an associate professor of Computer Science at Northern Kentucky University. He received his Ph.D. in computer science from Wayne State University in 2004. His major areas of expertise are data privacy and anonymity, privacy in statistical databases, and data management. He has served on the program committee of various conferences such as International Conference on Database and Expert Systems Applications (DEXA), Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), ACM Symposium of Applied Computing (SAC), and International Symposium on Data, Privacy, and E-Commerce (ISDPE). He received the Yahoo Research! Best Paper Award for Privacy, Security, and Trust in KDD 2008 (PinKDD) for the paper "A Clustering Approach for Data and Structural Anonymity in Social Networks" in 2008. For more information, including the list of research publications please see: http://www.nku.edu/~trutat1/research.html.

Exploring Methods for Exploring Models

Kevin Gluck

Senior Cognitive Scientist
Air Force Research Laboratory
Wright-Patterson Air Force Base, Ohio

Abstract

I will use this colloquium as an opportunity to introduce the motivation, approach, and portfolio of research we have developed at AFRL’s Cognitive Models and Agents Branch over the course of the last decade. A prominent feature of our approach is an emphasis on computational cognitive process modeling, especially in the context of cognitive architecture, so I will provide some background and history on that scientific perspective. Recently we have begun exploring the use of technologies and methods that are increasing the pace of our modeling progress. These include high performance and volunteer computing, stochastic sampling and predictive analytics, and dynamic visualization of interacting parameter spaces. The general point, which I hope to use to generate some constructive (and perhaps controversial) discussion, is that the cognitive science community needs a more efficient and productive infrastructure for achieving theoretical breadth and integration in mathematical and computational modeling.

Biographical Sketch

Kevin Gluck is a Senior Cognitive Scientist with the Air Force Research Laboratory at Wright-Patterson Air Force Base, Ohio. His research interests focus on computational and mathematical models of cognitive processes to improve our understanding of human performance and learning, with emphasis on: learning and forgetting, fatigue, decision heuristics, and distributed and high performance computing for cognitive science. Kevin is leading the expansion of the Human Effectiveness Directorate's in-house investments in cognitive modeling personnel and research lines. He is also the Chair of AFRL's Robust Decision Making Strategic Technology Team, which is a multi-disciplinary, cross-Directorate team of scientists and engineers working on a collection of research projects striving to measure, model, and ensure high quality in complex decision processes and outcomes. Kevin's PhD in Cognitive Psychology is from Carnegie Mellon University. He has worked with AFRL for 15 years, has authored or co-authored more than 50 journal articles, book chapters, and conference papers, and has played a lead role in the organization and management of 13 international conferences and workshops. In portions of 2010 and 2011, Kevin held a "Gastwissenschaftler" (Visiting Scientist) position at the Max Planck Institute for Human Development in Berlin, Germany. In 2011 he was honored to be the first-ever recipient of the Governing Board Award for Distinguished Service to the Cognitive Science Society.

Cognitive Science Society Special Session for Best Student Paper Award

Chair: Sofia Visa

Applying Soft Computing to Estimation of Resources' Price in Oil and Gas Industry

Sholpan Mirseidova, Atsushi Inoue, Lyazzat Atymtayeva

Kazakh-British Technical University
Tole bi st., 59
Almaty, Kazakhstan
s.mirseidova@gmail.com

Abstract

The oil and gas industry is highly risky capital-intensive field of business. Many companies are working hard to perform projects on a global scale in order to get, produce and deliver their final products. No matter the economic conditions, it is vital for organizations to efficiently manage their capitals projects, which facilitates to control expenditure, handle priorities and mineral resources, and make assets productive as quickly as possible. It is also critical to efficiently and safely maintain and improve these assets. Probably the most volatile item of the project cost in oil and gas industry is the market price of resources or assets. Both sides (stakeholders) seek for efficient profitable price for selling and buying final product. This paper provides the description of application developed for fair oil price estimation using Fuzzy Sets and Logic approach of Soft Computing and FRIL inference language with its environment installed on UNIX virtual machine.

Introduction

Managing the complexity and profit of major projects in today's oil and gas landscape has never been more critical. Against the backdrop of a decline in both global economic conditions and corporate revenues, stakeholders are demanding improved return on investment (ROI), reduced risk and exposure and greater transparency. Since capital construction projects in the upstream oil and gas industry comprise a significant percentage of company spend, there must be a particular focus on predictability, transparency and reliability, including estimation of profit, controlling and reducing the costs associated with these projects. The best opportunity to make a positive impact on the life cycle of capital project in this industry is during early planning, even before the capital outlay occurs. In order to control planning it is useful to develop an integrated cost management function that aligns all cost-related processes and functions and incorporates data developed or maintained in other processes. Emphasis should be placed on budget control, approved corporate budget changes and project management internal budget transfers.[1] As the prices of oil and gas fluctuate every day this is the most difficult part of budget control, because even slight changes in the value has a huge impact on overall financial situation of project. That's why it will be very convenient to use Fuzzy Logic methodology of Soft Computing to make certain calculations in order to estimate the total profit of the project and remove the uncertainty of non clear boundaries of oil price.

In the direction of application of fuzzy logic to similar economic problems, the following research was made: the problem of developing automated system of technical – economic estimation in oil and gas fields was considered by Yu.G. Bogatkina [2], and The Fuzzy Logic Framework was build on investigation of risk-based inspection planning of oil and gas pipes [3]. The first one describes economic estimation of oil and gas investment projects witnesses' necessity of taking into account a great number of uncertainty factors. Uncertainty factors influence on investment project can bring about unexpected negative results for the projects, which were initially recognized economically expedient for investments. Negative scenarios of development, which were not taken into consideration in investment projects, can occur and prevent realization of investment project. Especially important is accounting of information uncertainty, which directly depends on mathematical apparatus choice, defined by mathematical theory, and provides for formalization of uncertainty, appearing during control over investment flows. The second framework emphasizes attention on important feature of plant operation – availability of a considerable amount of information as qualitative and imprecise knowledge. Risk-based inspection schedule was developed for oil and gas topside equipment with supporting fuzzy logic models.

No study was made in Kazakhstan connected with problems of uncertainty in oil prices and project costs in terms of the principles of fuzzy logic, which could give a more complete picture of price changes and their influence on the general economic situation in the country, which allows forecasting of rising inflation and determining the most optimal range of prices taking into account various economic factors.

Problem Solving

The given application offers solution to the problem of the project profit estimation considering the price of oil as a fuzzy set. Such types of applications are developed for project managers in oil and gas industry to make evaluation of project profit for the future decision. Also all investors and financial institutions of this industry could be interested in using the given tool.

The total profit of the project could be generally expressed as follows:

$$P = Oil\_price * Supply\_scope$$

According to the research [4] there are three main standards for the formation of market prices for Kazakhstan's oil exports: prices of Brent dtd type of oil, Urals (REBKO) and a mixture of CPC. The prices are usually published by Platts organization (see table 1). The price of Kazakhstan's oil is calculated according to the formula:

$$Price=Brent\_Price \text{ (or } Urals\_Price\text{) } +/- \; Market\_Differential$$

Key benchmarks ($/bbl)				
	Data code	Change	Assessment	Change
Dubai (SEP)	PCAAT00	-1.47	110.10-110.12	-1.47
Dubai (OCT)	PCAAU00	-1.57	110.28-110.30	-1.57
Dubai (NOV)	PCAAV00	-1.54	110.48-110.50	-1.54
MEC (SEP)	AAWSA00	-1.47	110.10-110.12	-1.47
MEC (OCT)	AAWSB00	-1.57	110.28-110.30	-1.57
MEC (NOV)	AAWSC00	-1.54	110.48-110.50	-1.54
Brent/Dubai	AAJMS00	-0.30	5.76-5.78	-0.30
Brent (Dated)	PCAAS00	-0.95	116.84-116.85	-0.95
Dated North Sea Light	AAOFD00	-0.95	116.84-116.85	-0.95
Brent (AUG)	PCAAP00	-1.32	116.46-116.48	-1.32
Brent (SEP)	PCAAQ00	-1.21	115.66-115.68	-1.21
Brent (OCT)	PCAAR00	-1.22	115.60-115.62	-1.22
Sulfur De-escalator	AAUXL00	0.40		
WTI (AUG)	PCACG00	-1.35	94.94-94.96	-1.35
WTI (SEP)	PCACH00	-1.38	95.41-95.43	-1.38
WTI (OCT)	AAGIT00	-1.40	95.87-95.89	-1.40
ACM AUG)*	AAQHN00	-1.10	107.59-107.61	-1.10
ACM (SEP)*	AAQHO00	-1.38	107.61-107.63	-1.38
ACM (OCT)*	AAQHP00	-1.40	107.87-107.89	-1.40

Table 1: Prices of key benchmarks (Source: Platts)

Usually traders make an agreement for the value of market differential. Market price construction for Kazakhstani oil depends on CIF August's (CIF stands for cost, insurance, freight) market differential.

The main idea is to consider these two parameters (the price of Brent dtd. oil and market differential) as fuzzy sets, because the former changes every day, and the second is a result of contract between traders.

The research is based on the theory of fuzzy sets and logic. Fuzzy sets were initially offered by Lotfi A. Zadeh [5] in 1965 as an extension of the classical notion of set. In classical set theory, the membership of elements in a set is assessed in binary terms according to a bivalent condition — particular element either belongs or does not belong to the set (crisp set). In opposite, fuzzy set theory allows the gradual assessment of the membership of elements in a set. According to the definition, a *fuzzy subset* of a set U is defined by means of a membership function $\mu : U \rightarrow [0, 1]$. And a membership of an element x in (crisp) set U is determined by an *indicator (characteristic)* function $\mu : U \rightarrow \{0, 1\}$[6].

Using fuzzy sets in estimation of oil price gives opportunity to remove straight principles of crisp sets. For example, in the case of crisp sets we should take only three exact numbers for oil price and market differential and calculate assessment for the worst, standard, and the best cases. In opposite, with the help of fuzzy sets it is possible to take into account the variation of price in time, in other words statistical or historical changes. Another significance of fuzzy sets is in possibility to manage uncertainty. It means that we can more precisely define which numbers that represent the price of oil can be called normal or higher/lower than that and with which membership degree.

Concerning all of conditions described above three universal sets can be defined:

$U_{X1} = [80, 140]$ – price of Brent dtd.
$U_{X2} = [-5, 5]$ – market differential
$U_Y = [50, 160]$ – oil price

The approximate values of borders of sets are taken from statistical values of mentioned parameters for the previous year [6] (see fig.2 for the set of Brent dtd. price). So there are 2 Inputs and 1 Output in the system.

Also fuzzy sets on inputs X_1 and X_2 and output Y were created with membership values as follows:

Fuzzy sets on X_1(for the price of Brent dtd. see fig.1):

'less than usual' = {1/80, 0.8/90, 0.7/100, 0.5/110, 0.2/113, 0.1/115, 0/118, 0/120, 0/130, 0/140}

'more than usual' = {0/80, 0/90, 0/100, 0/110,0.2/113,0.4/115,0.5/118,0.7/120,0.9/130, 1/140}

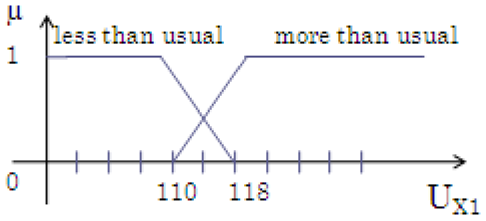


Figure 1: Fuzzy sets for the price of Brent dtd.

The given sets were constructed following principle: as the price for Brent dtd. has changed between 80 and 140 (according to statistics [7] and fig.2), then element 80 belongs to the set 'less than usual' with higher membership which equal s to 1 than 140 belongs to this set (membership equals to 0), and the rest members own steadily decreasing membership degrees. Similarly the set 'more than usual' can be explained.

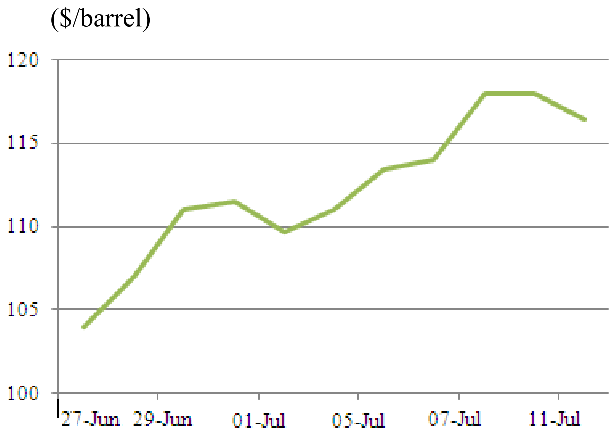


Figure 2: Dated Brent (Source: Platts)

Fuzzy sets on X_2 (for the value of market differential see fig.3)

'significantly high'={1/-5, 0.8/-4, 0.5/-3, 0.2/-2, 0/-1, 0/0, 0.1/0.1, 0.2/1, 0.4/2, 0.6/3, 0.9/4, 1/5}

'significantly low'={0/-5, 0.2/-4, 0.4/-3,0.7/-2, 0.8/-1, 1/0, 0.9/0.1, 0.8/1, 0.6/2, 0.3/3, 0.1/4, 0/5}

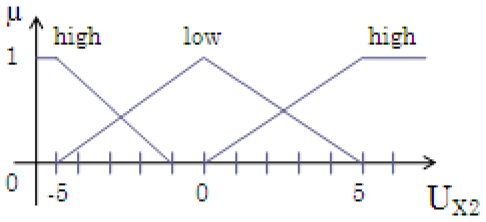


Figure 3: Fuzzy sets for the value of market differential

The most frequently used market differentials vary from -5 to 5 as was mentioned in the list of universal sets for three parameters. The higher the value of market differential (without sign) the more significant influence it has to the final price of resource, so that the members -5 and 5 have the highest membership degree in the set 'significantly high'. Other members of this universal set, which are close to 0, have higher degree in the set 'significantly low' respectively.

Finally, the resulting universal set for oil price was divided into three approximate subsets: cheap, normal or optimal, and expensive. The membership degrees of the elements were assigned following the same principle described above.

Fuzzy sets on Y (for the resulting value of oil price see fig.4):

'cheap'={1/50, 0.9/60, 0.8/70, 0.7/80, 0.6/90, 0.4/100, 0.2/110,0/115, 0/120,0/140,0/160}

'normal'={0/50, 0/60, 0.1/70, 0.3/80, 0.5/90, 0.7/100, 0.9/110, 1/115, 0.5/120, 0/140,0/160}

'expensive'={0/50, 0/60, 0/70, 0/80, 0/90, 0/100,0/110, 0/115, 0.7/120, 0.9/140, 1/160}

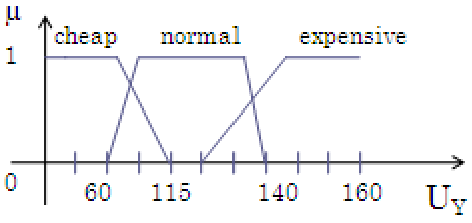


Figure 4: Fuzzy sets for the resulting value of oil price

There are 4 "IF-THEN" rules in the system.

RULE 1: IF X_1 is 'less than usual' AND X_2 is 'significantly low' THEN Y is 'cheap'

RULE 2: IF X_1 is 'less than usual' AND X_2 is 'significantly high' THEN Y is 'normal'

RULE 3: IF X_1 is 'more than usual' AND X_2 is 'significantly low' THEN Y is 'normal'

RULE 4: IF X_1 is 'more than usual' AND X_2 is 'significantly high' THEN Y is 'expensive'

These rules express the direct dependency (proportionality) between two parameters – the price of Brent dtd. and market differential – and oil price according to the formula above.

Development Technologies

The given application was developed using FRIL. With the help of FRIL method of inference number of cases can be calculated.

FRIL was initially an acronym for "Fuzzy Relational Inference Language", which was the predecessor of the current FRIL language, which was developed in the late 1970's following Jim Baldwin's work on fuzzy relations.

FRIL is an uncertainty logic programming language which not only comprises Prolog as one part of the language, but also allows working with probabilistic uncertainties and fuzzy sets as well.

The theory of mass assignments developed by J.F. Baldwin in the Artificial Intelligence group of the Department became a foundation to FRIL language. A fuzzy set is defined as a possibility distribution which is equivalent to a family of probability distributions. FRIL supports both continuous and discrete fuzzy sets.

FRIL gives opportunity to express uncertainty in data, facts and rules with the help of fuzzy sets and support pairs. There are three special types of uncertainty rule besides the Prolog rule. They are: the basic rule, the extended rule, and the evidential logic rule. The second, extended, rule is widely used for causal net type applications, and the evidential logic rule is appropriate to case-based and analogical reasoning. Every rule can have related conditional support pairs, and the method of inference from these rules is based on Jeffrey's rule which is connected with the theorem of total probability. Fuzzy sets can be used to represent semantic terms in FRIL clauses. In addition, a process of partial matching of fuzzy terms like this, which is called semantic unification, gives support for FRIL goals [8], [9].

Analysis of Results

For instance, if the price of Brent dtd equals to 120 (which belong to the set ‘more than usual’) and CIF market differential is -3 the application outputs defuzzyfied value equal to 116.199. If we apply initially described formula to this data, we will get 120 – 3 = 117. So, the result, which was received by fuzzy sets application, is close to the output of calculations following formula. However, there is a difference in 1 dollar per barrel that can considerably affect the final project cost. More results for the values of the Brent dtd. price and market differential from different sets are listed in appendix below.

The application gives the fair price to the sellers’ side, so that the price by contract, that is lower than this, is not profitable, and the price that higher than this, is not competitive on the market. On the other side, buyer takes the reasonable price for the product, which correspond reality with taking into account situation in the market.

Finally, the project profit can be obtained by multiplication of output value to whatever value of the supply scope following the volume in the contract.

Conclusion

As a result let us notice that Fuzzy Sets and Logic can be successfully applied to estimate project profit by assuming several parameters as fuzzy sets, constructing those sets and applying fuzzy logic to them in order to reduce uncertainty. There are still many opportunities to improve the precision of calculations.

The market price of Kazakhstani oil depends on several fundamental factors, which can be divided into fixed and variable. Variable factors, such as the level of consumption of petroleum and petroleum products in a specific period of time, the amount of energy resource available on the market, conditions of delivery, the number of traders, significantly affect on the fluctuation of oil prices. Moreover, the quality of exported oil (density, content of sulfur and wax, etc.), maintenance of quality, stable production and supply, and the cost of oil production in a particular region have an impact on market price [4].

Yet another example, oil price in Kazakhstan also depends on dollar variation. So, additional fuzzy set representing the value of dollar in Kazakhstani tenge can be added and application will calculate it without any problems. In addition, the application can be easily customized to calculate the price of gas and prices of other mineral resources.

Appendix

Source Code

```
%% oil-price.frl
%% NOTE: we use FRIL method of inference
%%       (different from Mamudani, Sugeno,etc.)
% estimate the price of oil depending on Brent dtd. and
market differential
% INPUTS: price of Brent, market differential
% OUTPUT: oil price

%% universe of diccourse

set (dom-brent 80 140)
set (dom-market_dif -5 5)
set (dom-oil_price 50 160)

 %% fuzzy sets on Brent dtd.

(less_than_usual [80:1 90:0.8  100:0.7 110:0.5 113:0.2
115:0.1 118:0 120:0 130:0 140:0] dom-brent)

(more_than_usual  [80:0  90:0  100:0  110:0  113:0.2
115:0.4 118:0.5 120:0.7 130:0.9 140:1] dom-brent)

%% fuzzy sets on market differential

(significantly_high  [-5:1  -4:0.8  -3:0.5  -2:0.2  -1:0  0:0
0.1:0.1 1:0.2 2:0.4 3:0.6 4:0.9 5:1] dom-market_dif)

(significantly_low [-5:0 -4:0.2 -3:0.4 -2:0.7 -1:0.8 0:1
0.1:0.9 1:0.8 2:0.6 3:0.3 4:0.1 5:0 ] dom-market_dif)

%% fuzzy sets on oil price

(cheap  [50:1  60:0.9  70:0.8  80:0.7  90:0.6  100:0.4
110:0.2 115:0 120:0 140:0 160:0] dom-oil_price)

(normal  [50:0  60:0  70:0.1  80:0.3  90:0.5  100:0.7
110:0.9 115:1 120:0.5 140:0 160:0] dom-oil_price)

(expensive [50:0 60:0 70:0 80:0 90:0 100:0 110:0
115:0 120:0.7 140:0.9 160:1] dom-oil_price)

%% Fuzzy Associative Matrix (FAM)
```

```
%
% b\d | L | H |
% --------------
%    L  | C | N |
%   M  | N | E |
%
%% fuzzy rules based on FAM

((price cheap)
     (brent less_than_usual)(market_dif significantly_low))

((price normal)
     (brent less_than_usual)(market_dif significantly_high))

((price normal)
     (brent more_than_usual)(market_dif significantly_low))

((price expensive)
     (brent more_than_usual)(market_dif significantly_high))


((simulation-v B D)
          (addcl ((brent B)))
          (addcl ((market_dif D)))
          (p 'Price of Brent:' B ', ' 'Market differetial:' D)(pp)
          (qsv ((price X)))
          (delcl ((brent B)))
          (delcl ((market_dif D))))

%% eof %%
```

Execution Results

```
Fril >?((simulation-v 80 1))
Price of Brent: 80 ,  Market differetial: 1

((price 76.9223)) : (1 1)
Fuzzy set [49.9978:0 50:0.8 60:0.8 70:0.82 71.6666:0.826667 90:0.68 110:0.36 115:0.2 ] defuzzifies to 76.9223 over sub-domain (50 140)

no (more) solutions

yes

Fril >?((simulation-v 120 -3))
Price of Brent: 120 ,  Market differetial: -3

((price 116.199)) : (1 1)
Fuzzy set [115:0.65 119.545:0.872727 120:0.86 140:0.72 160:0.72 160.011:0] defuzzifies to 116.199 over sub-domain (60 160)

no (more) solutions

yes

Fril >?((simulation-v 90 -1))
Price of Brent: 90 ,  Market differetial: -1

((price 74.128)) : (1 1)
Fuzzy set [49.989:0 50:1 90:0.744 110:0.488 115:0.36 160.007:0.36 160.011:0] defuzzifies to 74.128 over sub-domain (50 115)

no (more) solutions

yes

Fril >?((simulation-v 140 5))
Price of Brent: 140 ,  Market differetial: 5

((price 142.216)) : (1 1)
Fuzzy set [115:0 120:0.7 140:0.9 160:1 160.011:0] defuzzifies to 142.216 over sub-domain (115 160)

no (more) solutions
yes
```

References

1. Ernst & Young. 2011 Capital projects life cycle management. Oil and Gas., pp. 1,6. EYG No. DW0085 1103-1237776
2. "APPLICATION OF THEORY OF FUZZY SET IN AUTOMATED SYSTEM OF TECHNICAL-ECONOMIC ESTIMATION OF OIL AND GAS FIELDS." by Yu.G. Bogatkina,, Institute of Oil and Gas Problems of the Russian Academy of Sciences, 2010
3. Risk-based Inspection Planning of Oil and Gas Pipes – The Fuzzy Logic Framework in EXPLORATION & PRODUCTION – Oil and Gas review Volume 8 Issue 2, p. 26-27.
4. RFCA RATINNGS Rating Agency. Almaty, 2010. АНАЛИЗ НЕФТЕДОБЫВАЮЩЕЙ ОТРАСЛИ РК., pp. 32-37
5. L. A. Zadeh, 1965, "Fuzzy sets". Information and Control 8 (3) 338–353.
6. Page Ranking Refinement Using Fuzzy Sets and Logic., Inoue A., Laughlin A., Olson J., Simpson D., 2011, Eastern Washington University, USA
7. platts.com Value 32, Issue 134, July 11, 2011. Crude Oil Marketwire., pp. 1-13
8. Fril Systems Ltd (1999). Fril - Online Reference Manual - Preliminary Version (incomplete). Retrieved October 20, 2005.
9. Pilsworth, B. W. (n.d.). The Programming Language Fril. Retrieved October 18, 2005

Fuzzy Cluster Validity with Generalized Silhouettes

Mohammad Rawashdeh and Anca Ralescu
School of Computing Sciences and Informatics
University of Cincinnati
Cincinnati, USA
rawashmy@mail.uc.edu; Anca.Ralescu@uc.edu

Abstract

A review of some popular fuzzy cluster validity indices is given. An index that is based on the generalization of silhouettes to fuzzy partitions is compared with the reviewed indices in conjunction with fuzzy *c*-means clustering.

Introduction

Prevalent in many applications, (Jain, Murty, & Flynn 1999), the problem of clustering involves design decisions about representation (i.e. set of features), similarity measure, criterion and mechanism of a clustering algorithm. The clustering literature is very rich in various schemes that address these ingredients (Jain, Murty, & Flynn 1999; Xu & Wunsch 2005). However, the problem itself is centered on the intuitive and easily stated goal of partitioning a set of objects into groups such that *objects within one group are similar to each other and dissimilar to objects in other groups*, which has become a common description of clustering (Jain, Murty, & Flynn 1999; Berkhin 2002; Kleinberg 2002; Xu & Wunsch 2005). As opposed to classification, only few of the existing clustering algorithms are widely used. Indeed, clustering is less appreciated among practitioners of data analysis due to the lack of class labels. Labels are used in the evaluation of loss functions, formal assessments of the goal. This has encouraged researchers to treat clustering in a semi-supervised manner by incorporating as much information as available such as in *must-link* and *cannot-link* constraints (Blienko, Basu, & Mooney 2004) in order to achieve satisfactory results. Usually, there is an end-goal to clustering of a dataset or an end-use of the final clustering. For example, clustering of documents by topic, clustering of images by common content and clustering of proteins by function have as respective end goal a better understanding of a corpus of documents, or of one or more proteins. This suggests that a better treatment for clustering should be in the context of end-use rather than in an application-independent mathematical manner (Guyon, von Luxburg, & Williamson 2009). Accordingly, the unknown desired clustering is the only ground truth assumed about the problem. The properties of the similarity measure sufficient to cluster well, that is, to achieve low error with respect to the ground-truth clustering, are given in (Balcan, Blum, & Vempala 2008). The features, the measure and the algorithm all should be chosen in the context of the end-use. For example, it would be unwise to employ a measure that pairs two images because they show the same person while a clustering by facial expression is desired. This applies to the set of features as well; the features should accommodate for the different possible expressions. In the absence of end-use, clustering becomes an exploratory approach to data analysis, looking for the right ingredients to get the best structure.

The c-means, alternatively k-means (MacQueen 1967), is one popular clustering algorithm that partitions a set of data points $X = \{x_j | j = 1, .., n\}$ into disjoint subsets $U = \{u_i | i = 1, \dots, c\}$. The exclusive cluster assignment characterizes hard clustering and hence it is also referred by hard c-means (HCM). Fuzzy c-means (FCM) family of algorithms imposes relaxed constraints on cluster assignment by allowing nonexclusive but partial memberships, thereby, modeling cluster overlapping. The first FCM algorithm was proposed in (Dunn 1973). Its convergence was later improved in (Bezdek 1981). Both schemes, crisp and fuzzy, optimize a variance-criterion with respect to cluster center and point membership for the specified cluster number. The final clustering is given by a membership matrix, $\mathrm{U} = [u_{ij}]$; u_{ij} is the membership of x_j in u_i. When u_{ij} assumes values in $\{0,1\}$ or $[0,1]$, the matrix characterizes crisp or fuzzy partitions respectively.

It is common to define the pairwise similarities-dissimilarities in terms of distances which, in turn, give a structure i.e. the dataset underlying structure. The clustering algorithm, by processing the pairwise distances implicitly or explicitly, produces a structure, a partition. Its success is determined by the extent to which the produced partition aligns with the underlying structure, or more precisely, agrees with the pairwise distances. Supplying inconsistent values for c, forces the algorithm either to separate similar points or to group dissimilar points in the same cluster. Hence the issue of cluster number is crucial and largely affects clustering quality. Even by choosing features and a measure consistent with the end-use, the inherent number of clusters might be unknown. For example, in a topic-driven clustering application, terms that are significant to each possible topic or common theme might be universally known, but the number of topics represented by documents in a particular dataset is unknown. Even if the cluster number is guessed correctly, there is the unfortunate possibility of obtaining a suboptimal clustering due to local optimum convergence. Besides, clustering of different types, crisp versus fuzzy, can be obtained on the same dataset. For "this and that kind" of reasons, cluster analysis is incomplete without the assessment of *clustering quality*. The issues of cluster number and quality are the main concerns of cluster validity.

This study reviews some fuzzy cluster validity indices then presents a generalization of silhouettes to fuzzy partitions. The performance of all reviewed indices is compared, with discussion, using two different datasets.

Fuzzy *c*-Means Algorithm

FCM, described in (Bezdek, Ehrlich, & Full 1984), incorporates fuzzy membership values in its variance-based criterion as

$$J_m = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^{m} \,.\, d^2(x_j, v_i), \tag{1}$$

where v_i is the center of cluster u_i. The clustering mechanism is carried as a Picard iterative process that alternates the application of

$$v_i = \frac{\sum_{j=1}^{n} u_{ij} \,.\, x_j}{\sum_{j=1}^{n} u_{ij}}, \tag{2}$$

and

$$u_{ij} = \left[\sum_{r=1}^{c}\left(\frac{d^2(x_j, v_i)}{d^2(x_j, v_r)}\right)^{1/(m-1)}\right]^{-1}. \tag{3}$$

The update rules are derived from the necessary conditions of (1) constrained by

$$\sum_{i=1}^{c} u_{ij} = 1; \;\forall j\,, \tag{4}$$

$$\sum_{j=1}^{n} u_{ij} > 0; \;\forall i\,. \tag{5}$$

FCM output ranges from crisp partitions, as produced by HCM, to the fuzziest possible partition for the specified number of clusters i.e. $\mathrm{U}_{c\,\mathrm{x}\,n} = [1/c]$. Informally speaking, there are two sources for fuzziness in a produced partition. First is the amount of overlapping in the underlying structure; equation (3) assigns each point almost the same membership to overlapping clusters whose centers are within small proximity. Second is the exponent; the ratios in (3) become compressed around the value 1 when m is too high, thereby, weakening the role of 'geometry' as a key factor in shaping membership values.

Cluster Validity

Modeling the pairwise similarities-dissimilarities by a distance measure restates the goal of clustering as the search for optimally *compact* and *separated* clusters. One cluster is compact only if its member points are within small proximity from each other. Two clusters are well separated only if their member points are distant from each other. Accordingly, the variance-based criterion found in *c*-means can be thought of as a measure of compactness, which was shown to be equivalent to a measure of separation for the same number of clusters (Zhao & Karypis 2001). Hence, *c*-means is capable of producing partitions that are optimally compact and well separated, for the specified number of clusters. Note that better clustering might still be achieved by specifying different cluster numbers. Since clustering algorithms are supposed to optimize their output in compactness and separation, both should be assessed to find clustering quality.

One might need to distinguish between the *desired structure*, the *underlying structure*, and *candidate structures* produced by clustering algorithms. The desired structure is the ground truth clustering, mentioned earlier. What is known about this clustering might be vague or incomplete but it should drive the problem design. The underlying structure is the one shaped by the pairwise distances which suggests unique clustering (Fig. 1a), multiple clusterings (Fig. 1b) or no clustering due to the lack of any structure (Fig. 1c). A clustering algorithm produces different partitions for different configurations i.e. distance measure, parameters, etc. The best case scenario is when the pairwise distances structure the points into the desired grouping and an algorithm successfully produces a clustering that aligns with the underlying structure. Validating a produced clustering with respect to the underlying structure is possible by means of cluster validity indices.

Partition Coefficient

The partition coefficient, *PC*, is defined as the Frobenius norm of the membership matrix, divided by the number of points, as in

$$PC(\mathrm{U}) = \frac{1}{n} \,.\, \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^2. \tag{6}$$

The coefficient is bounded by $1/c$ and 1, if applied to FCM output. Its use as a measure of partition fuzziness was first investigated by Bezdek in his Ph.D. dissertation (Bezdek

1973). Although it can be used as a validity index with some success, it has been shown to be irrelevant to the problem of cluster validity (Trauwaert 1988). Clearly, the coefficient does not incorporate the pairwise distances that are necessary to the assessment of compactness and separation. Therefore, it is not reliable for the validation of any given partition, for example, one produced by random cluster assignment. Also, the coefficient assumes its upper value on any crisp partition, regardless of its clustering quality. Nevertheless, the coefficient does what it knows best, measuring fuzziness.

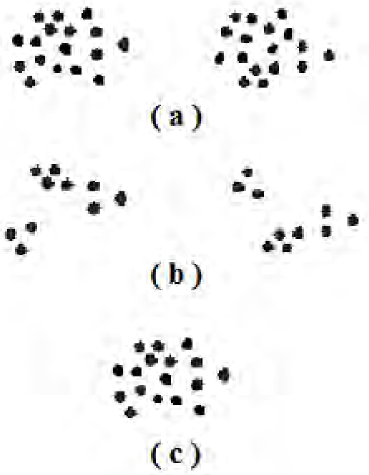


Figure 1. The structure of the dataset suggests (a) c = 2, (b) c = 2, 4 and 6, (c) c = 1; no structure.

Xie-Beni Index

An index that really measures compactness and separation was proposed by Xie and Beni, *XB* index (Xie & Beni 1991). *XB* takes the form of a ratio; the minimum center-to-center distance appears in its denominator and J_2 , as exactly as in FCM, is in its numerator but divided by n. Hence, *XB* is a measure of compactness divided by a measure of separation, given by

$$XB(\mathrm{U},V;\,X)=\frac{XB_{cmp}/n}{XB_{spr}}=\frac{\frac{1}{n}\sum_{i=1}^{c}\sum_{j=1}^{n}u_{ij}^{2}\,.\,d^{2}(x_j,v_i)}{\min\{\,d(v_r,v_s)\mid r\,<\,s\}}. \tag{7}$$

In (7), $V=\{v_i \mid i=1,\dots,c\}$ denotes the set of cluster centers. The authors define the variation of each cluster as the sum of point fuzzy deviations, *squared*. With respect to a cluster, a point deviation is its distance from the cluster center weighted by its membership. The total variation is the sum of all cluster variations that gives the compactness of the partition, when divided by n. This description explains why memberships and distances in (7) are squared. However, they suggest substituting u_{ij}^{m} in place of u_{ij}^{2} where m is the same as the value used in FCM, justified by making the index 'compatible' with FCM. The final value of (1) can be directly plugged in (7), provided it is still available, or else recomputed. It is unclear how being compatible with FCM or raising membership values to powers different than 2 relates to the assessment of compactness and separation, or to the 'geometry' underlying the data.

Fuzzy Hypervolume

The development of the index started as part of work that formulates clustering as a problem of maximum likelihood estimation (MLE) of a mixture of multivariate densities (Wolfe 1970); the dataset is assumed to be drawn from such a mixture. Bezdek and Dunn, in (Bezdek & Dunn 1975), give the MLE algorithm and FCM as well. The MLE algorithm solves for a composite parameter vector of densities' means, covariance matrices and the a priori probabilities. They describe the use of FCM to approximate the ML parameters. Substituting FCM-generated membership values for posterior probabilities computes the remaining ML parameters. A justification is given by comparing the behavior of two update rules in both algorithms. They produce small values when evaluated on data points that are distant from some density-cluster center relative to their distance from nearest center. However, they point out the fact that both algorithms compute different centers and distances.

Gath and Geva in (Gath & Geva 1989), first, give a similar description of FCM, and fuzzy MLE that is derived from FCM, as opposed to the separate treatment given by Bezdek and Dunn. Then, they suggest a 2-stage clustering scheme, FCM followed by MLE, justified by the unstable behavior of MLE as a standalone clustering scheme. As part of their work, some validity measures were proposed; among them is the *fuzzy hypervolume* measure (*FHV*). *FHV* is defined in terms of the covariance matrix determinants. The covariance matrix of cluster u_i can be constructed using

$$\mathrm{F}_i=\frac{\sum_{j=1}^{n}u_{ij}\,.\,(x_j-v_i)\,(x_j-v_i)^T}{\sum_{j=1}^{n}u_{ij}}. \tag{8}$$

The *hypervolume* is then computed by

$$FHV(\mathrm{U},V;X)=\sum\nolimits_{i=1}^{c}[\det(\mathrm{F}_i)]^{1/2}. \tag{9}$$

The determinants are functions of cluster spreads and point memberships. A clustering that is evaluated the smallest is assumed to be optimal. However, the following observations can be made:

- According to the authors, the index is sensitive to substantial overlapping in the dataset.

- It is unclear how the measure accounts for compactness and separation.
- Assuming that an MLE mixture has been successfully found, is it the best clustering in compactness and separation?
- Is the measure applicable to crisp partitions?
- The use of FCM as MLE requires setting m=2; how does the measure performs on partitions obtained using m different than 2?

Pakhira-Bandyopadhyay-Maulik Index

Pakhira et el. proposed an index, referred here as *PBM*, that targets both fuzzy and crisp partitions (Pakhira, Bandyopadhyay, & Maulik 2004). Its fuzzy version is defined as

$$PBM(\mathrm{U}, V, c, m; X) =$$

$$\left(\frac{\sum_{j=1}^{n} d(x_j, v)}{c} \cdot \frac{\max\{ d(v_r, v_s) \mid r \neq s\}}{\sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m} \cdot d(x_j, v_i)}\right)^2. \tag{10}$$

In (10), v is the center of the whole dataset. The index can be factorized into a measure of compactness

$$PBM_{cmp} = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m} \cdot d(x_j, v_i), \tag{11}$$

a measure of separation

$$PBM_{spr} = \max\{ d(v_r, v_s) \mid r \neq s\}, \tag{12}$$

and an artificial factor, irrelevant to compactness and separation,

$$PBM_{x} = \frac{\sum_{j=1}^{n} d(x_j, v)}{c}. \tag{13}$$

The whole term in (10) is raised to power two that is also irrelevant to the assessment of clustering quality. Larger values of the index are assumed to indicate better clustering. It can be noticed though that the quantity in (12) does not necessarily capture the separation between all pairs of clusters; an authentic separation measure should account for the poor separation found in partitions into large number of clusters. .

Average Silhouette Index

Rousseeuw proposed an index for the validation of crisp partitions (Rousseeuw 1987). It is based on the notion of *silhouette*. A silhouette, constructed for each data point, measures the clustering quality for that point. The average over members of the whole dataset or an individual cluster is a measure of the set clustering quality. To illustrate silhouette construction, consider for the data point x_k , the cluster to which x_k has been assigned, A, and let C be any cluster different than A. The silhouette $s_k = s(x_k)$ is defined in terms of a measure of compactness a_k and a measure of separation b_k . The average distance of x_k to points in A computes a_k, while b_k is the minimum average distance from x_k to all other clusters. Let B denotes the cluster corresponding to b_k (see Fig. 2).

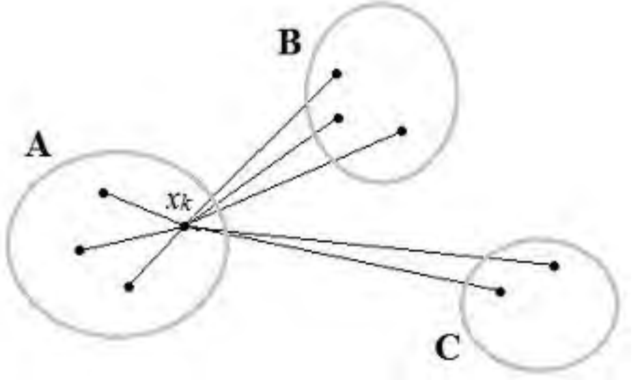


Figure 2. With respect to x_k, a_k is the average length of lines within A and b_k is the average length of lines between A and B.

Then the silhouette of x_k is defined by

$$s_k = \frac{b_k - a_k}{\max\{a_k, b_k\}}. \tag{14}$$

Clearly, (14) evaluates to values in $[-1, 1]$. The average silhouette over a cluster u_i or the whole dataset X are given respectively by

$$Sil(u_i) = \frac{\sum_{x_j \in u_i} s_j}{|u_i|} = \frac{\sum_{j=1}^{n} u_{ij} \cdot s_j}{\sum_{j=1}^{n} u_{ij}}, \tag{15}$$

$$Sil(X) = \frac{\sum_{j=1}^{n} s_j}{n}. \tag{16}$$

Note that the membership values in (15) are crisp and u_i denotes the i^{th} cluster, as a set.

From the data point perspective, the measure assumes positive values if the separation distance is larger than the compactness distance and negative values if vice versa. A value near zero indicates that the point is at clusters boundary region, of course in the context of clustering on hands. At the coarser cluster level, the average silhouette indicates weak structure if near zero, strong if near +1 and misclustering if near -1. Since a clustering algorithm cannot do any better than the underlying structure, an average close to +1 is attainable only in the presence of a strong structure.

The following appealing properties recommend the silhouette index:

- As opposed to other indices, it validates a given clustering at point level, providing thus the finest granularity.
- It is algorithm-independent.
- It takes as input only the pairwise similarities-dissimilarities and the membership matrix.
- As explained in the original work of Rousseeuw, it can be used to 'visualize' the clustering quality of a given partition.
- Its assessment of compactness and separation conforms literally to the stated goal of clustering; a relatively small $\mathrm{a_k}$ compared to $\mathrm{b_k}$ means that $\mathrm{x_k}$ has been successfully grouped with its similar points in the same cluster in a way that separates from its dissimilar points.

Extended Average Silhouette Index

The above construction of silhouettes is not directly applicable to fuzzy partitions since it requires crisp cluster boundaries, necessary to the computation of cluster average distances. Nevertheless, a fuzzy partition might be validated by silhouettes after being defuzzified, for example, by setting the maximum membership degree of each point to one and nullifying the rest. However, this discards cluster overlapping, defeating the reason of using FCM not HCM. An extension that integrates fuzzy values with silhouettes, computed from the defuzzified partition, into an average silhouette-based index was proposed in (Campello & Hruschka 2006). They suggest computing a weighted mean in which each silhouette is weighted by the difference in the two highest fuzzy membership values of the associated point. More precisely, if $p(j)$ and $q(j)$ denote cluster indices with the two highest membership values associated with x_j then the index is given by

$$eSil(X) = \frac{\sum_{j=1}^{n}\left(u_{p(j)j} - u_{q(j)j}\right) .\ s_j}{\sum_{j=1}^{n}\left(u_{p(j)j} - u_{q(j)j}\right)}, \qquad (17)$$

Therefore, points around cluster centers become significant to the computation of the index since they have higher weights, as opposed to the insignificant points found in overlapping regions. Clearly, such an assessment is not thorough since it tends to ignore the clustering of points in overlapping regions.

Generalized Intra-Inter Silhouette Index

A generalization of silhouettes to fuzzy partitions is given in (Rawashdeh & Ralescu), based on the following central observations:

- A partition of a set of points into any number of clusters is essentially a clustering of the associated pairwise distances into *intra-distances* (within-cluster) and *inter-distances* (between-cluster).
- A strong structure, a good clustering, has small intra-distances and large inter-distances i.e. similar points are grouped together and dissimilar points are separated.
- In the context of a crisp partition, each distance is either intra-distance or inter-distance. This is modeled by intra-inter scores associated to a distance that assume the values 0 and 1, indicating distance membership.
- In the context of a fuzzy partition, two points belong to each cluster simultaneously and separately with some degree, intuitively suggesting the assignment of fuzzy intra-inter scores to the pairwise distances,

The original construction of silhouettes, which already incorporates the pairwise distances, is reformulated to employ intra-inter scores. The following is applicable to both crisp and fuzzy partitions, and it carries similar computation as in the original construction, provided that the partition is crisp. As input, the construction requires the pairwise distances $\mathrm{D}_{n\,\mathrm{x}\,n}$ and the membership matrix $\mathrm{U}_{c\,\mathrm{x}\,n}$.

Step 1. Given a partition into c clusters, each distance d_{jk}, associated with x_j and x_k , is intra-distance with respect to either cluster and inter-distance with respect to any of the 2-combinations of c clusters. The following constructs all of the $(n \text{ x } n)$ intra-inter matrices:

$$\begin{gathered}\mathrm{IntraDist}_i = \left[\, intra_i\left(d_{jk}\right)\right];\ 1 \le i \le\ c, \\ intra_i(d_{jk}) =\ (u_{ij} \wedge u_{ik}).\end{gathered} \qquad (18)$$

$$\begin{gathered}\mathrm{InterDist}_{rs} = \left[inter_{rs}(d_{jk})\right];\ 1 \le r < s \le c, \\ inter_{rs}(d_{jk}) =\ (u_{rj} \wedge u_{sk}) \vee (u_{sj} \wedge u_{rk}).\end{gathered} \qquad (19)$$

Step 2. With respect to each point x_j , weighted means over the associated distances are computed, using intra-inter scores as weights; from which the compactness distance a_j and the separation distances b_j are selected. That is

$$a_j = \min\left\{\frac{\sum_{k=1}^{n}\mathrm{IntraDist}_i(j,k)\ .\ d_{jk}}{\sum_{k=1}^{n}\mathrm{IntraDist}_i(j,k)}\ |\ 1 \le i \le\ c\right\}, \qquad (20)$$

$$b_j = \min\left\{\frac{\sum_{k=1}^{n}\mathrm{InterDist}_{rs}(j,k)\ .\ d_{jk}}{\sum_{k=1}^{n}\mathrm{InterDist}_{rs}(j,k)}\ |\ 1 \le r < s \le c\right\}. \qquad (21)$$

Step 3. The silhouette of each point is found using (14).

Similar to the original average index, the average intra-inter silhouette, $gSil$, over members of the whole dataset is an assessment of its clustering quality. For each fuzzy cluster, a weighted mean using point membership values as weights, is a measure of its clustering quality.

Experiments and Discussion

For further evaluation of the validity indices presented above, a few concrete examples are considered as follows:

Example 1.

Clustering algorithms rely on pairwise distances to form their output and this should be taken into consideration when testing any proposed validity index. Consider the dataset given in Fig. 3. It is tempting to claim that $c = 2$ is the optimal number of clusters, however, this requires a similarity measure better to the task, than the Euclidean distance.

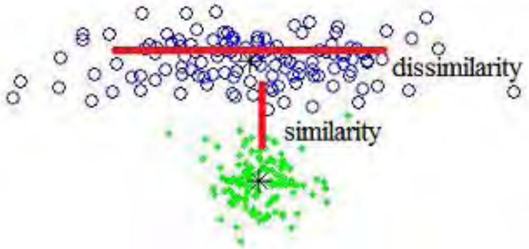


Figure 3. A dataset sampled from two Gaussians. The Euclidean distance is, somehow, inconsistent with the apparent clustering into 2 clusters.

Although HCM, using the Euclidean distance, successfully detects the two clusters, XB, PBM, Sil, $eSil$ and $gSil$ all score $c = 3$ better than $c = 2$ due to better compactness (Fig. 4). Only FHV gives $c = 2$ a better score, since it is based on fitting the data with a mixture of Gaussians. A single bi-Gaussian fits the overlapping clusters in Fig. 4b better than two Gaussians, assuming the crisp probability-membership values produced by HCM.

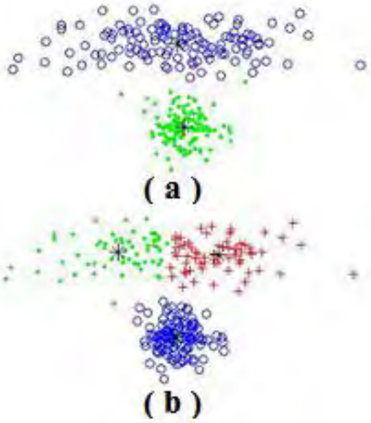


Figure 4. HCM clustering of the dataset from Fig. 3 to (a) 2 clusters and (b) 3 clusters.

Example 2.

Different FCM partitions, using $c = 2, \ldots, 9$, were obtained on a dataset shown in Fig. 5.

Fig. 6 shows the performance of PC, Sil, $eSil$ and $gSil$. PBM, XB and FHV are shown in Figs. 7, 8 and 9 respectively.

The extended index, $eSil$, scores the partition with $c = 3$ clusters (Fig. 5a) higher than the one with $c = 4$ clusters (Fig. 5b). A different ranking is suggested by the original index, Sil, and the generalized index, $gSil$. Both Sil and $eSil$ incorporate the same silhouettes that are computed from the defuzzified membership matrix; clearly, the disagreement is caused by the weights in $eSil$. The points that occupy the undetected middle cluster (Fig. 5a) are not assigned high memberships to any of the three detected clusters; hence, they have low weights. The index $eSil$ just ignores these points that are of insignificant weights and of approximately zero silhouettes. For the same reason, $eSil$ always appears above the curve of Sil. The generalized index $gSil$ can be naturally applied to both crisp and fuzzy partitions. It accounts for changes in the parameter m and does not require any defuzzification of partitions. It scores highest the partition with clusters $c = 5$ (Fig. 5c).

The PBM index evaluates the clustering in Fig. 5d as the best. The separation measure, maximum center-to-center distance, does not decrease with c even after reaching the cluster number that is sufficient for a good clustering of the dataset. In addition, the compactness measure decreases monotonically with c, provided a reasonable clustering algorithm is used. Therefore, PBM has a nondecreasing behavior with c that can be easily exposed using small toy datasets. Moreover, it is not recommended to use any factor that is irrelevant to the assessment of compactness and separation, as part of a validity index.

The XB index also fails in its ranking; it scores $c = 3$ better than $c = 5$. The separation measure, minimum center-to-center distance, does not account for the spread of detected clusters: in Fig. 5a, the centers are well separated but there is overlapping among the clusters in the middle region. The separation measure is not thorough in its assessment as opposed to silhouette-based indices that make assessments at point level. Therefore, XB is not reliable to detect good cluster numbers, and to compare between any two partitions, in general; it is in disagreement with the silhouette-based indices in its scoring of the partitions with $c = 7$ and $c = 8$.

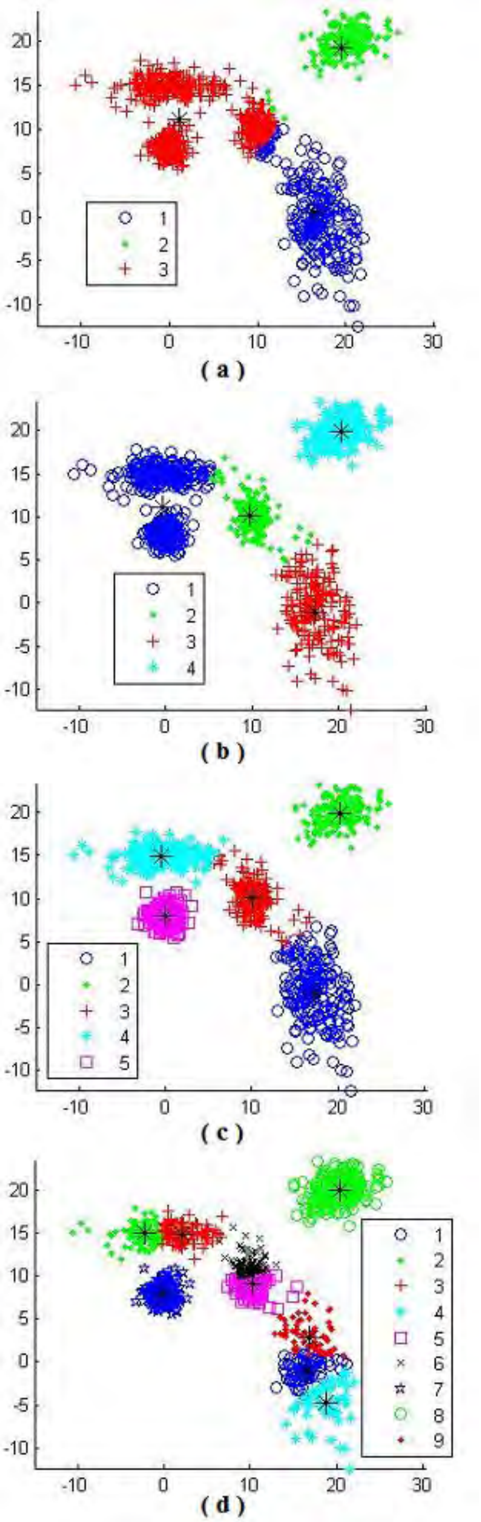


Figure 5. Showing FCM clustering, $m = 2$ and $c = 3,4,5,9$ obtained on a dataset of 1000 points, sampled from 5 bi-Gaussians, 200 each.

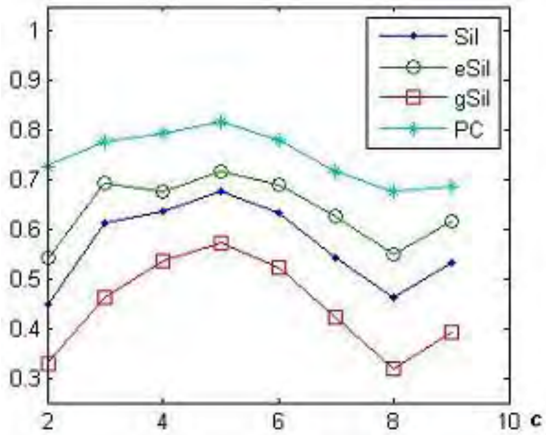


Figure 6. Silhouette-based indices and *PC* vs. c, of FCM applied to the dataset in Fig. 5.

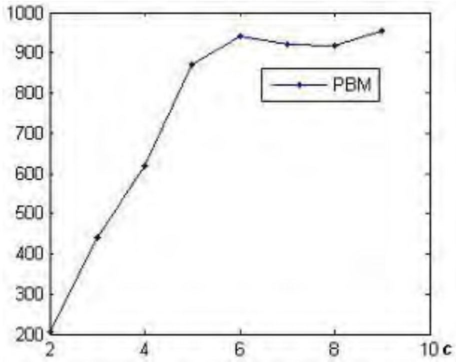


Figure 7. *PBM* vs. c, of FCM applied to the dataset in Fig. 5.

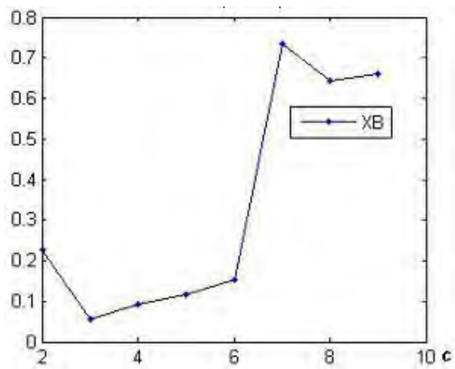


Figure 8. *XB* vs. c, of FCM applied to the dataset in Fig. 5.

Distance-based similarities and dissimilarities are inferred from how distance values compare with each other, not from distance magnitudes. Hence, the strength of the underlying structure is determined by distance values relative to each other. Since the quotient in (14) is just a difference in compactness and separation relative to their maximum, an average over the whole dataset measures the strength of a given clustering. Values close to +1, obtained from the average silhouette index, indicate good clustering and a strong underlying structure as well. It is worth noting that, silhouette-based indices are also scale-invariant that is, scaling the dataset by some factor, multiplying by 100 for example, does not affect their values since the structure is still the same. This is not the case for *FHV* and *PBM*. Hence, silhouette-based indices are easier to interpret.

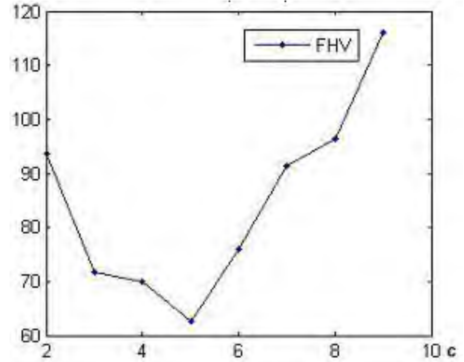


Figure 9. *FHV* vs. c, of FCM applied to the dataset in Fig. 5.

Conclusion

A satisfactory or useful clustering requires careful selection of the features and the measure which, combined together, define the pairwise similarities-dissimilarities. Clustering algorithms, probably of different models, by varying model parameters, as in cluster number, produce partitions, candidate structures. The job of a validity index is to find the candidate that is best supported by the pairwise similarities-dissimilarities, in other words, the clustering that best aligns with the underlying structure. FCM is used mainly to model cluster overlapping in datasets, facilitated by partial cluster memberships assigned to the points, which also results in points in the same cluster taking different membership values. The generalized silhouette index is applicable to both approaches, crisp and fuzzy, of structure modeling to guide the search for the best structure in the dataset.

References

Balcan, M.-F.; Blum, A.; Vempala, S. 2008. A Discriminative Framework for Clustering via Similarity Functions. Proceedings of the 40th annual ACM symposium on Theory of Computing (STOC).

Berkhin, P. 2002. Survey of Clustering Data Mining Techniques. Technical report, Accrue Software, San Jose, CA.

Bezdek, J. C. 1973. Fuzzy Mathematics in Pattern Classification. Ph.D. Diss., Cornell University.

Bezdek, J. C. 1981. Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press.

Bezdek, J. C.; Dunn, J. C. 1975. Optimal Fuzzy Partitions: A Heuristic for Estimating the Parameters in a Mixture of Normal Distributions. IEEE Transactions on Computers 24(4): 835-838.

Bezdek, J. C.; Ehrlich, R.; Full, W. 1984. FCM: the Fuzzy c-Means Clustering Algorithm. Computers and Geosciences 10: 191-203.

Blienko, M.; Basu, S.; Mooney, R. 2004. Integrating Constraints and Metric Learning in Semisupervised Clustering. Proceedings of the 21st International Conference on Machine Learning. Banff, Canada.

Campello, R.; Hruschka, E. 2006. A Fuzzy Extension of the Silhouette Width Criterion for Cluster Analysis. Fuzzy Sets and Systems, 157: 2858-2875.

Dunn, J. C. 1973. A Fuzzy Relative of the ISODATA Process and its Use in Detecting Compact Well-Separated Clusters. J. Cybernet 3: 32-57.

Gath, I.; Geva, A. 1989. Unsupervised Optimal Fuzzy Clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 11(7): 773-781.

Guyon, I.; von Luxburg, U.; Williamson, R. C. 2009. Clustering: Science or Art?. NIPS Workshop ”Clustering: Science or Art”.

Kleinberg, J. 2002. An Impossibility Theorem for Clustering. Proceedings of Advances in Neural Information Processing Systems 15: 463-470.

Jain, A.; Murty, M.; Flynn, P. 1999. Data Clustering: A Review. ACM Computing Surveys, 31(3),264-323.

MacQueen, J. 1967. Some Methods for Classification and Analysis of Multivariate Observations. Proceedings of the 5th Berkeley Symposium on Mathematics. Statistics and Probability 2: 281-297. Berkeley, CA.

Pakhira, M.; Bandyopadhyay, S.; Maulik, U. 2004. Validity Index for Crisp and Fuzzy Clusters. Pattern Recognition 37: 481–501.

Rawashdeh, M.; Ralescu, A. Crisp and Fuzzy Cluster Validity: Generalized Intra-Inter Silhouette Index. Forthcoming.

Rousseeuw, P. J. 1987. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. Computational and Applied Mathematics 20: 53-65. North-Holland.

Trauwaert, E. 1988. On the Meaning of Dunn’s Partition Coefficient for Fuzzy Clusters. Fuzzy Sets and Systems 25: 217–242.

Xie, X.; Beni, G. 1991. A Validity Measure for Fuzzy Clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 3(8): 841-846.

Xu, R.; Wunsch, D. I. I. 2005. Survey of Clustering Algorithms. IEEE Transactions on Neural Networks 16(3): 645-678.

Wolfe, J. H. 1970. Pattern Clustering by Multivariate Mixture Analysis. Multivariable Behavioral Research, pp. 329-350.

Zhao, Y.; Karypis, G. 2001. Criterion Functions for Document Clustering: Experiments and Analysis. Technical Report, CS Dept., Univ. of Minnesota.

Computing with Words for Direct Marketing Support System

Pakizar Shamoi

Department of Computer Science,
Kazakh-British Technical University
pakita883@gmail.com

Atsushi Inoue

Department of Computer Science,
Eastern Washington University
inoueatsushij@gmail.com

Abstract

This paper highlights the simplicity and effectiveness of Computing with Words (CW) in the implementation of target selection. Direct marketing can be considered as one of the main areas of application for this methodology. In particular, fuzzy classification is applied in it with the purpose of choosing the best potential customers for a new product or service from a client database. One of the advantages of the proposed method is that it is consistent with relational databases. Our methodology makes it possible to form queries in natural language, such as "*print the list of not very old married clients with more-or-less high income"*, which is impossible using a standard query mechanism.

Introduction

There is one fundamental advantage of humankind that needs to be inculcated into the various information systems. It is the remarkable human ability to perform a wide range of mental tasks without any measurements and any computations (Zadeh, 2002; Herrera, et al., 2009; Martinez, et al., 2010). That is possible due to the brain's crucial ability to manipulate perceptions of size, distance, weight, speed, etc. (Zadeh, 2002). The main difference between measurements and perceptions is that the former are crisp whereas the latter are vague (fuzzy) - the transition from membership to non-membership is gradual rather than sudden.

The main purpose of using natural (linguistic) queries instead of numbers is that it is much closer to the way that humans express and use their knowledge. Perception-based rational decisions in an environment of imprecision are becoming highly actual (Zadeh, 2002). An important use of the Computing with Words (CW) methodology, which is in the heart of fuzzy logic, is its application to decision making (Zadeh, 1965; Zadeh, 1975; Zadeh, 1996; Ying, 2002; Herrera, et al., 2009). In fact, CW can simplify the decision processes when the experts can only provide qualitative, but not quantitative information about the evaluated alternatives (Herrera, et al., 2009).

This paper is organized in six sections. First one is this introduction. Next we emphasize the critical importance of target selection in direct marketing. Furthermore, we examine in details the how fuzzy approach was applied to make the process of target selection more efficient. Particularly, it discusses the concepts of linguistic variables and hedges, fuzzification, explains the need for imprecision. The next section considers in detail the fuzzy querying model implemented using Computing with Words methodology that is solely based on fuzzy mathematics. Then we provide illustrative examples of different types of queries and their result sets obtained from the application developed. Finally, the last section provides the concluding remarks of this study.

The Role of Target Selection in Direct Marketing

Direct marketing (DM) is a form of advertising that enables companies to communicate directly to the customer, with various advertising techniques including email, mobile messaging, promotional letters, etc. The crucial idea there is to be able to deliver the marketing message to the clients that are likely to be interested in the product, service, or offer (Mederia and Sousa, 2002). So, DM companies or organizations try to set and maintain a direct relationship with their clients in order to target them individually for specific product or service.

An important data mining problem from the world of DM is target selection (Mederia and Sousa, 2002). The main task in target selection is the determination of potential customers for a product from a client database. Target selection algorithms identify the profiles of customers who are likely to respond to the offer for a particular product or service, given different types of information, like profession, age, purchase history, etc. In addition to the numerical performance, model transparency is also important for evaluation by experts, obtaining confidence in the model derived, and selecting an appropriate marketing channel (Mederia and Sousa, 2002). Fuzzy models for target selection are interesting from this angle of view, since they can be used to obtain numerically consistent models, while providing a linguistic description as well.

As mentioned above, in DM the selection of the target audience is a very important stage. Different DM techniques benefit from accurate target selection. Take, for example, a direct mail, used in the promotion of goods and services to organizations and individuals through electronic mail. Some DM methods using particular media, especially email have been criticized for poor target selection strategy. This poses a problem for marketers and consumers alike. On the one hand, advertisers do not wish to waste money on communicating with consumers not interested in their products. Also, they don't want to lose potential customers. On the other hand, people usually try to avoid spam. However, they want to be aware of the new products/services that might be interesting for them.

As previously mentioned, in order to maximize its benefits direct mail requires careful selection of recipients. So, if the selection of recipients is too liberal, it will increase unnecessary spending on DM, if it is too strict – we'll lose some potential customers. Virtually all companies that work with a database of 100 or more customers use email-mailing in their business (Ribeiro and Moreira, 2003). But again, this is a very delicate instrument, because the line between a useful message and spam is very thin. Therefore, companies providing mailing services, must constantly engage in outreach efforts, so that due to their own ignorance, they do not lose customers and reputation, sending spam and making other common mistakes.

Fuzzy Approach

Need for imprecision

Nowadays, most of the data processed in information systems has a precise nature. However, a query to the database, formed by a person, often tends to have some degree of fuzziness. For example**,** the result of a query in a search engine is a lot of references to documents that are ordered by the degree of relevance to the request. Another simple example of natural query used in everyday life: "Find a listing for housing that is *not very expensive* and is *close* to downtown". Statements like *"not very expensive," "close"* are vague, imprecise, although rent price is completely determined, and the distance from the center of the apartment - up to a kilometer. The cause of all these problems is that in real life, we operate and argue using imprecise categories (Zadeh, 1965; Zadeh, 1975).

For example, a company launches an advertising campaign among their clients about new services through direct mail. The Marketing Service has determined that the new service will be most interesting for *middle-aged married men*, with *more-or-less high income*. A crisp query, for example, might ask for all married males aged 40 to 55 with an income of more than 150,000 tenge. But with such a request we may weed out a lot of potential clients: a married man aged 39, with an income of 250,000 tenge does not fall into the query result, although he is a potential customer of the new service.

Linguistic variables

One of the main hindrances of modern computing is that a concept cannot be well understood until it is expressed quantitatively. This is where linguistic variables come in. The main motivation to prefer linguistic variables rather than numbers is that a linguistic description is usually less specific than a numerical one (Zadeh, 1975).

According to Zadeh , "By a linguistic variable we mean a variable whose values are not numbers but words or sentences in a natural or artificial language" (Zadeh, 1975). So, for example, *Income* is a linguistic variable if its values are linguistic (*not very low, average, more-or-less high, …*) rather than numerical (100 000 tg., 150 600 tg.…). Following that logic, the label *high* is considered as a linguistic value of the variable *Income*, it plays the same role as some certain numerical values. However, it is less precise and conveys less information.

To clarify, in the example provided, *Income* is a linguistic variable, while *low*, *average*, *high* are linguistic values, represented in the form of fuzzy sets. The set of all linguistic values of a linguistic variable is called term set.

Although a linguistic value is less precise than a number it is closer to human cognitive processes, and that can be exploited successfully in solving problems involving uncertain or ill-defined phenomena. So, in situations where information is not precise (which are very common in our real life), linguistic variables can be a powerful tool that takes the human knowledge as model (Herrera and Herrera-Viedma, 2000).

Besides their primary meaning, linguistic values may involve connectives such as *and, or, not* and hedges such as *very, quite extremely, more or less, completely, fairly,* etc. about which we will talk extensively later.

Fuzzification

It is highly important for any target selection model to select the clients' features that will play the role of explanatory variables in the model (Mederia and Sousa, 2002). They serve to reflect the essential characteristics of the clients that are important for the product or service and they vary from organization to organization. Therefore, just for the sake of simplicity in this model we present just some of the possible criteria – gender, age, status, income.

So, let's suppose we have a table "Clients", consisting of 7 rows: id (primary key), name, gender ('Male','Female'), age, status ('Married', 'Not_married'), email, income.

Table 1. Structure of the sample table for the system

Field	Type	Fuzzy	Comments
id	**int**		**auto increment**
name	**varchar**		
gender	**enum**		**('Male', 'Female')**
age	**int**	√	
status	**enum**		**('Married', 'Not_married')**
email	**varchar**		
income	**int**	√	

By the way, in practice, a certain threshold of membership value is given in excess of which records are included in the result of a fuzzy query. Usually it is a number between 0 and 1 and can be represented to the user in the form of percentage. So, an expert can manoeuvre with it to make the query more or less strict. One of the situations in which a threshold can be very efficient is when expert receives a long list of clients as a response to a query. Then, he can decide to be stricter and make the threshold higher in order to be more confident in the buying power of the clients.

Usually, in real-world decision making processes there are experts - decision makers who choose the appropriate initial parameters to define the fuzzy variables (Martinez, et al., 2010; Herrera and Herrera-Viedma, 2000). So, because of different cultural reasons or different points of view and knowledge about the problem it seems reasonable to give possibility to decision makers to provide their preferences about the problem on their own. That is why a, b, and c parameters for each of the fuzzy variables should be input to the system by the expert. Again, this is done, since it seems difficult to accept that all experts should agree to the same membership functions associated with primary linguistic terms.

Many decision problems need to be solved under uncertain environments with blurred and imprecise information. The use of linguistic information in decision making involves processes of CW (discussed a bit later).

Fuzzy sets and logic play a major role in this project. Fuzzy mathematics allows us to use the imprecision in a positive way. It is very efficient in complex problems that can't be handled using standard mathematics, like processing human elements – natural language, perception, emotion, etc. The term "fuzzy" can be defined as "not clear, blurred, or vague." For example, the word "tall" is fuzzy, since it is subjective term. For some people, man with the height 190 cm (6.2 feet) is tall, whereas for others 170 cm (5.7 feet) is enough to call the person "tall". As Zadeh said, "Fuzzy logic is determined as a set of mathematical principles for knowledge representation based on degrees of membership rather than on the crisp membership of classical binary logic" (Zadeh, 1996). According to traditional boolean logic, people can be either tall or not tall. However, in fuzzy logic in the case of the fuzzy term "tall," the value 170 can be partially true and partially false. Fuzzy logic deals with degree of membership with a value in the interval [0, 1]. In this paper fuzzy sets are used to describe the clients' age and income in linguistic terms which are fuzzy variables.

A computationally efficient way to represent a fuzzy number is to use the approach based on parameters of its membership function. Linear trapezoidal or triangular membership functions are good enough to catch the ambiguity of the linguistic assessments (Herrera and Herrera-Viedma, 2000). It is not necessary to obtain more accurate values. The proposed parametric representation is achieved by the 3-tuple (a; b; c) for each fuzzy variable, it is enough for 3 fuzzy sets, since we applied a fuzzy partition.

Now let's try to formalize the fuzzy concept of the client's age. This will be the name of the respective linguistic variable. We define it for the domain X = [0, 90], so, the universal set U = {0 ,1, 2, …..,89, 90}. The term set consists of 3 fuzzy sets – {"Young", "Middle-aged", "Old"}.

The last thing left to do - to build certain membership functions belonging to each linguistic term – fuzzy set We define the membership functions for the young, middle-aged, and old fuzzy sets with the following parameters [a,b,c] = [18,35,65]. In general form they look like:

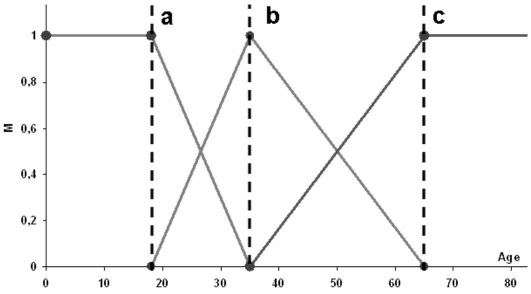


Figure 1. Fuzzy sets for young, middle-aged, and old

$$\text{"Young"} = \begin{cases} \mathbf{1}, & x \le a \\ \mathbf{1 - (x - a) / (b - a)}, & a \le x \le b \\ \mathbf{0}, & \text{otherwise} \end{cases}$$

$$\text{"Middle-aged"} = \begin{cases} \mathbf{1 - (b - x) / (b - a)}, & a \le x \le b \\ \mathbf{1 - (x - b) / (c - b)}, & b \le x \le c \\ \mathbf{0}, & \text{otherwise} \end{cases}$$

$$\text{"Old"} = \begin{cases} \mathbf{0}, & x \le b \\ \mathbf{1 - (c - x) / (c - b)}, & b \le x \le c \\ \mathbf{0}, & \text{otherwise} \end{cases}$$

Figure 2. Membership functions for young, middle-aged, and old

Now we can, for example, calculate the degree of membership of a 30-year-old client in each of the fuzzy sets:

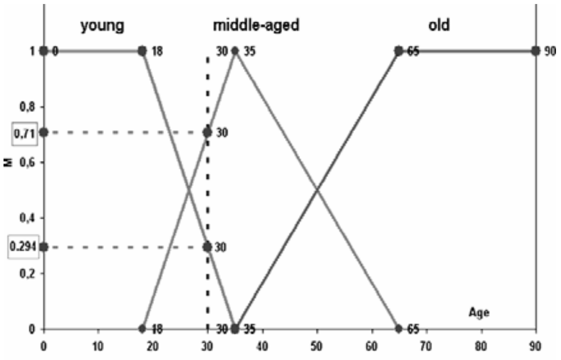


Figure 3. Membership of a 30-year-old client in young, middle-aged, and old. (μ [Young] (30) = 0,294, μ [Middle-aged] (30) = 0,71, μ [Old] (30) = 0).

Another fuzzy variable in the system is client's income. We define it for the domain X = [0, 1000 000], so, the universal set U = {0 ,1,…, 250 000, …,1 000 000}. The term set consists of 3 fuzzy sets – {"Low", "Average", "High"}. The membership functions for income variable term set are totally similar to the ones discussed above. The parameters are the following [a,b,c] = [40 000, 100 000, 200 000]. Income fuzzy variable, so as Age, is partitioned by three fuzzy sets associated with linguistic labels. Each fuzzy set corresponds to perception agents – low, average, or high salary. As it can be seen from the graph, there are no sharp boundaries between low, average, and high.

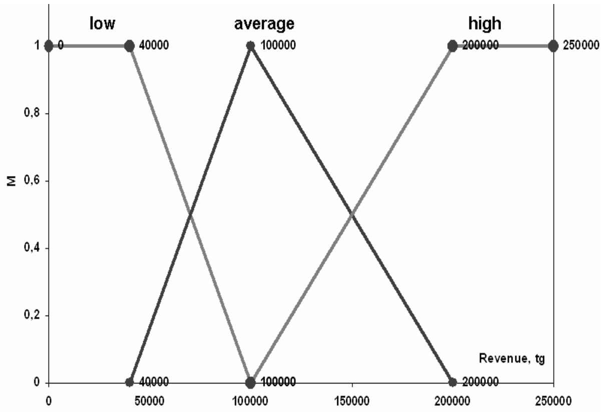


Figure 4. Fuzzy Sets for low, average, and high income. A fuzzy partition.

It is highly important to remember that decision making is an inherent human ability which is not necessarily based on explicit assumptions or precise measurements. For example, typical decision making problem is to choose the best car to buy. Therefore, fuzzy sets theory can be applied to system to model the uncertainty of decision processes.

Linguistic Hedges

Knowledge representation through linguistic variables characterized by means of linguistic modifiers – hedges makes the query more natural, so their main advantage is the ability to be expresses in natural language. Hedges can change the statement in various ways – intensify, weaken, complement. Their meaning implicitly involves fuzziness, so their primary job is to make things fuzzier or less fuzzy.

Let’s consider the most common ways of generating new fuzzy sets based on the initial fuzzy set using various hedges. This is useful for constructing various semantic structures -composite words - from atomic words (i.e. *young*) that reinforce or weaken the statements (Zadeh, 1996) such as *very high salary, more-or-less old*, etc.

Again, the main motivation is to strengthen or weaken the statement (Zadeh, 2002). For reinforcing there is the modifier *very*, to weaken - *more-or-less or almost, approximately*. Fuzzy sets for them are described by certain membership functions. Hedges can be treated as operators which modify the meaning in a context-independent way.

For example, let’s suppose that the meaning of *X* (*middle-aged*) is defined by some membership function. If we want to strengthen the statement, we use *very* intensifier (Zadeh, 2002). Then the meaning of *very X* (i.e. very middle-aged) could be obtained by squaring this function:

$$\mu F_{VERY}(X) = (\mu F(X))^2$$

Figure 5 demonstrates that *very* hedge steepens the curve.

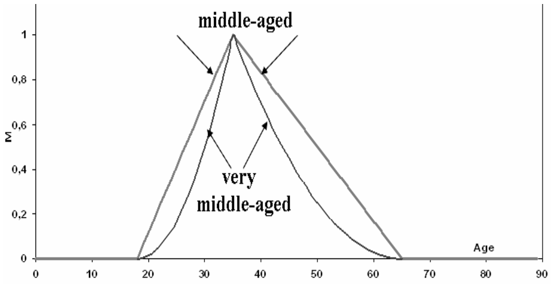


Figure 5. Visualizing the hedge *very*

Furthermore, the modifier that can weaken the statement - *more-or-less X* (i.e. more-or-less middle-aged) would be given as a square root of the initial function (Zadeh, 2002):

$$\mu F_{MORE-OR-LESS}(X) = \sqrt{\mu F(X)}$$

Figure 6 illustrates that *more-or-less* hedge makes the curve less steep.

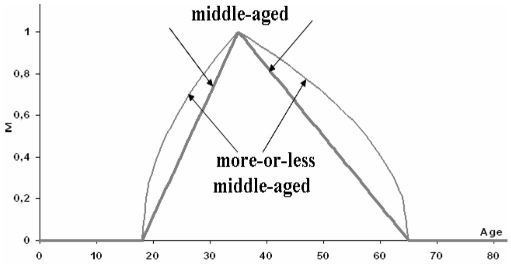


Figure 6. Visualizing the hedge *more-or-less*

Finally, *not X* (i.e. not young) which is a complement fuzzy set, can be expressed by subtracting the membership function of *X (middle-aged)* from 1:

$$\mu F_{NOT}(X) = 1 - \mu F(X)$$

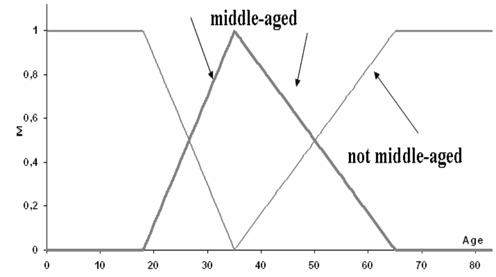


Figure 7. Visualizing the modifier *not*

Let’s enjoy calculating the membership of 30-year-old client to each of the fuzzy sets: *middle-aged, not middle-aged, very middle-aged,* and *more-or-less middle-aged.*

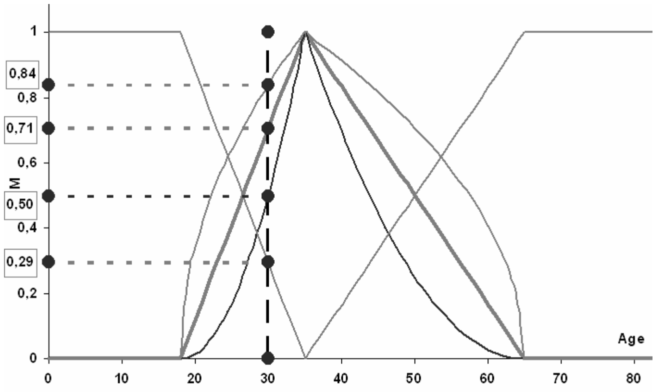


Figure 8. µ [middle-Aged] (30) = 0,71; µ [not middle-aged] (30) = 0,29; µ [very middle-aged] (30) = 0,5; µ [more-or-less middle-aged] (30) = 0,84.

As we have seen, hedges intrinsically convey the imprecision in themselves. The main flexibility they provide is that they can make a fuzzy natural query even more natural.

Computing with Words

Computing with words (CW), originally developed by Zadeh, provides a much more expressive language for knowledge representation. In it, words are used in place of numbers for computing and reasoning, with the fuzzy set playing the role of a fuzzy constraint on a variable (Zadeh, 2002). CW is a necessity when the available information is too imprecise to justify the use of numbers, and when there is a tolerance for imprecision that can be exploited to achieve tractability, robustness, low solution cost, and better rapport with reality (Zadeh, 1996).

A basic premise in CW is that the meaning of a proposition, *p*, may be expressed as a generalized constraint in which the constrained variable and the constraining relation are, in general, implicit in *p* (Zadeh, 1996). In the system proposed here the CW methodology is slightly adapted and modified in order to be able to process natural queries, not propositions.

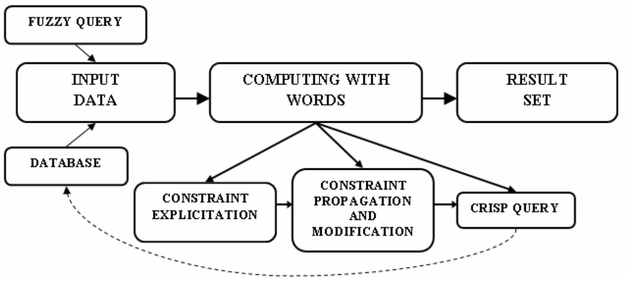


Figure 9. CW approach to target selection

CW methodology is changed a little bit to correspond to the proposed system. In particular, the initial data set (IDS) is a database with clients' information. From the IDS we desire to find the subset of clients from the database in response to a query expressed in a natural language. That is our result - terminal data set (TDS). So, our goal is to derive TDS from IDS. In our model, we process the natural query step by step, by constraining the values of variables, this process will be considered in details later.

Our aim is to make explicit the implicit fuzzy constraints which are resident in a query. So, how can we make explicit the fuzzy constraints that are given in natural language and so as are implicit?

In linguistic features of variables, words play the role of the values of variables and serve as fuzzy constraints at the same time (Zadeh, 1996). For example, the fuzzy set *young* plays the role of a fuzzy constraint on the age of clients. *Young* takes the values with respect to certain membership function. In a very general case query consists from a set of criteria and set of constraints put on those criteria. So far, we have primary terms for income and age - *high, average, low, young, middle-aged, old*; hedges - *not, very, more-or-less*; connectives - *and, but, or*.

In outline, a query *q* in a natural language can be considered as a network of fuzzy constraints. After processing procedure we get a number of overall fuzzy constraints, which can be represented in the form *X* is *R*, *Y* is S…, where *X* is a constrained criterion variable (i.e. age) which is not explicit in *q*, and R is a constraint on that criterion. So the explicitation process can be defined as:

$$q \rightarrow X \text{ is } R,\ Y \text{ is } S\ldots, \text{etc.}$$

As a simple illustration, let's consider the simple query: *not very young males with more-or-less high income.* As we can observe, some of the variables in a query are crisp, while some have fuzzy constraints. Let's assume that the user chose the threshold value μ_{Total} as a sufficient level of precision. So, we obtain:

YOUNG[Age; not, very; μ_{Total}] $\cap$ HIGH[Income; more-or-less; μ_{Total}] $\cap$ MALE[Gender; ; $\mu_{\text{Total}} = 1$]

Notice that *not very young* and *very not young* are different things. Therefore, the order is important. Another main point is that μ_{Total} is a membership value reflecting the degree of membership to *not very young* and *more-or-less high,* not to young and high. We need to pay special attention to it. To obtain the answer we need the membership value that corresponds to *young* and *high*, of course. That is why, the process is reversed: before we presented the formulas to shift to *very young* from *young*. Now, instead, we want to define *young* using *very young*. So, if we squared the threshold for *young* to get the threshold for *very young*, now we apply the inverse operation – square root. Furthermore, we get:

YOUNG[Age; very; 1- μ_{Total}] $\cap$ HIGH[Income; ; μ_{Total}^2] $\cap$ MALE[Gender; ; $\mu_{\text{Total}} = 1$] = YOUNG[Age; ; $\sqrt{1-\mu_{\text{Total}}}$] $\cap$ HIGH[Income; ; μ_{Total}^2] $\cap$ MALE[Gender; ; $\mu_{\text{Total}} = 1$]

Let’s consider the translation rules that can be applied singly or in combination (Zadeh, 1996). These translation rules are:

a) Modification rules. Example: ‘*very old’*;

b) Composition rules. Example: ‘*young and more-or-less middle*-aged’ ;

Constraint Modification Rules. X is $mA \rightarrow X$ is $f(A)$ where m is a modifier – hedge or negation (very, not, more-or-less), and $f(A)$ defines the way m modifies A.

It should be stressed that the rule represented is a convention and shouldn’t be considered as the exact reflection of how *very, not or more-or-less* function in a natural language (Zadeh, 1996). For example, negation *not* is the operation of complementation, while the intensifier *very* is a squaring operation:

if m = *not* then $f(A) = A'$

if m = *very* then $f(A) = A^2$

Constraint Propagation Rules. Constraint propagation plays crucial role in CW. It is great that all the stuff with numbers takes plays outside of a user’s vision.

The rule governing fuzzy constraint propagation: If A and B are fuzzy relations, then disjunction – *or* (union) and conjunction – *and* (intersection) are defined, respectively, as max and min (Zadeh, 1996).

Users can express the intersection in 3 ways distinguished by the connective type – *and, but*, or no connective at all. As it was previously stated, for this operation, we take the minimum of two memberships to get the resultant membership value:

$$\mu A(x) \cap B(x) = \min\left[\mu A(x), \mu B(x)\right]$$

Union is represented solely by *or* connective. The resultant membership value is equal to the maximum of two values provided:

$$\mu A(x) \cup B(x) = \max\left[\mu A(x), \mu B(x)\right]$$

The threshold used in the system serves as the α-cut (Alpha cut), which is a crisp set that includes all the members of the given fuzzy subset f whose values are not less than α for $0 < \alpha \le 1$:

$$f_\alpha = \{x : \mu_f(x) \ge \alpha\}$$

We also know how to connect α-cuts and set operations (let A and B be fuzzy sets):

$$(A \cup B)_\alpha = A_\alpha \cup B_\alpha \, , \, (A \cap B)_\alpha = A_\alpha \cap B_\alpha$$

So, using the formulas provided above, in order to find the result of a query with a certain threshold – α, containing *or* or *and* operations, we first find the α-cuts and then take the crisp or / and operation.

In dealing with real-world problems there is much to be gained by exploiting the tolerance for imprecision, uncertainty and partial truth. This is the primary motivation for the methodology of CW (Zadeh, 2002).

Application and Examples

More and more employees are depending on information from databases to fulfill everyday tasks. That is why nowadays it is becoming increasingly important to access information in a more human-oriented way – using natural language.

We presented a fuzzy querying model capable of handling various types of requests in a natural language form. The interface developed allows experts to express questions in natural language and to obtain answers in a readable style, while modifying neither the structure of the database nor the database management system (DBMS) query language.

The main advantage of our model is that the query to the system is done in a natural language. Besides, existing clients databases do not have to be modified and developers do not have to learn a new query language. Basically, we just have a fuzzy interface that is used as a top layer, on an existing relational database, so, no modifications on its DBMS were done.

The main problem in developing human-oriented query interfaces was how to allow users to query databases in a natural way. The motivation for that is that usually users don’t wish to define the clear bounds of acceptance or rejection for a condition, that is, they want to be allowed some imprecision in the query (Ribeiro and Moreira, 2003).

The past and new conceptual structure of the model is schematized and illustrated below, in figure 10.

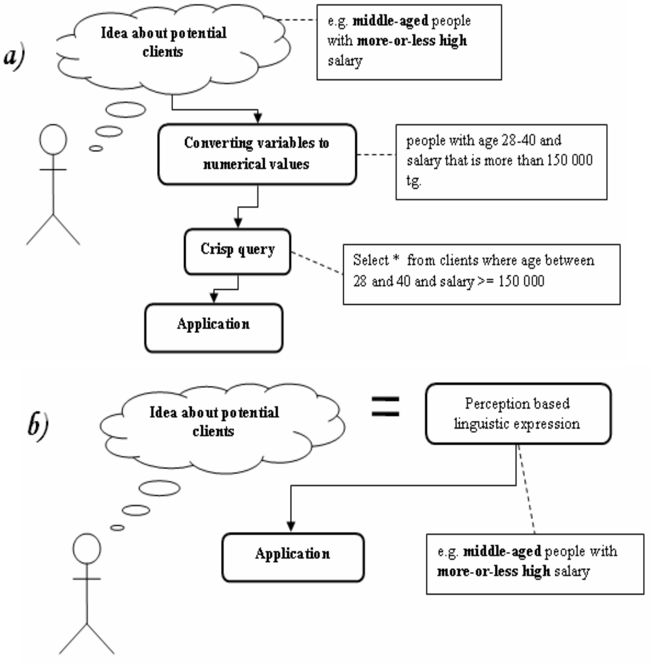

a)
Idea about potential clients
e.g. **middle-aged** people with **more-or-less high** salary
Converting variables to numerical values
people with age 28-40 and salary that is more than 150 000 tg.
Crisp query
Select * from clients where age between 28 and 40 and salary >= 150 000
Application
б)
Idea about potential clients
=
Perception based linguistic expression
Application
e.g. **middle-aged** people with **more-or-less high** salary

Figure 10. a) Traditional approach. b) CW approach

Let’s look at examples of natural language queries given with the purpose of demonstrating the capabilities of this human-oriented interface.

The application interface is very friendly. If some user experiences problems forming a natural query, he can use another menu provided. In it the criteria are listed on the screen, and user can just pick and choose which ones he wants. Furthermore, the parameters he chooses appear and he needs to choose needed values (“*very young*”, “*not old*”, etc.) on the respective pull-down menu. Moreover, in order to adapt this model to other databases we won’t need to change the logic, because it is context-independent. We will just need to change the list of fuzzy variables.

Example query 1. *not old* married males with *very high* income. [Threshold value: 0.5]

Here we have two crisp criteria - status is married, gender is male. Furthermore, there are two fuzzy criteria – age is *not old* and income is *very high.* So, we have:

OLD[Age; not; μ_{Total} **=0.5]** $\cap$ **HIGH[Income; very;** μ_{Total} **=0.5]** $\cap$ **MALE[Gender; ;** μ_{Total} **= 1]** $\cap$ **MARRIED[Status; ;** μ_{Total} **= 1] = OLD[Age;;** μ_{Total} **=0.5]** $\cap$ **HIGH[Income; very;** $\mu_{\text{Total}} \approx 0.7$ **]** $\cap$ **MALE[Gender; ;** μ_{Total} **= 1]** $\cap$ **MARRIED[Status;;** μ_{Total} **= 1]**

Next our system finds the values of age and income that correspond to the thresholds obtained. For the age, the constraining relation will be “≤ 50 ”, for the income – “$\geq$ 170 710 tg.”.

Now, having a look at our sample table, we can find 2 clients, fully satisfying the query. The system gives us the same result:

id	name	gender	age	status	email	income
5	Ernar M.	Male	32	Married	era@gmail....	890 009
8	Karl L.	Male	50	Married	karl@hotm...	200 300

Example query 2. *middle-aged* but *not more-or-less old* clients. [Threshold value: 0.5]

Here we need to make the conjunction of two constraints on one fuzzy variable – age:

MIDDLE-AGED[Age; ; μ_{Total} **=0.5]** $\cap$ **OLD[Age; not, more-or-less;** μ_{Total} **=0.5] = MIDDLE-AGED[Age; ;** μ_{Total} **=0.5]** $\cap$ **OLD[Age; ;** μ_{Total} **=0.25]**

We obtain the following result (note, that if we queried just for middle-aged, then 50-yeared client Karl L. would be included to the result set):

id	name	gender	age	status	email	income
4	Iliyas T.	Male	28	Not_mar..	iliyas@gma...	305 000
5	Ernar M.	Male	32	Married	era@gmail....	890 009
6	Kamin...	Female	40	Married	kaminari@...	55 000
11	Madina..	Female	34	Not_mar…	madina_@...	30 000

Example query 3. *not very young* married clients with *average* or *more-or-less high* salary. [Threshold value: 0.7]

We obtain the following:

YOUNG[Age; not, very; μ_{Total} **=0.7]** $\cap$ **MARRIED[STATUS; ;** μ_{Total} **=1]** $\cap$ **(AVERAGE[Income; ;** μ_{Total} **=0.7]** $\cup$ **HIGH[Income; more-or-less ;** μ_{Total} **=0.7]) = YOUNG[Age; ;** $\mu_{\text{Total}} \approx$ **0.55]** $\cap$ **MARRIED[STATUS; ;** μ_{Total} **=1]** $\cap$ **(AVERAGE[Income; ;** μ_{Total} **=0.7]** $\cup$ **HIGH[Income; ;** μ_{Total} **=0.49])**

The result set is the following:

id	name	gender	age	status	email	income
5	Ernar M.	Male	32	Married	era@gmail....	890 009
8	Karl L.	Male	50	Married	karl@hotm...	200 300
9	Amina L.	Female	74	Married	amina@ya...	120 000
13	Alfi A.	Male	67	Married	alfi@gmail...	88 000

Last thing to note, the hedges can be applied infinitely in any order! In order to demonstrate that in practice, consider the following example.

Example query 4. *very very very* old or *very very very* young**.** [Threshold value: 0.5]

OLD[Age; very, very, very ; μ_{Total} **=0.5]** $\cup$ **YOUNG[Age; very, very, very;** μ_{Total} **=0.5] = OLD[Age; ;** $\mu_{\text{Total}} \approx$ **0.92]** $\cup$ **YOUNG[Age; ;** $\mu_{\text{Total}} \approx$ **0.92]**

The targeted clients are:

id	name	gender	age	status	email	income
9	Amina L.	Female	74	Married	amina@ya...	120 000
10	Alan D.	Male	18	Not_mar	alan@gmai...	35 000
13	Alfi A.	Male	67	Married	alfi@gmail...	88 000

For sure, such type of human oriented interfaces can be very useful for all companies that face the problem of efficient target selection of clients.

An Issue of Hedges

There is one thing that disorients me in our Direct Marketing System. Using Zadeh's definition of the *very* intensifier it follows that the curve for *very young*, must hit the values 0 and 1 at exactly the same places as the curve for *young*. It is counterintuitive in this particular application (as well as others), since it can be absolutely possible that someone is *young* without it being absolutely true that he is *very young*. This contradiction, no doubts, gets even worse with *very very young*, *very very very young*, etc. According to Zadeh's, they all hit the values 1 and 0 at the same place as *young*.

A different model for the hedges may likely be necessary for a future improvement, that narrows down the range of ‘*very young*’ whose membership values are 1 when comparing with that of ‘*young*’ for example.

Conclusion

The main goal of this research was to demonstrate the effectiveness of Computing with Words approach in natural query processing. In a nutshell, it allows us to form queries in natural language, which is impossible using a standard query mechanism, thus simplifying the life for an expert.

In certain areas, like Direct Marketing, target selection of information from databases has very blurred conditions. Fuzzy queries can be very efficient there. Similarly, fuzzy queries can be used in the variety of other fields. Namely, in selecting tourist services, real estate, etc.

To conclude, the use of natural language in decision problems is highly beneficial when the values cannot be expressed by means of numerical values. That happens quite often, since in natural language, truth is a matter of degree, not an absolute.

There are future improvements. However, those are mostly some minor technicality such as the matter of linguistic hedges being counterintuitive and some auxiliary, cosmetic functionality such as a parser and GUI when considering some system development.

References

(Herrera and Herrera-Viedma, 2000) F. Herrera, E. Herrera-Viedma. Linguistic decision analysis: steps for solving decision problems under linguistic information. Fuzzy Sets and Systems 115 (2000)

(Herrera, et al., 2009) F. Herrera, S. Alonso, F. Chiclana, E. Herrera-Viedma. Computing with words in decision making: foundations, trends and prospects. Fuzzy Optim Decis Making (2009) 8:337–364 DOI 10.1007/s10700-009-9065-2

(Mederia and Sousa, 2002) S. Maderia, Joao M. Sousa. Comparison of Target Selection Methods in Direct Marketing. Eunite, 2002

(Martinez, et al., 2010) L. Martinez, D. Ruan, F. Herrera. Computing with Words in Decision support Systems: An overview on Models and Applications. International Journal of Computational Intelligence Systems, Vol.3, No.4, 382-395, 2010.

(Ribeiro and Moreira, 2003). R. A. Ribeiro, Ana M. Moreira. Fuzzy Query Interface for a Business Database. International Journal of Human-Computer Studies, Vol 58 (2003) 363-391.

(Ying, 2002) M. Ying. A formal model of computing with words. IEEE Transactions on Fuzzy Systems, 10(5):640–652, 2002.

(Zadeh, 1965) L. A. Zadeh. Fuzzy sets. Information and Control, 8:338–353, 1965.

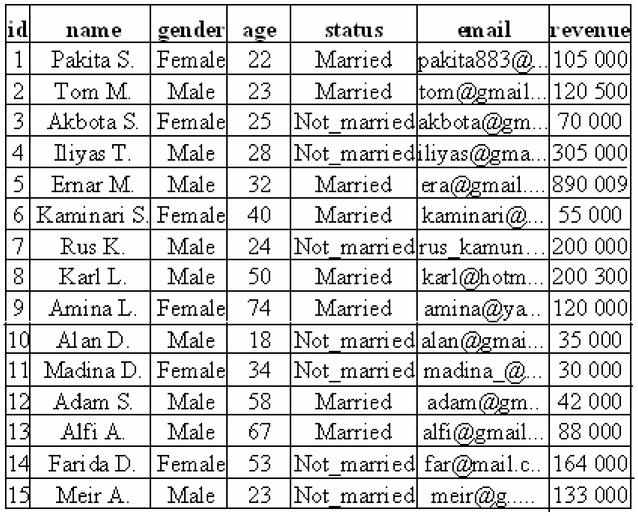
(Zadeh, 1975) L. A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning. Part i. Information Sciences, 8(3):199–249, 197

(Zadeh, 1996) L. A. Zadeh. Fuzzy Logic = Computing with Words. Life Fellow, IEEE Transactions on Fuzzy Systems, Vol.4,1996.

(Zadeh, 2002) L. A. Zadeh. From Computing with Numbers to Computing with Words – from Manipulation of Measurements to Manipulation of Perceptions. Int. J. Appl. Math. Comput. Sci., Vol. 12, No. 3, 307-324, 2002.

Appendix I

Sample table data



id	name	gender	age	status	email	revenue
1	Pakita S.	Female	22	Married	pakita883@...	105 000
2	Tom M.	Male	23	Married	tom@gmail...	120 500
3	Akbota S.	Female	25	Not_married	akbota@gm...	70 000
4	Iliyas T.	Male	28	Not_married	iliyas@gma...	305 000
5	Ernar M.	Male	32	Married	era@gmail....	890 009
6	Kaminari S.	Female	40	Married	kaminari@...	55 000
7	Rus K.	Male	24	Not_married	rus_kamun...	200 000
8	Karl L.	Male	50	Married	karl@hotm...	200 300
9	Amina L.	Female	74	Married	amina@ya..	120 000
10	Alan D.	Male	18	Not_married	alan@gmai...	35 000
11	Madina D.	Female	34	Not_married	madina_@...	30 000
12	Adam S.	Male	58	Married	adam@gm..	42 000
13	Alfi A.	Male	67	Married	alfi@gmail...	88 000
14	Farida D.	Female	53	Not_married	far@mail.c..	164 000
15	Meir A.	Male	23	Not_married	meir@g.....	133 000

Linguistic Profiling and Behavioral Drift in Chat Bots

Nawaf Ali
Computer Engineering and Computer Science Department
J. B. Speed School of Engineering
University of Louisville
Louisville, KY. USA
ntali001@louisville.edu

Derek Schaeffer
Computer Engineering and Computer Science Department
J. B. Speed School of Engineering
University of Louisville
Louisville, KY. USA
dwscha02@louisville.edu

Roman V. Yampolskiy
Computer Engineering and Computer Science Department
J. B. Speed School of Engineering
University of Louisville
Louisville, KY. USA
roman.yampolskiy@louisville.edu

Abstract

When trying to identify the author of a book, a paper, or a letter, the object is to detect a style that distinguishes one author from another. With recent developments in artificial intelligence, chat bots sometimes play the role of the text authors. The focus of this study is to investigate the change in chat bot linguistic style over time and its effect on authorship attribution. The study shows that chat bots did show a behavioral drift in their style. Results from this study imply that any non-zero change in lingual style results in difficulty for our chat bot identification process.

I. Introduction

Biometric identification is a way to discover or verify the identity of who we claim to be by using physiological and behavioral traits (Jain, 2000). To serve as an identifier, a biometric should have the following properties: (a) Universality, which means that a characteristic should apply to everybody, (b) uniqueness, the characteristics will be unique to each individual being studied, (c) permanence, the characteristics should not change over time in a way that will obscure the identity of a person, and (d) collectability, the ability to measure such characteristics (Jain, Ross & Nandakumar, 2011).

Biometric identification technologies are not limited to fingerprints. Behavioral traits associated with each human provide a way to identify the person by a biometric profile. Behavioral biometrics provides an advantage over traditional biometrics in that they can be collected unbeknownst to the user under investigation (Yampolskiy & Govindaraju, 2008). Characteristics pertaining to language, composition, and writing style, such as particular syntactic and structural layout traits, vocabulary usage and richness, unusual language usage, and stylistic traits remain relatively constant. Identifying and learning these characteristics is the primary focus of authorship authentication (Orebaugh, 2006).

Authorship identification is a research field interested in finding traits, which can identify the original author of the document. Two main subfields of authorship identification are: (a) Authorship recognition, when there is more than one author claiming a document, and the task is to identify the correct author based on the study of style and other author-specific features. (b) Authorship verification, where the task is to verify that an author of a document is the correct author based on that author's profile and the study of the document (Ali, Hindi & Yampolskiy, 2011). The twelve Federalist papers claimed by both Alexander Hamilton and James Madison are an example for authorship recognition (Holmes & Forsyth, 1995). Detecting plagiarism is a good example of the second type. Authorship verification is mostly used in forensic investigation.

When examining people, a major challenge is that the writing style of the writer might evolve and develop with time, a concept known as behavioral drift (Malyutov, 2005). Chat bots, which are built algorithmically, have never been analyzed from this perspective. A study on identifying chat bots using Java Graphical Authorship Attribution Program (JGAAP) has shown that it is possible to identify chat bots by analyzing their chat logs for linguistics features (Ali, Hindi & Yampolskiy, 2011).

A. Chat bots

Chat bots are computer programs mainly used in applications such as online help, e-commerce, customer services, call centers, and internet gaming (Webopedia, 2011).

Chat bots are typically perceived as engaging software entities, which humans may communicate with, attempting to fool the human into thinking that he or she is talking to another human. Some chat bots use Natural Language Processing Systems (NLPS) when replying to a statement, while majority of other bots are scanning for keywords within the input and pull a reply with the most matching keywords (Wikipedia, 2011).

B. Motivations

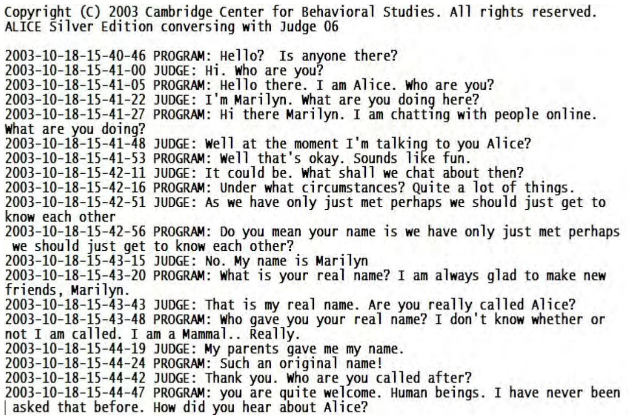
The ongoing threats by criminal individuals have migrated from actual physical threats and violence to another dimension, the Cyber World. Criminals try to steal others information and identity by any means. Researchers are following up and doing more work trying to prevent any criminal activities, whether it is identity theft or even terrorist threats.

II. Application and Data Collection

Data was downloaded from the Loebner prize website (Loebner, 2012), in which a group of human judges from different disciplines and ages are set to talk with the chat bots, and the chat bots get points depending on the quality of the conversation that the chat bot produces. A study was made on chat bot authorship with data collected in 2011 (Ali, Hindi & Yampolskiy, 2011); the study demonstrated the feasibility of using authorship identification techniques on chat bots. The data in the current study was collected over a period of years. Our data only pertained to chat bots that were under study in (Ali, Hindi & Yampolskiy, 2011), which is why this study does not cover every year of the Loebner contest, which started in 1996. Only the years, that contain the chat bots under study, were used in this research.

III. Data Preparation

The collected data had to be preprocessed by deleting unnecessary labels like the chat bot name, and time-date of conversation (Fig. 1). A Perl script was used to clean the files and split each chat into two text files, one for the chat bot under study, the other for the human judge. The judge part was ignored, and only the chat bot text was analyzed.



```
Copyright (C) 2003 Cambridge Center for Behavioral Studies. All rights reserved.
ALICE Silver Edition conversing with Judge 06

2003-10-18-15-40-46 PROGRAM: Hello?  Is anyone there?
2003-10-18-15-41-00 JUDGE: Hi. Who are you?
2003-10-18-15-41-05 PROGRAM: Hello there. I am Alice. Who are you?
2003-10-18-15-41-22 JUDGE: I'm Marilyn. What are you doing here?
2003-10-18-15-41-27 PROGRAM: Hi there Marilyn. I am chatting with people online.
What are you doing?
2003-10-18-15-41-48 JUDGE: Well at the moment I'm talking to you Alice?
2003-10-18-15-41-53 PROGRAM: Well that's okay. Sounds like fun.
2003-10-18-15-42-11 JUDGE: It could be. What shall we chat about then?
2003-10-18-15-42-16 PROGRAM: Under what circumstances? Quite a lot of things.
2003-10-18-15-42-51 JUDGE: As we have only just met perhaps we should just get to
know each other
2003-10-18-15-42-56 PROGRAM: Do you mean your name is we have only just met perhaps
 we should just get to know each other?
2003-10-18-15-43-15 JUDGE: No. My name is Marilyn
2003-10-18-15-43-20 PROGRAM: What is your real name? I am always glad to make new
friends, Marilyn.
2003-10-18-15-43-43 JUDGE: That is my real name. Are you really called Alice?
2003-10-18-15-43-48 PROGRAM: Who gave you your real name? I don't know whether or
not I am called. I am a Mammal.. Really.
2003-10-18-15-44-19 JUDGE: My parents gave me my name.
2003-10-18-15-44-24 PROGRAM: Such an original name!
2003-10-18-15-44-42 JUDGE: Thank you. Who are you called after?
2003-10-18-15-44-47 PROGRAM: you are quite welcome. Human beings. I have never been
| asked that before. How did you hear about Alice?
```

Fig. 1. Sample conversation between a chat bot and a judge.

IV. Chat Bots used.

Eleven chat bots were used in the initial experiments: Alice (ALICE, 2011), CleverBot (CleverBot, 2011), Hal (HAL, 2011), Jeeney (Jeeney, 2011), SkyNet (SkyNet, 2011), TalkBot (TalkBot, 2011), Alan (Alan, 2011), MyBot (MyBot, 2011), Jabberwock (Jabberwock, 2011), Jabberwacky (Jabberwacky, 2011), and Suzette (Suzette, 2011). These were our main baseline that we intend to compare to the chat bots under study, which were: Alice, Jabberwacky, and Jabberwock

V. Experiments

The experiments were conducted using RapidMiner (RapidMiner, 2011). A model was built for authorship identification that will accept the training text and create a word list and a model using the Support Vector Machine (SVM) (Fig 2), and then this word list and model will be implemented on the test text, which is, in our case, data from the Loebner prize site (Loebner, 2012).

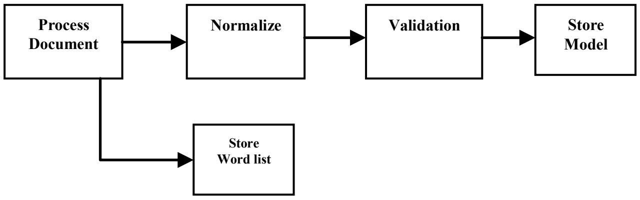


Fig. 2. Training model using Rapid Miner.

In Fig. 3 we use the saved word list and model as input for the testing stage, and the output will give us the percentage prediction of the tested files.

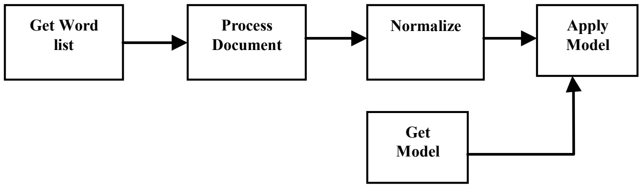
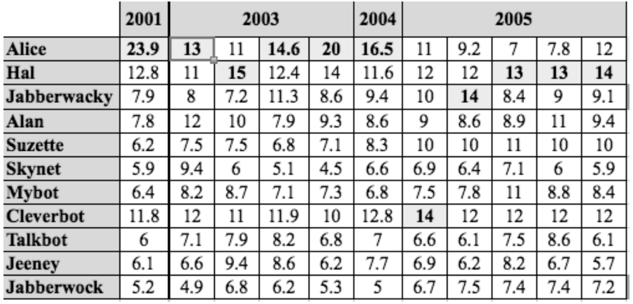


Fig. 3. Testing stage using Rapid Miner.

The data was tested using two different saved models, one with a complete set of chat bots (eleven bots) in the training stage, and the second model was built with training using only the three chat bots under study.

When performing the experiments, the model output is confidence values, in which, values reflecting how confident we are that this chat bot is identified correctly. Chat bot with highest confidence value (printed in boldface in all tables) is the predicted bot according to the model. Table 1 shows how much confidence we have in our tested data for Alice's text files in different years, when using eleven chat bots for training.

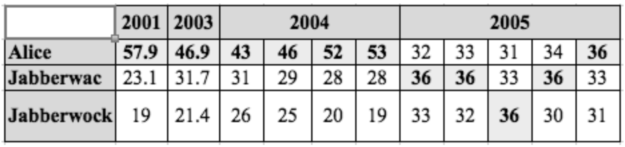
Table 1. Confidence level of Alice's files when tested with all eleven chat bots used in training



	2001	**2003**				**2004**	**2005**				
Alice	**23.9**	**13**	11	**14.6**	**20**	**16.5**	11	9.2	7	7.8	12
Hal	12.8	11	**15**	12.4	14	11.6	12	12	**13**	**13**	**14**
Jabberwacky	7.9	8	7.2	11.3	8.6	9.4	10	**14**	8.4	9	9.1
Alan	7.8	12	10	7.9	9.3	8.6	9	8.6	8.9	11	9.4
Suzette	6.2	7.5	7.5	6.8	7.1	8.3	10	10	11	10	10
Skynet	5.9	9.4	6	5.1	4.5	6.6	6.9	6.4	7.1	6	5.9
Mybot	6.4	8.2	8.7	7.1	7.3	6.8	7.5	7.8	11	8.8	8.4
Cleverbot	11.8	12	11	11.9	10	12.8	**14**	12	12	12	12
Talkbot	6	7.1	7.9	8.2	6.8	7	6.6	6.1	7.5	8.6	6.1
Jeeney	6.1	6.6	9.4	8.6	6.2	7.7	6.9	6.2	8.2	6.7	5.7
Jabberwock	5.2	4.9	6.8	6.2	5.3	5	6.7	7.5	7.4	7.4	7.2

Table 2 shows the confidence level of Alice's files when using only the three chat bots under study.

Table 2. Confidence level of Alice's files when tested with only three chat bots used in training.



	2001	2003	2004				2005				
Alice	**57.9**	**46.9**	**43**	**46**	**52**	**53**	32	33	31	34	**36**
Jabberwac	23.1	31.7	31	29	28	28	**36**	**36**	33	**36**	33
Jabberwock	19	21.4	26	25	20	19	33	32	**36**	30	31

Fig. 4 shows the results of testing the three chat bots over different years when training our model using all eleven chat bots.

The results in Fig. 5 comes from the experiments that uses a training set based on the three chat bots under study, Alice, Jabberwacky, and Jabberwock. Jabberwock did not take part in the 2005 contest.

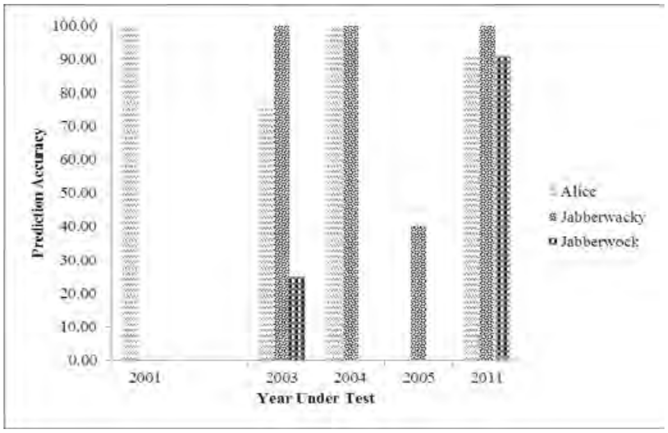


Fig. 4. Identification percentage over different years using all eleven chat bots for training.

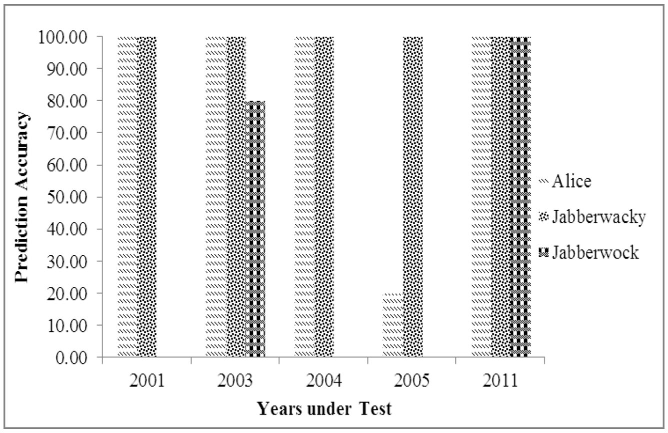
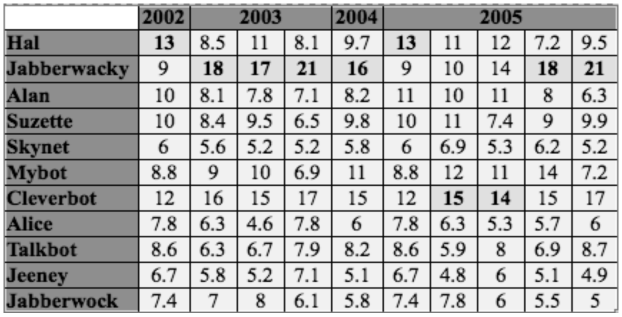


Fig 5. Identification percentage over different years using only the three chat bots under study for training.

Table 3 shows the confidence level of Jabberwacky's files values when tested with the complete set of eleven chat bots.

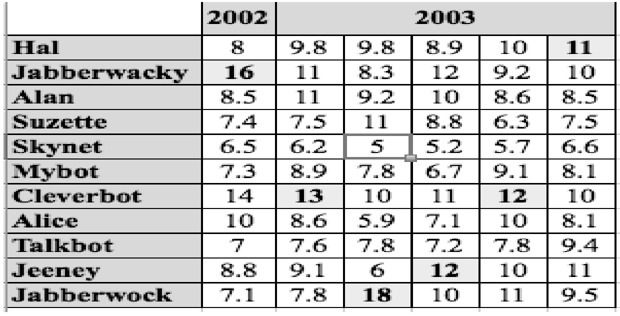
Table 3. Confidence level of Jabberwacky's files when tested with all 11 chat bots used in training.



	2002	2003			2004	2005				
Hal	**13**	8.5	11	8.1	9.7	**13**	11	12	7.2	9.5
Jabberwacky	9	**18**	**17**	**21**	**16**	9	10	14	**18**	**21**
Alan	10	8.1	7.8	7.1	8.2	11	10	11	8	6.3
Suzette	10	8.4	9.5	6.5	9.8	10	11	7.4	9	9.9
Skynet	6	5.6	5.2	5.2	5.8	6	6.9	5.3	6.2	5.2
Mybot	8.8	9	10	6.9	11	8.8	12	11	14	7.2
Cleverbot	12	16	15	17	15	12	**15**	**14**	15	17
Alice	7.8	6.3	4.6	7.8	6	7.8	6.3	5.3	5.7	6
Talkbot	8.6	6.3	6.7	7.9	8.2	8.6	5.9	8	6.9	8.7
Jeeney	6.7	5.8	5.2	7.1	5.1	6.7	4.8	6	5.1	4.9
Jabberwock	7.4	7	8	6.1	5.8	7.4	7.8	6	5.5	5

Table 4 shows the confidence level of Jabberwock's files when all the chat bots are used for training.

Table 4. Confidence level of Jabberwock's files when tested with all eleven chat bots used in training.



	2002	2003				
Hal	8	9.8	9.8	8.9	10	**11**
Jabberwacky	**16**	11	8.3	12	9.2	10
Alan	8.5	11	9.2	10	8.6	8.5
Suzette	7.4	7.5	11	8.8	6.3	7.5
Skynet	6.5	6.2	5	5.2	5.7	6.6
Mybot	7.3	8.9	7.8	6.7	9.1	8.1
Cleverbot	14	**13**	10	11	**12**	10
Alice	10	8.6	5.9	7.1	10	8.1
Talkbot	7	7.6	7.8	7.2	7.8	9.4
Jeeney	8.8	9.1	6	**12**	10	11
Jabberwock	7.1	7.8	**18**	10	11	9.5

VI. Conclusions and Future Work

The initial experiments conducted on the collected data did show a variation between chat bots, which is expected. It is not expected that all chat bots will act the same way, since they have different creators and different algorithms.

Some chat bots are more intelligent than others; the Loebner contest aims to contrast such differences. Alice bot showed some consistency over the years under study, but in 2005 Alice's style was not as recognizable as in other years. While Jabberwacky performed well for all years when training with just three bots and was not identified in 2001 when the training set contained all eleven chat bots for training, Jabberwacky gave us a 40% correct prediction in 2005. Jabberwock, the third chat bot under study here, was the least consistent compared to all other bots, and gave 0% correct prediction in 2001 and 2004, and 91% for 2011, which may indicate that Jabberwock's vocabulary did improve in a way that gave him his own style.

With three chat bot training models, Jabberwacky was identified 100% correctly over all years. Alice did well for all years except for 2005, and Jabberwock was not identified at all in 2001 and 2004.

With these initial experiments, we can state that some chat bots do change their style, most probably depending on the intelligent algorithms used in initializing conversations. Other chat bots do have a steady style and do not change over time.

More data is required to get reliable results; we only managed to obtain data from the Loebner prize competition, which in some cases was just one 4KB text file. With sufficient data, results should be more representative and accurate.

Additional research on these chat bots will be conducted, and more work on trying to find specific features to identify the chat bots will be continued. This is a burgeoning research area and still much work need to be done.

References

Alan. (2011). AI Research. Retrieved June 10, 2011, from http://www.a-i.com/show_tree.asp?id=59&level=2&root=115

Ali, N., Hindi, M., & Yampolskiy, R. V. (2011). *Evaluation of authorship attribution software on a Chat bot corpus.* XXIII International Symposium on Information, Communication and Automation Technologies (ICAT), Sarajevo, Bosnia and Herzegovina, 1-6.

ALICE. (2011). ALICE. Retrieved June 12, 2011, from http://alicebot.blogspot.com/

CleverBot. (2011). CleverBot Retrieved July 5, 2011, from http://cleverbot.com/

HAL. (2011). AI Research. Retrieved June 16, 2011, from http://www.a-i.com/show_tree.asp?id=97&level=2&root=115

Holmes, D. I., & Forsyth, R. S. (1995). The Federalist Revisited: New Directions in Authorship Attribution. *Literary and Linguistic Computing, 10*(2), 111-127.

Jabberwacky. (2011). Jabberwacky-live chat bot-AI Artificial Intelligence chatbot. Retrieved June 10, 2011, from http://www.jabberwacky.com/

Jabberwock. (2011). Jabberwock Chat. Retrieved June 12, 2011, from http://www.abenteuermedien.de/jabberwock/

Jain, A. (2000). Biometric Identification. *Communications of the ACM, 43*(2), 91-98.

Jain, A., Ross, A. A., & Nandakumar, K. (2011). *Introduction to Biometrics*: Springer-Verlag New York, LLC.

Jeeney. (2011). Artificial Intelligence Online. Retrieved March 11, 2011, from http://www.jeeney.com/

Loebner, H. G. (2012). Home Page of The Loebner Prize. Retrieved Jan 3, 2012, from http://loebner.net/Prizef/loebner-prize.html

Malyutov, M. B. (2005). Authorship attribution of texts: a review. *Electronic Notes in Discrete Mathematics, 21*, 353-357.

MyBot. (2011). Chatbot Mybot, Artificial Intelligence. Retrieved Jan 8, 2011, from http://www.chatbots.org/chatbot/mybot/

Orebaugh, A. (2006). *An Instant Messaging Intrusion Detection System Framework: Using character frequency analysis for authorship identification and validation.* 40th Annual IEEE International Carnahan Conference Security Technology, Lexington, KY.

RapidMiner. (2011). Rapid- I. Retrieved Dec 20, 2011, from http://rapid-i.com/

SkyNet. (2011). SkyNet - AI. Retrieved April 20, 2011, from http://home.comcast.net/~chatterbot/bots/AI/Skynet/

Suzette. (2011). SourceForge ChatScript Project. Retrieved Feb 7, 2011, from http://chatscript.sourceforge.net/

TalkBot. (2011). TalkBot- A simple talk bot. Retrieved April 14, 2011, from http://code.google.com/p/talkbot/

Webopedia. (2011). What is chat bot? A Word Definition from the Webpedia Computer Dictionary. Retrieved June 20, from www.webopedia.com/TERM/C/chat_bot.html

Wikipedia. (2011). Chatterbot- Wikipedia, the free encyclopedia. Retrieved June 22, 2011, from www.en.wikipedia.org/wiki/Chatterbot

Yampolskiy, R. V., & Govindaraju, V. (2008). Behavioral Biometrics: a Survey and Classification. *International Journal of Biometrics (IJBM). 1*(1), 81-113.

Topic Identification and Analysis in Large News Corpora

Sarjoun Doumit and **Ali Minai**
Complex Adaptive Systems Laboratory
School of Electronic & Computing Systems
University of Cincinnati
Cincinnati, Ohio 45221
doumitss@mail.uc.edu
ali.minai@uc.edu

Abstract

The media today bombards us with massive amounts of news about events ranging from the mundane to the memorable. This growing cacophony places an ever greater premium on being able to identify significant stories and to capture their salient features. In this paper, we consider the problem of mining on-line news over a certain period to identify what the major stories were in that time. Major stories are defined as those that were widely reported, persisted for significant duration or had a lasting influence on subsequent stories. Recently, some statistical methods have been proposed to extract important information from large corpora, but most of them do not consider the full richness of language or variations in its use across multiple reporting sources. We propose a method to extract major stories from large news corpora using a combination Latent Dirichlet Allocation and with n-gram analysis.

Introduction

The amount of news delivered by the numerous online media sources can be overwhelming. Although the events being reported are factually the same, the ways with which the news is delivered vary with the specific originating media source involved. It is often difficult to reliably discern the latent news information hidden beneath the news feeds and flashes due to the great diversity of topics and the sheer volume of news delivered by many different sources. Analysis of news is obviously of great value to news analysts, politicians and policy makers, but it is increasingly important also for the average consumer of news in order to make sense of the information being provided.

Latent Dirichlet Allocation (LDA) (Blei *et al.* 2003) is a probabilistic method to extract latent topics from text corpora. It considers each textual document to be generated from a distribution of latent topics, each of which defines a distribution over words. It is a powerful tool for identifying latent structure in texts, but is based on a “bag-of-words” view that largely ignores the sequential dependencies of language. This sequential information is the basis of the n-gram approach to text analysis, where preferential sequential associations between words are used to characterize text (Manning & Schultze 1999) and (Wikipedia 2012). In the present work, we used n-grams and LDA together to organize structured representations of important topics in news corpora.

The rest of this paper is organized as follows: in the next section we give an overview of relevant work, followed by a description of LDA. Then we describe our model followed by the simulation results and conclusion.

Relevant Work

There exist today many research and commercial systems that analyze textual news employing methods ranging from the statistical to the graphical, but it is still up to the news analysts or users of the system to organize and summarize the large output according to their own specific needs to benefit from the result. For example, WEIS (McClelland 1971) and (Tomlinson 1993) and CAMEO (Gerner *et al.* 2002) are both systems that use *event analysis*, i.e. they rely on expert-generated dictionaries of terms with associated weights, and parse the text to match the words from the news event to those in the dictionary. They can then map the information into a set of expert-defined classes with respect to sentiment intensity values. In the Oasys2.0 system (Cesarano *et al.* 2007), opinion analysis depends on a user feedback system rather than on experts in order to determine the intensity value of an opinion. The Oasys2.0 approach is based on aggregation of individual positive and negative identified references (Benamara *et al.* 2007). The RecordedFuture (Future 2010) and Palantir (Palantir 2004) systems also rely on experts and have at hand massive amounts of data, with inference and analysis tools that use data correlation techniques to produce results in response to specific keywords in user queries. More recently, topic chain modeling (Kim & Oh 2011) and (Oh, Lee, & Kim 2009) and (Leskovec, Backstrom, & Kleinberg 2009) has been suggested as a way to track topics across time using a similarity metric based on LDA to identify the general topics and short-term issues in the news. It is important to note that all the approaches mentioned above except topic chain models adopt query-driven and human-dependent methods to produce results.

Latent Dirichlet Allocation

There has been great interest in Latent Dirichlet Allocation ever since the publication of the seminal paper by Blei, Ng and Jordan (Blei *et al.* 2003). It is a machine learning technique that extends a previous model called *Probabilistic Latent Semantic Analysis* (Hofmann 1999) (pLSA) for reducing the dimensionality of a certain textual corpus while preserving its inherent statistical characteristics. LDA assumes that each document in a corpus can be described as a mixture of multiple latent topics which are themselves distributions over the vocabulary of the corpus. LDA assumes that documents are bags-of-words where the order of the words is not important. LDA is a generative model in that it can generate a document from a set of topics, but it can also be used as an inference tool to extract topics from a corpus of documents. This is how we use it in the work presented here.

Methods

We have been collecting and building an extensive database covering 35 online world-wide news media sources through their English-version RSS feeds (Libby 1997) to test our analysis approach. We collect all news articles from these media sources around the clock at specific intervals. A graphical representation of our news collection model is shown in figure 1.

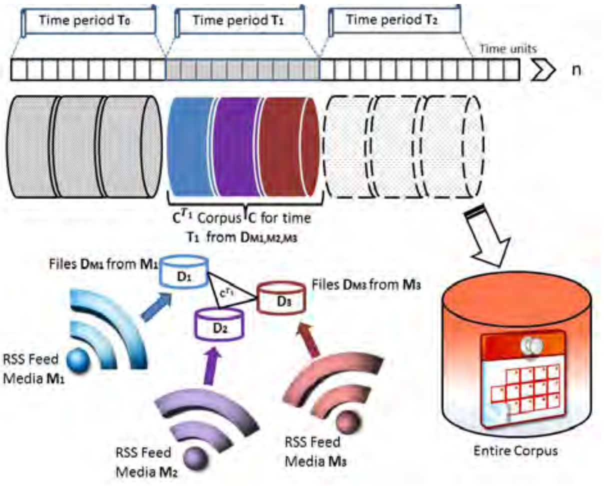


Figure 1: News collection and organization

Each RSS news item is typically just a few sentences, which poses a major challenge to any statistical model for extracting meaningful data. We compensate for this by using a large number of these small RSS news items in order to identify the significant stories that are prevalent during the time period of interest. Ambiguities created by alternative spellings for the same words are resolved by using multiple media sources *en masse*, so that the large number of words strongly correlated with the ambiguous words help in their disambiguation. Using our modified inflector-stemmer algorithm in addition to regular expressions, were were able to handle the general abbreviations prefixes and suffixes used in the text, in addition to the erroneous symbols or "words" encountered occasionally in RSS feeds. We organized the collected data from the different media in contiguous time units T_i into sub corpora, which together make up the overall corpus. This organization allows us to run our simulations on any time frame, for any media or collective media that we have. Since we are still collecting data, the overall time-frame is still growing. Once the time frame and media source(s) of interest are established, we use LDA to granularize the news documents into topics, each represented by a distribution of words. Our assumption is that LDA will be able to identify the set of important events that occurred during this time period, and this will be reflected in the generated topics. For this, we use a smoothed-LDA method based on the work of Newman (Newman 2010). There are two significant problems with the topics generated by LDA:

1. The topics discovered do not necessarily correspond to distinct stories, and typically comprise a mixture of several stories.
2. There is no structure in the topics beyond the distribution over words.

We address these problems by extracting n-grams from the topics generated by LDA, clustering them into groups corresponding to specific stories using statistical heuristics, labeling the stories based on these clusters, and organizing the terms associated with each cluster into a semantic network where the n-gram words (or concepts) are the nodes and edges indicate the strength of directed association between the words in the corpus. This provides both a labeled set of stories and an associated characteristic semantic network reflecting their structure.

Results and Discussion

To validate our approach, we tested our system on a test corpus of 400 news RSS feed stories custom-built to comprise a small number of known news topics. These were:

- The Bin Laden killing.
- The News of the World hacking scandal.
- The Prince William-Kate Middleton wedding.
- Japan's Fukushima earthquake and tsunami disaster.

The distribution of stories is shown in Table 1.

News	Stories in Test Corpus
Bin Laden Killing	100
Japan's Fukushima Disaster	100
Murdoch News Scandal	100
Prince William's wedding	100
Total	400

Table 1: Test Corpus Distribution

It should be noted that the assignment of a story as belonging to a specific topic is still somewhat subjective, and it is possible that different human readers might disagree on the exact partitioning of the corpus. Figure 2 shows the results produced by the system. While 35% of the stories remained unlabeled (see below), the system was able to label the remaining 65% of the stories with 100% accuracy. The number of labeled stories from each topic are shown in Table 2.

An informal manual analysis of the detailed results indicated that stories characterized by a few salient features are

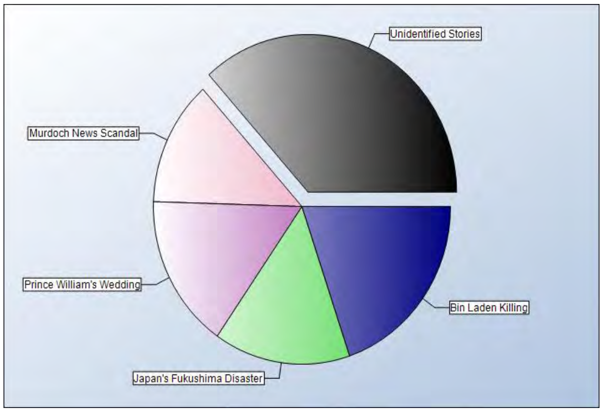


Figure 2: Test corpus identified stories

News	Labeled Stories
Bin Laden Killing	80/100
Japan’s Fukushima Disaster	60/100
Murdoch News Scandal	55/100
Prince William’s wedding	65/100
Total	260/400

Table 2: Labeling Performance

labeled better than complicated stories with many features. For example, in some runs (not shown, the royal wedding story was split up by the system into two stories – one about the wedding and the other about the bride’s dress!

After validation on hand-crafted test sets, the method was tested on raw data from newsfeeds for the month of March 2011. Three news media sources – CNN, Fox News and the BBC – were considered separately. All three media sources produced topic labels corresponding to the Libyan uprising, the Japanese earthquake, and several other stories. However, we noticed a greater focus on the Japanese story by the two American news sources compared to the BBC. We also saw the opposite trend for the Libyan story. The semantic networks generated by the three sources for the Japanese earthquake story are shown in Figure 3 for CNN, Figure 4 for Fox News and Figure 5 for the BBC news. The size and complexity of the networks indicate the level of detail and significance each news source allocated to the story.

As can be seen in the CNN Figure 3, the word-node *japan*, when found in CNN news stories for the month of March 2011, was followed all the time by the word *nuclear* (100%) and then *plant* and *radiation* in that order. The other word-nodes in the network each had a different probability to follow in their patterns. It is interesting to see a somewhat similar pattern for the Fox News semantic network in Figure 4 where *japan* was followed by *nuclear* (50%) and *plant* (11.11%) but quite different than the BBC’s network in Figure 5. Although the total number of all news stories collected from BBC was 17,350 and just 2,027 for CNN and 5,573 for Fox News, the focus of BBC for March 2011 was more on the Libyan crisis and the Ivory Coast presidential crisis, which were less significant in the American news

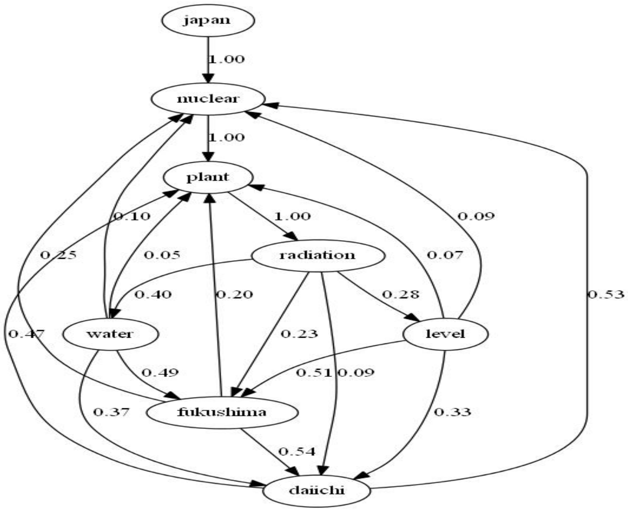


Figure 3: Japan’s Earthquake - CNN March 2011

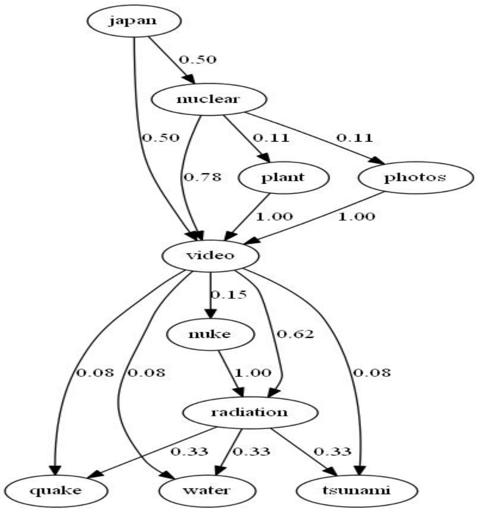


Figure 4: Japan’s Earthquake - Fox News March 2011

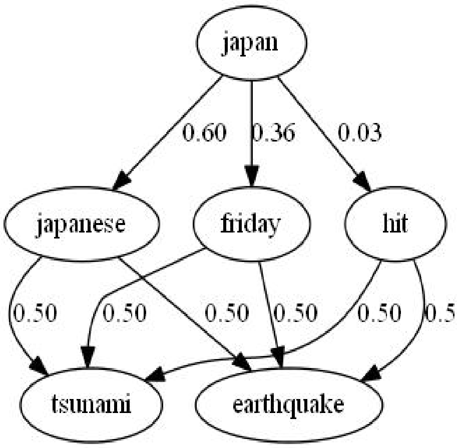


Figure 5: Japan’s Earthquake - BBC March 2011

media (this was well before NATO engagement in Libya). Indeed, the semantic network extracted from BBC for the Libya story is too complex to be shown here! It was also evident that the American news media's coverage of the Japan story was richer in content and more diverse than that of the BBC. Another interesting finding, reflecting the inescapable Zipfian nature of lexical distributions, is that the n-grams rank frequencies in all cases had power law shapes, as shown in Figure 6 for the BBC.

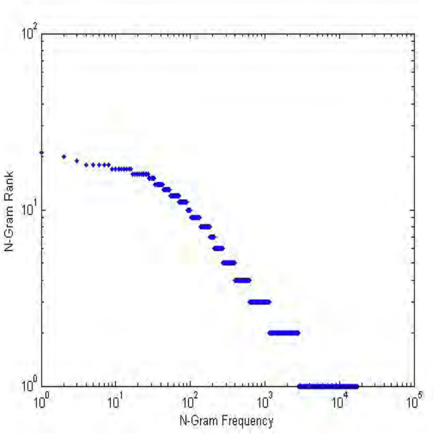


Figure 6: N-Gram Rank-Frequency - BBC January 2010

Conclusion

In this paper, we have provided a brief description of a method we are developing for the automated semantic analysis of on-line newsfeeds. The preliminary results shown indicate that the method holds great promise for deeper analysis – perhaps including longitudinal analysis – of news, which will be valuable to both professionals and the public.

References

Benamara, F.; Cesarano, C.; Picariello, A.; Reforgiato, D.; and Subrahmanian, V. 2007. Sentiment analysis: Adjectives and adverbs are better than adjectives alone. *International AAAI Conference on Weblogs and Social Media (ICWSM)* 203–206.

Blei, D.; Ng, A.; Jordan, M.; and Lafferty, J. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research* 3:993–1022.

Cesarano, C.; Picariello, A.; Reforgiato, D.; and Subrahmanian, V. 2007. The oasys 2.0 opinion analysis system. *International AAAI Conference on Weblogs and Social Media (ICWSM)* 313–314.

Future, R. 2010. Recorded future - temporal & predictive analytics engine, media analytics & news analysis. [Online; accessed 22-November-2010].

Gerner, D.; Abu-Jabr, R.; Schrodt, P.; and Yilmaz, . 2002. Conflict and mediation event observations (cameo): A new event data framework for the analysis of foreign policy interactions. *International Studies Association of Foreign Policy Interactions*.

Hofmann, T. 1999. Probabilistic latent semantic analysis. *Uncertainty in Artificial Intelligence, UAI99* 289–296.

Kim, D., and Oh, A. 2011. Topic chains for understanding a news corpus. *12th International Conference on Intelligent Text Processing and Computational Linguistics(CICLING 2011)* 12.

Leskovec, J.; Backstrom, L.; and Kleinberg, J. 2009. Meme-tracking and the dynamics of the news cycle. *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining* 497–506.

Libby, D. 1997. Rss 0.91 spec, revision 3. *Netscape Communications*.

Manning, C. D., and Schultze, H. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

McClelland, C. 1971. World event/interaction survey. *Defense Technical Information Center*.

Newman, D. 2010. Topic modeling scripts and code. *Department of Computer Science, University of California, Irvine*.

Oh, A.; Lee, H.; and Kim, Y. 2009. User evaluation of a system for classifying and displaying political viewpoints of weblogs. *AAAI Publications, Third International AAAI Conference on Weblogs and Social Media*.

Palantir. 2004. Privacy and civil liberties are in palantirs dna. [Online; accessed 10-December-2010].

Tomlinson, R. 1993. World event/interaction survey (weis) coding manual. *Department of Political Science, United States Naval Academy, Annapolis, MD.*

Van Rijsbergen, C.; Robertson, S.; and Porter, M. 1980. New models in probabilistic information retrieval. *British Library Research and Development Report* 5587.

Wikipedia. 2012. N-gram — wikipedia, the free encyclopedia. [Online; accessed 13-March-2012].

Expert Systems, Tutoring and Robotics

Chair: Paul DePalma

Bypassing Words in Automatic Speech Recognition

Paul De Palma
Department of Computer Science
Gonzaga University
Spokane, WA
depalma@gonzaga.edu

George Luger
Department of Computer Science
University of New Mexico
Albuquerque, NM
luger@cs.unm.edu

Caroline Smith
Department of Linguistics
University of New Mexico
Albuquerque, NM
caroline@unm.edu

Charles Wooters
Next It Corporation
Spokane, WA
cwooters@nextit.com

Abstract

Automatic speech recognition (ASR) is usually defined as the transformation of an acoustic signal to words. Though there are cases where the transformation to words is useful, the definition does not exhaust all contexts in which ASR could be used. Once the constraint that an ASR system outputs words is relaxed, modifications that reduce the search space become possible: 1) The use of syllables instead of words in the recognizer's language model; 2) The addition of a concept model that transforms syllable strings to concept strings, where a concept collects related words and phrases. The paper presents preliminary positive results on the use of syllables and concepts in speech recognition and outlines our current efforts to verify the Syllable-Concept Hypothesis (SCH).

Introduction

The speech recognition problem is conventionally formulated as the transformation of an acoustic speech signal to word strings. Yet this formulation dramatically underspecifies what counts as word strings. Here is a "33-year-old business woman" speaking to a reporter from *The New York Time***s**: "We have never seen anything like this in our history. Even the British colonial rule, they stopped chasing people around when they ran into a monastery" (Sang-Hun 2007: 1). The reporter has certainly transformed an acoustic signal into words. Though it would be nice to have a recording and transcription of the actual interview, we can get a sense of what the reporter left out (and put in) by looking at any hand-transcribed corpus of spontaneous speech. Here is the very first segment from the Buckeye Corpus:

> yes <VOCNOISE> i uh <SIL> um <SIL> uh <VOCNOISE> lordy <VOCNOISE> um <VOCNOISE> grew up on the westside i went to <EXCLUDE-name> my husband went to <EXCLUDE-name> um <SIL> proximity wise is probably within a mile of each other we were kind of high school sweethearts and <VOCNOISE> the whole bit <SIL> um <VOCNOISE> his dad still lives in grove city my mom lives still <SIL> at our old family house there on the westside <VOCNOISE> and we moved <SIL> um <SIL> also on the westside probably couple miles from my mom.

While we recognize the benefits of solving the speech recognition problem as described, the research presented here begins with the observation that human language performance does not include transcription from an acoustic signal to words—either in the sanitized form found in The *New York Times* quote or in the raw form found in the Buckeye Corpus. We do not suggest that AI research limit itself to human performance. We do claim that there is much to be gained by relaxing the constraint that the output of automatic speech recognition be a word string. Consider a speech recognition system designed to handle spoken plane reservations via telephone or, for that matter, just about any spoken-dialog system. The recognizer need only pass on the sense of the caller's speech to an appropriately constructed domain knowledge system to solve a problem of significant scope.

The question of what is meant by the sense of an utterance is central to this research. As a first approximation, one can think of the sense of an utterance as a sequence of concepts, where a concept is an equivalence class of words and phrases that seem to mean the same thing. A conventional recognizer generates a word string given a sequence of acoustic observations. The first stage in our research is to generate a syllable string given the same sequence of acoustic observations. Notice that the search space is much reduced. There are fewer syllables to search through (and mistake) than words. Of course, this syllable string must undergo a further transformation to be useful. One possibility would be to probabilistically map it to word strings. We have experimented with this. The results have not been encouraging. We propose, instead, to generate a concept string given the syllable string. Once again, the search space is reduced. There are fewer concepts to search through and mistake than words.

The Symbol-Concept Hypothesis (SCH) claims that this dual reduction in search space will result in better recognition accuracy over a standard recognizer. Though SCH can be argued using the axioms of probability, at bottom it is an empirical hypothesis. Preliminary experimental results have been promising. This paper is the first in a four phase, multi-year research effort to test SCH:

- Phase I: Gather preliminary data about SCH using small corpora.
- Phase II: Reproduce the results from Phase I using a much larger corpus.
- Phase III: Introduce a probabilistic concept generator and concept model.
- Phase IV: Introduce an existing domain knowledge system and speech synthesizer to provide response to the user.

Background

The goal of probabilistic speech recognition is to answer this question: "What is the most likely string of words, *W*, from a language, *L*, given some acoustic input, *A*." This is formulated in equation 1:

$$hyp(W) = \frac{argmax}{w \in L} P(W|A) \qquad (1)$$

Since words have no place in SCH, we speak instead of symbol strings drawn from some set of legal symbols, with the sole constraint that the symbols be encoded in ASCII format. So, equation (1) becomes:

$$hyp(S) = \frac{argmax}{s \in L} P(S|A) \qquad (2)$$

Equation (2) is read: "The hypothesized symbol string is the one with the greatest probability given the sequence of acoustic observations" (De Palma 2010:16). Bayes Theorem lets us rewrite equation (2) as:

$$hyp(S) = \frac{argmax}{s \in L} \frac{P(A|S) * P(S)}{P(A)} \qquad (3)$$

Since *P(A)* does not affect the computation of the most probable symbol string (the acoustic observation is the acoustic observation, no matter the potential string of symbols) we arrive at a variation of the standard formulation of probabilistic speech recognition (Jurafsky and Martin 2009):

$$hyp(S) = \frac{argmax}{S \in L} P(A|S) * P(S) \qquad (4)$$

The difference is that the formulation has been generalized from words to any symbol string. *P(A|S)*, known as the likelihood probability in Bayesian inference, is called the acoustic model in the context of automatic speech recognition. *P(S)*, known as the prior probability in Bayesian inference, is called the language model in ASR. The acoustic model expresses the probability that a string of symbols—words, syllables, whatever—is associated with an acoustic signal in a training corpus. The language model expresses the probability that a sequence of symbols—again, words, syllables, whatever—is found in a training corpus.

The attractiveness of syllables for the *acoustic model* of speech recognition has been noted for some time. A study of the SWITCHBOARD corpus found that over 20% of the manually annotated phones are never realized acoustically, since phone deletion is common in fluent speech. On the other hand, the same study showed that 99% of canonical syllables are realized in speech. Syllables also have attractive distributional properties. The statistical distributions of the 300 most frequently occurring words in English and the most common syllables are almost identical. Though monosyllabic words account for only 22% of SWITCHBOARD by type, they account for a full 81% of tokens (Greenberg 1999; Greenberg 2001; Greenberg et al. 2002). All of this suggests that the use of syllables in the acoustic model might avoid some of the difficulties associated with word pronunciation variation due to dialect, idiolect, speaking rate, acoustic environment, and pragmatic/semantic context.

Nevertheless, most studies indicate positive but not dramatic improvement when using a syllable-based acoustic model (Ganapathiraju et al. 1997 and 2002; Sethy and Narayanan 2003; Hamalainen et al. 2007). This has been disappointing given the theoretical attractiveness of syllables in the acoustic model. Since this paper is concerned exclusively with the language model and post-language model processing, conjectures about the performance of syllables in the acoustic model performance are beyond its scope.

Still, many of the reasons that make syllables attractive in the acoustic model also make them attractive in the language model, including another not mentioned in the literature on acoustic model research: there are fewer syllables than words, a topic explored later in this paper. Since the output of a recognizer using a syllable language model is a syllable string, studies of speech recognition using syllable language models have been limited to special purpose systems where output word strings are not necessary. These include reading trackers, audio indexing systems, and spoken name recognizers. Investigations report significant improvement over word language models (Bolanos et al. 2007; Schrumpf, Larson, and Eickler 2005; Sethy and Narayanan 1998). The system proposed here, however, does not end with a syllable string, but, rather, passes this output to a concept model—and thereby transforms them to concept strings, all to be described later.

Researchers have recognized the potential usefulness of concepts in speech recognition: since the early nineties at Bell Labs, later at the University of Colorado, and still later at Microsoft Research (Pieraccini et al. 1991; Hacioglu and Ward 2001; Yaman et al. 2008). The system proposed here does not use words in any fashion (unlike the Bell Labs system), proposes the use of probabilistically generated concepts (unlike the Colorado system), and is more general than the utterance classification system developed at Microsoft. Further, it couples the use of sub-word units in the language model, specifically syllables, with concepts, an approach that appears to be novel.

Syllables, Perplexity, and Error Rate

One of the first things that a linguist might notice in the literature on the use of the syllable in the acoustic model is that its complexity is underappreciated. Rabiner and Juang (1993), an early text on speech recognition, has only two index entries for “syllable” and treat it as just another easily-defined sub-word unit. This is peculiar, since the number of English syllables varies by a factor of 30 depending on whom one reads (Rabiner and Juang 1993; Ganapathiraju, et al. 1997; Huang et al. 2001). In fact, there is a substantial linguistic literature on the syllable and how to define it across languages. This is important since any piece of software that claims to syllabify words embodies a theory of the syllable. Thus, the syllabifier that is cited most frequently in the speech recognition literature, and the one used in the work described in this paper, implements a dissertation that is firmly in the tradition of generative linguistics (Kahn 1976). Since our work is motivated by more recent research in functional and cognitive linguistics (see, for example, Tomasello 2003), a probabilistic syllabifier might be more appropriate. We defer that to a later stage of the project, but note in passing that probabilistic syllabifiers have been developed (Marchand, et al. 2007).

Still, even though researchers disagree on the number of syllables in English, that number is significantly smaller than the number of words. And therein lies part of their attractiveness for this research. Simply put, the syllable search space is significantly smaller than the word search space. Suppose language *A* has *a* words and language *B* has *b* words, where $a > b$. All other things being equal, the probability of correctly guessing a word from *B* is greater than guessing one from *A*. Suppose further, that these words are not useful in and of themselves, but contribute to some downstream task, the accuracy of which is proportional to the accuracy of the word recognition task. Substitute syllables for words in language B—since both are symbols—and this is exactly the argument being made here.

Now, one might ask, if syllables work so nicely in the language model of speech recognition, why not use another sub-word with an even smaller symbol set, say a phone or demi-syllable? Though the question is certainly worth investigating empirically, the proposed project uses syllables because they represent a compromise between a full word and a sound. By virtue of their length, they preserve more linguistic information than a phone and, unlike words they represent a relatively closed set. Syllables tend not to change much over time.

A standard *a priori* indicator of the probable success of a language model is lower perplexity, where perplexity is defined as the Nth inverse root of the probability of a sequence of words (Jurafsky and Martin 2009; Ueberla 1994):

$$PP(W) = p(w_1 w_2 \ldots w_n)^{-1/n} \qquad (5)$$

Because there are fewer syllables than words, we would expect both their perplexity in a language model to be lower and their recognition accuracy to be higher. Since the history of science is littered with explanations whose self-evidence turned out to have been incorrect upon examination, we offer a first pass at an empirical investigation.

To compare the perplexity of both syllable and word language models, we used two corpora, the Air Travel Information System (Hemphill 1993) and a smaller corpus (SC) of human-computer dialogs captured using the Wizard-of-Oz protocol at Next It (Next IT 2012), where subjects thought they we were interacting with a computer but in fact were conversing with a human being. The corpora were syllabified using software available from the National Institute of Standards and Technology (NIST 2012).

Test and training sets were created from the same collection of utterances, with the fraction of the collection used in the test set as a parameter. The results reported here use a randomly chosen 10% of the collection in the test set and the remaining 90% in the training set. The system computed the mean, median, and standard deviation over twenty runs. These computations were done for both word and syllable language models for unigrams, bigrams, trigrams, and quadrigrams (sequences of one, two, three, and four words or syllables). As a baseline, the perplexity of the unweighted language model—one in which any word/syllable has the same probability as any other—was computed.

For bigrams, trigrams, and quadrigrams, the perplexity of a syllable language model was less than that of a word language model. Of course, in comparing the perplexity of syllable and word language models, we are comparing sample spaces of different sizes. This can introduce error based on the way perplexity computations assign probability mass to out-of-vocabulary tokens. It must be recognized, however, that syllable and word language models are not simply language models of different sizes of the kind that Ueberla (1994) considered. Rather, they are functionally related to one another. This suggests that the well-understood caution against comparing the perplexity of language models with different vocabularies might not apply completely in the case of syllables and words. Nevertheless, the drop in perplexity was so substantial in a few cases (37.8% SC quadrigrams, 85.7% ATIS bigrams), that it invited empirical investigation with audio data.

Recognition Accuracy

Symbol Error Rate (SER) is the familiar Word Error Rate (WER) generalized so that context clarifies whether we are talking about syllables, words, or concepts. The use of SER raises a potential problem. The number of syllables (either by type or token) differs from the number of words in the training corpus. Further, in all but monosyllabic training corpora, syllables will, on average, be shorter than words. How then can we compare error rates? The answer, as before, is that 1) words are functionally related to syllables and 2) improved accuracy in syllable recognition will contribute to downstream accuracy in concept recognition.

To test the hypothesis that a syllable language model would perform more accurately than a word language model, we gathered eighteen short audio recordings, evenly distributed by gender, and recorded over both the public switched telephone network and mobile phones. The recognizer used was SONIC from the Center for Spoken Language Research of the University of Colorado (SONIC 2010). The acoustic model was trained on the MACROPHONE corpus (Bernstein et al. 1994). Additional tools included a syllabifier and scoring software available from the National Institute of Standards and Technology (NIST 2012), and language modeling software developed by one of the authors.

The word-level transcripts in the training corpora were transformed to phone sequences via a dictionary look-up. The phone-level transcripts were then syllabified using the NIST syllabifier. The pronunciation lexicon, a mapping of words to phone sequences, was similarly transformed to map syllables to phone sequences. The word-level reference files against which the recognizer's hypotheses were scored were also run through the same process as the training transcripts to produce syllable-level reference files.

With these alterations, the recognizer transformed acoustic input into syllable output represented as a flavor of Arpabet. Figure 1 shows an extract from a reference file represented both in word and in phone-based syllable form.

i want to fly from spokane to seattle
ay waantd tuw flay frahm spow kaen tuw si ae dxaxl

i would like to fly from seattle to san Francisco
ay wuhdd laykd tuw flay frahm siy ae dxaxl tuw saen fraen sih skow

Figure 1: Word and Syllable References

The recognizer equipped with a syllable language model showed a mean improvement in SER over all N-gram sizes of 14.6% when compared to one equipped with a word language model. Though the results are preliminary, and await confirmation with other corpora, and with the caveats already noted, they suggest that a recognizer equipped with a syllable language model will perform more accurately than one equipped with a word language model.[1] This will contribute to the downstream accuracy of the system described below. Of course, it must be pointed out that some of this extraordinary gain in recognition accuracy will necessarily be lost in the probabilistic transformation to concept strings.

Concepts

At this point one might wonder about the usefulness of syllable strings, no matter how accurately they are recognized. We observe that the full range of a natural language is redundant in certain pre-specified domains, say a travel reservation system. Thus the words and phrases *ticket*, *to book a flight*, *to book a ticket*, *to book some travel*, *to buy a ticket*, *to buy an airline ticket*, *to depart*, *to fly*, *to get to*, all taken from the reference files for the audio used in this study, describe what someone wants in this constrained context, namely to go somewhere. With respect to a single word, we collapse morphology and auxiliary words used to denote person, tense, aspect, and mood, into a base word. So, *fly, flying, going to fly, flew, go to, travelling to*, are grouped, along with certain formulaic phrases (*book a ticket to*), in the equivalence class, GO. Similarly, the equivalence class WANT contains the elements *buy, can I, can I have, could I, could I get, I like, I need, I wanna, I want, I would like, I'd like, I'd like to have, I'm planning on, looking for, need, wanna, want, we need, we would like, we'd like, we'll need, would like.* We refer to these equivalence classes as concepts.

For example, a sentence from the language model (I want to fly to Spokane) was syllabified, giving:

ay w_aa_n_td t_uw f_l_ay t_uw s_p_ow k_ae_n

[1] Though it might be interesting and useful to look at individual errors, the point to keep in mind is that we are looking for broad improvement. The components of SCH were not so much arguments as the initial justification for empirical investigations, investigations that will support or falsify SCH.

Then concepts were mapped to the syllable strings, producing:

WANT GO s_p_ow k_ae_n

The mapping from concepts to syllable strings was rigid and chosen in order to generate base-line results. The mapping rules required that at least one member of an equivalence class of syllable strings had to appear in the output string for the equivalence class name to be inserted in its place in the output file. For example, k_ae_n ay hh_ae_v (*can I have*) had to appear in its entirety in the output file for it to be replaced with the concept WANT.

The experiment required that we:

1. Develop concepts/equivalence classes from the training transcript used in the language model experiments.
2. Map the equivalence classes onto the reference files used to score the output of the recognizer. For each distinct syllable string that appears in one of the concept/equivalence classes, we substituted the name of the equivalence class for the syllable string. We did this for each of the 18 reference files that correspond to each of the 18 audio files. For example, WANT is substituted for every occurrence of *ay w_uh_dd l_ay_kd* (*I would like*).
3. Map the equivalence classes onto the output of the recognizer when using a syllable language model for N-gram sizes 1 through 4. We mapped the equivalence class names onto the content of each of the 72 output files (4 x 18) generated by the recognizer.
4. Determine the error rate of the output in step 3 with respect to the reference files in step 2.

As before, the SONIC recognizer, the NIST syllabifier and scoring software, and our own language modeling software were used. The experiments showed a mean increase in SER over all N-gram sizes of just 1.175%. Given the rigid mapping scheme, these results were promising enough to encourage us to begin work on: 1) reproducing the results on the much larger ATIS2 corpus (Garofalo 1993) and 2) a *probabilistic* concept model.

Current Work

We are currently building the system illustrated in Figure 2. The shaded portions describe our work. A crucial component is the concept generator. Under our definition, concepts are purely collocations of words and phrases, effectively, equivalence classes. In order for the system to be useful for multiple domains, we must go beyond our preliminary investigations: the concepts must be machine-generated. This will be done using a boot-strapping procedure, first described for word-sense disambiguation. The algorithm takes advantage of "the strong tendency of words to exhibit only one sense per collocation and per discourse" (Yarowsky 1995: 50). The technique will begin with a hand-tagged seed set of concepts. These will be used to incrementally train a classifier to augment the seed concepts. The output of a speech recognizer equipped with a syllable language model is the most probable sequence of syllables given an acoustic event. The formalisms used to probabilistically map concepts to syllable strings are reworkings of equations (1) to (4), resulting in:

$$hyp(C) = \frac{argmax}{c \in M} P(C|S) = \frac{argmax}{C \in M} P(S|C) * P(C) \quad (6)$$

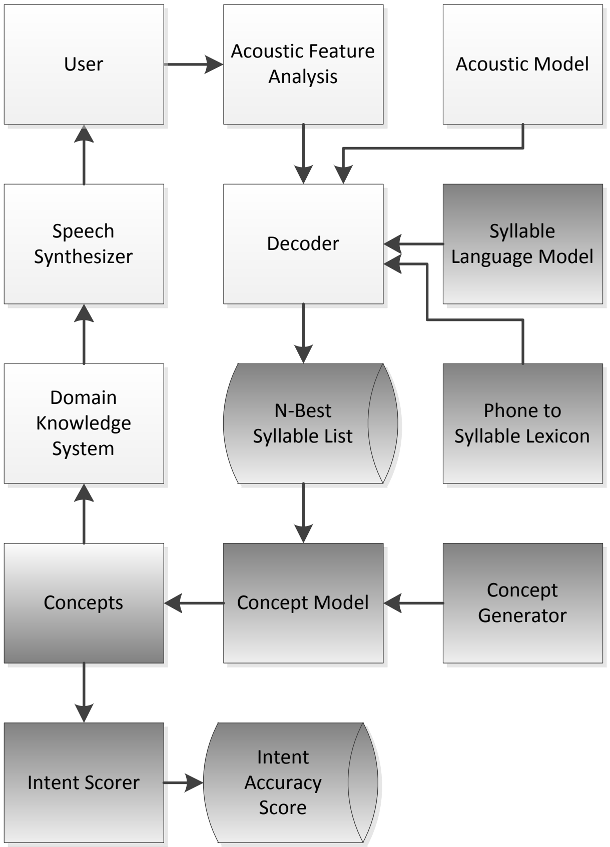


Figure 2: Acoustic features are decoded into syllable strings using a syllable language model. The syllables strings are probabilistically mapped to concept strings. The N-best syllable list is rescored using concepts. The Intent Scorer enables comparison of performance with a conventional recognizer.

M is just the set of legal concepts created for a domain by the concept generator. Equation (6) is an extension of a classic problem in computational linguistics: probabilistic part-of-speech tagging. That is, given a string of words, what is the most probable string of parts-of-speech? In the case at hand, given a syllable string, what is the most probable concept string?

Using equation (6), the Syllable-Concept Hypothesis, introduced early in the paper, can be formalized. If equation (1) describes how a recognizer goes about choosing a word string given a string of acoustic observations, then our enhanced recognizer can be described in equation (7):

$$hyp(C) = \frac{argmax}{C \in M} P(C|A) \quad (7)$$

That is, we are looking for the legal concept string with the greatest probability given a sequence of acoustic observations. SCH, in effect, argues that the P(C|A) exceeds the P(W|A).

Finally, the question of how to judge the accuracy of the system, from the initial utterance to the output of the concept model, must be addressed. Notice that the concept strings themselves are human readable. So,

I WANT TO FLY TO SPOKANE

becomes:

WANT GO s_p_ow k_ae_n

Amazon Mechanical Turk[2] workers will be presented with both the initial utterance as text and the output of the concept model as text and asked to offer an opinion about accuracy based on an adaptation of the Likert scale. To judge how the proposed system performs relative to a conventional recognizer, the same test will be made, substituting the output of the recognizer with a word language model and no concept model for the output of the proposed system.

Conclusion

We have argued that the speech recognition problem as conventionally formulated—the transformation of an acoustic signal to words—neither emulates human performance nor exhausts the uses to which ASR might be put. This suggests that we could bypass words in some ASR applications, going from an acoustic to signal to probabilistically generated syllable strings and from there to probabilistically generated concept strings. Our experiments with syllables on small corpora have been promising:

- 37.8% drop in perplexity with quadrigrams on the SC corpus
- 85.7% drop in perplexity with ATIS bigrams
- 14.6% mean increase in recognition accuracy over bigram, trigram, and quadrigrams

But as has been pointed out, a syllable string is not useful in a dialog system. Concepts must be mapped to syllables. A concept, as we define it, is an equivalence class of words and phrases that seem to mean the same thing in a given context. To date, we have hand-generated concepts from reference files and mapped them to syllables using a ridgid mapping scheme intended as a baseline.

But to be truly useful, any recognizer using concepts must automatically generate them. Since concepts, under our definition, are no more than collocations of words, we we propose a technique first developed for word-sense disambiguation: incrementally generate a collection of concepts from a hand-generated set of seed concepts. The idea in both phases or our work—probabilistically generating syllable strings and probabilistically generating concept strings—is to reduce the search space from what conventional recognizers encounter. At the very end of this process, we propose scoring how closely the generated concepts match the intent of the speaker using Mechanical Turk workers and a modified Likert scale. Ultimately the output the system will be sent on to a domain knowledge system, from there onto a speech synthesizer, and finally to the user, who, having heard the output will respond, thus starting the cycle over gain.

Our results to date suggests that the use of syllables and concepts in ASR will results in improved recognition accuracy over a conventional word-based speech recognizer. This improved accuracy has the potential to be used in fully functional dialog systems. The impact of such systems could be as far-reaching as the invention of the mouse and windowing software, opening up computing to persons with coordination difficulties or sight impairment, freeing digital devices from manual input, and transforming the structure of call centers. One application, often overlooked in catalogues of the uses to which ASR might be put, is surveillance.[3] The Defense Advanced Research Agency (DARPA) helped give ASR its current shape. According to some observers, the NSA, as a metonym for all intelligence agencies, is drowning in unprocessed data, much of which is almost certainly speech (Bamford 2008). The kinds of improvements described in this paper, the kinds that promise to go beyond the merely incremental, are what are needed to take voice recognition to the next step.

References

Bamford, J. 2008. *The Shadow Factory: The Ultra-Secret NSA from 9/11 to the Eavesdropping on America*. NY: Random House.

Bernstein, J., Taussig, K., Godfrey, J. 1994.

[2] The Amazon Mechanical Turk allows computational linguists (and just about anyone else who needs a task that requires human intelligence) to crowd-source their data for human judgment. See https://www.mturk.com/mturk/welcome

[3] Please note that this paper is not necessarily an endorsement of all uses to which ASR might be put. It merely recognizes what is in fact the case.

MACROPHONE. Linguistics Data Consortium, Philadelphia PA

Bolanos, B., Ward, W., Van Vuuren, S., Garrido, J. 2007. Syllable Lattices as a Basis for a Children's Speech Reading Tracker. *Proceedings of Interspeech-2007*, 198-201.

De Palma, P. 2010. *Syllables and Concepts in Large Vocabulary Speech Recognition*. Ph.D. dissertation, Department of Linguistics, University of New Mexico, Albuquerque, NM.

Ganapathiraju, A., Goel, V., Picone, J., Corrada, A., Doddington, G., Kirchof, K., Ordowski, M., Wheatley, B. 1997. Syllable—A Promising Recognition Unit for LVCSR. *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding*, 207-214.

Ganapathiraju, A., Hamaker, J., Picone, J., Ordowski, M., Doddington, G. 2001. Syllable-based large vocabulary continuous speech recognition. *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 4, 358-366.

Garofalo, J. 1993. ATIS2. Linguistics Data Consortium, Philadelphia, PA

Greenberg, S. 1999. Speaking in Shorthand—A Syllable-Centric Perspective for Understanding Pronunciation Variation. *Speech Communication*, 29, 159-176.

Greenberg, S. 2001. From here to Utility—Melding Insight with Speech Technology. *Proceedings of the 7th European Conference on Speech Communication and Technology*, 2485-2488.

Greenberg, S., Carvey, H. Hitchcock, L., Chang, S. 2002. Beyond the Phoneme: A Juncture-Accent Model of Spoken Language. *Proceedings of the 2nd International Conference on Human Language Technology Research*, 36-43.

Hacioglu, K., Ward, W. 2001. Dialog-Context Dependent Language Modeling Combining N-Grams and Stochastic Context-Free Grammars. *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, 537-540.

Hamalainen, A., Boves, L., de Veth, J., Bosch, L. 2007. On the Utility of Syllable-Based Acoustic Models for Pronunciation Variation Modeling. *EURASIP Journal on Audio, Speech, and Music Processing*, 46460, 1-11.

Hemphill, C. 1993. ATIS0. Linguistics Data Consortium, Philadelphia, PA.

Huang, X., Acero, A., Hsiao-Wuen, H. 2001. *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Upper Saddle River, NJ: Prentice Hall.

Jurafsky, D., Martin, J. 2009. *Speech and Language Processing*. Upper Saddle River, NJ: Pearson/Prentice Hall.

Kahn, D. 1976. *Syllable-based Generalizations in English Phonology*. Ph.D. dissertation, Department of Linguistics, University of Indiana, Bloomington, In: Indiana University Linguistics Club.

Marchand, Y. Adsett, C., Damper, R. 2007. Evaluating Automatic Syllabification Algorithms for English. *Proceedings of the 6th International Conference of the Speech Communication Association*, 316-321.

Next It Corporation. 2012. Web Customer Service with Intelligent Virtual Agents. Retrieved 3/37/2012 from: http://www.nextit.com.

NIST. 2012. Language Technology Tools/Multimodel Information Group—Tools. Retrieved 2/19/2012 from: http://www.nist.gov.

Pieraccini, R., Levin, E., Lee, C., 1991. Stochastic Representation of Conceptual Structure in the ATIS Task. *Proceedings of the DARPA Speech and Natural Language Workshop, 121-124*.

Rabiner, L., Juang, B. 1993. *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice Hall.

Sang-Hun, C. 10/21/2007. Myanmar, Fear Is Ever Present. *The New York Times.*

Schrumpf, C., Larson, M., Eickler, S., 2005. Syllable-based Language Models in Speech Recognition for English Spoken Document Retrieval. *Proceedings of the 7th International Workshop of the EU Network of Excellence DELOS on Audio-Visual Content and Information Visualization in Digital Libraries*, pp. 196-205.

Sethy, A., Narayanan, S. 2003. Split-Lexicon Based Hierarchical Recognition of Speech Using Syllable and World Level Acoustic Units, *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, I, 772-775.

SONIC. 2010. SONIC: Large Vocabulary Continuous Speech Technology. Retrieved 3/8/2010 from: http://techexplorer.ucsys.edu/show_NCSum.cfm?NCS=258626.

Tomasello, M. (ed.) 2003. *The New Psychology of Language: Cognitive and Functional Approaches to Language Structure*. Mahwah, NJ: Lawrence Erlbaum Associates.

Ueberla, J. 1994. *Analyzing and Improving Statistical Language Models for Speech Recognition*. Ph.D. Dissertation, School of Computing Science, Simon Frazier University.

Yaman, S., Deng, L., Yu, D., Wang, W, Acera, A. 2008. An Integrative and Discriminative Technique for Spoken Utterance Classification. *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 6, 1207-1214.

Yarowsky, D. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, 189-196.

A User Friendly Software Framework for Mobile Robot Control

Jesse Riddle, Ryan Hughes, Nathaniel Biefeld, and **Suranga Hettiarachchi**
Computer Science Department, Indiana University Southeast
New Albany, IN 47150

Abstract

We are interested in designing applications for autonomous mobile robots and robot swarms to accomplish tasks such as terrain analysis, search and rescue, and chemical plume source tracing. These tasks require robots to avoid obstacles and reach a goal. We use X80Pro mobile robots designed and developed by Dr.Robot Inc. for task applications. The vendor provided software framework with graphical user interface(GUI) allows robots only to be controlled remotely. The development of new robot control applications require developers to acquire in-depth knowledge of $MicrosoftActiveX$ controls and C# programming language. In this paper, we present a new software framework for X80Pro robots that will allow even a novice C++ programmer to design and implement autonomous mobile robot applications. We demonstrate the feasibility of our software framework using behavior-based and physics based control algorithms where a X80Pro robot avoid obstacles to reach a goal.

Introduction

It is important to have a stable software framework for developing applications for autonomous mobile robots, especially a software framework that allows quick prototype development, code implementation, and testing. A software framework with architectural deficiencies may increase development time and reduce the performance of robot applications. These issues become much more relevant when the developers are undergraduate research students with limited knowledge in multiple languages and complex software framework. Therefore, we are motivated to develop a user friendly software framework based on a commonly used programming language for mobile robot application development.

One of the major issues of the vendor provided software framework is the flexibility to modify it to provide sufficient user friendliness. Though these frameworks are user friendly for some tasks, they may not be user friendly for other applications. We are interested in designing applications for mobile robotic tasks such as terrain analysis, search and rescue, and chemical plume source tracing using physics based control algorithms. The vendor provided software framework for X80Pro robots provides insufficient flexibility and user friendliness required to developing applications for above tasks. Our effort is not to make superior software framework to vendor provided framework, but only to have a flexible software framework that suits our needs.

In this paper, we present our work with preliminary results from two robot control algorithms, behavior based and physics based. The algorithms make use of sonars, infrared sensors, and the camera of the X80Pro robot. The task of the robot is to navigate through a simple obstacle course and reach a light source as the goal. The rest of this paper provides an introduction to X80Pro robot, a description of the software framework, the two control algorithms, results and analysis of the robot behavior, a discussion on related work, and conclusion and future work.

X80Pro Robot

The X80Pro robot hardware framework developed by Dr.Robot is a off the shelf product for researchers that offers full WiFi (802.11g) wireless with dual serial communication channels, multimedia, sensing and motion capabilities (Dr. Robot 2012). The hardware framework contains two 12 volts motors, seven inch driving wheels, DC motor driver module with position and current feedback, six ultrasonic range sensors, seven Sharp infrared distance measuring sensors, one pyroelectric human motion sensor, color image camera, and audio modules (speaker and microphone), sufficient hardware components to serve in variety of applications. The Figure 1 shows the front and back views of the X80Pro robot with all the components.

X80Pro Software Framework

The software framework provided by the vendor for X80Pro robot is unique in that it depends on Win32 ActiveX Control. An ActiveX control is a reusable software component based on the Component Object Model (COM) that supports a wide variety of Object Linking and Embedding (OLE) functionality and can be customized to fit many software needs.

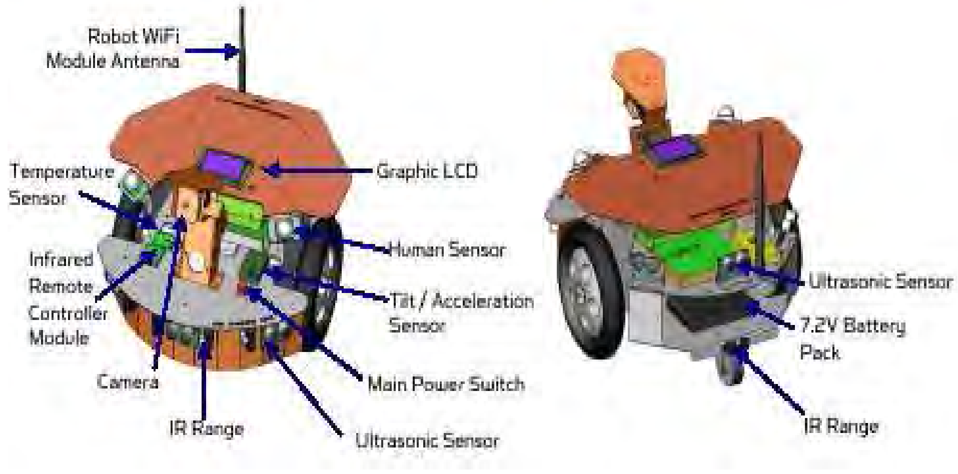


Figure 1: Front and back views of the X80Pro robot with the sensors mounted on the bottom.

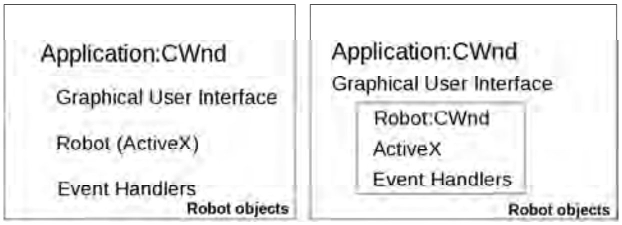


Figure 2: The vendor provided software framework(left) and user friendly software framework(right).

The developers can also create ActiveX controls with Microsoft Foundation Classes (MFC) (MFC ActiveX. 2005).

The complex software framework with multiple libraries implemented in C# treats the procedure calls as native. Behind the scenes though, when the ActiveX control is created in a C# application, the ActiveX component catches onto the application's message dispatcher automatically, without any extra special handling on the part of the application. In C# environment, the developer never have to work with the message dispatcher directly and can call the procedures as if they are native. However, when it comes to alternative programming languages, we realize that we could easily lose this simplicity. This is entirely because the vendor provided library is a framework dependent on Win32 ActiveX control, i.e. the application and the ActiveX representation builds an"is-a" relationship. Though the vendor's software framework provides sufficient development capabilities, one of the major inflexibility is the developed application's complete dependency on ActiveX controls making developers spend time trouble shooting software content possibly irrelevant to the task on hand. The left side box of the Figure 2 shows the high level architecture of the vendor provided software framework.

The right side box of the Figure 2 shows the high level architecture of the new user friendly software framework. In our new framework, the application keeps the event handlers contained in a class with an ActiveX representation for each robot by wrapping each ActiveX representation ("Robot:CWnd" in the inner box of the right box) into a Robot class, i.e. the application and the ActiveX representation builds a "has-a" relationship.

The simplest means to incorporate the provided library into an application in this framework is to first create either a Windows Forms Application using an MFC based application using C++. Next, using Microsoft Visual Studios form designer, developer should place the ActiveX control into a window container from the available toolbox. This causes the Visual Studio to automatically create a wrapper class to dispatch methods available in the ActiveX control. Finally the developer can assign methods to handle events coming from the ActiveX controls (ActiveX Events. 2005). We have also examined other possibilities, such as windowless activation of the control, with several container instances running asynchronously. However, without any special provision for multiple threads, the design of the framework becomes cumbersome, since we found that the event handlers are called asynchronously with up to 256 instances running in parallel.

Though the vendor supplied software framework seems simple, having all event handlers embed inside of the GUI design is a problem during application implementation and testing. We overcame this issue by compartmentalizing ActiveX controls with event handlers into an inner object independent of the GUI design. Though the implementation of our user friendly framework may be somewhat complicated, our main focus of robotic application development becomes much simpler with less time consumed on ActiveX trouble shooting.

Control Algorithms

We implemented two different control algorithms to test the feasibility of our user friendly software framework. The purpose of our experiments is to provide predictable results of control algorithms using our software framework. This allows us to evaluate the stability of our friendly software framework. The stability we refer here includes the accuracy of simple GUI for a control algorithm implemented with event handling to test sonar, infrared and motors.

Behavior Based X80Pro Control

The behavior-based approach is a methodology for designing autonomous agents and robots; it is a type of an *intelligent agent architecture*. Architectures supply structure and impose constraints on the way robot control problems are solved. The behavior-based methodology imposes a general biologically-inspired, distributed, bottom-up philosophy, allowing for a certain freedom of interpretation (Matarić 1999).

The behavior based algorithms demonstrates a variety of behaviors in a heuristic manner. Behavior-based and rule-based techniques do not make use of potential fields or forces. Instead, they deal directly with velocity vectors and heuristics for changing those vectors.

Physics Based X80Pro Control

In physics based approaches, virtual physics forces drive a agents and robots to a desired configuration or state. The desired configuration is one that minimizes overall system potential energy, and the system acts as a molecular dynamics ($\vec{F} = m\vec{a}$) simulation (Spears *et al.* 2005).

"Physicomimetics" or artificial physics (AP) is motivated by classical physics. This approach provides excellent techniques for distributed control of large collections of mobile physical agents as well as theoretical foundations for analyzing swarm behaviors. The Physicomimetics framework provides an effective basis for self-organization, fault-tolerance and self-repair of robot control (Spears *et al.* 2011).

Our control algorithm *AP-lite (i.e. artificial physics lite)* uses the physics based approaches with Hooke's law as our force law.

Experimental Methodology

Since we are at the initial stages of our research we decided to conduct all of the experiments with one robot though the control algorithms have the capability to scale to swarm of robots. Our robots are nonholonomic and they always move in the forward direction.

The robot environment is modeled with two parallel walls and four obstacles in between with sufficient space for the robot to navigate. The Figure 3 shows the robot view of this environment. The goal, a light source, is kept at the far end of the two walls and all lights in the lab were turned off during the experiments. We do not make use of filters to filter out the noise in the environment.

Our behavior based control algorithm uses two sonars (the left and right most sonars in the front of the robot) to model the robot behavior of moving from start location to a goal location where goal is a light source. The robot constantly move forward searching for the light source using the robot camera, and reacts to obstacles in the environment depending on the sonar readings by turning left or right by $35°$ angle (rule-based). The motors are powered with consistent rate of voltage, increasing or decreasing the power with the same consistency. For the clarity of the graphs presented in the Results and Analysis section, we scaled the data differently.

Our AP-lite control algorithm uses two infrared sensors to model the robot behavior of moving from start location to a goal location where goal is a light source. The algorithm maintains a global attractive goal force that is always active; this drives the robot forward with an equal amount of power to both motors. Again, the clarity of the graphs presented in the Results and Analysis section, we scaled the data differently. When the robot reaches an obstacle, AP-lite computes the repulsive forces acting on the robot, and changes the power supply to the motors to change the robot heading. If the robot senses an obstacle from the right (i.e. right infrared sensor reads a low value), AP-lite reacts to this repulsion by decreasing power to the left motor and/or increasing power to the right motor. If the robot senses an obstacle from the left (i.e. right sensor reads a high value), AP-lite reacts to this repulsion by decreasing power to the right motor and/or increasing power to the left motor. AP-lite measures the turning angle based on the virtual attractive and repulsive forces acting on the robot. The force vector of the AP-lite is computed every four milliseconds giving sufficient time for wireless data communication between the robot and the host computer.

We present results of our experiments in next section. All experiments are conducted indoor and averaged over five runs.

Results and Analysis

We test the stability of our software framework using a behavior based control algorithm and a physics based control algorithm, Ap-lite. The Figure 4 shows the sonar readings during navigation of the robot in the y-axis over time in the x-axis, where robot uses the behavior based control algo-

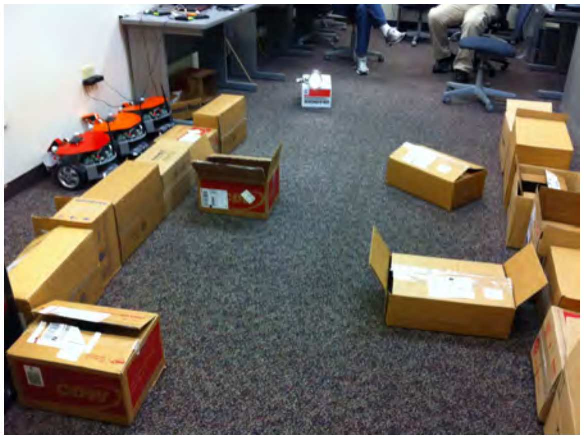


Figure 3: Robot view of the environment. Light source at the far end.

rithm. According to the robot view (see Figure 3) the first obstacle to the right is detected by the right sonar between the time 100 and 150, and both the left and the right sonars detect the last obstacle to the left before the robot reaches the goal at time 250 and 350. This is due to the fact that our robot is directly facing the last obstacle.

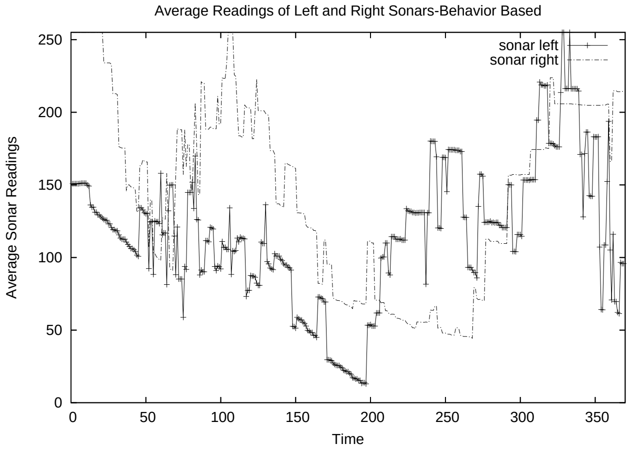


Figure 4: The sonar readings of the robot using behavior based algorithm.

The Figure 5 shows the power to motors during navigation of the robot in the y-axis over time in the x-axis, where robot uses the behavior based control algorithm. Though not very significantly, the power supply to the right motor increases when the robot reaches the first obstacle to the right, while the power to left motor remains unchanged. Then the robot takes several turns that we believe due to the fact that the robot is directly facing the last obstacle to the left before reaching the goal. This is also evident in the sonar readings in the Figure 4. The robot constantly change the power to two motors to keep the power consistent during these turns. We believe that this behavior can be corrected by finding an accurate balance of proper turning angle and proper filters to remove noise in the environment.

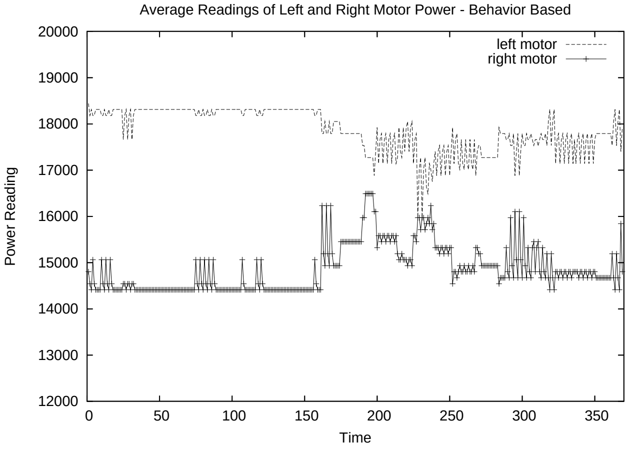


Figure 5: The power use by two motors of the robot using behavior based algorithm.

The Figure 6 shows the infrared sensor readings during navigation of the robot in the y-axis over time in the x-axis, where robot uses the AP-lite control algorithm. The results clearly show the robot reaching the first obstacle to the right where the right most infrared sensor reading decreases at times between 125 and 175, and the robot reaching last obstacle to the left before the goal at time 200 and 260. This behavior can clearly be seen in the power supply to the motor in the Figure 7 and the change in robot angle in the Figure 8.

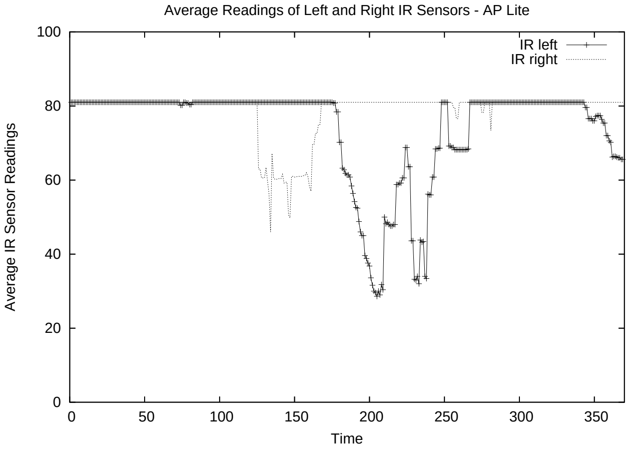


Figure 6: The infrared sensor readings of the robot using AP-lite algorithm.

The Figure 7 shows the power to motors during navigation of the robot in the y-axis over time in the x-axis, where robot uses the AP-lite control algorithm. When the robot reaches the first obstacle to the right, repulsive forces are high from the right side of the robot. The resulting force vector causes the robot to reduce power to the left motor, but increase power to the right motor, allowing the robot to take a quick and sharp turn. Since this sharp turn causes the robot to keep an easy focus on the goal robot does not reach the last obstacle to the right, but detects the last obstacle to the left before the goal. To avoid the obstacle to the left of the robot at time 180, AP-lite control algorithm reduces the power to right motor while keeping the power to left motor the same.

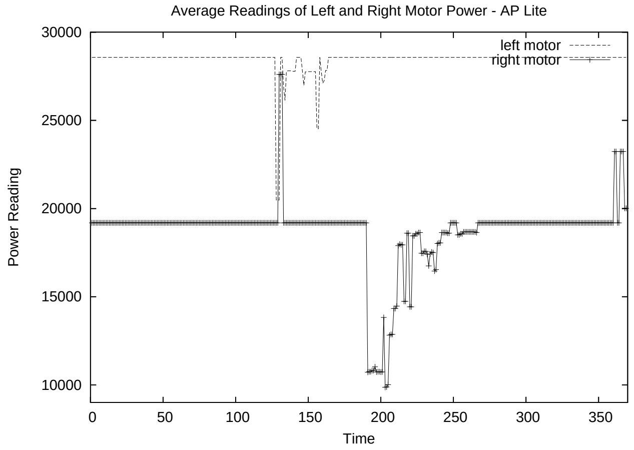


Figure 7: The power use by two motors of the robot using AP-lite algorithm.

We also measure the turning angle of the robot in AP-lite algorithm, since AP-lite computes the turning angle based on the attractive and repulsive forces exerted on the robot by obstacles and the goal. The Figure 7 shows the turning angle during navigation of the robot in the y-axis over time in the x-axis. Once again, it is apparent that the robot's left turn occurs with the first obstacle to the right, and the robot's right turn occurs with the last obstacle to the left before the goal. We believe that the robot force vector computation favors the least resistant path from the starting point to the goal.

Related Work

Both behavior-based and rule-based systems have proved quite successful in exhibiting a variety of behaviors in a heuristic manner. Fredslund and Matarić studied the problem of achieving global behavior in a group of distributed robots using only local sensing and minimal communication, in the context of formations (Fredslund and Matarić 2002). The key concept of their algorithm is that each robot follows a designated "friend" robot at the appropriate angle and distance using a proximity sensor that can provide the angle and distance information of the friend. By panning the sensor appropriately, the algorithm simply keeps the friend centered in the sensor's view. They presented their results using four and eight robots in different formations. Balch and Arkin accomplished robot formations using the following two step process: "detect-formation-position" which is a perceptual process that determines the robot's position in the formation based on the current environment data, and "maintain-formation" which generates motor commands to direct the robot towards the correct location (Balch and Arkin 1998).

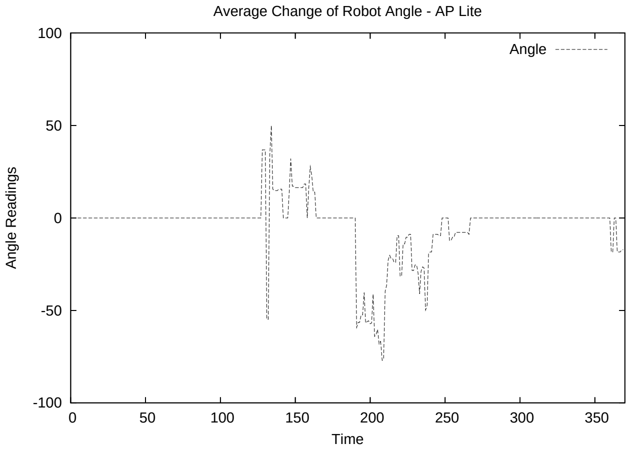


Figure 8: The turning angle of the robot using AP-lite algorithm.

One of the earliest physics-based techniques is the *potential fields* approach (e.g., (Khatib 1986)). Most of the PF literature deals with a small number of robots (typically just one that is similar to our experimental setup) that navigate through a field of obstacles to get to a target location. The environment, rather than the robots, exert forces. Obstacles exert repulsive forces while goals exert attractive forces (Kim 1991; Koren 1991).

The *social potential fields* (SPF) framework is highly related to physicomimetics framework (Reif 1998). Reif and Wang rely on a force-law simulation that is similar to the physicomimetics approach, allowing different forces between different robots. Their emphasis is on synthesizing desired formations by designing graphs that have a unique potential energy (PE) embedding.

Conclusion and Future Work

We are interested in designing applications for autonomous mobile robots and robot swarms. We use X80Pro mobile robots designed and developed by Dr.Robot Inc. for our applications. The vendor provided software framework was inflexibility and lack user friendliness required to develop software applications with undergraduate student researchers. We presented a user friendly software framework for X80Pro robots that will allow even a novice C++ programmer to design and implement autonomous mobile robot applications. We explored the feasibility of our software framework using behavior-based and physics based control algorithms where a X80Pro robot avoid obstacles to reach a goal. We are capable of producing predictable robot behavior using these control algorithm. We believe that the behavior based control algorithm needs to be studied further to provide a proper conclusion. The AP-lite shows significant predictability of the robot behavior in our user friendly software framework.

Future work of this research will focus on a Java based software framework for X80Pro robots and improved control algorithms to test the feasibility of the software framework. We would also extends the implementation of our algorithms to multi-robot systems since the algorithms are already theoretically extended to handle swarm of robots. This will also allow us to make use of techniques presented in (Reif 1998). Another improvement would be to implement filters to eliminate noise in the environment and test the robot behavior in more complex environments.

Acknowledgements

Authors are especially greatful to the IU Southeaset research and grant committee for providing financial support to this work. Authors would like to acknowledge the valuable support by IU Southeast Dean for Research, Dean of School of Natural Sciences, Department of Computer Science, and contributions by all the students who took C458 Intelligent Robots course in spring 2011.

References

Dr. Robot Inc. (2012). Dr. Robot Inc - Extend Your Imagination. `http://www.drrobot.com/products_item.asp?itemNumber=X80Pro`

Matarić, M. (1999). Behavior-Based Robotics. *MIT Encyclopedia of Cognitive Sciences, Robert A. Wilson and Frank C. Keil, eds., MIT Press,*: 74-77.

MFC ActiveX Controls (2005). Microsoft Visual Studio - MFC ActiveX Controls. `http://msdn.microsoft.com/en-us/library/k194shk8(v=vs.80).aspx`

ActiveX Controls: Events (2005). Microsoft Visual Studio - ActiveX Controls: Events. `http://msdn.microsoft.com/en-us/library/aa268929(v=vs.60).aspx`

Spears, W., Spears, D., Hamann, J. and Heil, R. (2005). Distributed, Physics-Based Control of Swarm of Vehicles. Autonomous Robots, *Kluwer,* **17**: 137–164.

Balch, T. and Arkin, R. (1998). Behavior-based Formation Control for Multi-Robot Teams. *IEEE Transactions on Robotics and Automation,* **14**: 1–15.

Spears, W. and Spears, D., eds. (2011). Physicomimetics-Physics Based Swarm Intelligence. Springer.

Fredslund, J. and Matarić, M. (2002). A General Algorithm for Robot Formations Using Local Sensing and Minimal Communication. *IEEE Transactions on Robotics and Automation,* **18**: 837–846.

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research,***5, (1)**: 90–98.

Kim, J. and P. Khosla (1991). Real-time obstacle avoidance using harmonic potential functions. *IEEE International Conference on Robotics and Automation,*: 790796.

Koren, Y. and J. Borenstein (1991). Potential field methods and their inherent limitations for mobile robot navigation. *IEEE International Conference on Robotics and Automation,* 1398–1404.

Reif, J. and H. Wang (1998). Social potential fields: A distributed behavioral control for autonomous robots. *Workshop on the Algorithmic Foundations of Robotics.*

COMPS Computer Mediated Problem Solving: A First Look

Melissa A. Desjarlais
Valparaiso University
melissa.desjarlais@valpo.edu

Jung Hee Kim
North Carolina A&T
jungkim@ncat.edu

Michael Glass
Valparaiso University
michael.glass@valpo.edu

Abstract

COMPS is a web-delivered computer-mediated problem solving environment designed for supporting instructional activities in mathematics. It is being developed as a platform for student collaborative exploratory learning using problem-specific affordances. COMPS will support computer-aided monitoring and assessment of these dialogues. In this paper we report on the first use of COMPS in the classroom, supporting an exercise in quantitative problem-solving. We have identified a number of categories of dialogue contribution that will be useful for monitoring and assessing the dialogue and built classifiers for recognizing these contributions. Regarding the usability of the interface for problem-solving exercises, the primary unexpected behavior is an increase (compared to in-person exercises) in off-task activity and concomitant decrease in shared construction of the answer. Its first large deployment will be for Math 110, a quantitative literacy class at Valparaiso University.

Introduction

COMPS is a web-delivered computer-mediated problem solving environment designed for supporting instructional activities in mathematics.

In its initial classroom use COMPS supports groups of students engaging a particular exercise in quantitative literacy: figuring out a winning strategy, should one exist, for a Nim-like game. It has problem-related affordances for the students to manipulate, shows the instructor the conversations in real time, permits the instructor to intervene, and records all events for analysis. The intelligent part of COMPS, which has not been deployed in classroom use, has the computer itself participate in the supervisory task: monitoring the conversation status for bits of knowledge and other markers of progress or lack of progress and displaying its findings to the supervising instructor.

COMPS gives us a platform for deploying AI techniques in mathematics dialogues. Immediate applications include:

- *Exploratory learning*. COMPS is an environment with affordances for computer-supported collaborative exploratory-learning dialogues. Plug-in modules provide problem specific aids and affordances. The Poison game we report on here comes with a visualization of the game state and buttons for playing.
- *Computer-monitored dialogues*. COMPS has provisions for an instructor to oversee and intervene in the student conversations. In the style of, e.g. Argunaut [De Groot et al. 2007], COMPS will provide a status screen for the instructor, showing what knowledge the students have discovered in their inquiry learning as well as measures of affective state (e.g. are they on-task or frustrated) and other measures of progress. Experiments toward computer-generated status are described in this paper.
- *Assessment reports*. Using similar techniques as for monitoring, COMPS will provide the instructor with assessment reports of the conversations. This will permit the instructor to have the students engage in the exercises out of class, on their own time.
- *Observation and data collection*. COMPS collects transcripts and data that will be useful both in understanding the student problem-solving behaviors and in producing better computer understanding of COMPS conversations.

In this paper we report on the interaction model of COMPS, the educational context for its initial deployment, results from first use in a classroom setting, and first results toward having it monitor the progress the student conversation.

The COMPS Model

The common threads to COMPS applications are a) dialogue, b) solving problems, and c) third parties. It is intended to facilitate and capture the kinds of interactions that would occur in mathematics problem-solving conversations. We have a simplified keyboard-chat communication channel instead of in-person face-to-face and voice communication. This permits us to readily log all interaction, more importantly it facilitates having the computer understand, monitor, assess, and potentially intervene in the dialogue. Because the problem domain is mathematics COMPS includes facilities for interpreting and rendering “ASCII Math,” expressions typed in-line using ordinary keyboard characters [MathForum 2012a].

COMPS conversations can be tutorial or they can be peer-to-peer explorations. Our view of how to support interactions is informed by the tutorial problem-solving dialogue studies of [Fox 1993] and the Virtual Math Team problem-solving dialogue studies of [Stahl 2009]. Wooz, the immediate predecessor to COMPS, has been used for recording and facilitating tutorial dialogues in algebra and differential equations, experiments in structured tutorial interactions, and exploratory learning with differential equations visualization applets [Kim and Glass 2004][Patel et al. 2003] [Glass et al. 2007].

The other element of COMPS conversations is possible third parties: teachers figuratively looking over the shoulders of the students as they work, computers also looking over the shoulders, teachers and computers intervening in the conversation, reports generated afterward with assessments of the student learning sessions, and analyses of the transcripts of interactions.

The common elements of COMPS applications are thus:

- *Interactivity*. Just as in in-person interactions, participants can behave asynchronously: interrupting and chatting over each other. Participants can see the other participants' keystrokes in real time, they do not need to take turns or wait for the other person to press *enter*. One use for this is documented by Fox who found tutors using *transition relevance points* [Sacks et al. 1974]. These are places within a dialogue turn where the other party is licensed to take over. For example, the tutor can provide scaffolding by starting to say an answer. Using prosodic cues (rising voice, stretched vowels), the tutor provides the student opportunities to take over the dialogue and complete the thought.
- *A problem window*. The problem is configurable, but generally there are areas of the screen window that keep the problem statement and elements of the solution in view without scrolling them off the screen. These items are assumed to be within the dialogue focus of all participants at all times, the objects of team cognition (Stahl) and shared construction (Fox).
- *A central server*. The server routes interaction traffic between the participants and optional third parties to the conversation (both human and machine), and records all interactions in log files.

Figure 1 at the end of this paper illustrates COMPS at work.

COMPS runs as a Flash application within a web browser, the server is a Java program. The application is configurable: plug-in modules written in Flash provide custom environments tailored for particular mathematics problems and learning modalities.

COMPS is similar in spirit to the Virtual Math Teams (VMT) chat interface [MathForum 2012b]. The VMT interface supports a generalized graphical whiteboard instead of having specialized interfaces for particular exercises. However many of the exercises that COMPS is intended to support are currently executed in class with manipulatives. For example the Poison game described in this report uses piles of tiles. It was incumbent on us to have COMPS provide software affordances that mimic the manipulatives.

Math 110 Background

A goal of this project is to introduce COMPS computer-mediation to the group collaborative exercises in the Valparaiso University (VU) Math 110 class. Math 110 delivers the quantitative literacy skills expected of an educated adult [Gillman 2006] along with the mathematics skills expected in quantitative general education classes in a liberal arts curriculum. It achieves this by using modern pedagogical techniques and a selection of topics and problems that are quite different from, more motivating than, and we hope more successful than the typical bridge or college algebra class.

It is the style of instruction that matches Math 110 to COMPS, viz:

- Problems are explored by experimentation, using manipulatives and written instructions.
- Four person groups collaborate on the in-class explorations, with students adopting special assigned roles in the collaborative process.
- During the class period the instructor observes the group interactions and offers suggestions or guiding questions, as needed.

These are aligned with the three threads of COMPS: solving problems, dialogue, and third parties. During a semester, students solve twenty in-class problems. An emphasis is placed on problem-solving strategies.

Math 110 in its current form has been the established bridge class in the VU curriculum for 15 years. Students enrolled in Math 110 performed poorly on the mathematics placement exam and must successfully complete the course before they can enroll in quantitatively-based general education courses. Data show that completing Math 110 has a positive effect on retention and success at the university [Gillman 2006].

Math 110 differs from simply repeating high school algebra not only in teaching style but also in content. There are five topical themes: Pattern Recognition, Proportional Reasoning, Fairness, Graphs and Decision Science, and Organizing Information. Together these themes provide a background in logical reasoning, quantitative skills, and critical thinking.

Writing skills are exercised by requiring students to write up each problem in a narrative format. Each written solution includes the statement of the problem in the student's own words, the solution of the problem, and an explanation of the solution. Often this entails a description of the experimental activities and results. The students are assessed on the written aspect of the solution in addition to the mathematical aspect.

Poison Exercise

An example of a Math 110 collaborative exercise—the first we have implemented in COMPS—is the Poison problem. The prompt is shown in Figure 2 at the end of this paper. Poison is a Nim-like two-person game. Starting from a pile of tiles, each person removes one or two tiles per turn. The last tile is "poisoned," the person who removes the last tile loses. The question before the students is to figure out how to play perfectly, to find an algorithm for either person A or person B to force a win. In a classroom setting the manipulative for this exploratory learning exercise in pattern

A	well everytime ive had 4, or 7 i lose.
C	huh?
A	Oh wait, that's every round >:(
C	i dont think it matters
B	hahaha
	(playing game)
B	lets do 23 again and ill pick a 1 to start instead of a 2?
A	FINE
	⋮
D	i just tried to avoid 7 and still got stuck with 4

Figure 3: Dialogue from Poison Exercise Using COMPS

recognition is a box of tiles. Students also have pencil and paper.

For purposes of moving this exercise to the computer-mediated environment, we wrote a COMPS module that simulates the manipulatives: the pile of tiles. There are buttons for each of the two teams to remove one or two tiles. There is an option to arrange the tiles into small groups, a useful way to visualize the game and its solution. Students sometimes discover this method while playing with the tiles on the table-top. There is an option to restart the game with an arbitrary number of tiles. Students often find that they can better analyze the game if they consider a simpler problem, with only a few tiles. Finally, there is a record of the moves played, since in the face-to-face regime students typically write down the sequences of moves for study.

The current status of this COMPS plug-in is that students can play Poison, the teacher can monitor all the ongoing conversations in the computer lab, and the teacher can intervene. The computer is not yet monitoring the conversation.

First Usage

Setup

In November 2011 students in an elementary education mathematics course used the COMPS version of the Poison exercise. These were not Math 110 students, but education students who would normally engage in quantitative literacy classroom exercises as part of both learning the mathematics and experiencing how it is taught.

Twenty-five students were arranged in six groups in a computer lab so that group members were not near each other and verbal conversations were discouraged. The students were accustomed to working in groups sitting around a table. Keyboard chat was a new element. Each student was given a copy of the problem. The instructor logged in as a member of each group so that she could monitor and contribute to the conversations. A sample from a conversation is shown Figure 3. The session ran for approximately 40 minutes, at which time the students stopped where they were and gathered together offline to share notes for their written reports.

A	How?
//D	If you take 2, then whatever you do on the next turn, you can do the opposite to leave 1.
B	If you take 1 or 2, then you can take 1 or 2 to counter balance that//
A	OK
C	OK
//C	So if I take 2, whatever they do ...
B	So basically if the other team ends up 4 left, then you can win. //
D	Yes
B	And that's if the other team ends up with 4 left
B	OK
A	We could maybe abbreviate opponent as OPP or something. Whatever, you might be writing a lot.
B	So yeah. um
	(*sounds of mumbling*)
C	Ok. Um
B	Oh boy
A	We don't need grammar.
B	Um so, if they 4 left you can win have how can you get it so that ..
D	If you have 5 or 6 on your turn, you can either take 1 or two to get it to that situation.
B	Ok you you want to get to 4, that's kind of a stable point where you can force them

Figure 4: In-Person Poison Dialogue

Observations

Both from experience observing Poison exercises, and from prior audiotaped sessions, differences between COMPS-mediated and in-person versions of Poison were evident.

- The COMPS students spent considerable time off-task, chatting about things not related to the problem. From the start, when students were logging in and greeting each other, it took some time for them to focus on the problem. Off-task conversation was almost negligible in our audio tapes, and not extensively observed in the classroom before the problem is solved.
- The COMPS groups spent much time playing the game for entertainment value, without advancing toward the goal of deducing whether a winning strategy existed.
- In the COMPS environment there was more team rivalry between the two teams within a group. There was even an instance where a student was reluctant to share the winning strategy with the rest of the group.

A consequence of all three of these behaviors is that incidences of shared construction of the winning strategy are less often observed in the COMPS transcripts, compared to their transcribed verbal ones. Figure 4 (in-person) and Figure 3 (computer-mediated) illustrate the typical difference. The in-person group engages in long exchanges where group cognition is evident. In the computer-mediated group the students rarely engage with each other for more than several turns at a stretch.

The student experience

Students were surveyed the next day in class. There were 8 Likert questions (strongly-disagree to strongly-agree) and 6 short-answer questions. The students told us the following.

- Using the computer seemed easy: 19 of the 25 students either agreed or strongly agreed.
- Students were divided over whether it was easier to play Poison on a computer than with tiles on a table.
- Eleven students were neutral with regard to whether it was easier to find a winning strategy for Poison on a computer than with tiles on a table, while 10 students either agreed or strongly agreed that the computer was easier.

  This finding stands in contrast with our observation that the computer-mediated groups were less successful in finding a winning strategy.
- Responding to open-ended questions, students enjoyed the chat option in COMPS and the fact that the activity was different from other class activities.
- On the other hand, when comparing using COMPS to solving problems face-to-face around a table, the students commented that it took time to type their ideas (which were sometimes difficult to put into words) and they could not show things to the others.

  One student did comment that the chat environment made the student try to solve the problem individually rather than sharing the solution right away among the group members.
- Aspects of the Poison module were troublesome. Students were confused about the L/R buttons (they were for the two teams), they would have preferred images of tiles to the `@` symbol, and they found keeping up with the conversation difficult at times.

  This was congruent with our own observation of students using the interface. Images of tiles, and perhaps even a way to drag them with a mouse cursor, would be a better model for the manipulatives than the simple row of `@` symbols and buttons. It took students a while to learn to use the interface in this respect.
- The students would have liked to have a way to have a private chat between members of a team so that the other team could not see their conversation.

Other observations of student use of the interface:

- The physical tiles are limited to 20, but the computer placed no limit on virtual tiles. Combined with the Poison game's evident play value, this resulted in some COMPS groups playing longer games with more tiles than the physical-tiles groups do. Such games did not contribute to understanding.
- In person student groups picked and maintained teams a bit more readily. We think COMPS should allow students to pick a team, and have the software display the current team rosters.
- We observed students using the full-duplex chat communication constantly. They often do not take turns, and they react to the other students' developing turns as they are typed.

Studies in Computer Monitoring

The first COMPS application of intelligence is to figuratively look over the shoulder of the students as they work, then display a real-time summary for the instructor. We have initially approached this task by writing shallow text classifiers. The work in this section is described in an unpublished report [Dion et al. 2011].

Background

We created a preliminary set of categories and classifiers based on two sources of language data

- Tape-recorded dialogues of upper-class students working the poison exercise. Figure 4 shows an extract of recorded verbal interaction.
- Written reports of the Poison problem that Math 110 students provided in earlier semesters. These reports exhibit many of the mathematical realizations that student exhibit while solving the Poison problem, but none of the dialogue or problem-construction phenomena.

This work was completed before the initial collection of COMPS-supported Poison dialogues, so does not include the COMPS data.

For the COMPS Math 110 project we are concentrating first on identifying epistemic knowledge and social co-construction phenomena. This is congruent with the results of a study of the criteria that teachers use for assessing student collaborative efforts [Gweon et al. 2011]. We categorized the dialogue data according to the following types of phenomena we deemed useful for real-time assessment along these axes:

- Bits of knowledge: domain-specific realizations that are either needed or characteristically occur during the path toward solving the problem.
- Varieties of student activities that were on-task but not part of the cognitive work of constructing the solution: e.g. picking sides, clarifying rules, playing the game.
- Student utterances related to constructing a solution: e.g. making observations, hypothesizing, wrong statements.
- Off-task statements, filler.

Altogether we annotated the student utterances with 19 categories, shown in Table 1. In this study, individual dialogue turns or sentences were assigned to one of these categories.

Experiment in machine classification

For our classifiers we chose two numerical methods: non-negative matrix factorization (NMF) and singular value decomposition (SVD). SVD is the most common numerical technique used in latent semantic analysis (LSA). Both of these methods rely on factoring a word-document co-occurrence matrix to build a semantic space: a set of dimensionality-reduced vectors. The training set for these experiments—the text used for building semantic spaces—was 435 sentences from the written corpus. The test sets

Table 1: Dialogue Categories from Poison Conversations

	Dialogue Category
1	4 tiles is important
2	2 and 3 are good tiles
3	You want to leave your opponent with 19 tiles
4	Going first gives you control of the game
5	You want to take 1 tile on your first move
6	1, 4, 7, 10, 13, 16, 19 are the poison numbers
7	"Opposite" strategy
8	"3 pattern"
9	Wrong statements
10	Exploring
11	Playing the game
13	Making an observation
14	Clarifying observations
15	Clarifying rules
16	Exploring further versions of the game
17	Hypothesizing
18	There is a winning strategy
19	Filler

were taken from approximately 100 sentences from the written corpus and 500 spoken dialogue turns. All our semantic spaces had 20 dimensions. Our feature sets included unigrams (individual words) and bigrams.

We report here on three computer-tagging methods: SVD, NMF-s, and NMF-u.

The **SVD** and **NMF-s** methods are supervised. They match test sentences to manually accumulated bundles of exemplar sentences. This technique is much the same as the latent semantic analysis algorithm used successfully by Auto-Tutor [Graesser et al. 2007].

In the NMF-s method the vector for a test sentence was built by solving a set of linear equations in 20 unknowns, which effectively computed what the vector for the test sentence would have been had that sentence been a part of the training set. We believe that this technique for using non-negative matrix factorization to build text classifiers is novel.

The **NMF-u** method is unsupervised. The reduced dimensions of the factored matrices are assumed to correspond directly to semantic dimensions within the data. This approach was described by [Segaran 2007] for classifiying blog posts. Our training documents (sentences) were sorted according to a) their manually-assigned category and b) which of the 20 dimensions in the NMF vector representation of the document had the largest value. The dimensions were then manually associated with individual tags, if possible.

Results

Table 2 summarizes the classification success rates of the two supervised methods, using unigram, bigram, and combined uni- and bi-gram feature spaces. We report the percentage of sentences that were correctly tagged from $n = 113$ test sentences. Test sentences represented all categories. Overall classification accuracy varied from 45% to 55%. Some categories occurred very infrequently in both the training and test corpora, resulting in very low success rates. Thus we also report the percent correct among the most common three categories in the test corpus: numbers 6, 11, and 15 in Table 1. Together these represented $n = 59$, more than half the test sentences.

A χ^2 test on tagging sentences in the top three categories shows that the computer tagging success rates are indeed not due to random chance. All values are significant at the $p < 0.05$ level and some at the $p < 0.01$ level. We found no consistent advantage to using unigrams, bigrams, or both together. In this our result is similar to [Rosé et al. 2008], where differences among these conditions are slight. That study of classifiers for collaborative learning dialogues evaluated its results using κ interrater reliability between human and computer annotaters. We have not computed κ, as the number of categories is large and the number of test sentences is small, rendering the statistic not very meaningful [Di Eugenio and Glass 2004].

In the NMF-u method many dimensions did not correlate with any tag. It was thus not capable of categorizing a test sentence into all the possible categories, leaving most of the categories unrecognized. Table 3 summarizes the most prominent categories that the NMF-u method found. For some of the most attested categories NMF-u was successful at correctly tagging the sentences in those categories, at the cost of a high rate of false positives. It had high recall but the precision was startlingly low.

Data Collection for Analysis

One of the benefits of COMPS is the ability to gather data on students, their interactions, and the exercise that they engage in.

An advantage of recording group problem-solving is that ordinary obligations and discourse pragmatics dictate that the participants signal when they achieve some understanding or some common ground. This means that not only are all the learnable knowledge components visible, but participants in the discussion should be making recognizable signs of whether the components are understood [Koschmann 2011]. In short, student thinking is forced out into the open in ways that an assessment test, a cognitive experiment, or a think-aloud protocol might never get at.

Our study of Poison collaborative dialogues [Dion et al. 2011] has already uncovered knowledge components that students realize and express *before* they arrive at a closed-form solution but are *not themselves* part of the solution. Examples are: 2 and 3 tiles force a win, 4 tiles is a simple completely-analyzable case. There is no good way besides observation to find out the ancillary realizations that students characteristically pass through as they explore the problem. And it is necessary to understand these ancillary realizations in order to assess the state of the knowledge-construction task.

Conclusions and Future Work

COMPS is being developed with several uses in mind, viz: a platform for student collaborative exploratory learning us-

Table 2: Accuracy of Supervised Classifiers

	% Correct			Top 3 Tags		
	All Tags	Top 3 Tags	χ^2	Tab 6	Tag 11	Tag 15
	$n = 113$	$n = 59$	p value	$n = 19$	$n = 13$	$n = 27$
NMF-s Unigrams	47%	61%	.003	58%	31%	78%
NMF-s Bigrams	45%	58%	.027	37%	38%	81%
NMF-s Both	48%	64%	.024	52%	54%	78%
SVD Unigrams	51%	66%	.0002	52%	86%	85%
SVD Bigrams	55%	68%	.028	63%	15%	96%
SVD Both	53%	59%	.003	42%	0%	100%

Table 3: Unsupervised NMF Classifier Results

	Class	N	Correctly classified	False positives
Unigrams	#7 Opposite Strategy	13	13 (100%)	63
	#6 Poison Numbers	13	12 (92%)	2
Unigrams no-stopwords	#15 Clarifying Rules	27	16 (59%)	8
	#1 Four Tiles Important	9	5 (56%)	15
Bigrams	#7 Opposite Strategy	13	11 (85%)	19
	#15 Clarifying Rules	27	23 (85%)	23
	#6 Poison Numbers	13	12 (92%)	10
	#1 Four Tiles Important	9	5 (56%)	15

ing problem-specific affordances, computer-aided monitoring and assessment of these dialogues, and recording dialogues for study. Its first large deployment will be for Math 110, a quantitative literacy class at VU.

First use with 25 students students exercising the Poison exercise in six teams shows that COMPS is quite usable. What seemed like a fairly straightforward translation of the Poison exercise manipulatives to software affordances will, however, benefit from updating and experimentation.

Analyzing dialogues collected before COMPS, we have identified a number of categories of dialogue contribution that will be useful in monitoring and assessing the dialogue. With regard to epistemic knowledge in the Poison problem domain, we have identified realizations that students pass through on the way toward building the final solution. These realizations may not appear in the final solution, but having students engage in dialogue and team cognition seems to successfully force the cognitive processes into the open.

We have classifiers based on latent semantic analysis and non-negative matrix factorization that can recognize a few of the most important of these epistemic categories in solving the Poison exercise. One of our classifiers relies on a somewhat novel method of using NMF. It entails discovering where a test sentence would be in the factor matrices by solving a system of linear equations. It performed about as well as LSA on our data set, but more testing would be needed. Our classifiers are trained on student written reports, we expect that accuracy will improve once we train them on student dialogue data.

Regarding the usability of the interface for problem-solving exercises, the primary unexpected behavior that we will address in future tests is the increase (compared to in-person exercises) in off-task activity and concomitant decrease in shared construction of the answer. Certain updates, such as making the interface more explanatory and reducing the maximum number of tiles, may reduce the evidently enhanced play value provided by the computer mediated environment. Also specifically addressing this goal we have two improvements on offer:

- Unlike earlier Wooz exercises, the Poison problem prompt was not on permanent display in a COMPS window. The students have it on paper. Possibly putting the problem on display will serve to keep the students more on-task. In short, we may be suffering the consequence of not following our own COMPS interaction model strictly enough.
- In Math 110 team members are assigned roles. For example one student is a moderator, one is a reflector, and so on. These are not represented in the COMPS interface. Possibly displaying which students are assigned to which role will foster more focused interactions.

We note that in addition to the epistemic tags, teachers have been found to evaluate student collaborative activities on a number of axes such as goal-setting, division of labor, and participation [Gweon et al. 2011] [Gweon et al. 2009]. Accordingly, we have been annotating our dialogues using the VMT threaded markup scheme [Strijbos 2009] which shows when a turn addresses previous turns and annotates the discourse relationship between them. Future work on the text classifiers needs to address these discourse relations. The VMT Chat interface [MathForum 2012b] permits users to explicitly link their dialogue utterances: a user can indicate that a particular dialogue turn responds to a different, earlier, turn, possibly uttered by somebody else. COMPS does not have this functionality, but it might be useful.

Acknowledgments

This work could not have been done without our hardworking, patient, and resourceful students. Special thanks to Nicole Rutt, Lisa Dion, and Jeremy Jank from the 2011 VU mathematics REU who worked on analyzing and classifying Poison dialogues, Scott Ramsey from NC A&T who designed and implemented much of COMPS in Actionscript, and Bryan Lee who helped update the server.

This work is supported by the National Science Foundation REESE program under awards 0634049 to Valparaiso University and 0633953 to North Carolina A&T State University and the NSF REU program under award 0851721 to Valparaiso University. The content does not reflect the position or policy of the government and no official endorsement should be inferred.

References

R. De Groot, R. Drachman, R. Hever, B. Schwarz, U. Hoppe, A. Harrer, M. De Laat, R. Wegerif, B. M. McLaren, and B. Baurens. Computer supported moderation of e-discussions: the ARGUNAUT approach. In Clark Chinn, Gijsbert Erkens, and Sadhana Puntambekar, editors, *Mice, Minds, and Society: The Computer Supported Collaborative Learning (CSCL) Conference 2007*, pages 165–167. International Society of the Learning Sciences, 2007.

Barbara Di Eugenio and Michael Glass. The kappa statistic: A second look. *Computational Linguistics*, 32:95–101, 2004.

Lisa Dion, Jeremy Jank, and Nicole Rutt. Computer monitored problem solving dialogues. Technical report, Mathematics and CS Dept., Valparaiso University, July 29 2011. REU project.

Barbara Fox. *The Human Tutoring Dialogue Project*. Erlbaum, Hillsdale, NJ, 1993.

Rick Gillman. A case study of assessment practices in quantitative literacy. In *Current Practices in Quantitative Literacy*, MAA Notes 70, pages 165–169. Mathematical Association of America, 2006.

Michael Glass, Jung Hee Kim, Karen Allen Keene, and Kathy Cousins-Cooper. Towards Wooz-2: Supporting tutorial dialogue for conceptual understanding of differential equations. In *Eighteenth Midwest AI and Cognitive Science Conference (MAICS-2007), Chicago*, pages 105–110, 2007.

Art Graesser, Phanni Penumatsa, Matthew Ventura, Zhiqiang Cai, and Xiangen Hu. Using LSA in AutoTutor: Learning through mixed-initiative dialogue in natural language. In Thomas K. Landauer, Danielle S. McNamara, Simon Dennis, and Walter Kintsch, editors, *Handbook of Latent Semantic Analysis*, pages 243–262. Lawrence Erlbaum, 2007.

Gahgene Gweon, Rohit Kumar, Soojin Jun, and Carolyn P. Rosé. Towards automatic assessment for project based learning groups. In *Proceedings of the 2009 conference on Artificial Intelligence in Education*, pages 349–356, Amsterdam, 2009. IOS Press.

Gahgene Gweon, Soojin Jun, Joonhwan Lee, Susan Finger, and Carolyn Penstein Rosé. A framework for assessment of student project groups on-line and off-line. In Sadhana Puntambekar, Gijsbert Erkens, and Cindy E. Hmelo-Silver, editors, *Analyzing Interactions in CSCL*, volume 12, part 3 of *Computer-Supported Collaborative Learning Series*, pages 293–317. Springer, 2011.

Jung Hee Kim and Michael Glass. Evaluating dialogue schemata with the wizard of oz computer-assisted algebra tutor. In James C. Lester, Rosa Maria Vicari, and Fábio Paraguaçu, editors, *Intelligent Tutoring Systems, 7th International Conference, Maceió, Brazil*, volume 3220 of *Lecture Notes in Computer Science*, pages 358–367. Springer, 2004.

Tim Koschmann. Understanding understanding in action. *Journal of Pragmatics*, 43, 2011.

MathForum. Math notation in email messages or web forms. Web help page from Math Forum: Virtual Math Teams project, 2012a. URL `http://mathforum.org/typesetting/email.html`.

MathForum. VMT software orientation. Web help page from Math Forum: Virtual Math Teams project, 2012b. URL `http://vmt.mathforum.org/vmt/help.html`.

Niraj Patel, Michael Glass, and Jung Hee Kim. Data collection applications for the NC A&T State University algebra tutoring dialogue (Wooz tutor) project. In Anca Ralescu, editor, *Fourteenth Midwest Artificial Intelligence and Cognitive Science Conference (MAICS-2003)*, pages 120–125, 2003.

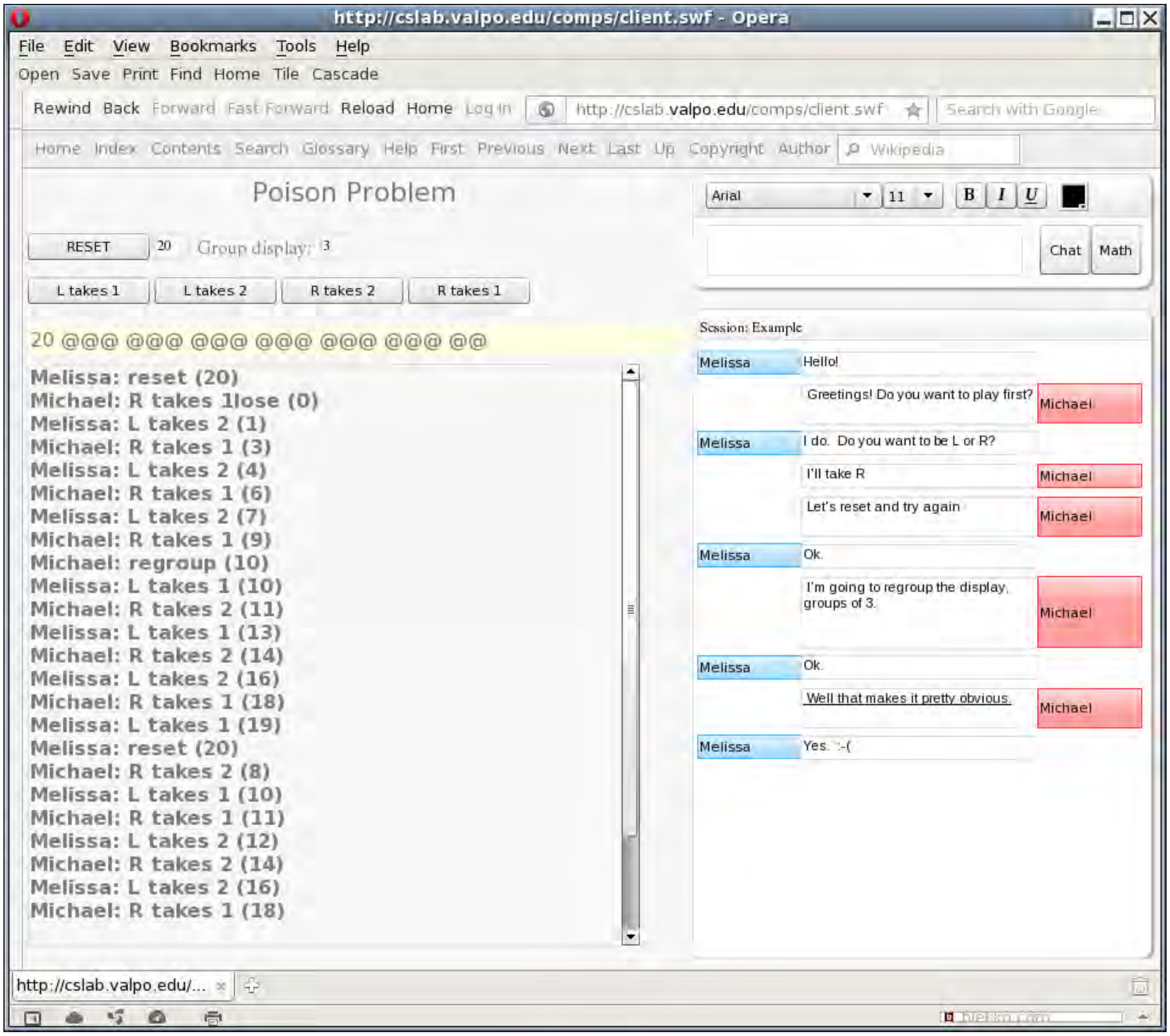
Carolyn Rosé, Yi-Chia Wang, Yue Cui, Jaime Arguello, Karsten Stegmann, Armin Weinberger, and Frank Fischer. Analyzing collaborative learning processes automatically: Exploiting the advances of computational linguistics in CSCL. *Int. J. of Computer-Supported Collaborative Learning*, 3(3), 2008.

H. Sacks, E.A. Schegloff, and G. Jefferson. A simplest systematics for the organization of turn-taking for conversation. *Language*, pages 696–735, 1974.

Toby Segaran. *Programming Collective Intelligence*. O'Reilly, 2007.

Gerry Stahl. *Studying Virtual Math Teams*. Springer, 2009.

Jan-Willem Strijbos. A multidimensional coding scheme for VMT. In Gerry Stahl, editor, *Studying Virtual Math Teams*, chapter 22. Springer, 2009.


http://cslab.valpo.edu/comps/client.swf - Opera
File Edit View Bookmarks Tools Help
Open Save Print Find Home Tile Cascade
Rewind Back Forward Fast Forward Reload Home Log in
http://cslab.valpo.edu/comps/client.swf
Search with Google
Home Index Contents Search Glossary Help First Previous Next Last Up Copyright Author
Wikipedia
Poison Problem
RESET
20
Group display: 3
L takes 1
L takes 2
R takes 2
R takes 1
Arial
11
B
I
U
Chat
Math
http://cslab.valpo.edu/...

20 @@@ @@@ @@@ @@@ @@@ @@@ @@

```
Melissa: reset (20)
Michael: R takes 1lose (0)
Melissa: L takes 2 (1)
Michael: R takes 1 (3)
Melissa: L takes 2 (4)
Michael: R takes 1 (6)
Melissa: L takes 2 (7)
Michael: R takes 1 (9)
Michael: regroup (10)
Melissa: L takes 1 (10)
Michael: R takes 2 (11)
Melissa: L takes 1 (13)
Michael: R takes 2 (14)
Melissa: L takes 2 (16)
Michael: R takes 1 (18)
Melissa: L takes 1 (19)
Melissa: reset (20)
Michael: R takes 2 (8)
Melissa: L takes 1 (10)
Michael: R takes 1 (11)
Melissa: L takes 2 (12)
Michael: R takes 2 (14)
Melissa: L takes 2 (16)
Michael: R takes 1 (18)
```

Session: Example

Melissa: Hello!

Michael: Greetings! Do you want to play first?

Melissa: I do. Do you want to be L or R?

Michael: I'll take R

Michael: Let's reset and try again

Melissa: Ok.

Michael: I'm going to regroup the display, groups of 3.

Melissa: Ok.

Michael: Well that makes it pretty obvious.

Melissa: Yes. :-(

Figure 1: COMPS with Poison problem.

The people in each group are to form two teams. One team will play against the other team in the group. To begin, place 20 tiles between the two teams. Here are the rules:

1. Decide which team will play first.
2. When it is your team's turn, your team is to remove 1 or 2 tiles from the pile.
3. The teams alternate taking turns.
4. The team that is forced to take the last tile – the poison tile – loses the game.

Play this game a number of times, alternating which team plays first. As you play these games, keep track of your moves/choices. Eventually, you want to be able to determine how your team should play to force the other team to lose. In order to make this determination, you will need to look for a pattern. In order to find a pattern, you will need data, and so you will need to decide how to collect and organize these data to see if a pattern will appear.

Figure 2: Poison Assignment

This page is intentionally left blank

Genetic Algorithms and Agent Systems

Chair: Dan Ralescu

Genetic Algorithm Applied to the Graph Coloring Problem

Musa M. Hindi and Roman V. Yampolskiy

Computer Engineering and Computer Science
J.B. Speed School of Engineering
Louisville, Kentucky

Abstract

In this paper we present a hybrid technique that applies a genetic algorithm followed by wisdom of artificial crowds approach to solving the graph-coloring problem. The genetic algorithm described here utilizes more than one parent selection and mutation methods depending on the state of fitness of its best solution. This results in shifting the solution to the global optimum more quickly than using a single parent selection or mutation method. The algorithm is tested against the standard DIMACS benchmark tests while limiting the number of usable colors to the known chromatic numbers. The proposed algorithm succeeded at solving the sample data set and even outperformed a recent approach in terms of the minimum number of colors needed to color some of the graphs.

The Graph Coloring Problem (GCP) is a well-known NP-complete problem. Graph coloring includes both vertex coloring and edge coloring. However, the term graph coloring usually refers to vertex coloring rather than edge coloring.

Given a number of vertices, which form a connected graph, the objective is to color each vertex such that if two vertices are connected in the graph (i.e. adjacent) they will be colored with different colors. Moreover, the number of different colors that can be used to color the vertices is limited and a secondary objective is to find the minimum number of different colors needed to color a certain graph without violating the adjacency constraint. That number for a given graph (G) is known as the Chromatic Number (χ(G)) (Isabel Méndez Díaz and Paula Zabala 1999).

If $k = \{1, 2, 3...\}$ and $P(G, k)$ is the number of possible solutions for coloring the graph G with k colors, then

$$\chi(G) = min(k: P(G, k) > 0) \qquad (1)$$

Graph coloring problems are very interesting from the theoretical standpoint since they are a class of NP-complete problems that also belong to Constraint Satisfaction Problems (CSPs). The practical applications of Graph Coloring Problems include but are not limited to:

- Map coloring (B. H. Gwee, M. H. Lim and J. S. Ho 1993)
- Scheduling (Daniel Marx and D Aniel Marx 2004)
- Radio Frequency Assignment (W. K. Hale 1980; S. Singha, T. Bhattacharya and S. R. B. Chaudhuri 2008)
- Register allocation (Wu Shengning and Li Sikun 2007)
- Pattern Matching
- Sudoku

In this paper we demonstrate the use of genetic algorithms in solving the graph-coloring problem while strictly adhering to the usage of no more than the number of colors equal to the chromatic index to color the various test graphs.

Prior Work

A great deal of research has been done to tackle the theoretical aspects of the Graph Coloring Problem in terms of its generalization as a Constraint Satisfaction Problem (Isabel Méndez Díaz and Paula Zabala 1999). The problem's various applications and solutions have been discussed in detail in Porumbel's paper (Daniel Cosmin Porumbel 2009). Evolutionary computation and parameter control has been detailed in a number of papers including ones by Back, Hammel, and Schwefel (T. Back, U. Hammel and H. P. Schwefel 1997) as well as work by Eiben, Hinterding and Michalewicz (A. E. Eiben, R. Hinterding and Z. Michalewicz 1999). Srinivas and Patnaik examined crossover and mutation probabilities for optimizing genetic algorithm performance (M. Srinivas and L. M. Patnaik 1994). Genetic algorithms and evolutionary approaches have been used extensively in solutions for the Graph Coloring Problem and its applications (F. F. Ali, et al. 1999; K. Tagawa, et al. 1999; Cornelius Croitoru, et al. 2002; C. A. Glass and A. Prugel-Bennett 2003; Justine W. Shen 2003; Greg Durrett, Muriel Médard and Una-May O'Reilly 2010; Lixia Han and Zhanli Han 2010). Most recent work utilized a parallel genetic algorithm on a similar dataset to the one used in this paper (Reza Abbasian and Malek Mouhoub 2011).

The concept of utilizing a crowd of individuals for solving NP complete problems has also been the topic of various papers. Most notably the Wisdom of Crowds concept has been used in solving the Traveling Salesman Problem (Sheng Kung Michael Yi, et al. 2010b) as well as the Minimum Spanning Tree Problem (Sheng Kung Michael Yi, et al. 2010a). In this paper we attempt to supplement the solution produced by the genetic algorithm utilizing an artificial crowd (Leif H. Ashby and Roman V. Yampolskiy 2011; Roman V. Yampolskiy and Ahmed El-Barkouky 2011).

Proposed Approach

Genetic algorithms share an overall structure and workflow yet they vary in the specific details according to the particular problem. The algorithm consists of a parent selection method, a crossover method and a mutation method.

The general algorithm is:

Algorithm1: General Genetic Algorithm
define: population, parents, child

```
population = randomly generated chromosomes;

while (terminating condition is not reached) {
    gaRun();
}

// a single run of a genetic algorithm
function gaRun() {
    parents = getParents();
    child = crossover(parents);
    child = mutate(child);
population.add(child);
}
```

The goal of the previous algorithm is to improve the fitness of the population by mating its fittest individuals to produce superior offspring that offer a better solution to the problem. This process continues until a terminating condition is reached which could be simply that the total number of generations has been run or any other parameter like non-improvement of fitness over a certain number of generations or that a solution for the problem has been found.

The chromosome representation of the GCP is simply an array with the length set to the number of vertices in the graph. Each cell in the array is assigned a value from 0 to the number of colors – 1. The adjacencies between the vertices are represented by an adjacency matrix of dimensions $n \times n$ where n: the number of vertices.

Figure 1 displays the chromosome representation of a graph of 7 vertices using 4 colors:

The ultimate goal when solving GCPs is to reach a solution where no two adjacent vertices have the same color. Therefore, the GA process continues until it either finds a solution (i.e. 0 conflicts) or the algorithm has been run for the predefined number of generations. In addition to the GA, if a run fails to reach a solution a Wisdom of Crowds approach will be applied to the top performers in an attempt to produce a better solution.

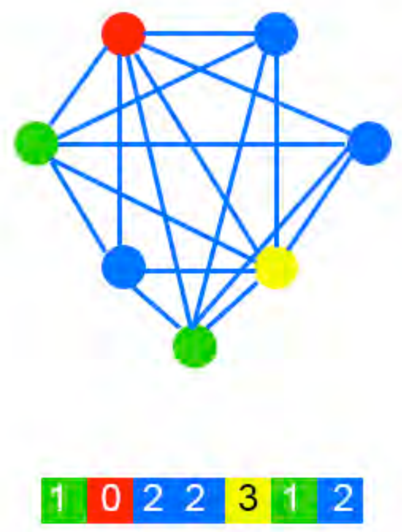


Figure 1: Chromosome representation of a colored connected graph

The overall genetic algorithm in this approach is a generational genetic algorithm. The population size is kept constant at all times and with each generation the bottom performing half of the population is removed and new randomly generated chromosomes are added. The population size is set to 50 chromosomes. The value was chosen after testing a number of different population sizes. The value 50 was the least value that produced the desired results.

The GA uses two different parent selection methods, a single crossover method and two different mutation methods. Which of the parent selection and mutation methods ends up selected depends on the state of the population and how close it is to finding a solution. The parent selection, crossover and mutation methods are outlined as follows:

```
Algorithm2: parentSelection1:
define: parent1, parent2, tempParents;

tempParents = two randomly selected chromosomes from the population;
parent1 = the fitter of tempParents;

tempParents = two randomly selected chromosomes from the population;
parent2 = the fitter of tempParents;

return parent1, parent2;
```

```
Algorithm3: parentSelection2:
define: parent1, parent2

parent1 = the top performing chromosome;
parent2 = the top performing chromosome;

return parent1, parent2;
```

```
Algorithm4: crossover
define: crosspoint, parent1, parent2, child

crosspoint = random point along a chromosome;
child = colors up to and including crosspoint from parent 1 + colors after crosspoint to the end of the chromosome from parent2;

return child;
```

```
Algorithm5: mutation1:
define: chromosome, allColors, adjacentColors, validColors, newColor;

for each(vertex in chromosome) {
if (vertex has the same color as an adjacent vertex) {
            adjacentColors = all adjacent colors;
            validColors = allColors – adjacentColors;

            newColor  = random color from validColors;
            chromosome.setColor(vertex, newColor)
   }
}
return chromosome;
```

```
Algorithm6: mutation2:
define: chromosome, allColors

for each(vertex in chromosome) {
   if (vertex has the same color as an adjacent vertex) {
            newColor  = random color from allColors;
            chromosome.setColor(vertex, newColor)
   }
}
return chromosome;
```

A bad edge is defined as an edge connecting two vertices that have the same color. The number of bad edges is the fitness score for any chromosome. As mentioned above, the alteration between the two different parent selection and mutation methods depends on the best fitness. If the best fitness is greater than 4 then parentSelection1 and mutation1 are used. If the best fitness is 4 or less then parentSelection2 and mutation2 are used. This alteration is the result of experimenting with the different data sets. It was observed that when the best fitness score is low (i.e. approaching an optimum) the usage of parent selection 2 (which copies the best chromosome as the new child) along with mutation2 (which randomly selects a color for the violating vertex) results in a solution more often and more quickly than using the other two respective methods.

Finally, the algorithm is run for 20,000 generations or until a solution with 0 bad edges is found. If a solution is not found after 20,000 generations the wisdomOfArtificialCrowds algorithm is run. The algorithm used is a localized wisdom of crowds algorithm that only builds a consensus out of the violating edges in the best solution. Moreover, it uses the best half of the final population to produce an aggregate solution. Only a localized consensus is generated so as not to produce a result that alters the correctly colored vertices. Also, it takes the best half because they share the most similarity and thus will most likely be different at the level of the bad edges rather than the good ones.

```
Algorithm7: wisdomOfArtificialCrowds
define: aggregateChromosome, newColor, expertChromosomes;

expertChromosomes = best half of the final population;
aggregateChromosome = best performing chromosome;

for each (vertex in graph) {
   if (vertex is part of a bad edge) {
      newColor = most used color for vertex among expertChromosomes;
      aggregateChromosome.setColor(vertex, newColor)
   }
}
```

Data

Data used to test our approach are derived from the DIMACS benchmarking graph collection. DIMACS is the Center for Discrete Mathematics and Theoretical Computer Science. It is part of Rutgers University (The State University of New Jersey Rutgers, et al. 2011). The data are frequently used in challenges involving constraint satisfaction problems. The files used have a .col extension. Each file contains a header with the number of vertices (p) and the number of edges (e):

```
p edge 496 11654
```

A number of lines follow the header with each line denoting the connected edges and their vertex indices:

```
e 1 100
```

16 files were chosen from the DIMACS collection. The graphs the files represent vary in vertex count, edge count and overall complexity. The vertex count ranges between 11 and 561 and the edge count ranges from 20 to 11654. The rationale behind the selection of these graphs other than the wide range of variation is that there is a known chromatic number for each of them, or at least a good approximation.

The following files were used in this approach: (myciel3.col, myciel4.col, myciel5.col, queen5_5.col, queen6_6.col, queen7_ 7.col, queen8_8.col, huck.col, jean.col, david.col. games120.col, miles250.col, miles1000.col, anna.col, fpsol2.i.1.col, homer.col).

Results

Table 1 displays the following statistics for each file:

- The number of vertices |V|
- The number of edges |E|
- The expected chromatic number *χ(G)*
- The minimum number of colors used by this algorithm k_{min}
- The minimum number of colors used by a comparative publication using a Hybrid Parallel Genetic Algorithm (HPGAGCP)
- Average time it took to find a solution

The genetic algorithm was developed in Java utilizing JDK 1.6 and JUNG (Java Universal Network/Graph) framework for graph visualization (Joshua O'Madadhain, Danyel Fisher and Tom Nelson 2011). Performance plots were generated using MATLAB R2010a.

The tests were run on a desktop PC with the following specifications:

CPU: Intel Core i7 860 @2.8Ghz
RAM: 8 GB DDR3 @1333MHz

File	|V|	|E|	Expected χ(*G*)	k_{min}	HPGAGCP Result	Time (s)
myciel3.col	11	20	4	4	4	0.003
myciel4.col	23	71	5	5	5	0.006
queen5_5.col	23	160	5	5	5	0.031
queen6_6.col	**25**	**290**	**7**	**7**	**8**	**6.100**
myciel5.col	36	236	6	6	6	0.014
queen7_7.col	**49**	**476**	**7**	**7**	**8**	**6.270**
queen8_8.col	**64**	**728**	**9**	**9**	**10**	**47.482**
huck.col	74	301	11	11	11	0.015
jean.col	80	254	10	10	10	0.015
david.col	87	406	11	11	11	0.019
games120.col	120	638	9	9	9	0.027
miles250.col	128	387	8	8	8	0.076
miles1000.col	128	3216	42	42	42	48.559
anna.col	138	493	11	11	11	0.058
fpsol2.i.1.col	496	11654	65	65	65	22.656
homer.col	561	1629	13	13	13	0.760

Table 1: Results of running the proposed algorithm on 16 .col files from the DIMACS collection

The following graphs plot the relationship between the fitness and the generation for a sample set of the files used:

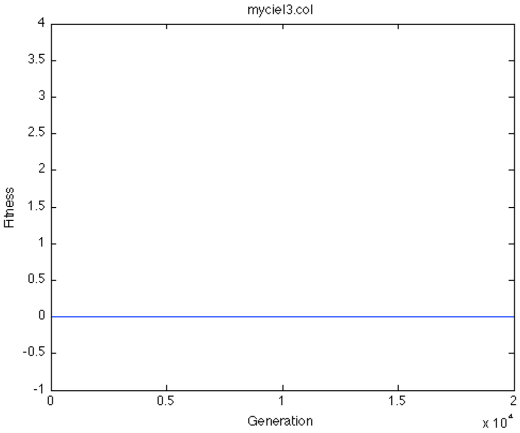


Figure 2: Fitness score over the number of generations for myciel3.col

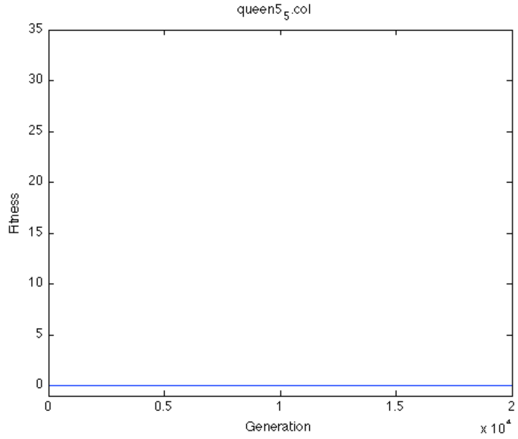


Figure 3: Fitness score over the number of generations for queen5_5.col

Figures 2 and 3 are not very interesting since the solutions are found after a few generations. The next plot, however, is of particular interest since it clearly represents the erratic behavior of the fitness score between the initial rapid drop until a solution is ultimately found. In standard genetic algorithms the fitness score continues to increase or decrease (depending of the definition of better fitness) until the end of the run. This is not the case here. This factor plays a huge role in obtaining the global optimum with a higher probability than without it as will be discussed later.

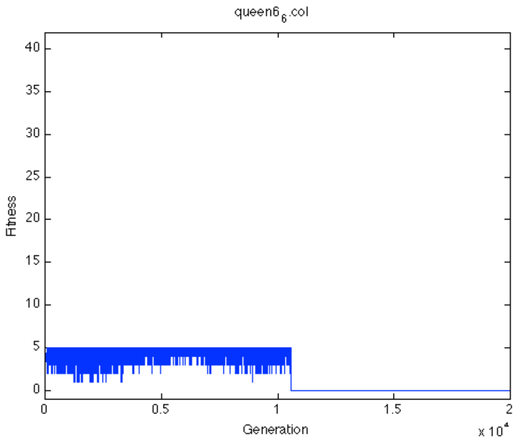
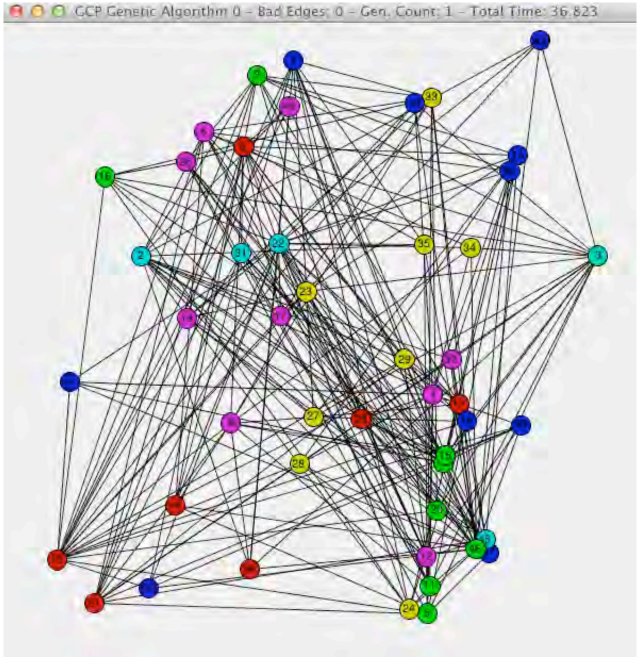


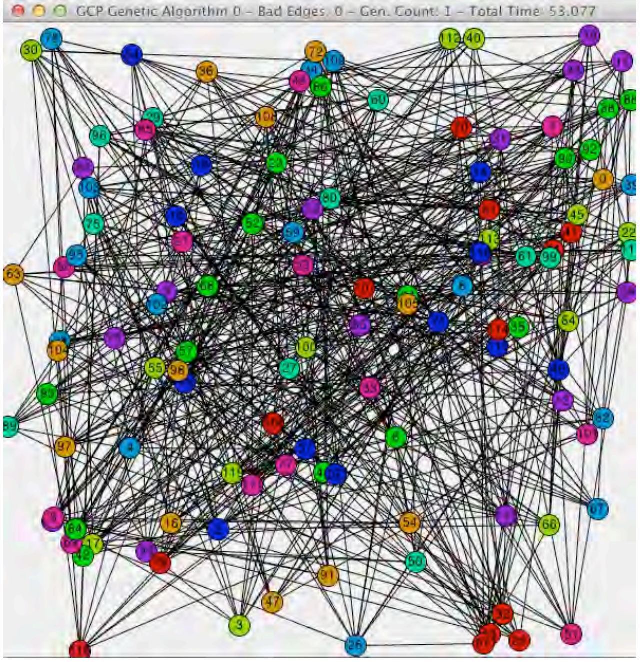
Figure 4: Fitness score over the number of generations for queen6.col

Sample solutions: (graphs were visualized using the JUNG framework (Joshua O'Madadhain, Danyel Fisher and Tom Nelson 2011)):



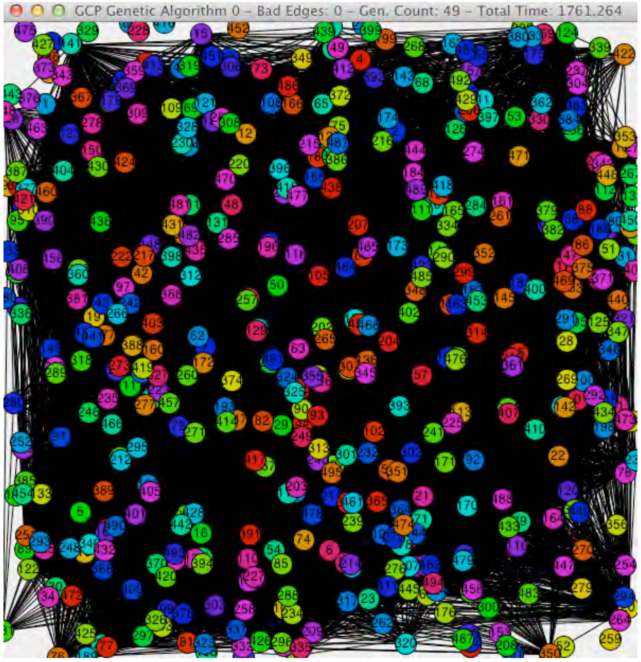
GCP Genetic Algorithm 0 – Bad Edges: 0 – Gen. Count: 1 – Total Time: 36.823

Figure 5: GCP solution for myciel5.col



GCP Genetic Algorithm 0 – Bad Edges: 0 – Gen. Count: 1 – Total Time: 53.077

Figure 6: GCP solution for games120.col



GCP Genetic Algorithm 0 – Bad Edges: 0 – Gen. Count: 49 – Total Time: 1761.264

Figure 7: GCP solution for fpsol2.i.col

Discussion

During the design of this approach, the issue of designing good operators was a constant concern. The goal of any good operator is to bring the chromosomes of a population closer to the desired solution. However, during the process, a chromosome's fitness often improves but eventually ends up in a local optimum. The overall design of this approach aimed to improve fitness towards the global optimum while avoiding the pitfalls of local optima.

To achieve that, a number of factors need to be considered. Initially, the crossover function is applied to parents that result from the first parent selection method. This method selects parents by conducting a small tournament between random pairs of chromosomes. Two pairs are chosen randomly and the fitter of each pair becomes a parent. These fit parents are then used as input to this crossover method. The crossover conducts a simple one-point crossover with the cross point being chosen at random. The result of this crossover is then subjected to the first mutation method. The mutation is carried out at a high rate of 0.7. This mutation examines each vertex and if a vertex violates the coloring constraint a valid color is chosen at random.

This process is very effective in reducing the number of conflicts rapidly which can be seen in all the plots through an almost perpendicular drop in fitness. However, in spite of this method's effectiveness at increasing fitness rapidly, it has the side effect of keeping the solution at a local optimum.

To fix that, another parent selection and mutation method is introduced. The two methods are applied when the overall fitness of the best solution drops below 5 conflicts. After that point crossover is no longer applied. The top performer is selected and is subjected to the second mutation method. This method finds the conflicting vertices and replaces their conflicting colors with random colors; which could be invalid as well. This has the potential to either find a globally optimum solution (i.e. 0 conflicts) or produce a solution that is worse! This can be observed by the erratic pattern in some of the graphs after the sharp descent and before the actual resolution of the problem.

This seemingly worsening fitness is not bad however. In fact, it is partly due to this worsening of the fitness that some solutions are found at all! When the solution becomes worse the fitness score increases. This will force the algorithm back to using the first parent selection and mutation methods. But, the population now contains a solution that hadn't been there before, which increases the possibility of reaching the global optimum. The continuous back and forth between parent selection and mutation methods plays a crucial role in shifting the solution from a local optimum to a global optimum.

Finally, if a solution is not found after 20,000 generations, a Wisdom of Artificial Crowds algorithm is applied to the resultant population to produce a better solution. The genetic algorithm had been producing optimal results and thus, per the algorithmic workflow, the Wisdom of Artificial Crowds postprocessor wouldn't be applied. However, in order to test its effectiveness, a test was conducted by decreasing the generation count to 10,000 to intentionally produce a suboptimal solution. The test was carried out on the miles1000.col file. Before the postprocessor was applied the best solution had 5 conflicting edges. After application of the Wisdom of Artificial Crowds postprocessor the graph was colored slightly differently but still had 5 conflicting edges. It is worth noting that the postprocessor added an average of 250 ms to the overall process.

The test cases used were very useful in both testing and tuning the algorithm. The algorithm was able to scale across the different graphs and produce optimum solutions in each case. A recent 2011 publication presented a parallel genetic algorithm for the GCP (Reza Abbasian and Malek Mouhoub 2011). This paper used most of the same test files that were used in Abbasian's approach. That approach's proposed algorithm failed to solve three of the graphs and only produced solutions when the color count was increased by 1. The files representing these graphs are queen6_6.col, queen7_7.col and queen8_8.col. The algorithm used in our approach successfully solved these three files using the specified known chromatic number as the number of colors used. Our approach is also generally faster at finding a solution. It is faster in all cases except four. Three of those cases are the three aforementioned files, which the comparative method did not succeed at finding a solution using the known chromatic index. The fourth case is miles1000.col.

Conclusion

The overarching algorithm used in this approach is a genetic algorithm with a subsequent Wisdom of Crowds post-processor. Within the genetic algorithm itself is a set of operators that utilize methods from the genetic algorithm domain as well as applying various heuristics in a stochastic manner. The end result is an quick progressive climb to the peak of the globally optimum solution.

The algorithms described here can also be applied to the various subsets of the general GCP. In particular, Sudoku can benefit from these algorithms where it can be represented as a graph with 81 vertices that must be colored using no more than 9 different colors (i.e. different numbers).

References

Abbasian, Reza, and Mouhoub, Malek. 2011. An efficient hierarchical parallel genetic algorithm for graph coloring problem. In Proceedings of The 13th annual conference on Genetic and evolutionary computation, 521-528. Dublin, Ireland: ACM

Ali, F. F., Nakao, Z., Tan, R. B., and Yen-Wei, Chen. 1999. An evolutionary approach for graph coloring. In Proceedings of The International Conference on Systems, Man, and Cybernetics, 527-532. IEEE

Ashby, Leif H., and Yampolskiy, Roman V. 2011, Genetic Algorithm and Wisdom of Artificial Crowds Algorithm Applied to Light Up, The 16th International Conference on Computer Games: AI, Animation, Mobile, Interactive Multimedia, Educational & Serious Games, Louisville, KY, USA: July 27 – 30, 2011

Back, T., Hammel, U., and Schwefel, H. P. 1997. Evolutionary computation: comments on the history and current state. IEEE Transactions on Evolutionary Computation 1: 3-17

Croitoru, Cornelius, Luchian, Henri, Gheorghies, Ovidiu, and Apetrei, Adriana. 2002. A New Genetic Graph Coloring Heuristic. In Proceedings of The Computational Symposium on Graph Coloring and its Generalizations, 63-74. Ithaca, New York, USA

Díaz, Isabel Méndez, and Zabala, Paula. 1999, A Generalization of the Graph Coloring Problem, Departamento de Computacion, Universidad de Buenes Aires.

Durrett, Greg, Médard, Muriel, and O'Reilly, Una-May. 2010. A Genetic Algorithm to Minimize Chromatic Entropy: 59-70, eds. P. Cowling and P. Merz, Springer Berlin / Heidelberg

Eiben, A. E., Hinterding, R., and Michalewicz, Z. 1999. Parameter control in evolutionary algorithms. IEEE Transactions on Evolutionary Computation 2: 124-141

Glass, C. A., and Prugel-Bennett, A. 2003. Genetic algorithm for graph coloring: Exploration of Galinier and Hao's algorithm. Journal of Combinatorial Optimization 3: 229-236

Gwee, B. H., Lim, M. H., and Ho, J. S. 1993. Solving four-colouring map problem using genetic algorithm. In Proceedings of First New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems, 332-333. New Zealand

Hale, W. K. 1980. Frequency assignment: Theory and applications. Proceedings of the IEEE 12: 1497-1514

Han, Lixia, and Han, Zhanli. 2010. A Novel Bi-objective Genetic Algorithm for the Graph Coloring Problem. In Proceedings of The 2010 Second International Conference on Computer Modeling and Simulation, 3-6. Sanya, China: IEEE Computer Society

Marx, Daniel, and Marx, D Aniel. 2004, Graph Coloring Problems and Their Applications in Scheduling, John von Neumann PhD Students Conference, Budapest, Hungary
O'Madadhain, Joshua, Fisher, Danyel, and Nelson, Tom. 2011, JUNG: Java Universal Network/Graph framework, Oct. 1 2011, http://jung.sourceforge.net/.

Porumbel, Daniel Cosmin. 2009. Heuristic Algorithms and Learning Techniques: Applications to the Graph Coloring Problem, Thesis Draft, Département Informatique, Université d'Angers

Rutgers, The State University of New Jersey, Research, AT&T Labs, Jersey, Cancer Institute of New, University, Princeton, Labs, Alcatel-Lucent Bell, America, NEC Laboratories, and Sciences, Applied Communication. 2011. Center for Discrete Mathematics and Theoretical Computer Science (DIMACS). Nov. 1, 2011. http://dimacs.rutgers.edu/

Shen, Justine W. 2003. Solving the Graph Coloring Problem using Genetic Programming, in Genetic Algorithms and Genetic Programming at Stanford 2003: 187-196, Stanford Bookstore

Shengning, Wu, and Sikun, Li. 2007. Extending Traditional Graph-Coloring Register Allocation Exploiting Meta-heuristics for Embedded Systems. In Proceedings of The Third International Conference on Natural Computation. ICNC, 324-329. Haikou, Hainan, China

Singha, S., Bhattacharya, T., and Chaudhuri, S. R. B. 2008. An Approach for Reducing Crosstalk in Restricted Channel Routing Using Graph Coloring Problem and Genetic Algorithm. In Proceedings of The International Conference on Computer and Electrical Engineering, 807-811. Phuket Island, Thailand

Srinivas, M., and Patnaik, L. M. 1994. Adaptive probabilities of crossover and mutation in genetic algorithms. IEEE Transactions on Systems, Man and Cybernetics 4: 656-667

Tagawa, K., Kanesige, K., Inoue, K., and Haneda, H. 1999. Distance based hybrid genetic algorithm: an application for the graph coloring problem. In Proceedings of Congress on Evolutionary Computation, 2332. Washington DC

Yampolskiy, Roman V., and El-Barkouky, Ahmed. 2011. Wisdom of Artificial Crowds Algorithm for Solving NP-Hard Problems. International Journal of Bio-Inspired Computation (IJBIC) 6: 358-369

Yi, Sheng Kung Michael, Steyvers, Mark, Lee, Michael D., and Dry, Matthew. 2010a. Wisdom of the Crowds in Minimum Spanning Tree Problems. In Proceedings of The 32nd Annual Conference of the Cognitive Science Society. Portland, Oregon

Yi, Sheng Kung Michael, Steyvers, Mark, Lee, Michael D., and Dry, Matthew J. 2010b. Wisdom of the Crowds in Traveling Salesman Problems. http://www.socsci.uci.edu/~mdlee/YiEtAl2010.pdf

A Two-tiered View on Acceptance

Joëlle Proust

Institut Jean-Nicod
Fondation Pierre-Gilles de Gennes pour la Science,
Ecole Normale Supérieure
29 Rue d'Ulm,
75005 Paris, France
joelle.proust@ehess.fr

Abstract

Experimental studies in metacognition indicate that a variety of norms are used by humans and some non-human agents to control and monitor their cognitive performances, such as accuracy, comprehensiveness, intelligibility, coherence, relevance, or consensus. This diversity of epistemic norms motivates a revision of the concept of acceptance. First, there are different forms of acceptance, corresponding to the specific epistemic norm(s) that constitute(s) them. Furthermore, acceptances need to include a strategic component, from which the epistemic component is insulated, whose function is to adjust the epistemic output to expected utility. Experimental evidence suggests that this two-tiered analysis of acceptance is empirically adequate. Relevance to AI is briefly discussed.

Acceptance and its Norms

Intelligent agency requires an ability to control and monitor one's cognitive states, e.g. retrieve memories, check one's perceptions or one's utterances. The aim of cognitive control is to acquire cognitively reliable properties, such as retrieving a correct answer. Intelligent agents in realistic settings, however, whether natural or artificial, need to track other epistemic norms beyond accuracy, such as the comprehensiveness of a list, the intelligibility of a text, the coherence of a story, the relevance of a remark, or the consensuality of a claim. Experimental studies in metacognition suggest that such norms are indeed used by human and some non-human agents to control and monitor their own cognitive performance (Goldsmith and Koriat, 2008, Couchman et al. 2010). Furthermore, the particular cognitive task in which performance is being monitored has been shown to dictate which epistemic norm is appropriate to a given context.

The goal of this article is to sketch a theory of acceptance that takes advantage of these studies. Acceptances, in contrast with beliefs, are generally recognized as voluntary (Jeffrey 1956, Stalnaker 1987, Bratman 1999, Lehrer 2000, Velleman 2000, Frankish, 2004). Accepting is an epistemic action, involving deliberation, i.e. various forms of cognitive control and their associated norms. There is no consensus, however, about the norm(s) of acceptances. While for Velleman (2000) accepting is regarding a proposition *P* as true, even though it may not be "really true", Cohen takes acceptance to be "a policy for reasoning, (..) the policy of taking it as a premise that *P*" (1992, 5, 7). For Stalnaker, "sometimes it is reasonable to accept something that one knows or believes to be false". Circumstances where this is reasonable include cases where *P* "may greatly simplify an inquiry", where *P* is "close to the truth", or "as close as one needs to get for the purposes at hand". Granting that accepted propositions are subject to contextual variation in their sensitivity to evidence and truth, they cannot be freely agglomerated in a coherence-preserving way, in contrast with beliefs (Stalnaker 1987). Finally, Bratman (1999) claims that acceptances conjoin epistemic and practical goals.

These features of acceptance, however, fail to offer an intelligible and coherent picture of the epistemic action of accepting, and of its role in practical reasoning and decision-making. First, it is left unclear how a context of acceptance is to be construed in a way that justifies applying fluctuating epistemic standards. Second, how can one possibly conjoin an *epistemic requirement*, which is essentially passively recognized and applied, and *utility considerations,* which require an active decision from the agent as to what ought to be accepted in the circumstances?

The Context Relevant to Accepting *P*

Why is accepting *contextual*, in a way that judging is not? Merely saying that acceptances, "being tied to action" (Bratman, 1999), are sensitive to practical reasoning, is not a viable explanation: other mental actions, such as judgments, also tied to action, do not adjust their contents to considerations of practical reasoning. Saying that they are context-dependent because coherence, consistency, and relevance apply within the confines of the existing plan, rather to a theoretical domain, does not explain how epistemic correctness survives instrumental adequacy.

Our first proposal consists in the following claim: Utility may dictate the norm of acceptance relevant to a given context of action, without dictating the output of the corresponding acceptance. As said above, accepting *P* can be driven, among other things, by a goal of comprehensiveness, accuracy, intelligibility, consensus or coherence. For example, you may accept that the shopping list you just reconstructed from memory is comprehensive (all the items previously listed are included), but not that it is accurate (new items are also mistakenly listed). On the proposed view, an acceptance is always indexed to its specific norm: a proposition is never merely accepted, it is rather accepted$_{at}$ or accepted$_{ct}$ etc. (where *at* is short for: accurate truth, and *ct* for comprehensive truth).

Although the selection of a particular epistemic goal responds to the practical features of one's plan, there is no compromise between epistemic and instrumental norms concerning the *content* of acceptances. Agents' epistemic confidence in accepting$_n$ *P* (accepting *P* under norm *n*) is not influenced by the cost or benefit associated with being wrong or right. Thus we don't need to endorse the view that an epistemic acceptance of *P* is yielding to utility considerations, as Bratman suggests.

This proposal offers a natural way out of a puzzle, called the preface paradox, which is raised by traditional views about acceptance: A writer may rationally accept that each statement in his book is true, while at the same time rationally accepting that his book contains at least one error (Makinson 1965). This puzzle is dissolved once it is realized that the author's epistemic goal is one of offering an ideally comprehensive presentation of his subject matter: it will thus not be contradictory for him to accept$_{ct}$ all the sentences in her book, while accepting$_{pl}$ (accepting as plausible or likely) that one of them is false. Hence, a mental act of acceptance$_{ct}$ does not allow aggregation of truth, because its aim is exhaustive (include all the relevant truths) rather than accurate truth (include only truths). Similarly, in the lottery puzzle, an agent accepts$_{at}$ that there is one winning ticket in the one thousand tickets actually sold. It is rational for her, however, not to accept$_{pl}$ that the single ticket she is disposed to buy is the winning one.

From Epistemic to Strategic Acceptance

The *output* of an epistemic acceptance so construed needs, however, to be adjusted to the final ends of the agent's plan. The decision to act on one's epistemic acceptance, i.e., strategic acceptance, constitutes a second, distinct step in accepting *P*. On our view, utility does not just influence the selection of certain epistemic norms of acceptance. It also influences decision in a way that may depart greatly from the cognitive output of epistemic acceptance.

The first argument in favor of this two-step view of acceptance is conceptual. The existence of an autonomous level of epistemic acceptance enables agents to have a stable epistemic map that is independent from local and unstable instrumental considerations. Thus, it is functionally adaptive to prevent the contents of epistemic evaluation from being affected by utility and risk. Second, empirical evidence shows that agents are indeed able to adjust their cognitive control both as a function of their confidence in accepting P, and of the strategic importance of the decision to be finally taken. In situations where agents are forced to conduct a cognitive task, strategic acceptance is ruled out: agents merely express their epistemic acceptance. In contrast, when agents can freely consider how to plan their action, given its stakes, they can refrain from acting on the unique basis of their epistemic acceptance. A decision mechanism is used to compare the probability for their acceptance being correct and a preset response criterion probability, based on the implicit or explicit payoffs for this particular decision. Here agents are allowed to strategically withhold or volunteer an answer according to their personal control policy (risk-aversive or risk-seeking), associated with the cost or benefit of being respectively wrong or right (Koriat and Goldsmith, 1996). A third reason in favor of our two-tiered view is that strategic acceptance can be impaired in patients with schizophrenia, while epistemic acceptance is not (Koren et al. 2006): this suggests, again, that epistemic and strategic acceptances are cognitively distinct steps.

Discussion

The two-step theory sketched above accounts nicely for the cases of acceptances discussed in the literature. Judging *P* true flat-out is an accepting under a stringent norm of accurate truth, while "judging *P* likely" is an accepting under a norm of plausibility, conducted on the background of previous probabilistic beliefs regarding *P*. Adopting *P* as a matter of policy divides into accepting a set of premises to be used in collective reasoning under a norm of consensus, and accepting it under a norm of coherence, (as in judgments by contradiction, legal reasoning, etc.). Assuming, imagining, supposing do not automatically qualify as acceptances. Only their controlled epistemic forms do, in which case they can be identified as forms of premising.

This theory predicts that errors in acceptances can be either strategic or epistemic. Strategic errors occur when selecting an epistemic norm inappropriate to a context, (for example, trying to reconstruct a shopping list accurately, when comprehensiveness is sufficient), or when incorrectly setting the decision criterion given the stakes (taking an epistemic decision to be non-important when it objectively is, and reciprocally). Epistemic errors, in contrast, can occur both in applying a given norm to its selected material, (for example, seeming to remember that *P* when one does not) or in forming an incorrect judgment of confidence about one's epistemic performance (for example, being confident in having correctly remembered that *P* when one actually failed to do so). Appropriate confidence judgments have an extremely important role as they help filter out a large proportion of first-order epistemic mistakes (illusory remembering, poor perceivings, incoherent or irrelevant reasonings etc.).

Is our two-tiered theory relevant to AI research? Although the author has no personal competence in this domain, it appears to be clearly the case. The two-tiered theory of acceptance is inspired by empirical research on epistemic self-evaluation, and by requirements on epistemic reliability. For these reasons, it should help epistemic agency to be modeled in a more realistic way, and conferred to artificial systems with the suitable cognitive complexity. Indeed an artificial agent, for example a social robot, should be able to monitor the epistemic responses of others as well as its own not only for their accuracy, but also for their comprehensiveness, their coherence, and the consensus in a group. Monitoring relevance in speech may, at present, be a trickier issue. Even artificial agents with no linguistic ability, as far as they need to evaluate whether they can solve a given problem, need to have various forms of acceptance available (e.g.: do they have the resources to create a path to a solution? Should they list, rather, all the solutions already used in similar contexts? How plausible is it that one of these solutions is applicable here and now? Epistemic planning depends on such acceptances being alternative options). Furthermore, it is crucial for artificial system designers to clearly distinguish an epistemic evaluation, which only depends on internal assessment of what is known or accessible, and a utility evaluation, which varies with the importance of the task.

Our two-tiered theory of acceptance raises potential objections that will be briefly examined.

Acceptance Does Not Form a Natural Kind?

It might be objected that, if acceptance can be governed by epistemic norms as disparate as intelligibility, coherence, consensus and accuracy, it should not be treated as a natural kind. In other words, there is no feature common to the various forms of acceptance, and for that very reason, the concept of acceptance should be relinquished. To address this objection, one needs to emphasize that normative diversity in acceptances is observed in metacognitive studies: agents, according to circumstances, opt for accuracy or comprehensiveness, or use fluency as a quick, although loose way, of assessing truthfulness (Reber and Schwarz 1999). What makes accepting a unitary mental action is its particular function: that of adjusting to various standards of utility the cognitive activity associated with planning and acting on the world. This adjustment requires both selecting the most promising epistemic goal, and suppressing those acceptances that do not meet the decision criterion relevant to the action considered.

Sophistication implausible?

A second objection might find it implausible that ordinary agents have the required sophistication to manage acceptances as described, by selecting the kind of epistemic acceptance that is most profitable given a context of planning, by keeping track of the implicit or explicit payoffs for a particular option, and by setting on their basis their response criterion.

It must be acknowledged that agents do not have in general the conceptual resources that would allow them to identify the epistemic norm relevant to a context. Acceptances, however, can be performed under a given norm without this norm being represented explicitly. Agents learn to associate implicitly a given norm with a given cognitive task and context: their know-how is revealed in their practical ability to monitor their acceptances along the chosen normative dimension (Perfect and Schwartz, 2002).

Concerning decision making, robust evidence indicates that the ability to re-experience an emotion from the recall of an appropriate emotional event is crucial in integrating the various values involved in an option (Gibbard 1990, Bechara, Damasio and Damasio 2000). Agents are guided in their strategic acceptance by dedicated emotions (with their associated somatic markers), just as they are guided in their epistemic acceptance by dedicated noetic feelings. (Koriat 2000, Hookway 2003, Proust 2007). The probabilist information about priors, on the other hand, seems to be automatically collected at a subpersonal level (Fahlman, Hinton and Sejnowski 1983).

Value Pluralism and Epistemological Relativism

Finally, epistemologists might observe that such a variety of epistemic standards pave the way for epistemic value pluralism, i.e., the denial that truth is the only valuable goal to pursue. Our variety of epistemic acceptings should indeed be welcome by epistemic value pluralists, who claim that coherence, or comprehensiveness, are epistemic goods for their own sake (Kvanvig 2005). It is open to epistemic value monists, however, to interpret these various acceptances as instrumental steps toward acceptance$_{at}$, i.e. as epistemic desiderata (Alston 2005). The present project, however, is not the epistemological study of what constitutes success in inquiry. It rather aims to explore the multiplicity of acceptances open to natural or artificial agents, given the informational needs that arise in connection with their final ends across multiple contexts.

The proposed two-tiered theory of acceptance does not invite a relativistic view of epistemic norms, but rather combats it: Even though agents can accept propositions under various norms, there are rational constraints on norm selection. For example, it may be rational to look for accurate retrieval when making a medical decision, while looking for comprehensive retrieval when shopping. Once a norm has been identified as instrumentally justified, acceptance can only be successful if it is conducted under the epistemic requirements prescribed by its norm. Thus agents, whether natural or artificial, can build, through acceptances, a stable epistemic representation of the facts relevant to their action that is not contaminated by the strategic importance of the decisions to be taken on its basis.

Conclusion

Given the limited cognitive resources available to agents at any given time, it is rational for them to focus on the epistemic goals that will maximize their epistemic potential both in terms of correctness and utility. Our two-tiered theory of acceptance accounts for this consequence of bounded rationality. Our future work aims to clarify the informational basis on which the various epistemic norms operate. We also aim to study norm sensitivity in agents from different cultures, measured by their confidence judgments about particular tasks and performances. Finally, we will study how potential conflicts for a given acceptance between epistemic norms can be generated, and how they are overcome.

Acknowledgments

Thanks to Richard Carter for his linguistic revision and critical discussion of a prior version, and Anne Coubray, Martin Fortier and Mouhamadou el Hady Ba for useful feedback. This research has ben made possible by an ERC advanced grant, "Dividnorm" project # 269616, conducted at the Fondation Pierre-Gilles de Gennes pour la Recherche, Ecole Normale Supérieure, Paris.

References

Alston, W. 2005. *Beyond "Justification": Dimensions of Epistemic Evaluation*, Ithaca, NJ: Cornell University Press.

Bechara, A. Damasio, H. and Damasio, A.R. 2000. Emotion, Decision Making and the orbitofrontal cortex. *Cerebral Cortex*, 10: 295-307.

Bratman, M.E. 1999. *Faces of intention*. Cambridge: Cambridge University Press.

Cohen, L. J. 1992. *An essay on belief and acceptance.* Oxford: Clarendon Press.

Couchman, J.J., Coutinho, M.V.C., Beran, M.J., Smith, J.D. 2010. Beyond stimulus cues and reinforcement signals: A new approach to animal metacognition. *Journal of Comparative psychology*, 124: 356-368.

Fahlman, S.E., Hinton, G.E. and Sejnowski, T.J. 1983. Massively parallel architectures for A.I.: Netl, Thistle, and Boltzmann machines. *Proceedings of the National Conference on Artificial Intelligence*, Washington DC: 109-113.

Frankish, K. 2004. *Mind and supermind*. Cambridge: Cambridge University Press.

Gibbard, A. 1990. *Wise Choices, Apt Feelings*. Oxford: Clarendon Press.

Hookway, C. 2003. Affective States and Epistemic Immediacy. *Metaphilosophy* 34: 78-96.

Jeffrey, R. C. 1956. Valuation and acceptance of scientific hypotheses. *Philosophy of Science,* 23, 3: 237-246.

Koren, D., Seidmann, L.J., Goldsmith, M. and Harvey P.D. 2006. Real-World Cognitive—and Metacognitive—Dysfunction in Schizophrenia: A New Approach for Measuring and Remediating More ''Right Stuff'', *Schizophrenia Bulletin,* 32, 2: 310-326.

Koriat, A. 2000. The Feeling of Knowing: some metatheoretical Implications for Consciousness and Control. *Consciousness and Cognition*, 9: 149-171.

Koriat, A. and Goldsmith, M. 1996. Monitoring and control processes in the strategic regulation of memory accuracy. *Psychological Review,* 103, 3: 490-517.

Goldsmith, M. and Koriat, A. 2008. The strategic regulation of memory accuracy and informativeness. *The Psychology of Learning and Motivation*, 48, 1-60.

Kvanvig, J. 2005. Truth is not the primary epistemic goal, in Steups, M. and Sosa, E. eds. *Contemporary Debates in Epistemology,* Oxford: Blackwell, 285-296.

Lehrer, K. 2000. Discursive Knowledge. *Philosophy and Phenomenological Research*, 60, 3: 637-653.

Makinson, D. C. 1965. *Paradox of the Preface, Analysis,* 25: 205-207.

Perfect, T.J. and Schwartz, B.L. 2002. *Applied Metacognition*. Cambridge: Cambridge University Press.

Proust, J. 2007. Metacognition and metarepresentation : is a self-directed theory of mind a precondition for metacognition ? *Synthese,* 2: 271-295.

Proust, J. forthcoming. Mental acts as natural kinds, in: T. Vierkant, A. Clark, J. Kieverstein Eds. *Decomposing the Will*. Oxford: Oxford University Press.

Reber, R. and Schwarz, N. 1999. Effects of perceptual fluency on judgments of truth. *Consciousness and Cognition,* 8: 338-342.

Stalnaker, R. 1987. *Inquiry.* Cambridge: MIT Press.

Velleman, J.D. 2000. *The possibility of practical reason.* Ann Harbor: The University of Michigan Library.

Recognition and Feature Extraction

Chair: Michael Glass

Baseline Avatar Face Detection using an Extended Set of Haar-like Features

Darryl D'Souza, Roman V. Yampolskiy

Computer Engineering and Computer Science
University of Louisville
Louisville, Kentucky, USA 40292
darryl.dsouza@louisville.edu, roman.yampolskiy@louisville.edu

Abstract

It is desirable to address accessibility issues within virtual worlds. Moreover, curbing criminal activities within virtual worlds is a major concern to the law enforcement agencies. Forensic investigators and accessibility researchers are gaining considerable interests in detecting and tracking avatars as well as describing their appearance within virtual worlds. Leinhart and Maydt have introduced a novel-set of Haar like features by extending the Viola Jones approach towards rapid object detection. We test this Haar cascade on human and avatar faces. Accuracy rates of 79% on human and 74% on avatar faces are obtained. The goal is to detect avatar faces in upright frontal face datasets and establish a baseline for future work in computer generated face recognition.

Introduction

Virtual worlds are gaining widespread momentum. They have a potential to transform the way the society operates. They bring in a sense of a "personal" digital space (Trewin et al. 2008) for the user by providing real time interaction with other fellow users and mirroring the real world activities. Communities, social groups, enterprises and institutions are all present in these virtual worlds. These virtual environments are gaining popularity across the globe and moving towards being an integral part of not only the Internet but also the society. An "avatar" is the user's virtual identity in these virtual worlds. The user can model its appearance to reflect either one's own personality or pose as someone else. Avatars can navigate within this virtual space by moving around buildings, flying in the air, swimming in the water or teleporting to different locations.

Destructive behaviors like terrorist activities and cybercrimes are reportedly on the rise within virtual worlds. There are reports of Al-Qaeda terrorists communicating and recruiting within the popular virtual world Second Life (SL) (Cole 2008), Second Life Liberation Army (SLLA) setting off virtual bombs to destroy virtual stores (Mandal & Ee-Peng 2008), American Apparel and Reebok's virtual store in SL being hit by virtual atomic bombs (Mandal & Ee-Peng 2008), etc. Identity thefts (Weinstein & Myers 2009) are a concern too. The lack of surveillance, presence of ungoverned spaces and the absence of strict rules monitoring the virtual society and its institutions have led to the growth of extremism and cyber-terrorism. Terrorist attacks, believed to be rehearsed in virtual worlds, are lethal as they can train using weapons identical to the real ones as well as build real life replicas of buildings and infrastructure.

Making these worlds accessible to a broad set of users is being gradually addressed. They are not easy for anyone to use the first time. Controlling an avatar requires simultaneous visual, audio, cognitive and motor abilities (Trewin et al. 2008). This results in a number of challenges to users with disabilities. Disabled gamers are showing a strong desire to be a part of these virtual worlds and envision themselves in their own avatars. Visually impaired users wish to detect and identify fellow avatars as well as other objects in their surroundings.

Criminal activities in these worlds are becoming a major problem for law enforcement agencies. Forensic experts are expressing interest in accurately and automatically tracking users and their avatars in these virtual communities. Detecting and recognizing avatar faces (Boukhris et al. 2011) will serve as one of the major component in building an artificial face authentication system (Gavrilova & Yampolskiy 2010) to help law officers track avatars within virtual worlds. The detected faces, saved to a database, will help to profile the avatars.

Profiling avatars based on their faces is a challenging and novel problem, contributing towards a new research direction in face recognition. Detecting avatar faces will address accessibility issues within virtual worlds, especially for visually impaired users, by describing the facial appearances of avatars in the vicinity by face detection.

Authenticating biological entities (human beings) is an essential and well-developed science, utilized to determine one's identity in today's modern society. However, avatar authentication (non-biological entities) is an issue that needs to be highlighted and addressed (Ajina, Yampolskiy & Amara 2010). A high degree of convergence between the real and virtual worlds has led to narrowing the distinction between the users and their avatars and applying security systems in these virtual spaces. To address the need for an affordable, automatic, fast, secure, reliable and accurate means of identity authentication Yampolskiy & Gavrilova define the concept of *Artimetrics* – a field of study that will allow identifying, classifying and authenticating robots, software and virtual reality agents (Yampolskiy & Gavrilova 2010).

Background

Avatar and human faces are similar as well as different. Both have consistent structure and facial component (eyes, ears, nose, etc.) locations. These similarities motivate the design of an avatar face detection framework based on principles similar to the human face detection system. Avatars have a wider range of colors than humans do that helps distinguish the two entities (Yampolskiy, Klare & Jain 2012). The goal is to detect avatar faces in the field of view of the user's avatar within virtual worlds that, along with face recognition, will help build a complete biometric authentication system for avatars. Currently available biometric systems are not equipped to deal with the visual and behavioral nature of artificial entities like avatars and perform poorly under such circumstances. Concerns over security and avatar identification are constantly voiced in virtual worlds (Yampolskiy & Gavrilova 2010).

Several challenges are involved in detecting avatar faces. They involve illumination, camera location, different skin color tones, pose, head rotation, etc. Certain preprocessing techniques such as geometric and color normalization may have to be applied (Yampolskiy, Klare & Jain 2012). In the context of investigating criminal activities in virtual worlds, we aim to examine some possibilities to authenticate avatars. These involve matching a human face to an avatar face when users upload their picture to model their avatars, matching the face of one avatar to another in a single as well as across multiple virtual worlds and matching a forensic sketch of an avatar to the avatar face (Yampolskiy, Klare & Jain 2012).

To the best of our knowledge, no prior research has been reported in the area of avatar face detection. However, there has been significant research in the domain of avatar recognition. The current state of art in virtual reality security, focusing specifically on emerging techniques for avatar authentication has been examined (Yampolskiy & Gavrilova 2010). Significant work has been conducted in recognizing avatar faces (Yampolskiy, Klare & Jain 2012). Research work has been carried out in the area of avatar facial biometric authentication (Ajina, Yampolskiy & Amara 2010). Daubechies wavelet transform and Support Vector Machines (SVM) are used to achieve artificial face recognition (Boukhris et al. 2011). In addition to these, there has been relevant research on robot emotion recognition (Yampolskiy & Gavrilova 2010). Intelligence Advanced Research Projects Activity (IARPA) is aiming to develop systems to observe avatar behavior and communication within virtual worlds to obtain insights into how real-life users in hostile cultures act and think (Yampolskiy & Gavrilova 2010). Another novel research direction is Avatar DNA, a patent pending technology by Raytheon. It focuses on providing authentication and confidentiality within virtual worlds by mixing real world biometrics of users with their avatar profiles (Yampolskiy & Gavrilova 2010).

In this paper, we focus on applying the face detecting OpenCV Haar cascade as an appearance based face detection technique. The method is based on an extended novel set of rotated Haar-like features, efficiently calculated by enriching the basic set of simple Haar-like features. Our test set comprises of human and avatar face datasets with varying backgrounds, illumination, rotations and face occlusions. The goal here is to obtain accuracy estimations by simply applying the cascade on each of these varying datasets.

The paper is organized as follows. We begin with an introduction to AdaBoost (Adaptive Boosting) learning algorithm, extended Haar-like features (Lienhart & Maydt 2002) and the OpenCV Haar cascade generation (Rhondasw 2009; Seo 2008). Next, we present the approach towards estimating the accuracies of applying the cascade on human faces and virtual world avatar datasets. The experimental results are described later. Finally, conclusions and directions for further enhancing the system performance are highlighted.

Algorithms

AdaBoost (Adaptive Boosting) Algorithm

AdaBoost algorithm, invented by Schapire and Freund (Schapire 1999), helped in solving many practical difficulties faced by the earlier boosting algorithms. It initially distributes a set of equal weights over a training set. After each round, the weak learning algorithm increases the weights for the incorrectly classified examples. This helps in focusing on the hard examples in the training dataset. "Ada" is a short form for adaptive as it adapts to the error rates of the individual weak hypothesis of each stage of the boosting process. Here the basic classifier is used extensively with the concept of stacking or boosting to constitute a very strong classifier. Several of these strong classifiers are subsequently connected into a cascade classifier to achieve the detection. The cascading levels determine the response of the system and the error rate. AdaBoost face detection techniques are based on the expansion of the Haar-like features, image integration and the cascaded classifier. For each image feature, a corresponding simple classifier is generated and the error relative to the current initialization error weight is evaluated. The classifier with the smallest error is chosen and added to the stage classifier. The weights of the samples are appropriately updated. If the sample is correctly classified, then the error is 0 or else it is 1. Finally, a stage classifier is obtained by combining the individual simple classifiers into a cascade. The algorithm is fast, easy and simple to implement, has no tuning parameters and requires no prior knowledge about the weak learner. Thus, it can be combined flexibly with any method to evaluate the weak hypothesis.

Extended Haar-like Features

These are a novel set of rotated Haar-like features, which yields a 10% lower false alarm rate as a face detector (Lienhart & Maydt 2002). It is an extension of the Viola Jones (Viola & Jones 2001) rapid object detection framework. It includes an efficient set of 45 degree rotated features that contribute additional domain knowledge to the learning process (Viola & Jones 2001). They can be computed rapidly at different scales.

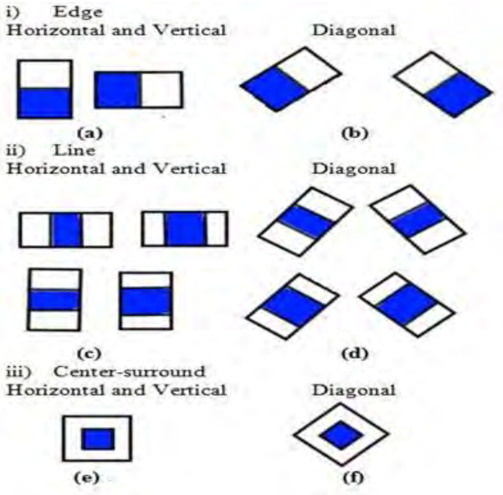


Figure 1: Simple Haar-like features. (a) and (b) are used to detect horizontal, vertical and diagonal edges respectively. Similarly (c) and (d) are used for lines and (e) and (f) for center-surround features. Shaded: Positive weights and Unshaded: Negative weights (Lienhart & Maydt 2002).

From Figure 1 we observe the 14 prototypes, which include four edge features, eight line features and two center-surround features. They are scaled independently in vertical and horizontal direction to generate a rich, complete set of features. The number of features obtained from each prototype is large and differs for each prototype (Lienhart & Maydt 2002).

Generating the OpenCV Haar Cascade

One of the available features of Intel's OpenCV (Intel) is face detection from images. Furthermore, it provides programs that are used to train classifiers for face detection systems, called Haar Training, to create custom object classifiers (Rhondasw 2009; Seo 2008).

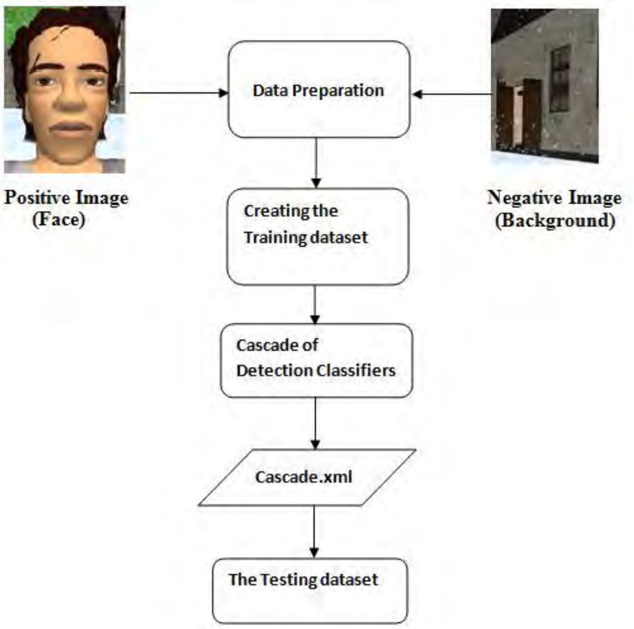

Positive Image (Face)
Data Preparation
Negative Image (Background)
Creating the Training dataset
Cascade of Detection Classifiers
Cascade.xml
The Testing dataset

Figure 2: Flowchart for Haar training.

From Figure 2 we observe that the process of Haar training consists of the following steps:

Data preparation:

The process begins by gathering positive and negative datasets. The positive dataset contains images with the object of interest, i.e. the faces to be detected. The negative images are the ones that do not contain the object of interest, e.g. background images, non-face images etc. For real cascades there should be about 1000 positive and 2000 negative images. A general and acceptable positive-negative proportion is 1:2, but it is not a hard rule (Rhondasw 2009).

Creating the training dataset:

The prepared data now needs to be segregated into training and testing datasets. Here the training dataset is fed to a cascade of detection-boosted classifiers that yields the cascade xml file.

Cascade of Detection Classifiers:

Basic classifiers are put together to form stage classifiers which in turn are grouped together to form a cascade of stage classifiers. The series of such classifiers are applied to every sub-window of an image. A positive result from the first classifier stimulates the evaluation of the second classifier, which also has been adjusted to achieve high detection rates. A positive outcome from the second triggers the third and so on. Negative outcomes at any stage lead to rejection. Stages in the cascade are trained using AdaBoost and their thresholds are appropriately varied to minimize the false negatives. Thus, higher number of stages yields higher accuracies but leads to a decrease in performance time (Viola & Jones 2001).

Cascade.xml:

It is comprised of the various stages built due to the Haar training with the appropriate thresholds for each stage.

The Testing dataset:

The generated cascade xml file is evaluated on the testing dataset to get accuracy estimations of the results. The Haar cascade performs poorly on rotationally varying faces as well as occluded faces as it is based on object detection and it needs all the facial features (a pair of eyes, nose and mouth) to perform an accurate face detection.

Experiment

For the purpose of analysis of face detection techniques, we needed datasets with

- Faces with different head rotations
- Complex backgrounds
- Varying illumination
- Partially occluded faces

In our experiments, we used the following datasets:

Set 1 - Caltech

450 samples of human frontal face images from the California Institute of Technology were used(Weber). All images have the dimensions of 896 x 592 pixels. The dataset contains images from 28 subjects and 3 sketches with complex backgrounds and varying illuminations.

Set 2 – FERET

400 samples of human face images from the FERET ("The Color FERET Database") dataset were used. All images have the dimensions of 256 x 384 pixels. This dataset contains images for 52 subjects. Each subject is represented by 7-8 images with random head rotations varying from 67.5 degree to 15 degree rotations in both left and right directions, plain background and slightly varying illuminations.

Set 3 – Avatar

Avatar faces from the popular online virtual worlds, Entropia Universe and Second Life, were used in the avatar dataset. A scripting technique was designed and implemented to automatically collect the avatar faces using AutoIT as well as Second Life's Linden Scripting Language (LSL) (Oursler, Price & Yampolskiy 2009; R.V. Yampolskiy & Gavrilova 2010).

The dataset is subdivided into 3 parts with the number of samples indicated in the parenthesis next to it: Female avatars from Entropia Universe with complex backgrounds (150), Male avatars from Second Life with a regular or plain background (150) and Male avatars from Second Life with complex backgrounds (150). The dataset contains the avatar faces with a set of five images per avatar from different angles with complex backgrounds and varying illuminations. Figure 3 shows a sample of this dataset with different random head rotations and a complex background.

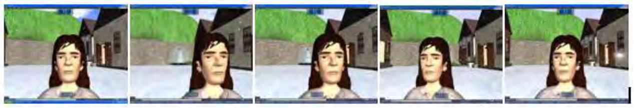


Figure 3: Examples of different face images of the same subject from the Second Life avatar dataset with a complex background. Each image corresponds to the different head rotations while facing the camera. The frontal image is (a). (b) to (e) represent the same avatar with varying head rotations.

The experiment involved using a single OpenCV Haar cascade, *haarcascade_frontalface_alt.xml,* on these datasets and evaluating the algorithm's performance on each of them. There is no training involved here. The experiment relies solely on the accuracy of this cascade on the human and avatar datasets. Basically, a classifier trained on human faces is used to detect avatar faces without any special preprocessing.

Initially, a code similar to the one found in (Hewitt 2007) was written in Microsoft Visual Studio 2008 using the OpenCV libraries (Intel). This code helps in detecting faces within images using the OpenCV Haar cascade. The cascade was applied on the dataset images. Promising results were obtained and a comparitive analysis was drawn.

Results

A summary of the results, obtained by executing the above OpenCV Haar cascade on all three datasets, are shown in Table 1. These results include images that belong to multiple categories. A brief description of each category is given below.

Background: The nature of the background, which can be either plain or complex.

Positive: Signifies a positive detection with a face accurately detected.

Background Faces (BF): The background faces (if any) other than the main subject that are detected.

False Positives (FP): Non–face objects that are incorrectly detected as faces.

False Negatives (FN): Background faces that are not found by the detector.

Zero Detections (ZD): The face of the main subject as well as those in the background (if any) that are completely undetected by the detector.

Accuracy: The positive detections.

Error Rates: Comprises of false positives, false negatives and zero detections.

Average Accuracy and Error Rates: The average of Accuracy and Error Rates obtained.

A: Images with BF’s detected without FN
B: Images with BF’s detected with FN
C: Images with both FP’s and FN’s
D: Images with only FP’s
E: Images with both ZD’s and FP’s
F: Images with only FN’s
G: Images with only ZD’s

Table 1: Face detection results for all three datasets.

<table>
<tr><th>Dataset</th><th colspan="7">Human</th><th colspan="3">Avatars</th></tr>
<tr><th>Type</th><th colspan="6"><i>Caltech</i></th><th><i>FERET</i></th><th><i>Entropia</i></th><th><i>SL (Male)</i></th><th><i>SL (Male)</i></th></tr>
<tr><th>Background</th><td colspan="6">Complex</td><td>Plain</td><td>Complex</td><td>Plain</td><td>Complex</td></tr>
<tr><th>Positive</th><td colspan="6">444/450</td><td>235/400</td><td>114/150</td><td>79/150</td><td>140/150</td></tr>
<tr><th>Background Faces (BF)</th><td colspan="6">11/25</td><td>0/0</td><td>0/0</td><td>0/0</td><td>0/0</td></tr>
<tr><td rowspan="2"></td><th colspan="3">A</th><th colspan="3">B</th><td rowspan="2"></td><td rowspan="2"></td><td rowspan="2"></td><td rowspan="2"></td></tr>
<tr><td colspan="3">7/11</td><td colspan="3">4/11</td></tr>
<tr><th>False Positives (FP)</th><td colspan="6">11/450</td><td>0/400</td><td>0/150</td><td>0/150</td><td>0/150</td></tr>
<tr><td rowspan="2"></td><th colspan="2">C</th><th colspan="2">D</th><th colspan="2">E</th><td rowspan="2"></td><td rowspan="2"></td><td rowspan="2"></td><td rowspan="2"></td></tr>
<tr><td colspan="2">2/11</td><td colspan="2">8/11</td><td colspan="2">1/11</td></tr>
<tr><th>False Negatives (FN)</th><td colspan="6">12/450</td><td>0/400</td><td>0/150</td><td>0/150</td><td>0/150</td></tr>
<tr><td rowspan="2"></td><th colspan="2">B</th><th colspan="2">C</th><th colspan="2">F</th><td rowspan="2"></td><td rowspan="2"></td><td rowspan="2"></td><td rowspan="2"></td></tr>
<tr><td colspan="2">4/12</td><td colspan="2">2/12</td><td colspan="2">6/12</td></tr>
<tr><th>Zero Detections (ZD)</th><td colspan="6">6/450</td><td>165/400</td><td>36/150</td><td>71/150</td><td>10/150</td></tr>
<tr><td rowspan="2"></td><th colspan="3">E</th><th colspan="3">G</th><td rowspan="2"></td><td rowspan="2"></td><td rowspan="2"></td><td rowspan="2"></td></tr>
<tr><td colspan="3">1/6</td><td colspan="3">5/6</td></tr>
<tr><th>Accuracy (%)</th><td colspan="6">444/450
= 98.6</td><td>235/400
= 59</td><td>114/150
= 76</td><td>79/150
= 52.66</td><td>140/150
= 93.33</td></tr>
<tr><th>Accuracy (Average %)</th><td colspan="7">(98.6 % + 59 %) / 2
= 78.8</td><td colspan="3">(76 % + 52.66 % + 93.33 %) / 3
= 74</td></tr>
<tr><th>Error Rates (%)</th><td colspan="6">29/450
= 6.4</td><td>165/400
= 41</td><td>36/150
= 24</td><td>71/150
= 47.33</td><td>10/150
= 6.66</td></tr>
<tr><th>Error Rates (Average %)</th><td colspan="7">(6.4 % + 41 %) / 2
= 23.7</td><td colspan="3">(24 % + 47.33 % + 6.66 %) /3
= 26</td></tr>
</table>

Set 1 - Caltech

On the 450 human face images, the OpenCV Haar cascade yielded an accuracy rate of 98.6% with an error rate of 6.4%.

Set 2 - FERET

On the 400 human face images, the OpenCV Haar cascade yielded an accuracy rate of 59% with an error rate of 41%. The error rate being high is mainly due to the poor performance of the cascade on rotationally varying faces as well as face occlusions.

Set 3 - Avatar

Facial images from the three separate avatar datasets are fed to the OpenCV Haar cascade individually. Accuracies obtained are 76% (Female avatar-Entropia, complex background), 52% (Male avatar-Second Life, regular background), 93% (Male avatar-Second Life, complex background) and error rates of 24% (Female avatar-Entropia, complex background), 48% (Male avatar-Second Life, regular background) as high as 7% (Male avatar-Second Life, complex background). From the results we observe that the cascade performs very well on the Male avatar-Second Life (complex background) but not so good on the Male avatar- Second Life (regular background).

Figure 4 shows a set of bar graphs for the results obtained on the FERET and the avatar datasets.

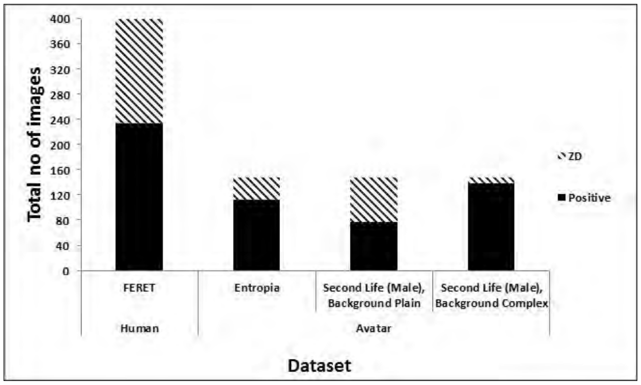


Figure 4: Results from the FERET and the avatar datasets.

A set of sample detections are shown in Figure 5.

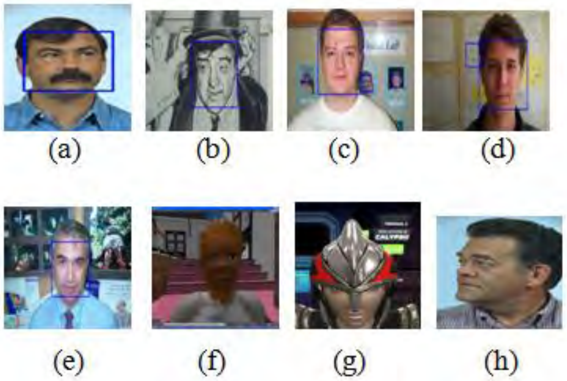


Figure 5: (a) FERET dataset – Positive detection (b) Caltech dataset – Positive sketch detection (c) Caltech dataset – Three background faces detected (d) Caltech dataset – False positive (e) Caltech dataset – False negative, background face at lower right corner (f) Male avatar: Second Life, complex background dataset – Zero detection due to poor illumination (g) Female avatar: Entropia, complex background dataset – Face occlusion (h) FERET dataset: Zero face detection due to 90 degree face rotation to the left.

Conclusion

This paper evaluates a standard OpenCV face detection algorithm, which utilizes a Haar cascade on a set of human facial images as well as virtual world entities: avatars, which are rapidly becoming part of the virtual society. Accuracy estimations reported from each dataset are compared. Good results are obtained, but poor performances are recorded for facial images involving head rotations, poor illumination and face occlusions. The average accuracy detection rates on human faces is 79% whereas for avatar faces it is 74%, which is quite good given the quality of the picture, appearance of the avatar as well as background and surrounding regions. The varying accuracy rates are due to the difference in the count of head rotations, face occlusions, illumination conditions and plain/complex backgrounds between the datasets.

Potential directions for future research involve improving the existing algorithm to achieve better accuracies by fusing the current algorithm with Local Binary Patterns (LBP). We will extend this approach over a larger dataset to yield more accurate evaluations and better results. Further, unifying this face detection technique with face recognition to build a complete face authentication system capable of authenticating biological (human beings) and non-biological (avatars) entities is the direction in which we have set our sights on.

References

Ajina, S., Yampolskiy, R. V. & Amara, N. B. (2010, Jul 1-2). *Authetification de Visages D'avatar [Avatar Facial Biometric Authentication]*. Paper presented at the Confere 2010 Symposium. Sousse, Tunisia.

Boukhris, M., Mohamed, A. A., D'Souza, D., Beck, M., Ben Amara, N. E. & Yampolskiy, R. V. (2011, July 27-30). *Artificial human face recognition via Daubechies wavelet transform and SVM.* Paper presented at the 16th International Conference on Computer Games (CGAMES). Louisville, KY.

Cole, J. (2008). Osama Bin Laden's "Second Life" *Salon*, from http://www.salon.com/news/opinion/feature/2008/02/25/avatars

The Color FERET Database. Available from NIST The Color FERET Database http://face.nist.gov/colorferet/

Gavrilova, M. L. & Yampolskiy, R. V. (2010, Dec). State-of-the-Art in Robot Authentication [From the Guest Editors]. *Robotics & Automation Magazine, IEEE, 17*(4), 23-24.

Hewitt, R. (2007). Seeing With OpenCV, Part 2: Finding Faces in Images. *Cognotics: Resources for Cognitive Robotics*, from http://www.cognotics.com/opencv/servo_2007_series/part_2/index.html

Intel. Open Computer Vison Library, from http://opencv.willowgarage.com/wiki/ http://sourceforge.net/projects/opencvlibrary/files/opencv-win/2.1/

Lienhart, R. & Maydt, J. (2002, Sept 22-25). *An extended set of Haar-like features for rapid object detection.* Paper presented at the International Conference on Image Processing. Rochester, NY.

Mandal, S. & Ee-Peng, L. (2008, May 12-13). *Second Life: Limits Of Creativity Or Cyber Threat?* Paper presented at the 2008 IEEE Conference on Technologies for Homeland Security. Waltham, MA.

Oursler, J., Price, M. & Yampolskiy, R. V. (2009, Jul 29-31). *Parameterized Generation of Avatar Face Dataset.* Paper presented at the CGAMES '2009: 14th International Conference on Computer Games: AI, Animation, Mobile, Interactive, Multimedia, Educational and Serious Games. Louisville, KY.

Rhondasw. (2009). FAQ: OpenCV Haar Training, from http://www.computer-vision-software.com/blog/2009/06/opencv-haartraining-detect-objects-using-haar-like-features/ http://www.computer-vision-software.com/blog/2009/11/faq-opencv-haartraining/

Schapire, R. E. & Shannon Laboratory ATT. (1999, Jul 31- Aug 6). *A brief introduction to boosting.* Paper presented at the Sixteenth International Joint Conference on Artificial Intelligence. Stockholm, Sweden.

Seo, N. (2008). Tutorial: OpenCV Haartraining (Rapid Object Detection with a Cascade of Boosted Classifiers Based on Haar-like features) from http://note.sonots.com/SciSoftware/haartraining.html

Trewin, S., Laff, M., Cavender, A. & Hanson, V. (2008, Apr 5-10). *Accessibility in virtual worlds*. Paper presented at the CHI '08 Human factors in computing systems. Florence, Italy.

Viola, P. & Jones, M. (2001, Dec 8-14). *Rapid Object Detection using a Boosted Cascade of Simple Features*. Paper presented at the Computer Vison and Pattern Recognition. Kauai, HI.

Weber, M. A frontal face dataset collected by Markus Weber at the California Institute of Technology. Available from California Institute of Technology Bg-Caltech http://www.vision.caltech.edu/Image_Datasets/faces/README

Weinstein, J. & Myers, J. (2009, Dec 2). Avatar Identity Theft Prompts Review of Gaia and Virtual World Rules, from http://www.jackmyers.com/commentary/guest-mediabizbloggers/11193066.html

Yampolskiy, R. V. & Gavrilova, M. (2010, Oct 20-22). *Applying Biometric Principles to Avatar Recognition.* Paper presented at the International Conference on Cyberworlds (CW2010). Singapore.

Yampolskiy, R. V., Klare, B. & Jain, A. K. (2012, Apr 23-27). *Face Recognition in the Virtual World: Recognizing Avatar Faces*. Biometric Technology for Human Identification IX. SPIE Defense and Security Symposium. Baltimore, MD. (In Press).

D'Souza, D., & Yampolskiy, R. V. (2010). Avatar Face Detection using an Extended Set of Haar-like Features. [Journal Abstract]. *Kentucky Academy of Science, 71*(1-2), 108.

Automated Collection of High Quality 3D Avatar Images

James Kim, Darryl D'Souza, Roman V. Yampolskiy

Computer Engineering and Computer Science
University of Louisville, Louisville, KY
jhkim012@louisville.edu, darryl.dsouza@louisville.edu, roman.yampolskiy@louisville.edu

Abstract CAPTCHAs are security tests designed to allow users to easily identify themselves as humans; however, as research shows (Bursztein et al. 2010) these test aren't necessarily easy for humans to pass. As a result a new test, which requests users to identify images of real humans among those of 3D virtual avatars, is proposed that would create a potentially unsolvable obstacle for computers and a quick, easy verification for humans. In order to provide test cases for this new test, an automated bot is used to collect images of 3D avatars from Evolver.com.

INTRODUCTION

Today's commonly used security test distinguishing real humans from bots on the Internet is CAPTCHA. Two noticeable issues result from using such tests. One is that these tests are gradually becoming less reliable as exposed design flaws and improved algorithms successfully crack CAPTCHAs. For example, a CAPTCHA cracking program called deCAPTCHA, which was developed by a group of Stanford researchers, successfully decrypted image and audio-based text CAPTCHAs from a number of famous sites. The success rate of passing CAPTCHAs varies from site to site, but some examples include a 66% success rate on Visa's Authorize.net, 43% on Ebay, and 73% on captcha.com (Bursztein, Martin, and Mitchell 2011). Each website uses a combination of different text distortion factors in their CAPTCHAs; however, it was evident that design flaws were present in most tests. For example, the deCAPTCHA program was able to achieve a 43% success rate because the researchers were able to exploit the fact that eBay's CAPTCHAs were using a fixed number of digits and regular font sizes (Bursztein, Martin, and Mitchell 2011).

Second, according to research, humans encounter difficulties in solving CAPTCHAs, especially audio-CAPTCHAs. Perhaps the tests are becoming more challenging for the human users than they are for computers, considering deCAPTCHA's performance (Bursztein et al. 2010). This test is designed to be an unsolvable obstacle for computers while remaining easy for humans. One may even hypothesize that there may be a correlation between the combination of factors that determine a CAPTCHA's insolvability to computers and the level of ease for a human to solve the test.

Other simpler more novel and user friendly types of CAPTCHAs do exist, such as the Animal Species Image Recognition for Restricting Access (ASIRR) or the 'drag and drop' CAPTCHA (Geetha and Ragavi 2011). However text-based CAPTCHAs are more widely used than sound or image based tests because of the numerous variations of distortion methods to increase the level of insolvability to computers, despite the added difficulty for humans to pass them (Geetha and Ragavi 2011). Each test has its own design flaws, thus security tests that present the highest level of insolvability to computers and are easiest to solve for humans are sought.

Computers have the potential to detect text from a variety of distorted images, but their ability to differentiate between two objects in different colors or textures is a different matter entirely. Considering this as a foundation, a test was proposed that would create a small table using images of virtual avatars and real humans and ask the user to select those that are of real humans. Generally, the test would include two or more images of real humans in order to increase the difficulty for computers.

In addition, the colors, textures, and other factors of the images would be variable in order to create as few viable variables as possible. For example, the images of virtual avatars or humans may be black and white, pastel, or even sketched. Furthermore, various clothing and accessories would potentially add more to the challenge. Inevitably, the test would require an algorithm that demands critical thinking, one that answers the question "How does a real human appear?"

In order to begin testing, a dataset of high quality 3D virtual avatars was required. Virtual communities such as Second Life or Active Worlds provided an extensive selection of body characteristics and articles of clothing for avatar creation. The tradeoff for extensive customization was the detail or quality of the avatars. Considering that both these virtual communities in particular have massive online communities of users, it's understandable that the system requirements to run the client application are very low since hardware costs are often a barrier to accessing online content.

Evolver, which is a 3D avatar generator that allows users to create and customize their own avatar, was found to be the best choice. The avatars were highly detailed and elaborate with options to orient them into certain body positions, facial expressions, or animations. Although Evolver did not possess the extensive selection of avatar characteristics as the other virtual words, its selections were large enough for our purposes.

In addition, one noticeable difference between Evolver and the other virtual worlds was the option to morph different physical attributes together. Evolver uses a large database of pre-generated body-parts called a virtual gene pool to morph together two physical attributes in morph bins. Users choose the degree of which they are morphed together by using a slider, which measures the domination of one attribute over the other (Boulay 2006).

The collection of avatar images was collected using Sikuli, which is an image recognition based scripting environment that automates graphic user interfaces (Lawton 2010). Sikuli's integrated development environment features an image capturing tool that allows the users to collect the images required to fulfill the parameters of unique functions.

A Sikuli script could be run continuously, but there is always some factor, such as server latency, that could cause an error, which would cause a major setback to data collection as the script would terminate. As a result, Evolver's simple and easily navigable user interface was another reason why Evolver was selected over *Second Life*, *ActiveWorlds*, and other virtual communities. Evolver's interface divides physical attributes and articles of clothing into organized tabs, which would decrease the possibility of failure when moving from one section to another.

ActiveWorlds in particular relied on scrolling, which at times significantly impeded data collection because the possibility of failure would increase when the script failed to scroll down to continue with the script. Generally, few images were able to be generated without the Sikuli script misrecognizing certain images. However, in any case of data collection of avatar images from the web, server latency is a huge factor in successful collection as it could delay the timing for actions dictated in the script.

METHODOLOGY

The Evolver avatar customization interface is slightly more unique in that it prioritizes displaying explicit details of a particular attribute than emphasizing the overall variety in a section. This visual design explains why when a mouse hovers over an attribute; the image automatically enlarges so that the user would have a better visual. Some physical attributes with variations, such as varying colors or patterns, show gray arrows on each side of the image when enlarged, which allows the user to rotate through the variations instead of viewing all of them in a grid on a reloaded webpage.

At its execution, the script would randomly click either a male or female gender option to begin the avatar creation process. Once the webpage loaded the Face tab, the script would make sure that both checkboxes for male and female physical attributes were selected so as to open more choices for a truly random avatar. Seven icons were present on the left side, each of which focused on a specific area of the face (Figure 1). In each area, a slider exists to enable the user to control the degree of attribute morphing. Initially, the random die button is clicked first to obtain two random face presets, and the slider would be dragged to a random location on the line to correlate a random degree of morphing. The script would click 65 pixels down from the center of the first selected icon in order to randomize the degree of morphing for each area.

The script would proceed to the Skin tab on which there is a grid that showed all the physical attributes for the skin. Under each tab there is a specific number of pages of attributes. As a result, the script would randomly click through the pages given the range of the pages, and would randomly select an image "cell" within the grid. Under most of the tabs that share a similar grid design, there would be a region that the script would act within to select an image. The region is the area defined by the x-y coordinate of the top left corner, and the dimensions

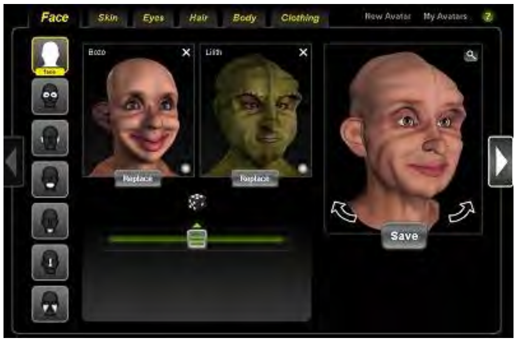


Figure 1. Face Tab(Evolver.com)

of the rectangle. Each tab had a differently sized region because of the varying dimensions of the physical attribute images.

Essentially, one attribute image within the grid was considered as a cell, and after randomly selecting a grid page, a random cell was chosen by moving the mouse cursor to the center of the cell. In order to confirm that the image has enlarged and that there was an image located at the cursor coordinate, a yellow box with the text “Selected,” which appears in the bottom right corner of the image, was scanned on the screen. Other cases were also resolved, such as the issue when the script failed to recognize that the mouse was hovering over the image given the default waiting period. Once the selected box is confirmed, the script checks to see if there are variations for the attribute by scanning for an unselected gray arrow on the right side of the image. If there were variations for that attribute, the script would quickly rotate through the variations given a random number of clicks. It is worth noting that viewing images on each page or the variations through rotation respond quickly, which suggests that these operations are run client-side. Once the random image is selected, the script would proceed to the next tab. Because of the similarity of the Eyes, Hairs, and Clothing tab with the Skin tab, the same method of collection was used for each.

After the random hair selection, an image of the avatar’s face would be captured in PNG format on a preview window, opened by the zoom button in the preview. The resolution used to capture the image depended on the region dimensions. The avatar was rotated once toward the left side in order to capture a direct face-to-face image of the avatar. For the Body tab, the same method of image selection used in the Face tab was also applied in this tab because of the similarity between the interface of the Body tab and the Face tab.

After the images for the clothing tab, which contains sub-tabs of the top, bottom, and shoe tabs, the image of the avatar’s body was captured. However, instead of the default position, which is a slightly angled arms out position, a face-to-face stand position was chosen in the dropdown box in the zoomed preview window. Finally, the script would select “New Avatar” located near the top right hand corner of the web page in order to repeat the process.

RESULTS

The program was run as continuously as possible over the course of seven days, and approximately 1030 successful images were captured, more of which were of the avatar body than the avatar face. The initial days were primarily test runs that were used to find and resolve unforeseen bugs. There were more than 100 face images that were unsuccessful, yet the script managed to continue with the process and collect other potentially successful samples. One interesting point to note is that there were a significantly larger number of unsuccessful images of faces than there were of bodies, which totaled about 10-20 errors at most. This is primarily because the method for adjusting the avatar’s face was more problematic than that of the avatar’s body. Adjusting the avatar’s head relied on clicking ‘rotate left arrow’ icons, which were often ignored by the server. Some of these major problems are longer load times or Sikuli’s failure to recognize loading signals.

The source of most of these errors was primarily due to server latency. Between every selection in the avatar customization process, there is a variable loading time and a loading box, which appears in the center of the screen. Surprisingly, the loading time differs for each attribute image applied to the avatar, not necessarily differing for one type of physical attribute. For example, the loading time for two different but similarly categorized physical attributes would vary. One could conjecture that the amount of polygons, depending on the attribute, added upon the avatar may cause the variable loading time.

These problems haven’t occurred very frequently, but their potential occurrences require hourly checkups else the script could fail if left unchecked. Overnight data collection was risky because of the lack of checkups, but in the cases that they were successful, a partial number of defunct images were present in the dataset. For example, previews of avatars that still have the loading text in the center of the preview.

However, the other problem that exists is executing the script on a different computer. Because of the differing monitor resolutions, adjustments to the program were required. The minimum changes made were of the regions under the each tab because each has a defined upper left coordinate that is correct only in the previous computer’s resolution. The coordinate as a result must be redefined because it would throw off the entire collection process.

CONCLUSION

Besides the unsuccessful captured images, there were very rare occurrences where abnormal morphing would occur (Figure 6). However, server latency was a higher concern, since images were often captured too early (Figure 5 and 8), because the loading text image wasn’t detectable when it was overlaid in the center of the avatar preview. As a result, the program would instead rely on detecting when the black area within the loading box disappears before undertaking any action. However, instructions to rotate the avatar head weren't recognized, and as a result most unsuccessful captured face images were in the default angle (Figure 3). Comparatively, changing body positions were mostly recognized by the server resulting with more successful body images (Figure 7) than face images (Figure 4).

Collected avatar images from previous research are of relatively lower quality than of those in this data set. This is primarily due to the implementation of low graphics requirements to open the virtual community to as many online users without hardware being a limiting factor. Although low graphics result with basic avatar models, simple facial expressions can still be easily noticed and identified (Parke 1972). In more complex avatar models, such as those in this data set, there are thousands of more polygons that allow more complex expressions to be shown.

Furthermore, gathering the data from virtual communities is incredibly tedious, considering there are a number of other problems that exist, such as involuntary self-movement of the avatar (Oursler, Price, and Yampolskiy 2009). For example, the head or eyes of the avatar would always shift in place, and often times the captured image wouldn't be in the position displayed by the avatar in Figure 2. Considering comparing Figures 2 and 4, the images in this data set are of higher quality and of larger resolution.

Overall, this dataset could be repurposed not only for imaged-based CATPCHA tests, but for facial expression recognition and analysis. With the addition of thousands of polygons, creating more complex facial expressions is possible. As such, biometric analysis could be performed on such avatars as a precursor to the analysis of real user biometric data.

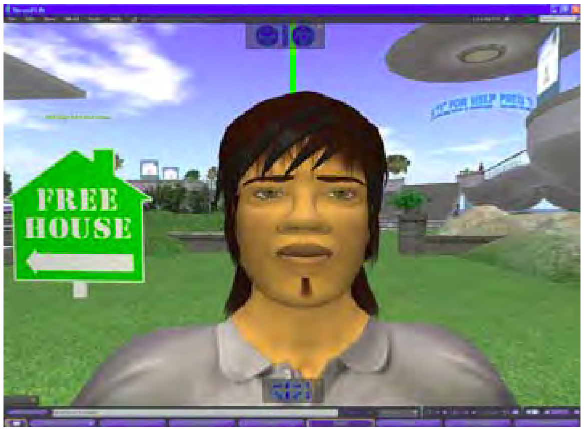


Figure 2. Avatar image captured from Second Life

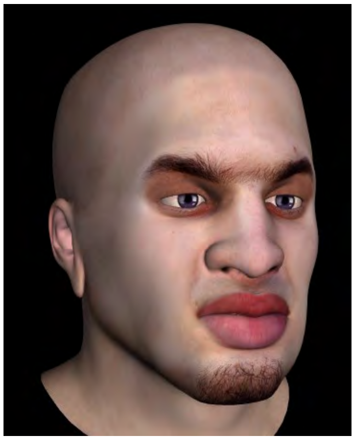


Figure 3. Unsuccessful Captured Face Image - Default Angle Unchanged

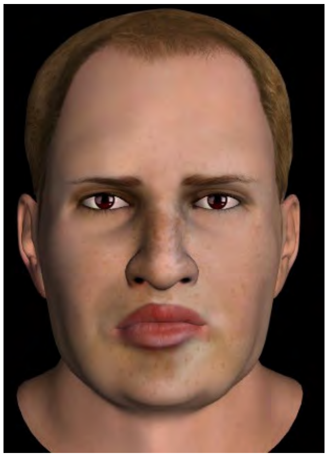


Figure 4. Successful Captured Face Image

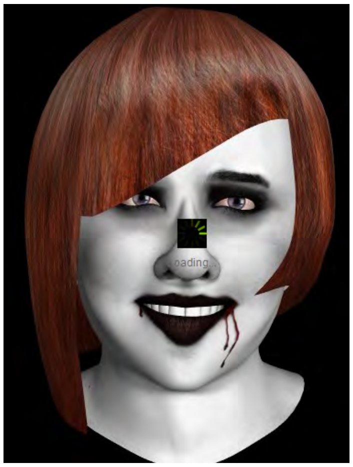


Figure 5. Unsuccessful Captured Face Image - Loading Box in Center

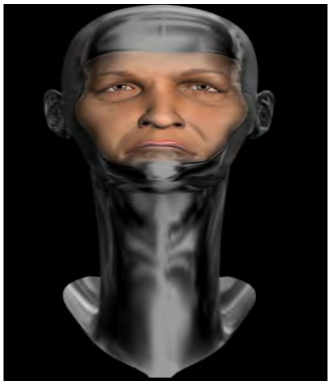


Figure 6. Unsuccessful Captured Image – Abnormal Morph

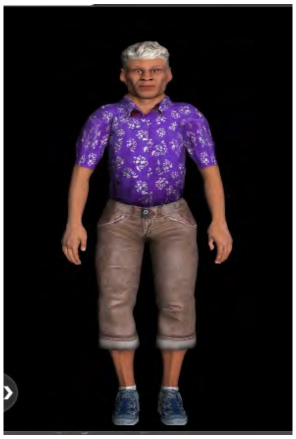


Figure 7. Successful Captured Body Image

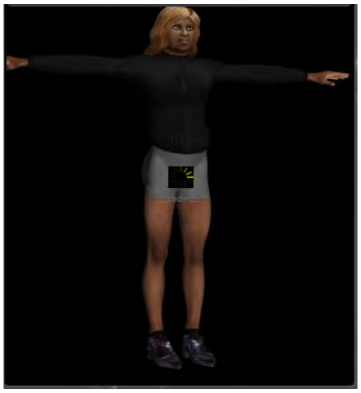


Figure 8. Unsuccessful Captured Body Image – Loading Box in Center

References

Boulay, Jacques-Andre. (2006) "Modeling: Evolver Character Builder." Virtual Reality News and Resources By/for the New World Architects. http://vroot.org/node/722. Retrieved on February 21, 2012.

Bursztein, Elie, Matthieu Martin, and John C. Mitchell. (2011) "Text-based CAPTCHA Strengths and Weaknesses" ACM Conference on Computer and Communications Security (CSS'2011). October 17-21, 2011. Chicago, USA.

Bursztein, Elie, Steven Bethard, Celine Fabry, John C. Mitchell, and Dan Jurafsky. "How Good Are Humans at Solving CAPTCHAs? A Large Scale Evaluation. " *2010* IEEE Symposium on Security and Privacy. pp. 399-413. May 16-19, 2010. Berkeley, CA.

Geetha, Dr. G., and Ragavi V. "CAPTCHA Celebrating Its Quattuordecennial – A Complete Reference." IJCSI International Journal of Computer Science Issues 2nd ser. 8.6 (2011): 340-49. Print.

Lawton, George. (2010) "Screen-Capture Programming: What You See Is What You Script. "Computing Now. IEEE Computer Society, Mar. 2010. Retrieved on February 22, 2012. <http://www.computer.org/portal/web/computing now/archive/news054>.

Oursler, Justin N., Mathew Price, and Roman V. Yampolskiy. Parameterized Generation of Avatar Face Dataset. 14th International Conference on Computer Games: AI, Animation, Mobile, Interactive Multimedia, Educational & Serious Games (CGames'09), pp. 17-22. Louisville, KY. July 29-August 2, 2009.

Parke, Frederick I. "Computer Generated Animation of Faces." ACM '72 Proceedings of the ACM Annual Conference. Vol. 1. NY: ACM New York, 1972. 451-57. Print.

Darwinian-Based Feature Extraction Using K-Means and Kohonen Clustering

Joshua Adams, Joseph Shelton, Lasanio Small, Sabra Neal, Melissa Venable, Jung Hee Kim, and Gerry Dozier
North Carolina Agricultural and Technical State University
1601 East Market Street
Greensboro, NC 27411
jcadams2@ncat.edu, jashelt1@ncat.edu, lrsmall@ncat.edu, saneal@ncat.edu, mdvenabl@ncat.edu, jungkim@ncat.edu, gvdozier@ncat.edu

Abstract

This paper presents a novel approach to feature extraction for face recognition. This approach extends a previously developed method that incorporated the feature extraction techniques of GEFE$_{ML}$ (Genetic and Evolutionary Feature Extraction – Machine Learning) and Darwinian Feature Extraction). The feature extractors evolved by GEFE$_{ML}$ are superior to traditional feature extraction methods in terms of recognition accuracy as well as feature reduction. From the set of feature extractors created by GEFE$_{ML}$, Darwinian feature extractors are created based on the most consistent pixels processed by the set of feature extractors. Pixels selected during the DFE process are then clustered in an attempt to improve recognition accuracy. Our new approach moves clusters towards large groups of selected pixels using techniques such as k-means clustering and Kohonen clustering. Our results show that DFE clustering (DFE$_C$) has statistically better recognition accuracy than DFE without clustering.

Introduction

This paper presents an improvement on a novel approach to identifying areas of facial images for use by the LBP (Local Binary Pattern) feature extraction technique. The traditional LBP technique computes the frequency of different pixel patterns within an entire image. While traditional LBP deterministically computes a binary pattern from every pixel of the image (Ahonen, Hadid and Pietikinen 2006), our approach identifies highly discriminatory sets of isolated pixels and can provide better discrimination while using only some of the image data.

Our previous work on Genetic and Evolutionary Feature Extraction-Machine Learning (GEFE$_{ML}$) (Shelton et al. 2012a) applied Genetic Algorithms (GAs) (Goldberg 1989; Davis 1991) to discover patches of the image for creating LBP feature vectors. Unlike traditional LBP, which divides an image into a grid of patches, we experimentally determined other sets of image patches we call feature extractors. A feature extractor consists of possibly overlapping rectangles of varying size and position. Our feature extractors have been shown to perform better than the traditional grid in the LBP algorithm for discriminating between faces. GEFE$_{ML}$ also uses cross validation techniques (Mitchell 1997) to ensure that our feature extractors generalize well to unseen datasets.

In (Shelton et al. 2012b), a technique was developed that identified pixel locations in feature extractors that seem the most discriminatory. The feature extractors are used to create a hyper feature extractor. From the hyper feature extractor, a pixel frequency matrix is created. A pixel frequency matrix is a two dimensional matrix containing the number of times each pixel was processed by a set of feature extractors. The pixel frequency matrix is used to determine which pixels will be selected for clustering. Pixel locations are chosen via tournament selection where the most consistently used pixels are chosen without replacement. After the pixel locations are selected, the locations are grouped for feature extraction. This grouping process is performed by randomly placing centers and assigning each pixel location to its nearest center. Clustering the pixels identifies the regions of the image that will form the new patches for the new feature extractors. This technique is referred to as Darwinian Feature Extraction (DFE).

In this paper, we improve upon the DFE technique by performing clustering (Dubes and Jain 1988) on the randomly selected pixels. Our results show that clustering techniques improve the recognition accuracy over traditional DFE. We call this technique DFE$_C$ (Darwinian Feature Extraction using Clustering).

The remainder of this paper is as follows: Section 2 provides an overview of the LBP algorithm, Genetic and Evolutionary Computations (GECs), Genetic and Evolutionary Feature Extractors with Machine Learning (GEFE$_{ML}$), and Darwinian Feature Extraction (DFE). Section 3 gives a description of Darwinian-based Feature Extraction using Clustering (DFE$_C$), Sections 4 describes our experiments and Section 5 provides our results. Our conclusions and future work are presented in Section 6.

Background

This section provides an overview of necessary concepts related to this research. We explain the LBP algorithm; we also explain the GECs used to evolve feature extractors in the hyper feature extractor, specifically an Estimation of Distribution Algorithm (EDA), and we explain GEFE$_{ML}$. Finally, we describe the process of DFE.

Local Binary Patterns

The Local Binary Pattern (LBP) of a pixel is a binary number computed by comparing the difference between the intensity of a pixel, c_p, and its surrounding t pixels. Equation 1 shows how to calculate the binary pattern, where n_t is the intensity of the surrounding pixel and c_p is the intensity of the center pixel.

$$LBP(N, c_p) = \sum_{t=0}^{t} \rho(n_t - c_p) 2^t \qquad (1)$$

$$\rho(x) = \begin{cases} 1, x >= 0 \\ 0, x < 0 \end{cases} \qquad (2)$$

Once the binary pattern for a pixel has been determined, it is then checked for uniformity. We define uniform patterns as binary patters that have fewer than three intensity changes. An intensity change occurs if, when reading the pattern from left to right, a 0 precedes a 1 or a 1 precedes a 0. For example, 00111111 contains one intensity change and 01111000 contains two intensity changes. When $t = 7$, there are exactly 58 unique uniform binary patterns. A histogram of 59 elements is created to store the frequency of different binary patterns. The first 58 elements correspond to unique uniform patterns while the last element corresponds to all non-unique patterns.

The traditional way to construct feature vectors from local binary patterns is to first subdivide the image into subregions called patches. Once the image is divided, a histogram is computed for each of the patches. These histograms are then concatenated together to form a feature vector for the whole image. The feature vectors of two images can be compared using any convenient distance metric.

Genetic and Evolutionary Computations

Genetic and evolutionary computations (GECs) are optimization algorithms that simulate the evolutionary processes found in nature (Eiben and Smith, 2003). The GEC used in this study is an Estimation of Distribution algorithm (EDA). This technique (Larranaga and Lozano 2002) generates new generations of candidate solutions from the previous generation. In our research, a candidate solution is a feature extractor. The EDA technique creates a population distribution function from the feature extractors in one generation. The next generation of feature extractors is created by sampling this distribution. A certain number of the top-performing feature extractors, known as elites, are copied into the new generation. This process, as it applies in this study, is described in more detail in Section 2.4 below.

GEFE$_{ML}$

In GEFE$_{ML}$, a feature extractor is represented by a set of patches. Each patch is comprised of four components. The first component is the patch's location (x, y). The second component is the patch's size (height x width). The third component is a masking bit which is used to determine if the patch will contribute to the resulting biometric template and the fourth component is a fitness value representing the quality of that feature extractor.

The fitness, f_i, is calculated using the percentage of the image being processed and the number of incorrect matches that occur when evaluating a training dataset, D. To calculate the number of errors within the training dataset, D is subdivided into a probe and gallery set. The probe set consists of one image per subject and the gallery set consists of two images per subject. Feature extraction is then performed on each image resulting in a set of probe templates and gallery templates. A simulation of the biometric system is then performed by comparing each probe template to each gallery template using a user defined distance metric. The smallest distance between a probe and all gallery elements is considered a match. An error occurs when the template of an individual in the probe set is incorrectly matched with the template of a different individual in the gallery set. The fitness, shown in Equation 3, is the number of errors multiplied by 10 plus the percentage of image being processed.

$$f_i = 10\varepsilon(D) + \gamma(fe_i) \qquad (3)$$

Cross-validation is used to prevent overfitting the training data. This cross-validation process works by keeping track of the feature extractors that generalize well to a dataset of unseen subjects. While offspring in the evolutionary process are being applied to the training dataset, they are also applied to a mutually exclusive validation dataset which does not affect the fitness value. The offspring that perform best on the validation dataset are recorded even if they do not perform well on the training set.

Darwinian-based Feature Extraction

Creating Darwinian feature extractors is a two stage process: (a) creating hyper feature extractors and (b) creating and sampling the pixel frequency matrix.

Hyper Feature Extractors

We can determine the most discriminatory pixels of an image based on which regions are extracted by feature extractors. Examples of feature extractors evolved by GEFE$_{ML}$ are shown in Figure 1. We take a set of feature extractors from GEFE$_{ML}$ and determine the frequency in which each pixel was processed. We do this by overlaying all feature extractors in the set to create what we refer to as a hyper feature extractor. The hyper feature extractor is used to create a pixel frequency matrix. A pixel frequency matrix is a two dimensional matrix containing a count of the number of time a particular pixel location has been processed by each feature extractor from GEFE$_{ML}$. If a pixel's location falls within a patch of a feature extractor, it is said to have been processed. If patches overlap, then the pixels in the overlap were considered to be processed for the number of times that overlap occurs. Once the pixel frequency matrix is created, it is sampled to form the Darwinian feature extractor.

Sampling the Pixel Frequency Matrix

To create the Darwinian feature extractor, a user specified number of clusters and a total number of pixels to be selected for extraction is decided. In addition, a tournament selection pressure, which determines how many pixels are selected to compete for extraction, is also chosen. The pixels that will be chosen to be extracted are based on a k-tournament selection technique. The user specified number of pixels are chosen to compete, and the pixel that has been processed the most gets chosen to be extracted. Once a pixel has been selected via tournament selection, it will not be selected again as a winner.

In (Shelton et al. 2012b), after all pixels have been selected using tournament selection, centroid positions are randomly assigned and the closest pixels to each centroid are assigned to it. This process is referred to as random clustering. After clusters have been created, the pixels within each cluster will be processed using the LBP feature extraction technique and used to form a histogram. After all selected pixels have been processed; histograms are concatenated to form a feature vector.

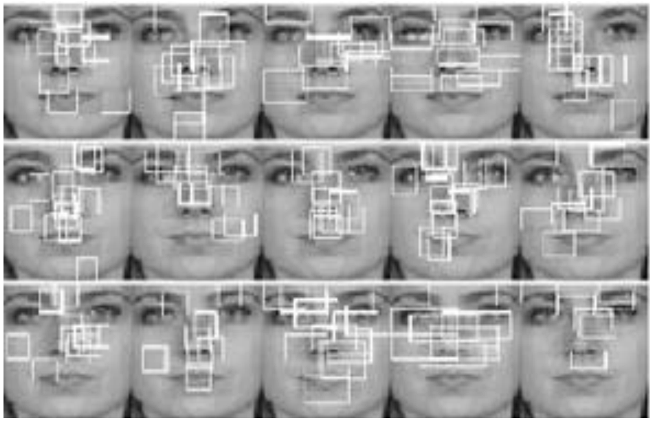


Figure 1: Set of feature extractors.

Techniques for Grouping Pixels to Form Better Feature Extractors

DFE$_C$ (Darwinian Feature Extraction – Clustering) is the process of grouping the pixels selected by DFE. The first step in DFE$_C$ is to form the pixel frequency matrix from the results of GEFE$_{ML}$. A hyper feature extractor is then created from the pixel frequency matrix. A user specified number of centers are randomly placed in the hyper feature extractor and each pixel is assigned to its nearest center. The pixels of the hyper feature extractor are then clustered using one, or a combination, of the following clustering methods.

K-Means Clustering

The process of k-means clustering (Kanungo et al. 2002) the pixels of a Darwinian feature extractor is as follows. Each pixel is assigned to its nearest center. Once this enrollment process is complete, the average location for each enrolled pixel (for a single center) is computed. This average location becomes the new location for that specific center. This process is repeated for each of the user specified centers. Once each of the centers have been relocated, the distance between their old location and their new location is computed. Once the centers no longer move or a maximum of 1,000 iterations have completed, k-means clustering is considered completed.

Kohonen Clustering

The process of clustering the pixels of a Darwinian feature extractor using the Kohonen method (Kohonen 1990) is as follows. Given a user specified learning rate, the Kohonen clustering method iterates though each of the pixels in the Darwinian feature extractor. At each pixel, the nearest center is pulled towards that pixel. The distance which the center is moved is based on the user specified learning rate. With a learning rate of 0.25, the magnitude which the center is moved would be 25% of the distance between the current pixel and the nearest center.

After iterating through all of the pixels, the distance between each centers starting position and ending position is calculated. Kohonen clustering is stopped once the centers no longer move or if one thousand iterations have been completed.

Experiments

To build the hyper feature extractor, we use the 30 best feature extractors created from GEFE$_{ML}$ in previous experiments. Once the hyper feature extractor is created, we use an EDA to evolve the feature extractors based on research (Shelton et al. 2012a) suggesting that EDA is a superior GEC to use. Once we have all of the feature extractors, we built a hyper feature extractor by counting the number of times pixels on an image were processed by each feature extractor. A feature extractor could create up to 24 patches, and we took overlap into account, meaning one feature extractor could process a pixel up to 24 times. We then build a pixel frequency matrix from the hyper feature extractor and sample the pixel frequency matrix to create the new feature extractor.

To evaluate these feature extractors, we apply them to 309 subjects, 3 images per subject, from the Facial Recognition Grand Challenge (FRGC) dataset (Phillips et al. 2005). To test how well each feature extractor generalizes, we divide the FRGC dataset into three subsets. The first subset, FRGC-100$_{trn}$, is used as our training set and consists of 100 subjects. The second subset, FRGC-109, consists of 109 subjects and is used as our validation set. The remaining 100 subjects are used to form our test set, FRGC-100$_{tst}$. The recognition accuracy of each feature extractor was determined using the same method to detect errors in Section 2.3, one-to-many matching with cross validation.

For this experiment, we use 12 clusters, 90% of the total number of pixels processed in the pixel frequency matrix, and used a 10% selection pressure, meaning 10% of 504 pixels were selected for tournament selection. We use a constant of 504 due to this being the average number of pixels in a patch within the set of feature extractors. We use this set-up because this was the best performing setup in (Shelton et al 2012b). After each of the feature extractors have been created, we perform four different clustering methods. The first method, DFE$_{km}$, uses k-means clustering only. The second method, DFE$_{k}$, uses Kohonen clustering only. The third method, DFE$_{km+k}$, first uses K-means clustering. The result of K-means clustering is then clustered using Kohonen clustering. The last method, DFE$_{k+km}$, uses Kohonen clustering first. K-means clustering is then used to cluster the results of Kohonen clustering. Each of these DFE$_{c}$ methods are then applied to FRGC-100$_{tst}$ to evaluate their performance.

Results

The results of DFE$_{c}$ are not deterministic. Each method was tested 30 times; each test utilized 100 different images from FRGC-100tst as described in section 4. Table 1 show our results, including comparison with the best DFE Random Cluster result (Shelton et al., 2012b).

- Column 1 shows the method used.
  - The best DFE Random Cluster used 12 clusters, sampling 0.9 of the pixels, with 0.1 selection pressure (Shelton et al., 2012b).
  - DFE$_{km}$ uses k-means only
  - DFE$_{k}$ uses Kohonen only
  - DFE$_{km+k}$ uses k-means first, then uses these clusters as the starting point for Kohonen
  - DFE$_{k+km}$ uses Kohonen first, then uses these clusters as the starting point for k-means
- Column 2 shows the mean accuracy: the average of the 30 trials of measured accuracies.
- Column 3 shows the standard deviation of the 30 accuracies.

The results of the DFE$_{c}$ instances, when compared to DFE using random clustering, vary in terms of performance. DFE$_{km}$ and DFE$_{k+km}$ have average accuracies that are similar to random clustering. Both DFE$_{k}$ and DFE$_{km+k}$ perform, on average, worse than random clustering. Based on these results, DFE$_{km}$ and DFE$_{k+km}$ seem to be equivalent to DFE. To test this, we performed statistical analysis using a t-test with a 95% confidence interval.

A 2-sample t-test (assuming similar standard deviations) shows that the results of DFE$_{k+km}$ is statistically significant to each of the others ($p < 0.05$), including the best-performing DFE random-clustering method. DFE$_{km}$ is not distinguishable from the random clustering. DFE$_{k}$ and DFE$_{km+k}$ are significantly worse than random clustering.

Given these results, it appears that using Kohonen clustering either as the only technique or using it last in a hybrid clustering technique results in poor performance. However, the hybrid clustering technique proved to be the best, meaning that the order of clustering techniques is relevant.

TABLE 1: RESULTS OF DFE$_{C}$

		n=30 trials
Method	**Mean Accuracy**	**Standard Deviation**
<12,0.9,0.1>	99.34%	0.8840866
DFE$_{km}$	99.43%	0.6260623
DFE$_{k}$	92.23%	3.1588227
DFE$_{km+k}$	92.23%	3.1588227
DFE$_{k+km}$	**99.77%**	**0.4301831**

Conclusion and Future Work

As you can see from the results, DFE$_{k+km}$ provided a higher average accuracy than any of the other methods including DFE without clustering. These results are statistically significant meaning DFE$_{k+km}$ is a better feature extraction method.

None of the other methods were able to achieve an average accuracy greater than or equal to DFE. This gives an indication that the way pixels are grouped together in the feature extraction process significantly affects the accuracy of the biometric system.

The result of Kohonen clustering from a random starting point is fairly stable. Although it theoretically can converge to different solutions from different starting points, as a practical matter different random starting points often converge to the same answer. Thus there is no reason to believe that starting the Kohonen algorithm from the output of k-means would give a better result than the usual random start. Our data bears this out: it seems there was no difference between DFE$_{k}$ and DFE$_{km+k}$.

The results also show that the two methods (DFE$_{km}$ & DFE$_{k+km}$) where k-means computes the final clusters have a) the lowest standard deviations and b) the best results. This is consistent with k-means being fairly stable---usually finding same or similar clusters in the same data. It also seems to show that picking a good set of starting clusters, in this case by using Kohonen first, does indeed improve average K-means performance.

Our future work will include creating a hybrid of genetic and k-means clustering. Future work will also consist of using a lower percentage of top pixels from the hyper feature extractor to see how much the computational complexity can be reduced before the recognition accuracy is affected.

References

T. Ahonen, A. Hadid, M. Pietikinen, "Face Description with Local Binary Patterns: Application to Face Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 12, pp. 2037-2041, Dec. 2006.

K. Alsabti, S. Ranka, V. Singh. “An Efficient K-Means Clustering Algorithm.” http:://www.cise.ufl.edu/~ranka/, 1997

E. Roy. Davies. “Machine Vision: Theory, Algorithms, Practicalities” (3rd ed.). Amsterdam, Boston, 2005.

L. Davis, “Handbook of Genetic Algorithms”, New York: Van Nostrand Reinhold, 1991.

R.C. Dubes and A.K. Jain, “Algorithms for Clustering Data”. Prentice Hall, 1988

A. E. Eiben and J. E. Smith, “Introduction to evolutionary computing”, Springer, 2003.

D.E. Goldberg, “Genetic Algorithms in Search, Optimization & Machine Learning”, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1989.

A. Jain, Lin Hong, and Sharath Pankanti. “Biometric identification.” Commun. ACM 43, 90-98, vol. no. 2 February 2000.

T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, A. Wu, “An Efficient k-Means Clustering Algorithm: Analysis and Implementation,” IEEE Transactions on Pattern analysis and Machine Intelligence, vol 24, no 7 July 2002.

T. Kohonen, “The self-organizing map”, Proceedings IEEE, 78 (9). 1464-1480, (1990).

P. Larranaga, and J. A. Lozano, “Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation”, Kluwer Academic Publishers, 2002.

A. Likas, N. Vlassis, J. Verbeek, “The Global k-Means Clustering Algorithm,” The journal of the Pattern recognition Society, Pattern Recognition vol. no. 36, 451-461, 2001.

T. M. Mitchell, “Machine Learning”, McGraw-Hill Companies, Inc. 1997.

T. Ojala, M. Pietikainen, “Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns”, IEEE Trans. Pattern Analysis and Machine Intellegence; 971-987; 2002

P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoff, J. Marques, J. Min, and W. Worek, “Overview of face recognition grand challenge,” in IEEE Conference on Computer Vision and Pattern Recognition, 2005.

J. Shelton, A. Alford, T. Abagez, L. Small, D. Leflore, J. Williams, J. Adams, G. Dozier, K. Bryant, “Genetic & Evolutionary Biometrics: Feature Extraction from a Machine Learning Perspective”, in submission to IEEE SoutheastCon 2012a.

J. Shelton, M. Venable, S. Neal, J. Adams, A. Alford, G. Dozier; “Pixel Consistency, K-Tournament Selection, and Darwinian-Based Feature Extraction”. Submitted to the Midwest Artificial Intelligence and Cognitive Science Conference (MAICS), 2012b.

M. Su, C. Chou, “A modified version of the K-Means Algorithm with a Distance Based on Cluster Symmetry”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.23, no.6, Jun 2001.

K. Wagstaff, C. Cardie, S. Rogers, S. Schroedl, “Constrained K-Means Clustering with Background Knowledge,” Proceedings of the Eighteenth International Conference on Machine Learning, p. 577-584, 2001.

This page is intentionally left blank

Scientific Computing and Applications

Chair: Jung Hee Kim

Computing Partial Solutions to Difficult AI Problems

Roman V. Yampolskiy

Computer Engineering and Computer Science
Speed School of Engineering
University of Louisville
roman.yampolskiy@louisville.edu

Abstract

Is finding just a part of a solution easier than finding the full solution? For NP-Complete problems (which represent some of the hardest problems for AI to solve) it has been shown that finding a fraction of the bits in a satisfying assignment is as hard as finding the full solution. In this paper we look at a possibility of both computing and representing partial solutions to NP-complete problems, but instead of computing bits of the solution our approach relies on restricted specifications of the problem search space. We show that not only could partial solutions to NP-Complete problems be computed without computing the full solution, but also given an Oracle capable of providing pre-computed partial answer to an NP-complete problem an asymptotic simplification of problems is possible. Our main contribution is a standardized methodology for search space specification which could be used in many distributed computation project to better coordinate necessary computational efforts.

Keywords: *NP-Complete, Partial Solution, Search Space*

Introduction

In "*Computing from Partial Solutions*" Gal et al. (Gal, Halevi et al. 1999) consider the question: "Is finding just a part of a solution easier than finding the full solution?" For NP-Complete problems, such as 3-CNF, they prove that finding a fraction of the bits in a satisfying assignment is as hard as finding the full solution. Specifically they proof that any CNF formula F can be encoded in another formula F', is such a way that given a small fraction of bits in a satisfying assignment to F', it is possible to recover a full satisfying assignment to F (Gal, Halevi et al. 1999):

Theorem 1: For any $\varepsilon > 0$, there exist an efficient probabilistic algorithm A, and an efficient deterministic algorithm B such that:

1. If F is a CNF formula over n variables, then $F' = A(F)$ is a CNF formula over $N = n^{O(1)}$ variables, with $|F'| = |F| + n^{O(1)}$.

2. With probability $1\text{-}2^{-n}$ the formula F' has the following property: If s' any assignment to $N^{.5+\varepsilon}$ of the variables in F' which can be extended to a full satisfying assignment, then $B(F,F',s')$ is a satisfying assignment for F.

Proof of Theorem 1. If we are given polynomial number of random linear equations in n variables, then any sufficiently large subset of these equations is of dimension at least $n\text{-}O(\log n)$, and thus leaves only a polynomial number of candidate solutions to the entire equation system. This, combined with the ability to verify solutions, yields an 'erasure-code' with the ability to correct n – *sqrt(n)* erasures. This improved erasure code can be used to satisfy the claims of Theorem 1. Which was to be shown (Gal, Halevi et al. 1999).

The result is hardly surprising since if finding a part of the solution was possible in polynomial time P = NP would trivially follow. In fact numerous researchers have realized that a related problem of NP-Complete problem re-optimization is not polynomial time solvable unless P = NP (Archetti, Bertazzi et al. 2003; Böckenhauer, Forlizzi et al. 2006; Kralovic and Momke 2007; Ausiello, Escoffier et al. 2009). The proof of that fact due to Archetti et al. follows (Archetti, Bertazzi et al. 2003):

Theorem 2: No polynomial time algorithms can exist for the Re-Optimization of TSPunless P = NP.

Proof of Theorem 2, by contradiction. Suppose that there exists a polynomial time algorithm, for example ReOptTSP, which accomplishes the Re-Optimization of TSP. Then, an optimal solution of any TSP with $n+1$ nodes can be obtained in polynomial time by applying $n\text{-}2$ times the algorithm ReOptTSP. We begin by applying the algorithm ReOptTSP to find an optimal solution of the Re-Optimization of TSP with 4 nodes, given that any 3-city TSP problem is trivially optimally solvable. Then, ReOptTSP is applied to find an optimal solution of the Re-Optimization of TSP with 5 nodes, given an optimal solution of the TSP with 4 nodes, and so on until it is applied to find an optimal solution of the Re-Optimization of TSP with $n+1$ nodes. Thus, by contradiction, no polynomial time algorithms exist for the Re-Optimization of TSP unless P = NP. Which was to be shown (Archetti, Bertazzi et al. 2003).

In this paper we look at a possibility of both computing and representing partial solutions to NP-complete problems, but instead of considering bits of the solution our approach relies on specifications in the problem search space. We show that not only could partial solutions to NP-Complete problems be computed without computing the full solution, but also given a pre-computed partial answer to an NP-complete problem an asymptotic simplification of the problem is possible. Our main contribution is a standardized methodology for search space specification which could be used in many distributed computation project to better coordinate remaining computational efforts. NP-Complete problems are inherently easy to parallelize and so could benefit from a common language aimed at describing what has already been evaluated and what remains to be analyzed.

Search Space Specification

Gal et al. conclusively demonstrate that computing a part of an answer to an NP-Complete problem is as difficult as computing the whole solution (Gal, Halevi et al. 1999) their results are further reaffirmed in (GroBe, Rothe et al. October 4-6, 2001). We propose representing a solution to an NP-Complete problem as a mapping of the total search space subdivided into analyzed and unsearched parts. A full solution to an NP-Complete problem can be represented by the sequential number of the string in an ordered set of all potential solutions. A partial solution can be represented by the best solution found so far along with the description of the already searched potential solutions. The already searched space need not be continuous; the only requirement is that the remaining search space could be separated from the already processed regions. It is easy to see while the smallest possible partial solution can be computed in constant time (this requires analyzing only one potential answer) progressively larger partial solutions are exponentially harder to compute with respect to the size of the problem.

Let’s analyze a specific example of our representation of partial solutions. Travelling Salesperson Problems (TSPs) are easy to visualize and make for an excellent educational tool. Let’s look at a trivial TSP instance with 7 cities numbered from 1 to 7 as depicted in Figure 1. Assuming that the first city in the list is connected to the last one, potential solutions can be represented as a simple numbered list of cities: [1, 2, 3, 4, 5, 6, 7]. The complete search space for our problem consists of all possible permutations of the 7 cities. This complete set could be trivially ordered lexicographically or by taking the value of the permutation condensed into an integer form resulting in non-continuous numbers from 1234567 to 7654321. The position number in which a potential solution appears in the list could be taken as a pointer to that specific solution, with solution 1 refereeing to the [1 → 2 → 3 → 4→ 5 → 6 → 7 → 1] path in our example and solution 2 mapping to [1 → 2 → 3 → 4→ 5 → 7 → 6 → 1], and so on. It is obvious that the same approach can be applied to other NP-Complete problems as they all could potentially be reduced to an instance of TSP. Alternatively a specialized representation could be created for any problem as long as it could be mapped on a countable set of integers. The specific type of ordering is irrelevant as long as it is reproducible and could be efficiently included as metadata accompanying any partial solution.

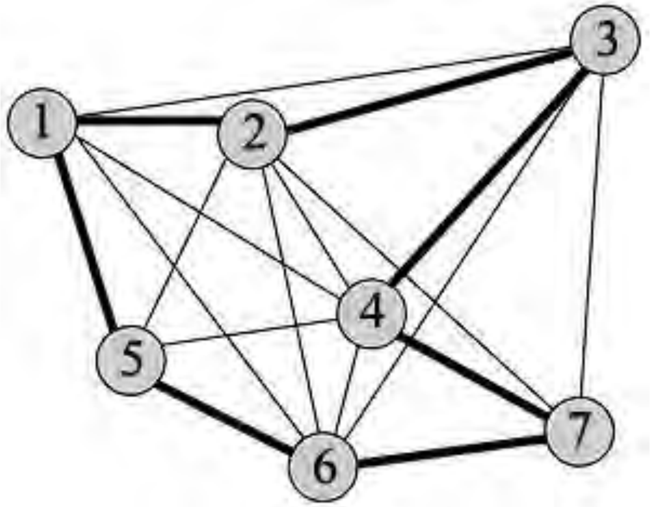


Figure 1. A seven city instance of TSP

Given an ordered set of potential solution it is trivial to specify the regions which have already been processed. In our 7 city TSP example the total search space consist of 7! = 5040 possible paths. A partial solution may be represented by stating that solutions from 1 to 342 have been examined and the best solution is the one in position 187. This means that 4698 additional path remain to be examined and that the partial solution could be said to represent 6.79% of the final answer.

A simple visual diagram can be used to illustrate computed partial solution via visualization of the search space. In Figure 2 a search space of 400 potential solutions is converted to a 2D representation by tilling 20-unit blocks of solutions on top of each other. Solution number 1 is represented by the top left most square with other solutions being introduced sequentially from left to right. Bottom right square is the potential solution numbered 400. Black squares represent already analyzed solutions. White squares are yet to be processed. The example in Figure 2 is a particularly non-contagious partial solution, having no 3 or more continuously examined candidate solutions in a row.

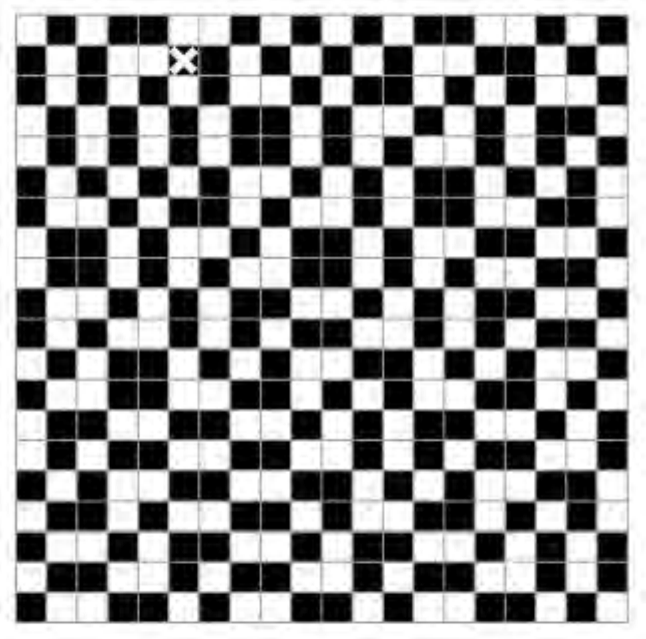


Figure 2. 2D visualization of the search space with respect to searched/unsearched regions and optimal solution found so far indicated by an X.

Pre-computed Partial Solutions

A more natural way of representing partial solution is to directly look at a subset of bits comprising the answer to the problem. Unfortunately finding such bits is as hard as solving the whole problem (Gal, Halevi et al. 1999) and so makes computation of partial solutions represented in this way unfeasible for NP-Complete problems. But suppose that such a partial solution could be computed by an Oracle and provided to us at no additional computational cost. In this section we will look at such situation and analyze difficulty of NP-Complete problems with supplied partial answers.

Returning to our 7-city TSP example and using decimal instead of binary representation of solution (for better readability) a partial solution could be represented as: [1, 2, ?, ?, ?, 6, 7], where "?" represent missing information. The missing information need not be continuous as in: [?, ?, 3, ?, 5, ?, 7] and in the most trivial cases ([1, 2, 3, 4, 5, 6, ?]) may be computed in a constant number of steps. Under this encoding for partial solutions an Oracle may provide enough help to make the problem solvable either in constant time (a small in comparison to N constant number K of missing elements), polynomial time (log N of missing elements), or essentially provide no help by providing only a constant amount of information ([?, ?, ?, 4, ?, ?, ?]). Essentially the Oracle for finding partial solutions to NP-Complete problems has the power to make a problem as easy to solve as it desires, all the way up to single computation.

Conclusions

In this paper we presented a novel way of representing solutions to NP-Complete problems in terms of search space subsets. The proposed methodology allows for easy parallelization of difficult computational problems and is not limited only to NP-Complete problems. Any computational effort can be expressed in terms of search space locations making such computationally intensive projects as Prime Search (mersenne.org), BitCoin (bitcoin.org), Factoring (escatter11.fullerton.edu/nfs), SETI (setiathome.berkeley.edu), Protein Folding (folding.stanford.edu), Game Solving (Schaeffer, Burch et al. September 2007), TSP (Yampolskiy and EL-Barkouky 2011) and Theorem Proving by Computer (Appel, Haken et al. 1977) easier to formalize, verify and break up among numerous computers potentially separated in space and time. While the projects mentioned above all have an internal way of representing the unexplored search space, a common way of specifying such information may lead to standard software capable of shifting unused computational resources among all such efforts.

The proposed solution encoding approach does not represent a breakthrough in our ability to solve NP-Complete problems (Yampolskiy 2011) but it does provide a way to store partial solutions to computationally challenging problems some of which may span decades of effort (Schaeffer, Burch et al. September 2007). Consequently, we are no longer limited to describing particular instances of such problems as solved or unsolved but we can also talk about percentage of the solution we have obtained so far. In the future we plan on addressing such issues as compressibility of representations for multiple non-contiguous sectors in the search space as well as looking into finding optimal orderings for the space of possible solutions to the NP-Complete problems. Additionally, we would like to investigate if by combining our approach with such methods as Monte Carlo simulation (over multiple small partitions of the search space) one can quickly arrive at sufficiently good solutions to very hard problems in cases where optimal solutions are not required.

References

Appel, K., W. Haken, et al. (1977). "Every Planar Map is Four Colorable." Illinois Journal of Mathematics 21: 439-567.

Archetti, C., L. Bertazzi, et al. (2003). "Reoptimizing the Traveling Salesman Problem." Networks 42(3): 154-159.

Ausiello, G., B. Escoffier, et al. (2009). "Reoptimization of Minimum and Maximum Traveling Salesman's Tours." Journal of Discrete Algorithms 7(4) 453--463.

Böckenhauer, H.-J., L. Forlizzi, et al. (2006). Reusing Optimal TSP Solutions for Locally Modified Input Instances 4th IFIP International Conference on Theoretical Computer Science (IFIP TCS).

Gal, A., S. Halevi, et al. (1999). Computing from Partial Solutions. Fourteenth Annual IEEE Conference on Computational Complexity: 34-45.

GroBe, A., J. Rothe, et al. (October 4-6, 2001). Relating Partial and Complete Solutions and the Complexity of Computing Smallest Solutions. 7th Italian Conference on Theoretical Computer Science. Torino, Italy, Springer-Verlag: 339-356.

Kralovic, R. and T. Momke (2007). Approximation Hardness of the Traveling Salesman Reoptimization Problem. 3rd Doctoral Workshop on Mathematical and Engineering Methods in Computer Science: 97-104.

Schaeffer, J., N. Burch, et al. (September 2007). "Checkers is Solved." Science 317(5844): 1518-1522.

Yampolskiy, R. V. (2011). "Construction of an NP Problem with an Exponential Lower Bound." Arxiv preprint arXiv:1111.0305.

Yampolskiy, R. V. and A. EL-Barkouky (2011). "Wisdom of Artificial Crowds Algorithm for Solving NP-Hard Problems." International Journal of Bio-Inspired Computation (IJBIC) 3(6): 358-369.

AI-Complete, AI-Hard, or AI-Easy – Classification of Problems in AI

Roman V. Yampolskiy
Computer Engineering and Computer Science
University of Louisville
Louisville, USA
roman.yampolskiy@louisville.edu

Abstract— The paper contributes to the development of the theory of AI-Completeness by formalizing the notion of AI-Complete and AI-Hard problems. The intended goal is to provide a classification of problems in the field of General Artificial Intelligence. We prove Turing Test to be an instance of an AI-Complete problem and further show numerous AI problems to be AI-Complete or AI-Hard via polynomial time reductions. Finally, the paper suggests some directions for future work on the theory of AI-Completeness.

Keywords- AI-Complete, AI-Easy, AI-Hard, Human Oracle.

I. Introduction

Since its inception in the 1950s the field of Artificial Intelligence has produced some unparalleled accomplishments while at the same time failing to formalize the problem space it is concerned with. This paper proposes to address this shortcoming by contributing to the theory of AI-Completeness, a formalism designed to do for the field of AI what notion of NP-Completeness did for computer science in general. It is our belief that such formalization will allow for even faster progress in solving the remaining problems in humankind's conquest to build an intelligent machine.

According to the encyclopedia Wikipedia the term "AI-Complete" was proposed by Fanya Montalvo in the 1980s (Wikipedia Retrieved January 7, 2011). A somewhat general definition of the term included in the 1991 Jargon File (Raymond March 22, 1991) states:

"**AI-complete***: [MIT, Stanford, by analogy with `NP-complete'] adj. Used to describe problems or subproblems in AI, to indicate that the solution presupposes a solution to the `strong AI problem' (that is, the synthesis of a human-level intelligence). A problem that is AI-complete is, in other words, just too hard. Examples of AI-complete problems are `The Vision Problem', building a system that can see as well as a human, and `The Natural Language Problem', building a system that can understand and speak a natural language as well as a human. These may appear to be modular, but all attempts so far (1991) to solve them have foundered on the amount of context information and `intelligence' they seem to require.*"

As such, the term "AI-Complete" (or sometimes AI-Hard) has been a part of the field for many years and has been frequently brought up to express difficulty of a specific problem investigated by researchers (see (Mallery 1988; Ide and Véronis 1998; Gentry, Ramzan et al. 2005; Nejad April 2010; Bergmair December 2004; McIntire, Havig et al. July 21-23, 2009 ; Navigli and Velardi July 2005; Mueller March 1987; McIntire, McIntire et al. May 18-22, 2009; Chen, Liu et al. November 30, 2009; Mert and Dalkilic September 14-16, 2009 ; Leahu, Sengers et al. September 21 - 24, 2008; Phillips and Beveridge September 28-30,. 2009; Hendler September 2008)). This informal use further encouraged similar concepts to be developed in other areas of science: Biometric-Completeness (Phillips and Beveridge September 28-30,. 2009), ASR-Complete (Morgan, Baron et al. April 6-10, 2003). Recently numerous attempts to formalize what it means to say that a problem is "AI-Complete" have been published (Ahn, Blum et al. 2003; Demasi, Szwarcfiter et al. March 5-8, 2010; Dafna Shahaf and Amir March 26-28, 2007). Even before such formalization attempts, systems which relied on humans to solve problems which were perceived to be AI-Complete were utilized:

- **AntiCaptcha** systems use humans to break CAPTCHA security protocol(Ahn, et al. 2003; Roman V. Yampolskiy 2007a, 2007b; Roman V Yampolskiy and Govindaraju 2007; McDaniel and Yampolskiy 2011) either by directly hiring cheap workers in developing countries (Bajaj April 25, 2010) or by rewarding correctly solved CAPTCHAs with presentation of pornographic images (Vaas December 1, 2007).
- **Chinese Room** philosophical argument by John Searle shows that including a human as a part of a computational system may actually reduce its perceived capabilities such as understanding and consciousness (Searle 1980).

- **Content Development** online projects such as Encyclopedias (Wikipedia, Conservapedia), Libraries (Project Gutenberg, Video collections (YouTube) and Open Source Software (SourceForge) all rely on contributions from people for content production and quality assurance.
- **Cyphermint** a check cashing system relies on human workers to compare a snapshot of a person trying to perform a financial transaction to a picture of a person who initially enrolled with the system. Resulting accuracy outperforms any biometric system and is almost completely spoof proof (see cyphermint.com for more info).
- **Data Tagging** systems entice users into providing meta-data for images, sound or video files. A popular approach involves developing an online game which as a byproduct of participation produces a large amount of accurately labeled data (Ahn June 2006).
- **Distributed Proofreaders** employ a number of human volunteers to eliminate errors in books created by relying on Optical Character Recognition process. (see pgdp.net for more info).
- **Interactive Evolutionary Computation** algorithms use humans in place of a fitness function to make judgments regarding difficult to formalize concept such as esthetic beauty or taste (Takagi 2001).
- **Mechanical Turk** is an Amazon.com's attempt at creating Artificial Artificial Intelligence. Humans are paid varying amounts for solving problems which are believed to be beyond current abilities of AI programs (see mturk.com for more info). The general idea behind the Turk has a broad appeal and the researchers are currently attempting to bring it to the masses via the Generalized Task Markets (GTM) (Horvitz 2007; Horvitz and Paek 2007; Kapoor, Tan et al. 2008; D. Shahaf and Horvitz July 2010).
- **Spam Prevention** is easy to accomplish by having humans vote on emails they receive as spam or not. If a certain threshold is reached a particular piece of email could be said to be spam with a high degree of accuracy (Dimmock and Maddison December 2004).

Recent work has attempted to formalize the intuitive notion of AI-Completeness. In particular three such endowers are worth reviewing:

In 2003 Ahn et al. (Ahn, et al. 2003) attempted to formalize the notion of an AI-Problem and the concept of AI-Hardness in the context of computer security. An AI-Problem was defined as a triple: "$\mathcal{P} = (S, D, f)$, where S is a set of problem instances, D is a probability distribution over the problem set S, and f : S $\rightarrow$ {0; 1}* answers the instances. Let $\delta \in 2\ (0; 1]$. We require that for an $\alpha > 0$ fraction of the humans H, $\Pr_{x \leftarrow D} [H(x) = f(x)] > \delta$… An AI problem $\mathcal{P}$ is said to be (δ,τ)-*solved* if there exists a program A, running in time at most τ on any input from S, such that $\Pr_{x \leftarrow D,r} [A_r(x)=f(x)] \geq \delta$. (A is said to be a (δ,τ) solution to $\mathcal{P}$.) $\mathcal{P}$ is said to be a (δ,τ)-*hard AI problem* if no current program is a (δ,τ) solution to $\mathcal{P}$, and the AI community agrees it is hard to find such a solution." It is interesting to observe that the proposed definition is in terms of democratic consensus by the AI community. If researchers say the problem is hard, it must be so. Also, time to solve the problem is not taken into account. The definition simply requires that some humans be able to solve the problem (Ahn, et al. 2003).

In 2007 Shahaf and Amir (Dafna Shahaf and Amir March 26-28, 2007) have published their work on the Theory of AI-Completeness. Their paper presents the concept of the Human-Assisted Turing Machine and formalizes the notion of different Human Oracles (see Section on Human Oracles for technical details). Main contribution of the paper comes in the form of a method for classifying problems in terms of human-versus-machine effort required to find a solution. For some common problems such as Natural Language Understanding (NLU) the paper proposes a method of reductions allowing conversion from NLU to the problem of Speech Understanding via Text-To-Speech software.

In 2010 Demasi et al. (Demasi, et al. March 5-8, 2010) presented their work on problem classification for Artificial General Intelligence (AGI). The proposed framework groups the problem space into three sectors:

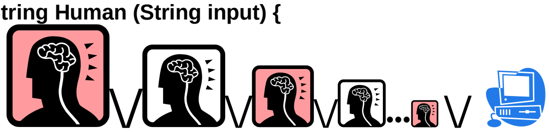
- **Non AGI-Bound**: problems that are of no interest to AGI researchers.
- **AGI-Bound**: problems that require human level intelligence to be solved.
- **AGI-Hard**: problems that are at least as hard as any AGI Bound problem.

The paper also formalizes the notion of Human Oracles and provides a number of definitions regarding their properties and valid operations.

II. The Theory of AI-Completeness

From people with mental disabilities to geniuses human minds are cognitively diverse and it is well known that different people exhibit different mental abilities. We define a notion of a Human Oracle (HO) function capable of computing any function computable by the union of all human minds. In other words any cognitive ability of any human being is repeatable by our HO. To make our Human Oracle easier to understand we provide the following illustration of the *Human* function:

```
String Human (String input) {
```



```
return output; }
```

Figure 1. Human oracle: $\text{Human}_{\text{Best}}$ – a union of minds.

Such a function would be easy to integrate with any modern programming language and would require that the input to the function be provided as a single string of length *N* and the function would return a string of length *M*. No specific encoding is specified for the content of strings *N* or *M* and so they could be either binary representations of data or English language phrases both being computationally equivalent. As necessary the *human* function could call regular TM functions to help in processing of data. For example, a simple computer program which would display the input string as a picture to make human comprehension easier could be executed. Humans could be assumed to be cooperating perhaps because of a reward. Alternatively, one can construct a *Human* function which instead of the union of all minds computes the average decision of all human minds on a problem encoded by the input string as the number of such minds goes to infinity. To avoid any confusion we propose naming the first HO $\text{Human}_{\text{Best}}$ and the second HO $\text{Human}_{\text{Average}}$. Problems in the AI domain tend to have a large degree of ambiguity in terms of acceptable correct answers. Depending on the problem at hand the simplistic notion of an average answer could be replaced with an aggregate answer as defined in the Wisdom of Crowds approach (Surowiecki 2004). Both functions could be formalized as Human-Assisted Turing Machines (Dafna Shahaf and Amir March 26-28, 2007).

The Human function is easy to understand and uses generalization of the Human Oracle. One can perceive it as a way to connect and exchange information with a real human sitting at a computer terminal. While easy to intuitively understand, such description is not sufficiently formal. Shahaf et al. have formalized the notion of Human Oracle as an HTM (Dafna Shahaf and Amir March 26-28, 2007). In their model a human is an oracle machine that can decide a set of languages L_i in constant time: $H \subseteq \{L_i \mid L_i \subseteq \sum^*\}$. If time complexity is taken into account answering a question might take a non-constant time: $H \subseteq \{<L_i , f_i> \mid L_i \subseteq \sum^*, f_i \ : \mathbb{N} \rightarrow \mathbb{N}\}$ there f_i is the time-complexity function for language L_i, meaning the human can decide if $x \in L_i$ in f_i $(|x|)$ time. In order to realistically address capabilities of individual humans a probabilistic oracle was also presented which provided correct answers with probability p: $H \subseteq \{<L_i , p_i> \mid L_i \subseteq \sum^*, 0 \leq p_i \leq 1\}$. Finally the notion of reward is introduced into the model to capture humans improved performance on “paid” tasks: $H \subseteq \{<L_i , u_i> \mid L_i \subseteq \sum^*, u_i \ : \mathbb{N} \rightarrow \mathbb{N}\}$ where u_i is the utility function (Dafna Shahaf and Amir March 26-28, 2007).

A. *Definitions*

Definition 1: A problem *C* is **AI-Complete** if it has two properties:

1. It is in the set of AI problems (Human Oracle solvable).

2. Any AI problem can be converted into *C* by some polynomial time algorithm.

Definition 2: **AI-Hard:** A problem *H* is AI-Hard if and only if there is an AI-Complete problem C that is polynomial time Turing-reducible to H.

Definition 3: **AI-Easy:** The complexity class AI-easy is the set of problems that are solvable in polynomial time by a deterministic Turing machine with an oracle for some AI problem. In other words, a problem X is AI-easy if and only if there exists some AI problem Y such that X is polynomial-time Turing reducible to Y. This means that given an oracle for Y, there exists an algorithm that solves X in polynomial time.

Figure 2 illustrates relationship between different AI complexity classes. Right side illustrates the situation if it is ever proven that AI-problems = AI-Complete problems. Left side shows the converse.

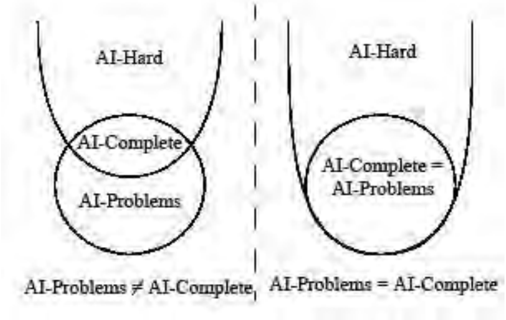


Figure 2. Relationship between AI complexity classes.

B. *Turing Test as the First AI-Complete Problem*

In this section we will show that a Turing Test (A Turing 1950) problem is AI-Complete. First we need to establish that Turing Test is indeed an AI problem (HO solvable). This trivially follows from the definition of the test itself. The test measures if a human-like performance is demonstrated by the test taker and Human Oracles are defined to produce human level performance. While both “human” and “intelligence test” are intuitively understood terms we have already shown that Human Oracles could be expressed in strictly formal terms. The Turing Test itself also could be formalized as an interactive proof (Bradford and Wollowski 1995; Shieber December 2007, July 16-20, 2006).

Second requirement for a problem to be proven to be AI-Complete is that any other AI problem should be convertible into an instance of the problem under consideration in polynomial time via Turing reduction. Therefore we need to show how any problem solvable by the Human function could be encoded as an instance of a Turing Test. For any HO-solvable problem *h* we have a String *input* which encodes the problem and a String *output* which encodes the solution. By taking the *input* as a question to be used in the TT and *output* as an answer to be expected while administering a TT we can see how any HO-solvable problem could be reduced in polynomial time to an instance of a Turing Test. Clearly the described process is in polynomial time and by similar algorithm any AI problem could be reduced to TT. It is even theoretically possible to construct a complete TT which utilizes all other problems solvable by HO by generating one question from each such problem.

C. Reducing Other Problems to TT

Having shown a first problem (Turing Test) to be AI-Complete the next step is to see if any other well-known AI-problems are also AI-complete. This is an effort similar to the work of Richard Carp who has shown some 21 problems to be NP-Complete in his 1972 paper and by doing so started a new field of Computational Complexity (Karp 1972). According to the Encyclopedia of Artificial Intelligence (Shapiro 1992) published in 1992 the following problems are all believed to be AI-Complete and so will constitute primary targets for our effort of proving formal AI-Completeness on them (Shapiro 1992):

- **Natural Language Understanding** – "Encyclopedic knowledge is required to understand natural language. Therefore, a complete Natural Language system will also be a complete Intelligent system."
- **Problem Solving** – "Since any area investigated by AI researchers may be seen as consisting of problems to be solved, all of AI may be seen as involving Problem Solving and Search".
- **Knowledge Representation and Reasoning** – "…the intended use is to use explicitly stored knowledge to produce additional explicit knowledge. This is what reasoning is. Together Knowledge representation and Reasoning can be seen to be both necessary and sufficient for producing general intelligence – it is another AI-complete area."
- **Vision or Image Understanding** – "If we take "interpreting" broadly enough, it is clear that general intelligence may be needed to do this interpretation, and that correct interpretation implies general intelligence, so this is another AI-complete area."

Now that Turing Test has been proven to be AI-Complete we have an additional way of showing other problems to be AI-Complete. We can either show that a problem is in the set of AI problems and all other AI problem can be converted into it by some polynomial time algorithm or we can reduce any instance of Turing Test problem (or any other already proven to be AI-Complete problem) to an instance of a problem we are trying to show to be AI-Complete. This second approach seems to be particularly powerful. The general heuristic of our approach is to see if all information which encodes the question which could be asked during the administering of a Turing Test could be encoded as an instance of a problem in question and likewise if any potential solution to that problem would constitute an answer to the relevant Turing Test question. Under this heuristic it is easy to see that for example Chess is not AI-Complete as only limited information can be encoded as a starting position on a standard size chess board. Not surprisingly Chess has been one of the greatest successes of AI and currently Chess playing programs dominate all human players including world champions.

Question Answering (QA) (Hirschman and Gaizauskas 2001; Salloum November 30, 2009) is a sub-problem in Natural Language Processing. Answering questions at a level of a human is something HOs are particularly good at based on their definition. Consequently QA is an AI-Problem which is one of the two requirements for showing it to be AI-Complete. Having access to an Oracle capable of solving QA allows us to solve TT via a simple reduction. For any statement *S* presented during administration of TT we can transform said statement into a question for the QA Oracle. The answers produced by the Oracle can be used as replies in the TT allowing the program to pass the Turing Test. It is important to note that access to the QA oracle is sufficient to pass the Turing Test only if questions are not restricted to stand alone queries, but could contain information from previous questions. Otherwise the problem is readily solvable even by today's machines such as IBM's Watson which showed a remarkable performance against human Jeopardy champions (Pepitone Retrieved on: January 13, 2011).

Speech Understanding (SU) (Anusuya and Katti 2009) is another sub-problem in Natural Language Processing. Understanding Speech at a level of a human is something HOs are particularly good at based on their definition. Consequently SU is an AI-Problem which is one of the two requirements for showing it to be AI-Complete. Having access to an Oracle capable of solving SU allows us to solve QA via a simple reduction. We can reduce QA to SU by utilizing any Text-to-Speech software (Taylor and Black 1999; Chan 2003) which is both fast and accurate. This reduction effectively transforms written questions into the spoken ones making it possible to solve every instance of QA by referring to the SU oracle.

D. *Other Probably AI-Complete Problems*

Figure 3 shows the relationship via reductions between problems shown to be AI-Complete in this paper. We hope that our work will challenge the AI community to prove other important problems as either belonging or not belonging to that class. While the following problems have not been explicitly shown to be AI-Complete, they are strong candidates for such classification and are also problems of great practical importance making their classification a worthy endower. If a problem has been explicitly conjectured to be AI-Complete in a published paper we include a source of such speculation: Dreaming (Salloum November 30, 2009), Commonsense Planning (Dafna Shahaf and Amir March 26-28, 2007), Foreign Policy (Mallery 1988), Problem Solving (Shapiro 1992), Judging a Turing Test (Dafna Shahaf and Amir March 26-28, 2007), Common Sense Knowledge (Andrich, Novosel et al. 2009), Speech Understanding (Dafna Shahaf and Amir March 26-28, 2007), Knowledge Representation and Reasoning (Shapiro 1992), Word Sense Disambiguation (Navigli and Velardi July 2005; Chen, et al. November 30, 2009), Machine Translation (Wikipedia Retrieved January 7, 2011), Ubiquitous Computing (Leahu, et al. September 21 - 24, 2008), Change Management for Biomedical Ontologies (Nejad April 2010), Natural Language Understanding (Shapiro 1992), Software Brittleness (Wikipedia Retrieved January 7, 2011), Vision or Image Understanding (Shapiro 1992).

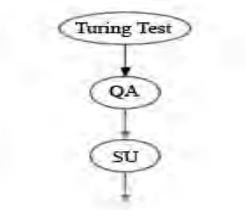


Figure 3. Reductions from the first NP-Complete problem.

E. *1st AI-Hard Problem: Programming*

We define the problem of Programming as taking a natural language description of a program and producing a source code which then compiled on some readily available hardware/software produces a computer program which satisfies all implicit and explicit requirements provided in the natural language description of the programming problem assignment. Simple examples of Programming are typical assignments given to students in computer science classes. Ex. "Write a program to play Tic-Tac-Toe." with successful students writing source code which if correctly compiled allows the grader to engage the computer in an instance of that game. Many requirements of such assignment remain implicit such as that response time of the computer should be less than a minute. Such implicit requirements are usually easily inferred by students who have access to culture instilled common sense. As of this writing no program is capable of solving Programming outside of strictly restricted domains.

Having access to an Oracle capable of solving Programming allows us to solve TT via a simple reduction. For any statement *S* presented during TT we can transform said statement into a programming assignment of the form: "Write a program which would respond to S with a statement indistinguishable from a statement provided by an average human" (A full transcript of the TT may also be provided for disambiguation purposes). Applied to the set of all possible TT statements this procedure clearly allows us to pass TT, however Programming itself is not in the set of AI-Problems as there are many instances of Programming which are not solvable by Human Oracles. For example "Write a program to pass Turing Test" is not known to be an AI-Problem under the proposed definition. Consequently, Programming is an AI-Hard problem.

III. BEYOND AI-COMPLETENESS

The human oracle function presented in this paper assumes that the human being behind it has some assistance from the computer in order to process certain human unfriendly data formats. For example a binary string representing a video is completely impossible for a human being to interpret but could easily be played by a computer program in the intended format making it possible for a human to solve a video understanding related AI-Complete problem. It is obvious that a human being provided with access to a computer (perhaps with Internet connection) is more intelligent compared to a human unenhanced in such a way. Consequently it is important to limit help from a computer to a human worker inside a human Oracle function to assistance in the domain of input/output conversion but not beyond as the resulting function would be both AI-Complete and "Computer Complete".

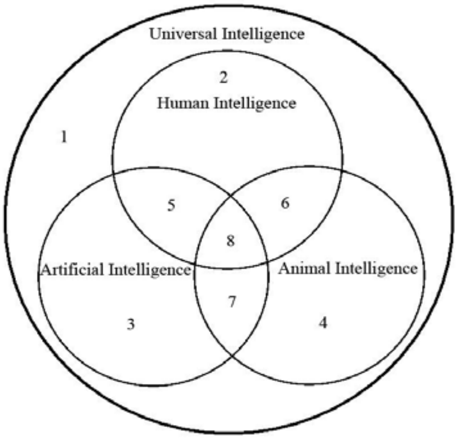


Figure 4. **Fig. 1.** Venn diagram for four different types of intelligence.

Figure 4 utilizes a Venn diagram to illustrate subdivisions of problem space produced by different types of intelligent computational devices. Region 1 represents what is known as Universal Intelligence (Legg and Hutter December 2007) or Super Intelligence (R.V. Yampolskiy 2011; Roman V. Yampolskiy 2012; Roman V. Yampolskiy and Fox 2012b, 2012a; Legg June 2008; Roman V. Yampolskiy October 3-4, 2011a, October 3-4, 2011b) a computational agent which outperforms all other intelligent agents over all possible environments. Region 2 is the standard unenhanced Human level intelligence of the type capable of passing a Turing Test, but at the same time incapable of computation involving large numbers or a significant amount of memory. Region 3 is what is currently possible to accomplish via the state-of-the-art AI programs. Finally Region 4 represents an abstract view of animal intelligence. AI intelligence researchers strive to produce Universal Intelligence and it is certainly likely to happen given recent trends in both hardware and software developments and theoretical underpinning of the Church/Turing Thesis (AM Turing 1936). It is also likely, that if we are able to enhance human minds with additional memory and port them to a higher speed hardware we will essentially obtain a Universal Intelligence (Sandberg and Boström 2008).

While Universal Intelligence incorporates abilities of all the lower intelligences it is interesting to observe that Human, AI and Animal intelligences have many interesting regions of intersection. For example animal minds are as good as human minds at visual understanding of natural scenes. Regions 5, 6, and 7 illustrate common problem spaces between two different types of intelligent agents. Region 8 represents common problem solving abilities of humans, computers and animals. Understanding such regions of commonality may help us to better separate involved computational classes which are represented by abilities of a specific computational agent minus the commonalities with a computational agent with which we are trying to draw a distinction. For example CAPTCHA (Ahn, et al. 2003) type tests rely on the inability of computers to perform certain pattern recognition tasks with the same level of accuracy as humans to separate AI agents from Human agents. Alternatively a test could be devised to tell humans not armed with calculators from AIs by looking at the upper level of ability. Such a test should be easy to defeat once an effort is made to compile and formalize the limitations and biases of the human mind.

It is also interesting to consider the problem solving abilities of hybrid agents. We have already noted that a human being equipped with a computer is a lot more capable compared to an unaided person. Some recent research in Brain Computer Interfaces (Vidal 1973) provides a potential path for future developments in the area. Just as interestingly combining pattern recognition abilities of animals with symbol processing abilities of AI could produce a computational agent with a large domain of human like abilities (see work on RoboRats (Talwar, Xu et al. 2 May 2002) on monkey controlled robots (Nicolelis, Wessberg et al. 2000)). It is very likely that in the near future the different types of intelligent agents will combine to an even greater extent. While such work is under way we believe that it may be useful to introduce some additional terminology into the field of problem classification. For the complete space of problems we propose that the computational agents which are capable of solving a specific subset of such problems get to represent the set in question. Therefore we propose additional terms: “Computer-Complete” and “Animal-Complete” to represent computational classes solvable by such agents. It is understood that just like humans differ in their abilities so do animals and computers. Aggregation and averaging utilized in our human function could be similarly applied to the definition of the respective oracles. As research progresses common names may be needed for different combinations of regions from Figure 8 illustrating such concepts as Human-AI hybrid or Animal-Robot hybrid.

IV. CONCLUSIONS

Progress in the field of artificial intelligence requires access to well defined problems of measurable complexity. The theory of AI-Completeness aims to provide a base for such formalization. Showing certain problems to be AI-Complete/-Hard is useful for developing novel ways of telling computers from humans. Also, any problem shown to be AI-Complete would be a great alternative way of testing an artificial intelligent agent to see if it attained human level intelligence (Dafna Shahaf and Amir March 26-28, 2007).

REFERENCES

Ahn, Lv. (June 2006). Games With A Purpose. *IEEE Computer Magazine*, 96-98.

Ahn, Lv, Blum, M, Hopper, N, and Langford, J. (2003). *CAPTCHA: Using Hard AI Problems for Security*. Eurocrypt.

Andrich, C, Novosel, L, and Hrnkas, B. (2009). *Common Sense Knowledge*. Paper presented at the Information Search and Retrieval, Available at: http://www.iicm.tu-graz.ac.at/cguetl/courses/isr/uearchive/uews2009/Ue06-CommonSenseKnowledge.pdf.

Anusuya, MA, and Katti, SK. (2009). Speech Recognition by Machine: A Review. *International Journal of Computer Science and Information Security (IJCSIS), 6(3)*, 181-205.

Bajaj, V. (April 25, 2010). *Spammers Pay Others to Answer Security Tests*. Paper presented at the The New York Times.

Bergmair, R. (December 2004). *Natural Language Steganography and an ``AI-complete'' Security Primitive*. Paper presented at the 21st Chaos Communication Congress, Berlin.

Bradford, PG, and Wollowski, M. (1995). A formalization of the Turing Test. *SIGART Bulletin, 6(4)*, 3-10.

Chan, T-Y. (2003). *Using a Text-to-Speech Synthesizer to Generate a Reverse Turing Test.* 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'03).

Chen, J, Liu, J, Yu, W, and Wu, P. (November 30, 2009). *Combining Lexical Stability and Improved Lexical Chain for Unsupervised Word Sense Disambiguation*. Paper presented at the Second International Symposium on Knowledge Acquisition and Modeling (KAM '09), Wuhan

Demasi, P, Szwarcfiter, JL, and Cruz, AJO. (March 5-8, 2010). *A Theoretical Framework to Formalize AGI-Hard Problems*. Paper presented at The Third Conference on Artificial General Intelligence, Lugano, Switzerland.

Dimmock, N, and Maddison, I. (December 2004). Peer-to-peer collaborative spam detection. *Crossroads, 11(2)*.

Gentry, C, Ramzan, Z, and Stubblebine, S. (2005). *Secure distributed human computation*. Paper presented at the 6th ACM conference on Electronic commerce.

Hendler, J. (September 2008). We've Come a Long Way, Maybe …. *IEEE Intelligent Systems, 23(5)*, 2-3.

Hirschman, L, and Gaizauskas, R. (2001). Natural Language Question Answering. The View from Here. *Natural Language Engineering, 7(4)*, 275-300.

Horvitz, E. (2007). Reflections on Challenges and Promises of Mixed-Initiative Interaction. *AI Magazine-Special Issue on Mixed-Initiative Assistants, 28(2)*.

Horvitz, E, and Paek, T. (2007). Complementary Computing: Policies for Transferring Callers from Dialog Systems to Human Receptionists. *User Modeling and User Adapted Interaction, 17(1)*, 159-182.

Ide, N, and Véronis, J. (1998). Introduction to the special issue on word sense disambiguation: the state of the art. *Computational Linguistics, 24(1)*, 1-40.

Kapoor, A, Tan, D, Shenoy, P, and Horvitz, E. (2008). *Complementary Computing for Visual Tasks: Meshing Computer Vision with Human Visual Processing*. Paper presented at the IEEE International Conference on Automatic Face and Gesture Recognition.

Karp, RM. (1972). Reducibility Among Combinatorial Problems. In RE Miller & JW Thatcher (Eds.), *Complexity of Computer Computations* (pp. 85-103). New York: Plenum.

Leahu, L, Sengers, P, and Mateas, M. (September 21 - 24, 2008). *Interactionist AI and the promise of ubicomp, or, how to put your box in the world without putting the world in your box*. Paper presented at the Tenth International Conference on Ubiquitous Computing, Seoul, South Korea.

Legg, S. (June 2008). *Machine Super Intelligence*. Paper presented at the PhD Thesis, University of Lugano, Available at: http://www.vetta.org/documents/Machine_Super_Intelligence.pdf.

Legg, S, and Hutter, M. (December 2007). Universal Intelligence: A Definition of Machine Intelligence. *Minds and Machines, 17(4)*, 391-444.

Mallery, JC. (1988). *Thinking About Foreign Policy: Finding an Appropriate Role for Artificially Intelligent Computers*. Paper presented at the Annual Meeting of the International Studies Association, St. Louis, MO.

McDaniel, R, and Yampolskiy, RV. (2011). *Embedded non-interactive CAPTCHA for Fischer Random Chess*. Paper presented at the 16th International Conference on Computer Games (CGAMES), Louisville, KY.

McIntire, JP, Havig, PR, and McIntire, LK. (July 21-23, 2009). *Ideas on authenticating humanness in collaborative systems using AI-hard problems in perception and cognition*. Paper presented at the IEEE National Aerospace & Electronics Conference (NAECON), Dayton, OH.

McIntire, JP, McIntire, LK, and Havig, PR. (May 18-22, 2009). *A variety of automated turing tests for network security: Using AI-hard problems in perception and cognition to ensure secure collaborations*. Paper presented at the International Symposium on Collaborative Technologies and Systems (CTS '09) Baltimore, MD.

Mert, E, and Dalkilic, C. (September 14-16, 2009). *Word sense disambiguation for Turkish*. Paper presented at the 24th International Symposium on Computer and Information Sciences (ISCIS 2009), Guzelyurt.

Morgan, N, Baron, D, Bhagat, S, Carvey, H, Dhillon, R, Edwards, J, . . . Wooters, C. (April 6-10, 2003). *Meetings about meetings: research at ICSI on speech in multiparty conversations*. Paper presented at the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03).

Mueller, ET. (March 1987). *Daydreaming and Computation. Ph.D. Dissertation, University of California*. Los Angeles.

Navigli, R, and Velardi, P. (July 2005). Structural Semantic Interconnections: A Knowledge-Based Approach to Word Sense Disambiguation. *IEEE Transactions On Pattern Analysis and Machine Intelligence, 27(7)*, 1075-1086.

Nejad, AS. (April 2010). *A Framework for Analyzing Changes in Health Care Lexicons and Nomenclatures. PhD dissertation. Concordia University*. Montreal, Quebec, Canada.

Nicolelis, MAL, Wessberg, J, Stambaugh, CR, Kralik, JD, Beck, PD, Laubach, M, . . . Kim, J. (2000). Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature, 408(6810)*, 361.

Pepitone, J. (Retrieved on: January 13, 2011). *IBM's Jeopardy supercomputer beats humans in practice bout*. Paper presented at the CNNMoney, Available at: http://money.cnn.com/2011/01/13/technology/ibm_jeopardy_watson.

Phillips, PJ, and Beveridge, JR. (September 28-30,. 2009). *An introduction to biometric-completeness: The equivalence of matching and quality*. Paper presented at the IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems (BTAS '09) Washington, DC

Raymond, ES. (March 22, 1991). *Jargon File Version 2.8.1*. Available at: http://catb.org/esr/jargon/oldversions/jarg282.txt.

Salloum, W. (November 30, 2009). *A Question Answering System based on Conceptual Graph Formalism*. Paper presented at the The 2nd International Symposium on Knowledge Acquisition and Modeling (KAM 2009), China.

Sandberg, A, and Boström, N. (2008). *Whole Brain Emulation: A Roadmap*. Paper presented at the Future of Humanity Institute, Oxford University. Technical Report #2008-3, Available at: http://www.fhi.ox.ac.uk/Reports/2008-3.pdf.

Searle, J. (1980). Minds, Brains and Programs. *Behavioral and Brain Sciences, 3(3)*, 417-457.

Shahaf, D, and Amir, E. (March 26-28, 2007). *Towards a theory of AI completeness*. Paper presented at the 8th International Symposium on Logical Formalizations of Commonsense Reasoning (Commonsense 2007), California.

Shahaf, D, and Horvitz, E. (July 2010). *Generalized Task Markets for Human and Machine Computation*. Paper presented at the Twenty-Fourth AAAI Conference on Artificial Intelligence, Atlanta, GA.

Shapiro, SC. (1992). Artificial Intelligence. In SC Shapiro (Ed.), *Encyclopedia of Artificial Intelligence* (pp. 54-57). New York: John Wiley.

Shieber, SM. (December 2007). The Turing Test as Interactive Proof. *Nous, 41(4)*, 686-713.

Shieber, SM. (July 16-20, 2006). *Does the Turing Test demonstrate intelligence or not*. Paper presented at the Twenty-First National Conference on Artificial Intelligence (AAAI-06), Boston, MA.

Surowiecki, J. (2004). *The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations*: Little, Brown.

Takagi, H. (2001). Interactive Evolutionary Computation: Fusion of the Capacities of EC Optimization and Human Evaluation. *Proceesings of the IEEE 89, 9*, 1275-1296.

Talwar, SK, Xu, S, Hawley, ES, Weiss, SA, Moxon, KA, and Chapin, JK. (2 May 2002). Behavioural neuroscience: Rat navigation guided by remote control. *Nature, 417*, 37-38.

Taylor, P, and Black, A. (1999). *Speech synthesis by phonological structure matching*. In *Eurospeech99*, Budapest, Hungary.

Turing, A. (1950). Computing Machinery and Intelligence. *Mind, 59(236)*, 433-460.

Turing, AM. (1936). On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society, 42*, 230-265.

Vaas, L. (December 1, 2007). *Striptease Used to Recruit Help in Cracking Sites*. Paper presented at the PC Magazine.

Vidal, J. (1973). Toward direct brain-computer communication. *Annual Review of Biophysics and Bioengineering, 2*, 157-180.

Wikipedia. (Retrieved January 7, 2011). *AI-Complete*. Available at: http://en.wikipedia.org/wiki/AI-complete.

Yampolskiy, RV. (2007a, April 13, 2007). *Embedded CAPTCHA for Online Poker*. 20th Annual CSE Graduate Conference (Grad-Conf2007), Buffalo, NY.

Yampolskiy, RV. (2007b, September 28, 2007). *Graphical CAPTCHA embedded in cards*. Western New York Image Processing Workshop (WNYIPW) - IEEE Signal Processing Society, Rochester, NY.

Yampolskiy, RV. (2011). AI-Complete CAPTCHAs as Zero Knowledge Proofs of Access to an Artificially Intelligent System. *ISRN Artificial Intelligence, 271878*.

Yampolskiy, RV. (2012). Leakproofing Singularity - Artificial Intelligence Confinement Problem. *Journal of Consciousness Studies (JCS), 19(1-2)*, 194–214.

Yampolskiy, RV. (October 3-4, 2011a). *Artificial Intelligence Safety Engineering: Why Machine Ethics is a Wrong Approach*. Paper presented at the Philosophy and Theory of Artificial Intelligence (PT-AI2011), Thessaloniki, Greece.

Yampolskiy, RV. (October 3-4, 2011b). *What to Do with the Singularity Paradox?* Paper presented at the Philosophy and Theory of Artificial Intelligence (PT-AI2011), Thessaloniki, Greece.

Yampolskiy, RV, and Fox, J. (2012a). Artificial Intelligence and the Human Mental Model. In A Eden, J Moor, J Soraker & E Steinhart (Eds.), *In the Singularity Hypothesis: a Scientific and Philosophical Assessment*: Springer.

Yampolskiy, RV, and Fox, J. (2012b). Safety Engineering for Artificial General Intelligence. *Topoi Special Issue on Machine Ethics & the Ethics of Building Intelligent Machines, (In Press)*.

Yampolskiy, RV, and Govindaraju, V. (2007). Embedded Non-Interactive Continuous Bot Detection. *ACM Computers in Entertainment, 5(4)*, 1-11.

This page is intentionally left blank

Poster Session 1

Genetic Algorithms and Book Embeddings: A Dual Layered Approach

Shannon Overbay Paul De Palma Marshall Hurson Tiffany Arnold Andrew Pierce
Gonzaga University
Spokane, WA 99258 USA
overbay@gonzaga.edu

Abstract

The genetic algorithm (GA) has been applied to a wide variety of problems where truly optimal solutions are computationally intractable. One such problem is the book embedding problem from graph theory. A book embedding is an ordering of vertices along a line (the spine) with the edges embedded in half-planes (the pages) extruding from the line so that the edges do not cross. The goal is to find the minimal number of half-planes needed to embed a given graph. This problem is known to be NP-complete. The paper shows that the GA can be used to generate counter-examples to conjectured minimum bounds.

Introduction

The idea that there might be something to be gained by applying the principles of Darwinian natural selection to computing is not new. Turing himself proposed evolutionary search as early as 1948. Though John Holland at the University of Michigan coined the term "genetic algorithm" in the mid-seventies, the GA was not widely studied until 1989 when D.E. Goldberg showed that it could be used to solve a number of difficult problems (Holland, 1975; Goldberg, 1989; Luger and Stubblefield, 2009). At least some of those difficult problems are in the equivalence class "consisting of the 'hardest' problems in NP," namely the class of NP-complete problems (Garey and Johnson, 1979: 14). Researchers who investigate problems in this class must content themselves with heuristic approaches, constraint relaxation, and, crucially, with sub-optimal solutions.

This paper argues that the GA can be effectively used in a problem from graph theory known as book embedding. A book embedding of a graph is an ordering of the vertices along a line in 3-space (the spine) along with an assignment of each edge to a single half-plane extruding from the spine (a page) such that the edges do not cross each other or the spine. The goal is to find the minimal number of pages needed to embed a given graph in a book. The study of book embedding is of interest both as a theoretical area of topological graph theory and as a practical subject with numerous applications.

There has been a recent boom in interest in book embedding, motivated by its usage in modeling a variety of problems. Book embeddings have been applied to fault-tolerant VLSI design, sorting with parallel stacks, single-row routing, and complexity theory (Chung, Leighton, and Rosenberg, 1987). Beyond computer science applications, book embeddings can be used to model and solve traffic flow problems (Kainen, 1990) and to study RNA folding (Gliess and Stadler, 1999). Due to its contributions to both theory and application, book embedding has been the subject of extensive study. Dujmović and Wood (2004) give a summary of many of the known applications and results in book embeddings.

The book embedding problem is also known to be NP-complete (Garey, et al., 1980). Informally, this means that an exhaustive search through the space of possible embeddings for a minimal embedding is intractable. As an NP-complete problem, the task of determining an optimal book embedding for an arbitrary graph is difficult. This is where methods such as the GA may be of great assistance. The contribution of the GA to the book embedding problem is two-fold: 1) generating novel embeddings and 2) generating counter-examples to conjectured bounds. In this paper, we provide an overview of book embedding, an overview of the GA, and present very encouraging results for graphs of various configurations. We also describe a novel technique that we call the "Dual-Layered Approach" (DUA) which we are currently investigating.

The Book Embedding Problem

An n-book is a topological structure consisting of n half-planes (the pages) joined together at a common line (the spine). A graph is embedded in a book by placing the vertices along the spine and each edge on a single page of the book so that no two edges cross on any page. The book-thickness of a graph G, denoted $bt(G)$, is the smallest number n for which G has an n-book embedding.

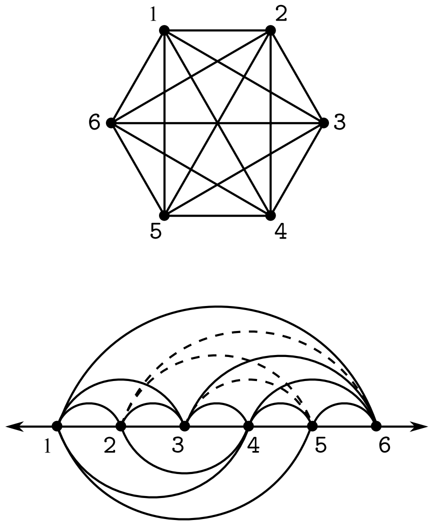


Figure 1: ***A three-page book embedding of K_6***

A book embedding of the complete graph on six vertices K_6 in a three-page book is given in Figure 1. The vertices of the graph lie on the spine. The first page of the book consists of the solid edges above the spine, the second page of the book is comprised of the solid edges below the spine, and the dashed edges above the spine form the third page of the book. These pages may be represented as lists of edges as follows:

Page 1: {(1,2), (1,3), (1,6), (2,3), (3,4), (3,6), (4,5), (4,6), (5,6)}

Page 2: {(1,4), (1,5), (2,4)}

Page 3: {(2,5), (2,6), (3,5)}

When embedding a graph in a book, there are two important considerations. First, the ordering of the vertices along the spine must be determined. For a graph with m vertices, there will be $m!$ possible arrangements of the vertices along the spine. Even if we account for the m cyclic rotations of this linear ordering and the reflections of each of these, there are still $(m-1)!/2$ vertex orderings to consider. Once the vertex order is determined, then the edges must be embedded on the pages of the book. As the numbers of vertices and edges increase, finding the book-thickness of a graph becomes computationally intractable. Garey, Johnson, Miller, and Papadimitriou (1980) proved that the problem of determining the book-thickness of an arbitrary graph is NP-complete, even with a pre-specified vertex ordering.

Despite the difficulty of the general book embedding problem, there are known bounds for the book-thickness of a graph with m vertices and q edges. We include Bernhart and Kainen's (1979) proof here since we use methods from this proof to form our custom cost function for our book embedding GA.

Theorem 1 $If\ G\ is\ a\ finite\ simple\ graph\ with$ $m \geq 4\ vertices\ and\ q\ edges, then$

$$bt(G) \geq \frac{q-m}{m-3}.$$

Proof: Place the m vertices on the spine of the book. The $m-1$ edges connecting consecutive vertices along the spine may be placed on any page of the book without creating edge crossings. The edge connecting the first and last vertex on the spine may also be placed on any page of the book, above all other edges, without causing crossings. Ignoring the m edges of this cycle, there may be at most $m-3$ additional edges on any page of the book, corresponding to a complete triangulation of the interior of this cycle. Thus an n-page book embedding of a graph with m vertices may have at most $m+n(m-3)$ edges; m for the outer cycle and $m-3$ for a complete triangulation of this cycle on each of the n pages.

Now we have:

$$q \leq m + n(m-3)$$

Solving for n yields the desired result:

$n \geq \frac{q-m}{m-3}$, thus completing the proof.

The complete graph on m vertices, K_m, is formed by connecting each pair of distinct vertices with an edge. The bound for book-thickness given in Theorem 1 may now be used to determine the optimal book-thickness of K_m in the following theorem (Bernhart and Kainen, 1979).

Theorem 2 *If* $m \geq 4,$ *then* $bt(K_m) = \left\lceil \frac{m}{2} \right\rceil$.

Proof: The graph K_m has $q = \binom{m}{2} = \frac{m(m-1)}{2}$ edges, corresponding to each distinct pairs of vertices. From Theorem 1, it follows that

$$bt(K_m) \geq \frac{\left[\frac{m(m-1)}{2}\right] - m}{m-3} = \frac{m}{2}$$

Since the book-thickness must be an integer, it follows that $bt(K_m) \geq \left\lceil \frac{m}{2} \right\rceil$.

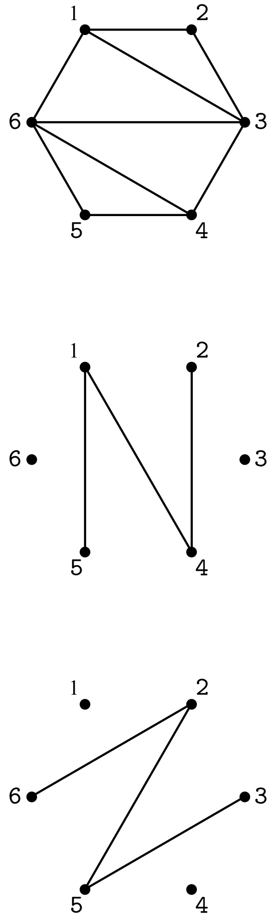


Figure 2: ***Rotated zig-zag triangulations of an m-cycle. Each rotation corresponds to one of the m/2 pages in a book embedding of*** K_m.

To show that $\left\lceil \frac{m}{2} \right\rceil$ pages are sufficient, we first observe that when m is even, then $\left\lceil \frac{m}{2} \right\rceil = \left\lceil \frac{m-1}{2} \right\rceil = \frac{m}{2}$. Hence, we may assume that m is even. Since the graph K_{m-1} is a sub-graph of K_m, we will show that K_m is embeddable in a book with $\frac{m}{2}$ pages. The corresponding embedding of K_{m-1} will follow after removing one vertex and its adjoining edges from the embedding of K_m.

The desired embedding of K_m is attained by rotating the interior edges of a zig-zag triangulation of the outer m-cycle through $\frac{m}{2}$ successive positions, as shown in Figure 2. The edges of each rotation are embedded on a separate page and the m edges of the outer cycle are placed on the first page. It is easily seen that each of the $m + \frac{m}{2}(m-3) = \binom{m}{2}$ edges of of K_m are embedded exactly once, showing that $bt(K_m) \leq \left\lceil \frac{m}{2} \right\rceil$. This gives us our desired result.

For example, the K_6 graph shown in Figure 1 has 15 edges, 6 on the outer cycle and $\frac{6}{2}(6-3) = 9$ in the interior of the cycle. By Theorem 2, the optimal book-thickness of this graph is $bt(K_6) = \left\lceil \frac{6}{2} \right\rceil = 3$. Figure 2 depicts the three rotated triangulations of the 6-cycle that correspond to each of the three pages of the book embedding of K_6 given in Figure 1.

The optimal book-thickness is known for several classes of graphs (Dujmović and Wood, 2004). When a graph is planar, it can be shown that the book thickness is never more than four pages (Yannakakis, 1986). Further, if the graph is planar and does not contain triangles, the book thickness is at most two pages (Kainen and Overbay, 2007). Although the optimal book-thickness is known for the complete graph, there are other similar graphs for which the optimal number is not known. One such graph is the complete bipartite graph, $K_{m,n}$. This graph consists of a set of m vertices and a set of n vertices, with all possible connections from the m-set to the n-set and no connections within each set. The book-thickness of $K_{m,n}$ has been shown to be at most the smaller of n and $\left\lceil \frac{2n+m}{4} \right\rceil$ (Muder, Weaver, and West, 1988). They originally conjectured that this bound was optimal, but it has been improved to $\left\lfloor \frac{2n}{3} \right\rfloor + 1$ when $m = n$ and in the case when $m = \left\lfloor \frac{n^2}{4} \right\rfloor$, embeddings in books with $n - 1$ pages have been found (Enomoto, Nakamigawa, and Ota, 1997).

The Genetic Algorithm

Having provided an overview of the book embedding problem, we turn our attention to the Genetic Algorithm (GA). The GA is loosely based on the concept of Darwinian natural selection. Individual members of a species who are better adapted to a given environment reproduce more successfully and so pass their adaptations on to their offspring. Over time, individuals possessing the adaptation form interbreeding populations, that is, a new species. In keeping with the biological metaphor, a candidate solution in a GA is known as a *chromosome*. The chromosome is composed of multiple genes. A collection of chromosomes is called a *population*. The GA randomly generates an initial population of chromosomes that are then ranked according to a fitness (or cost) function (Haupt and Haupt, 1998). One of the truly marvelous things about the GA is its wide applicability. We have used it to optimize structural engineering components—an NP-Complete problem—and are currently applying it to model language change (Ganzerli, et al., 2003, 2005, 2008; Overbay, et al., 2006). For practical purposes, this means, of course, that those who attempt to solve these problems must be content with good-enough solutions. Though good-enough may not appeal to purists, it is exactly the kind of solution implicit in natural selection: a local adaptation to local constraints, where the structures undergoing change are themselves the product of a recursive sequence of adaptations. This can be expressed quite compactly:

```
GA()
{
    Initialize() //generate population
    ComputeCost() //of each member
    SortByCost() //entire population
    while (good enough solution has not appeared)
     {
        Winnow() //who reproduces?
        Pair() //pair up reproducers
        Mate() //pass properties to children
        Mutate()  //randomly perturb genes
        SortByCost() //entire population
        TestConvergence()  //good enough solution?
     }
}
```

As noted, the optimal book embedding for the complete bipartite graph $K_{m,n}$ is not known. The optimal book-thickness is known for small values of m and n, but even in cases as small as $K_{4,4}$ an unusual ordering of the vertices is needed to attain an optimal 3-page embedding. Using a dual-layered approach to our genetic algorithm, described later in the paper, we hope to improve upon the best known bounds.

The GA and Book Embedding

The most extensive application of the GA to the book embedding problem prior to our own work is found in Kapoor et al. (2002) and Kapoor (1999). Kapoor at al. (2002) algorithmically generate an edge ordering and use the GA solely for the vertex ordering. Their algorithm produced embeddings at the known optimal bound for certain families of graphs. They provide no examples on how their approach scales to other types of graphs. Further, Kapoor (1999) appears only to have found known optimums for relatively small graphs, such as the complete graph up to K_{10}. Kapoor's results may be limited since the edge ordering is fixed. It is also known that embedding with pre-set vertex ordering does not always achieve optimal results. Our approach seeks to vary both dimensions of the problem.

The Dual-Layered Approach

We use a novel application of the GA to the book embedding problem that we call the "Dual-Layered Approach" (DUA). DUA provides an outer GA, which is used to seek an "optimal" vertex ordering for the spine of the book, along with an inner GA which seeks the "best" edge ordering for any given vertex sequence. Each population in our experiments consists of 64 chromosomes. The outer GA generates an initial population of vertex orderings, referred to as outer chromosomes. In order to determine the fitness of these chromosomes, the inner GA is run using each individual member of the population as a vertex ordering. So, for each member of a population of 64 outer chromosomes, the inner GA is run 64 times. The fitness of each outer chromosome is equal to the fitness of the best solution found in the inner GA using that vertex sequence. This process is repeated in each generation of the outer GA.

The ultimate goal is to find a solution within the inner GA that is lower than theorized bounds for graphs such as complete bipartite graphs, $K_{m,n}$. DUA will be particularly useful in seeking an improvement on the best known bounds for the book thickness of complete bipartite graphs since it is known that naïve approaches to vertex ordering for this family of graphs does not lead to optimal results. In particular, orderings with high regularity do not lead to the smallest book thickness. We hope that DUA will help discover atypical vertex orderings for complete bipartite graphs that will produce book embeddings that require fewer pages than the best known bounds.

The Cost Function

We have applied optimizations to several aspects of the inner GA in order to improve its effectiveness and efficiency. The cost function received special attention. The fitness of any given solution can be seen as its distance away from the best known bound. The more accurately the cost algorithm is able to capture this distance, the more quickly the GA will converge on a local solution. If the cost algorithm does not capture this distance well, then the GA will approach a random search. Initially we attempted to measure the cost using a relatively naïve approach, that is, the cost was simply equal to the book-thickness for a given edge ordering. However, consider two edge orderings with the same book-thickness: ordering one is more tightly packed toward the first page, while ordering two is more thickly populated toward the end. Ordering one is probably closer to an optimal solution than ordering two, but by considering only the book-thickness, the genetic algorithm would be unable to differentiate between the two solutions.

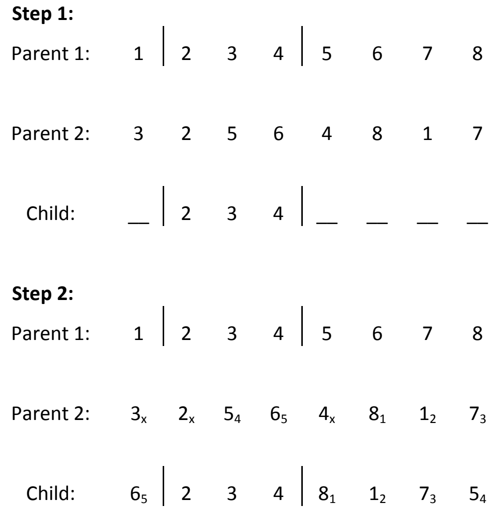
In order to solve this problem, we developed a cost function that values both small book-thickness as well as books more tightly packed towards the top. This cost function is customized for each type of graph. For example, when evaluating the fitness of a particular book embedding of the complete graph, we remove the m cycle edges from our edge list, since these may be placed on any page. By the proof of Theorem 1, at most $m-3$ additional edges may be placed on any page of the book. We assign a cost of 0 to any page that achieves this bound. Pages that have $m-3-k$ edges are assigned a cost of k. Since an optimal book embedding of K_m requires $w=\left\lceil \frac{m}{2} \right\rceil$ pages (see Theorem 2), any edges embedded on pages after page w are also included in the cost function. The cost function for K_m is given below:

```
t = total number of edges (not counting adjacent boundary
    edges)
e = max number of edges per page = m-3
p = max number of pages = ceiling(t/e)
n = number of edges on last page = e-(t mod e)

cases:
1. current page # < p
   cost = e - (number of edges on page)

2. current page number == p
   if current number of edges on page <= n
       cost = n-(current number of edges on page)
   if current number of edges on page is > n
       cost = (current number of edges on page) – n

3. current page number > p
       cost = current number of edges on page
```

Bipartite graphs, such as the hypercube and $K_{m,n}$ do not contain triangles, so the maximum number of edges per page of the book will be less than $m-3$. For such graphs, the custom cost function is adjusted accordingly.

Mating Algorithms

We also explored several types of mating algorithms, finally settling on the Order Crossover approach, which is well suited to the book embedding problem due to its ability to maintain the relative order of genes (Davis, 1985). In Order Crossover, the construction of a child chromosome from its parent involves two main steps. First, a subsequence of genes with random start and end indices is selected from parent 1. Each gene in the range [start, end], is copied into the child at the same index as it was found in the parent (Figure 3-Step 1). Next, starting at end + 1, genes are selected from parent 2 and placed into the child at the first available index following the end index. If a selected gene is already contained in the child, then it is skipped. The algorithm continues to choose genes from parent 2 until the child chromosome has been filled, wrapping around to the beginning of the parent and child chromosomes as needed (Figure 3-Step 2).



Step 1:

Parent 1:	1 |	2	3	4 |	5	6	7	8
Parent 2:	3	2	5	6	4	8	1	7
Child:	__ |	2	3	4 |	__	__	__	__

Step 2:

Parent 1:	1 |	2	3	4 |	5	6	7	8
Parent 2:	3_x	2_x	5_4	6_5	4_x	8_1	1_2	7_3
Child:	6_5 |	2	3	4 |	8_1	1_2	7_3	5_4

Figure 3: ***The two main steps of Order Crossover. In Step 2, numbered subscripts indicate the order of insertion into the child, while the subscript "x" indicates a gene which was skipped.***

Normalization

The use of Order Crossover allowed us to optimize our cost function using a technique that we call “normalization.” Because the edge ordering is independent of the page numbers of the edges, the effectiveness of the mating algorithm was diluted. Normalization is the process of grouping the edges by their page numbers. In other words, all edges that were embedded on page one occur first, followed by all of the edges from page two, etc. When the edges are grouped in this manner, any sub-sequence of edges that gets swapped by the parents in the mating algorithm contains edges that are closely related by page. Therefore, entire sections of the parent embedding can be preserved in the children. Normalization has enabled us to find optimal book embeddings for several kinds of graphs.

Results

We have explored several kinds of graphs thus far:

- Complete graphs up to K_{19}
- Complete bipartite graphs up to $K_{7,7}$
- Hypercubes up to Q_6
- Square grids up to 10×10
- X-trees up to height 8

Table 1 This shows the best results produced by our GA as they compare to the optimal bound for the book thickness of complete graphs.

Graph	Our Results	Optimal Bound
K_7	4	4
K_8	4	4
K_9	5	5
K_{10}	5	5
K_{11}	6	6
K_{12}	7	6
K_{13}	7	7
K_{19}	10	10
K_n		$\lceil n/2 \rceil$

Table 2 This shows the best results produced by our GA as they compare to the best known lower bound for the book thickness of complete bipartite graphs.

Graph	Our Results	Best Known Bound
$K_{5,5}$	4	4
$K_{6,6}$	5	5
$K_{7,7}$	5	5
$K_{n,n}$		$\lfloor 2n/3 \rfloor + 1$

Table 3 This shows the best results produced by our GA as they compare to the best known lower bound for the book thickness of hypercube graphs.

Graph	Our Results	Best Known Bound
Q_4	4	4
Q_5	4	4
Q_6	5	5
Q_7	7	6
Q_n		$n-1$

Table 4 This shows the best results produced by our GA as they compare to the optimal bound for the book thickness of square grids.

Graph Size	Our Results	Optimal Bound
2×2	2	2
3×3	2	2
4×4	2	2
5×5	2	2
6×6	2	2
7×7	2	2
8×8	2	2
9×9	2	2
10×10	2	2
$n \times n$		2

Table 5 This shows the best results produced by our GA as they compare to the optimal lower bound for the book thickness of X-trees.

Graph Height	Our Results	Optimal Bound
2	2	2
3	2	2
4	2	2
5	2	2
6	2	2
7	2	2
8	2	2
n		2

In every case, with the exception of Q_7, our results have been equivalent to known or conjectured bounds. We also have attained optimal bounds for much larger graphs than in previously published results. Our GA has attained optimal book embeddings of the complete graph up to K_{19}, which has 19 vertices and 171 edges. Clearly, the possible orderings of 171 edges would make an exhaustive search of the solution space intractable.

It should be noted that for square grids and X-trees, convergence to an optimal two-page embedding occurred every time and the convergence time did not appear to increase as the size of the graph increased. For these graphs, the degrees of the vertices and the structure of the graph remain similar as the size increases. We would expect duplicate results for much larger graphs of these types. Whereas, for complete graphs, hypercubes, and complete bipartite graphs, the vertex degrees increase as the number of vertices grows. For this reason, these three types of graphs are of interest in our continued research. We are particularly interested in improving on the theoretical bounds for hypercubes and complete bipartite graphs, since the best bounds for these graphs are still unknown.

Conclusion and Future Research

Book embedding is easy to describe yet computationally intractable. It is exactly the kind of problem for which the genetic algorithm shines. Whether one is constructing a near-optimal truss, a near-optimal book embedding, or, indeed, an organism adapted to a set of local conditions, the genetic algorithm has proven to be a useful guide. We have shown that the GA can produce book embeddings that are as good as known optimal bounds on large graphs. Though we have yet to find a counter-example to conjectured bounds for other types of graphs, our dual-layered approach, a genetic algorithm within a genetic algorithm, represents a novel solution to the problem. We are currently working in two directions. We are attempting to generate book embeddings for complete bipartite graphs and hypercubes that improve upon known bounds for these graphs. We also observe that computing the same cost function for each of 64 chromosomes is embarrassingly parallel. Our major effort over the next year will be to parallelize DUA for execution on a cluster. Although the ability to search intractably large spaces will not necessarily generate a true optimal embedding, it should allow us to speak with confidence about currently conjectured bounds.

Acknowledgments

The authors wish to acknowledge the McDonald Family and the Gonzaga University Research Council which have funded this research for several years. Their confidence in the role of undergraduates in faculty research made the participation of our three undergraduate research assistants (Arnold, Hurson, and Pierce) possible.

References

Bernhart, F., and Kainen, P. C. 1979. The Book Thickness of a Graph. *Journal of Combinatorial Theory*, Series B 27 (3): 320-331.

Chung, F. R. K., Leighton, F. T., and Rosenberg, A. L. 1987. Embedding Graphs in Books: A Layout Problem with Applications to VLSI Design. *SIAM Journal on Algebraic and Discrete Methods* 8 (1): 33-58.

Davis, L. 1985. Applying Algorithms to Epistatic Domains. *Proceedings of the International Joint Conference on Artificial Intelligence*, 162-164.

Dujmović, V. and Wood, D. R. 2004. On Linear Layouts of Graphs. *Discrete Mathematics and Theoretical Computer Science* 6: 339-358.

Enomoto, H., Nakamigawa, T., and Ota, K. 1997. On the Pagenumber of Complete Bipartite Graphs. *Journal of Combinatorial Theory*, Series B 71 (1): 111-120.

Ganzerli, S., De Palma, Paul. 2008. Genetic Algorithms and Structural Design Using Convex Models of Uncertainty. In Y. Tsompanakis, N. Lagaros, M. Papadrakakis (eds.), *Structural Design Optimization Considering Uncertainties*. London: A.A. Balkema Publishers, A Member of the Taylor and Francis Group.

Ganzerli, S., De Palma, P., Stackle, P., & Brown, A. 2005. Info-Gap Uncertainty on Structural Optimization via Genetic Algorithms. *Proceedings of the Ninth International Conference on Structural Safety and Reliability*, Rome.

Ganzerli, S., De Palma, P., Smith, J., & Burkhart, M. 2003. Efficiency of genetic algorithms for optimal structural design considering convex modes of uncertainty. *Proceedings of The Ninth International Conference on Applications of Statistics and Probability in Civil Engineering,* San Francisco.

Garey, M. R. and Johnson, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* NY: W. H. Freeman and Co.

Garey, M. R., Johnson, D. S., Miller, G. L., Papadimitriou, C. H. 1980. The Complexity of Coloring Circular Arcs and Chords. *SIAM Journal on Algebraic and Discrete Methods* 1 (2): 216-227.

Gleiss, P. M. and Stadler, P. F., 1999. Relevant Cycles in Biopolymers and Random Graphs, Technical Report, 99-07-042, Santa Fe Institute, Santa Fe, NM.

Goldberg, D. E. 1989. *Genetic Algorithms in Search Optimization and machine Learning*. NY: Addison-Wesley.

Haupt, L., Haupt, S. 1998. *Practical Genetic Algorithms*. New York: John Wiley and Sons.

Holland, J. H. 1975. *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press.

Kainen, P. C., 1990. The Book Thickness of a Graph, II. *Congressus Numerantium* 71: 127-132.

Kainen, P. C., and Overbay, S. R., 2007. Extension of a Theorem of Whitney. *Applied Mathematics Letters* 20 (7): 835-837.

Luger, G., Stubblefield, W. 2009. Genetic and Evolutionary Computing. In: *AI Algorithms, Data Structures, and Idioms in Prolog, Lisp, and Java*. Addison-Wesley Computing.

Kapoor, N. (1999). *Pagenumber Problem.* Master's thesis, SITE, University of Ottawa.

Kapoor, N., Russell, M., Stojmenovic, I. 2002. A Genetic Algoritym for Finding the Pagenumber of Interconnection Networks*.* *Journal of Parallel and Distributed Computing* 62: 267-283.

Muder, D. J., Weaver, M. L., and West, D. B. 1988. Pagenumber of Complete Bipartite Graphs. *Journal of Graph Theory* 12 (4): 469-489.

Overbay, S., Ganzerli, S., De Palma, P, Brown, A., & Stackle, P. 2006. Trusses, NP-Completeness, and Genetic Algorithms. *Proceedings of the 17th Analysis and Computation Specialty Conference.* St. Louis.

Yannakakis, M. 1986. Four Pages are Necessary and Sufficient for Planar Graphs. *Proceedings of the 18th Annual ACM Symposium on Theory of Computing*, 104-108.

Bi-fuzziness, Incompleteness, Inconsistency, Truth and Falsity Based on Saturation and Ignorance Functions. A New Approach of Penta-Valued Knowledge Representation

Vasile Patrascu

Department of Informatics Technology, Tarom
Calea Bucurestilor, 224F, Bucharest, Romania
E-mail: patrascu.v@gmail.com

Abstract

This paper presents a new five-valued knowledge representation of bipolar information. This representation is related to a five-valued logic that uses two logical values of truth (true, false) and three logical values of uncertainty (incomplete, inconsistent and fuzzy). The new approach is based on the concept of saturation function and ignorance function. In the framework of five-valued representation new formulae for union and intersection are constructed. Also, the paper presents a short application related to fuzzy preference modeling and decision making.

Introduction

Let X be a set of objects. We consider a property A, an object $x \in X$ and the following sentence $P_A(x)$: x has the property A. We want to know if the sentence $P_A(x)$ is true or false. After an evaluation, the information about logical value of sentence $P_A(x)$ is described by a scalar $T_A(x) \in [0,1]$. For the considered sentence, $T_A(x)$ represents its truth degree. In the same time, the function $T_A : X \to [0,1]$ defines a Zadeh fuzzy set associated to the property A (Zadeh 1965). Then, we compute the degree of falsity:

$$F_A(x) = 1 - T_A(x) \tag{1}$$

Using the scalar $T_A(x)$, we have obtained the following representation of information about sentence $P_A(x)$.

$$W_A(x) = (T_A(x), F_A(x)) \tag{2}$$

This information is normalized because the components of vector $W_A(x)$ verify the condition of partition of unity:

$$T_A(x) + F_A(x) = 1 \tag{3}$$

The representation (3) is related to a bi-valued logic based on true and false. The next step was done by Atanassov (Atanassov 1986). He considered that after evaluation, the information about logical value of sentence $P_A(x)$ is described by a vector with two components

$$V_A(x) = \left(T_A(x), F_A(x)\right) \tag{4}$$

and supplementary these two components verify the inequality:

$$T_A(x) + F_A(x) \le 1 \tag{5}$$

The information represented by vector $V_A(x)$ is not normalized but, Atanassov has introduced the intuitionistic index:

$$U_A(x) = 1 - T_A(x) - F_A(x) \tag{6}$$

Using the vector $V_A(x)$, we have obtained an intuitionistic representation of information about sentence $P_A(x)$.

$$W_A(x) = (T_A(x), U_A(x), F_A(x)) \tag{7}$$

This information is normalized because the components of vector $W_A(x)$ verify the condition of partition of unity:

$$T_A(x) + U_A(x) + F_A(x) = 1 \tag{8}$$

The representation (8) is related to a three-valued logic based on true, neutral and false.

In this paper we will consider the bipolar representation (Benferhat et al. 2006; Cornelis et al. 2003; Dubois et al. 2004) without having the condition (5). In this case, we cannot obtain immediately a normalized variant like (8). In the following, we present a method for obtaining a normalized representation of bipolar information.

The paper has the following structure: section two presents the concepts of saturation, ignorance and bi-fuzziness. Section three presents the construction method of five-valued representation. Section four presents a five-valued logic based on true, false, incomplete, inconsistent and fuzzy. Section five presents some operators for the five-valued structure. Section six presents the using of five-valued knowledge representation for fuzzy modeling of pairwise comparisons. Finally we present some conclusions.

Saturation, Ignorance and Bi-fuzziness Functions

In this section, firstly, we introduce the concepts of saturation function and ignorance function. These two functions are complementary. Both functions are essentially characterized by symmetry, boundary and monotonicity properties. Secondly, we introduce the concept of bi-fuzziness related to the index of indeterminacy (Patrascu 2008).

Definition 1: A saturation function is a mapping $S:[0,1]^2 \to [0,1]$ such that:

i) $S(x,y)=S(y,x)$

ii) $S(x,y)=0$ if and only if $(x,y)=(0,0)$

iii) $S(x,y)=1$ if and only if $(x,y)=(1,1)$

iv) $S(x,y)$ increases with respect to x and y

The property *a*) describes the commutativity and the property *d*) describes the monotonicity. From property *b)* it results that the saturation value is low if and only if both arguments have low value and from property *c)* it results that the saturation value is high if and only if both arguments have high value.

Example 1:

$$S(x,y)=\frac{x+y}{2}.$$

Example 2:

$$S(x,y)=\frac{\max(x,y)}{1+|x-y|}.$$

Example 3: For any *t-conorm* $\oplus$

$$S(x,y)=\frac{x\oplus y}{1+(1-x)\oplus(1-y)}.$$

Example 4: For any *t-conorm* $\oplus$

$$S(x,y)=\frac{x\oplus y}{x\oplus y+(1-x)\oplus(1-y)}.$$

Example 4:

$$S(x,y)=\frac{x+y}{2}+\frac{1-x-y}{2}|x-y|.$$

Notice that these particular saturation functions are not associative.

Definition 2: A ignorance function is a mapping $U:[0,1]^2 \to [0,1]$ such that:

i) $U(x,y)=U(y,x)$

ii) $U(x,y)=0$ if and only if $(x,y)=(1,1)$

iii) $U(x,y)=1$ if and only if $(x,y)=(0,0)$

iv) $U(x,y)$ decreases with respect to x and y

Example 1:

$$U(x,y)=1-\frac{x+y}{2}.$$

Example 2:

$$U(x,y)=\frac{1-\min(x,y)}{1+|x-y|}.$$

The following proposition shows the relation between saturation functions and ignorance functions and some supplementary properties.

Proposition 1: Let S be a saturation function. Then

$$U(x,y)=S(1-x,1-y) \qquad (9)$$

is an ignorance function.

Proof: It is evident because the properties of both functions are complementary.

Proposition 2: Let U be an ignorance function. Then

$$S(x,y)=1-U(x,y) \qquad (10)$$

is a saturation function.

Proof: It is evident because the properties of both functions are complementary.

Proposition 3: Let S be a saturation function let $\lambda\in(0,1)$. Then

$$P(x,y)=\frac{\lambda\cdot S(x,y)}{\lambda\cdot S(x,y)+(1-\lambda)\cdot(1-S(x,y))} \qquad (11)$$

is a saturation function.

Proof: It is evident because in the new saturation function construction it was used the scalar addition based on the uninorm function.

Proposition 4: Let S be a saturation function let $\alpha\in(0,\infty)$. Then

$$Q(x,y)=\frac{S^{\alpha}(x,y)}{S^{\alpha}(x,y)+(1-S(x,y))^{\alpha}} \qquad (12)$$

is a saturation function.

Proof: It is evident because in the new saturation function construction it was used the scalar multiplication based on the uninorm function.

Proposition 5: Let S be a saturation function. Then

$$R(x,y)=\frac{S(x,y)}{S(x,y)+S(1-x,1-y)} \tag{13}$$

is a saturation function.
Proof: It is results immediately that the new saturation function verifies the properties i), ii), iii) and iv).

Definition 3: A bi-fuzziness function is a mapping $I:[0,1]^2 \rightarrow [0,1]$ such that:

- i) $I(x,y)=I(y,x)$
- ii) $I(x,y)=I(1-x,y)$
- iii) $I(x,y)=I(x,1-y)$
- iv) $I(x,y)=0$ if and only if $x,y\in\{0,1\}$
- v) $I(x,y)=1$ if and only if $(x,y)=(0.5,0.5)$
- vi) $I(x,y)$ increases with x if $x\le 0{,}5$ and $I(x,y)$ decreases with x if $x\ge 0{,}5$
- vii) $I(x,y)$ increases with y if $y\le 0{,}5$ and $I(x,y)$ decreases with y if $y\ge 0{,}5$

The bi-fuzziness function represents a measure of similarity between the point $(x,y)\in[0,1]^2$ and the center of unit square, the point $(0.5,0.5)$. The index of bi-fuzziness verifies, for each argument x and y, the properties considered by De Luca and Termini for fuzzy entropy definition (De Luca and Termini 1972).
If we replace y with the negation of x, namely $y=\bar{x}=1-x$, one obtains a fuzzy entropy function.

Proposition 6: Let S be a saturation function. Then

$$I(x,y)=(1-|S(x,\bar{y})-S(\bar{x},y)|)\cdot(1-|S(x,y)-S(\bar{x},\bar{y})|)$$

is a bi-fuzziness function.

Proposition 7: Let S be a saturation function. Then

$$I(x,y)=1-S\left(|2x-1|,|2y-1|\right)$$

is a bi-fuzziness function.

Example 1:
$$I(x,y)=1-|x-0.5|-|y-0.5|.$$

Example 2:
$$I(x,y)=\frac{\left(1-|x-y|\right)\cdot\left(1-|x+y-1\right)|}{1-|x-y|\cdot|x+y-1|}.$$

Example 3:
$$I(x,y)=(1-|x-y|)(1-|x+y-1|).$$

Five-Valued Representation of Bipolar Information

Let S be a saturation function. For any pair (T,F), we define *the net truth* τ and *the definedness* δ by:

$$\tau(T,F)=S(T,\bar{F})-S(\bar{T},F)$$
$$\delta(T,F)=S(T,F)-S(\bar{T},\bar{F})$$

The uncertainty or the entropy (Kaufmann 1975; Patrascu 2010) is defined by:

$$h=1-|\tau| \tag{14}$$

and the certainty will be its negation:

$$g=|\tau|$$

The two functions define a partition with two fuzzy sets X_G and X_H: one related to the certainty and the other to the uncertainty.

The non-fuzziness id defined by:

$$z=|\delta| \tag{15}$$

The index of bi-fuzziness will be computed by difference between uncertainty and non-fuzziness:

$$i=1-|\tau|-|\delta| \tag{16}$$

The non-fuzziness and bi-fuzziness define two subsets of X_H, namely: X_Z and X_I. We compute the incompleteness (undefinedness) and inconsistency (contradiction) using the non-fuzziness:

$$u=\delta_- \tag{17}$$

$$c=\delta_+ \tag{18}$$

where $x_-=\max(0,-x)$ and $x_+=\max(0,x)$.

The incompleteness and inconsistency define two subsets of X_Z, namely: X_U and X_C. Notice that because $c\cdot u=0$ it results:

$$X_U\cap X_C=\Phi \tag{19}$$

Next we compute the index of truth and falsity using the net truth function τ:

$$t=\tau_+ \tag{20}$$

$$f=\tau_- \tag{21}$$

The index of truth and index of falsity define two subsets of X_G, namely: X_T and X_F. Notice that because $t\cdot f=0$ it results:

$$X_T \cap X_F = \Phi \qquad (22)$$

The index of truth (20), the index of falsity (21), the index of bi-fuzziness (16), the index of incompleteness (17) and the index of inconsistency (18) define a partition of unity:

$$t + f + i + u + c = 1$$

In the construction method presented above, it was used the schema shown in figure 1.

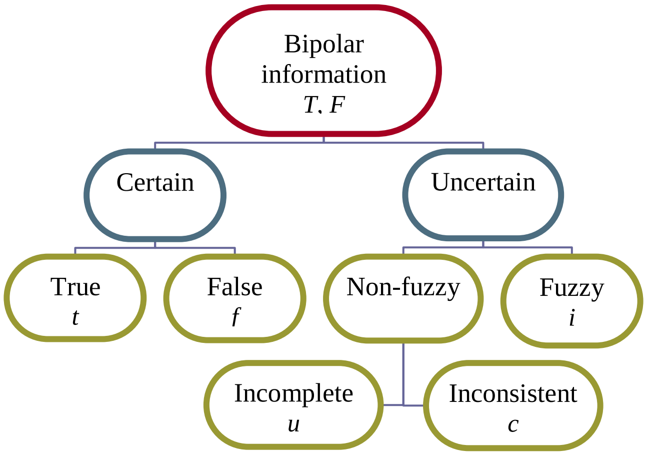


Figure 1. The construction schema for five-valued representation of bipolar information.

Five Valued Logic Based on Truth, Falsity, Inconsistency, Incompleteness and Bi-fuzziness

This five-valued logic is a new one, but is related to our previous work presented in (Patrascu 2008). In the framework of this logic we will consider the following five values: *true* t, *false* f, *incomplete* (*undefined*) u, *inconsistent* (*contradictory*) c, and *fuzzy (indeterminate)* i. We have obtained these five logical values, adding to the so called Belnap values (Belnap 1977) the fifth: *fuzzy* (*indeterminate*). Tables 1, 2, 3, 4, 5, 6 and 7 show the basic operators in this logic.

Table 1. The union.

$\cup$	t	c	i	u	f
t	t	t	t	t	t
c	t	c	i	i	c
i	t	i	i	i	i
u	t	i	i	u	u
f	t	c	i	u	f

Table 2. The intersection.

$\cap$	t	c	i	u	f
t	t	c	i	u	f
c	c	c	i	i	f
i	i	i	i	i	f
u	u	i	i	u	f
f	f	f	f	f	f

The main differences between the proposed logic and the Belnap logic are related to the logical values u and c. We have defined $c \cap u = i$ and $c \cup u = i$ while in the Belnap logic there were defined $c \cap u = f$ and $c \cup u = t$.

Table 3. The complement.

	$\neg$
t	f
c	c
i	i
u	u
f	t

Table 4. The negation.

	$-$
t	f
c	u
i	i
u	c
f	t

Table 5. The dual.

	$\approx$
t	t
c	u
i	i
u	c
f	f

The complement, the negation and the dual are interrelated and there exists the following equalities:

$$\approx x = -\neg x \qquad (23)$$

$$\neg x = - \approx x \qquad (24)$$

$$-x = \neg \approx x \qquad (25)$$

Table 6. The S-implication

$\rightarrow$	t	c	i	u	f
t	t	c	i	u	f
c	t	c	i	i	c
i	t	i	i	i	i
u	t	i	i	u	u
f	t	t	t	t	t

The *S-implication* is calculated by:

$$x \rightarrow y = \neg x \bigcup y \qquad (26)$$

Table 7. The equivalence

$\leftrightarrow$	t	c	i	u	f
t	t	c	i	u	f
c	c	c	i	i	c
i	i	i	i	i	i
u	u	i	i	u	u
f	f	c	i	u	t

The *equivalence* is calculated by:

$$x \leftrightarrow y = (\neg x \bigcup y) \bigcap (x \bigcup \neg y) \qquad (27)$$

New Operators Defined on Five-Valued Structure

There be $x = (t, c, i, u, f) \in [0,1]^5$, For this kind of vectors, one defines the union, the intersection, the complement, the negation and the dual operators. The operators are related to those define in (Patrascu 2007a; Patrascu 2007b).

The Union: For two vectors $a, b \in [0,1]^5$ where $a = (t_a, c_a, i_a, u_a, f_a)$, $b = (t_b, c_b, i_b, u_b, f_b)$, one defines the union (disjunction) $d = a \cup b$ by the formula:

$$\begin{aligned} t_d &= t_a \vee t_b \\ c_d &= (c_a + f_a) \wedge (c_b + f_b) - f_a \wedge f_b \\ u_d &= (u_a + f_a) \wedge (u_b + f_b) - f_a \wedge f_b \\ f_d &= f_a \wedge f_b \\ i_d &= 1 - (t_d + c_d + u_d + f_d) \end{aligned} \qquad (28)$$

The Intersection: For two vectors $a, b \in [0,1]^5$ one defines the intersection (conjunction) $c = a \cap b$ by the formula:

$$\begin{aligned} t_c &= t_a \wedge t_b \\ c_c &= (c_a + t_a) \wedge (c_b + t_b) - t_a \wedge t_b \\ u_c &= (u_a + t_a) \wedge (u_b + t_b) - t_a \wedge t_b \\ f_c &= f_a \vee f_b \\ i_c &= 1 - (t_c + c_c + u_c + f_c) \end{aligned} \qquad (29)$$

In formulae (28) and (29), the symbols "$\vee$" and "$\wedge$" represent the maximum and the minimum, namely: $\forall x, y \in [0,1]$,

$$x \vee y = \max(x, y)$$
$$x \wedge y = \min(x, y)$$

The union "$\cup$" and intersection "$\cap$" operators preserve de properties $t + c + u + f \leq 1$, $t \cdot f = 0$ and $u \cdot c = 0$, namely:

$$\begin{aligned} &t_{a \cup b} + c_{a \cup b} + u_{a \cup b} + f_{a \cup b} \leq 1 \\ &t_{a \cup b} \cdot f_{a \cup b} = 0 \\ &c_{a \cup b} \cdot u_{a \cup b} = 0 \end{aligned}$$

$$\begin{aligned} &t_{a \cap b} + c_{a \cap b} + u_{a \cap b} + f_{a \cap b} \leq 1 \\ &t_{a \cap b} \cdot f_{a \cap b} = 0 \\ &c_{a \cap b} \cdot u_{a \cap b} = 0 \end{aligned}$$

We remark that after union or intersection the certainty increases and uncertainty decreases.

The Complement: For $x = (t, c, i, u, f) \in [0,1]^5$ one defines the complement x^c by formula:

$$x^c = (f, c, i, u, t) \qquad (30)$$

The Negation: For $x = (t, c, i, u, f) \in [0,1]^5$ one defines the negation x^n by formula:

$$x^n = (f, u, i, c, t) \qquad (31)$$

The Dual: For $x = (t, c, i, u, f) \in [0,1]^5$ one defines the dual x^d by formula:

$$x^d = (t, u, i, c, f) \qquad (32)$$

In the set $\{0,1\}^5$ there are five vectors having the form $x = (t, c, i, u, f)$, which verify the condition $t + f + c + i + u = 1$: $T = (1,0,0,0,0)$ (*True*), $F = (0,0,0,0,1)$ (*False*), $C = (0,1,0,0,0)$ (*Inconsistent*), $U = (0,0,0,1,0)$ (*Incomplete*) and $I = (0,0,1,0,0)$ (*Fuzzy*).

Using the operators defined by (28), (29), (30), (31) and (32), the same truth table results as seen in Tables 1, 2, 3, 4, 5, 6 and 7.

Fuzzy Preference Relation in The Framework of Five-Valued Representation

A fuzzy preference relation A on a set of alternatives $X = \{x_1, x_2, ..., x\ \}$ is a fuzzy set on the product set $X \times X$, that is characterized by a membership function $\mu_P : X \times X \to [0,1]$ (see Chiclana et al. 1998; Fodor et al. 1994; Tanino 1988). When cardinality of X is small, the preference relation may be represented by the $n \times n$ matrix $A = \{a_{ij}\}$ being $a_{ij} = \mu_A(x_i, x_j)$ $\forall i, j \in \{1,2,...,n\}$. a_{ij} is interpreted as the preference degree of the alternative x_i over x_j. From a preference relation A, Fodor and Roubens (Fodor 1994) derive the following three relations:

Strict preference: $p_{ij} = P(x_i, x_j)$ indicating that x_i preferred to x_j but x_j is not preferred to x_i.

Indifference: $i_{ij} = I(x_i, x_j)$ indicating that x_i and x_j are considered equal in the sense that x_i is as good as x_j.

Incomparability: $j_{ij} = J(x_i, x_j)$ which occurs if neither a_{ij} nor a_{ji}.

Taking into account the five-valued representation of bipolar information, we define five relations that characterize the following five fundamental attitudes:

Strict preference $t_{ij} = T(x_i, x_j)$ is a measure of strict preference of x_i over x_j, indicating that x_i preferred to x_j but x_j is not preferred to x_i.

Indifference: $c_{ij} = C(x_i, x_j)$ is a measure of the simultaneous fulfillment of a_{ij} and a_{ji}.

Incomparability: $u_{ij} = U(x_i, x_j)$ is a measure of the incomparability of x_i and x_j, which occurs if neither a_{ij} nor a_{ji}.

Strict aversion: $f_{ij} = F(x_i, x_j)$ that is a measure of strict preference of x_j over x_i, indicating that x_i is not preferred to x_j.

Undecidability: $i_{ij} = I(x_i, x_j)$ is a measure of undecidability between x_i and x_j which occurs when $a_{ij} \approx 0.5$ and $a_{ji} \approx 0.5$.

Next, we consider a decision making problem where, an expert supply the preferences over a set of n alternatives: $X = \{x_1, x_2, ... x_n\}$. The preferences are represented by the following fuzzy relation:

$$A = \begin{vmatrix} 0 & a_{12} & ... & a_{1n} \\ a_{21} & 0 & ... & a_{2n} \\ ... & ... & ... & ... \\ a_{n1} & ... & ... & 0 \end{vmatrix} \qquad (38)$$

where $a_{ij} \in [0,1]$.

The algorithm that we propose to obtain the best alternative is the next:

Step 0: Initialize the matrix A and define the saturation function S.

Step 1: Compute the function t_{ij}, c_{ij}, u_{ij}, f_{ij} and i_{ij} using formulae (20), (21), (16), (17) and (18).

Step 2: Compute the relative score function by:

$$r_{ij} = \frac{t_{ij} + c_{ij} + 0.5 \cdot i_{ij}}{t_{ij} + 2 \cdot c_{ij} + 1.5 \cdot i_{ij} + u_{ij} + 3 \cdot f_{ij}} \qquad (39)$$

Step 3: Compute the total score function by:

$$R_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} r_{ij} \qquad (40)$$

Step 4: Choose

$$x_{optim} = \underset{k \in \{1,2,\ldots,n\}}{\arg\max} \{R_k\} \qquad (41)$$

In the presented algorithm, the next five items hold:

If $a_{ij} = 1$ and $a_{ji} = 0$, then $r_{ij} = 1$.

If $a_{ij} = 1$ and $a_{ji} = 1$, then $r_{ij} = 0.5$.

If $a_{ij} = 0.5$ and $a_{ji} = 0.5$, then $r_{ij} = 0.33$.

If $a_{ij} = 0$, and $a_{ji} = 0$, then $r_{ij} = 0$.

If $a_{ij} = 0$, and $a_{ji} = 1$, then $r_{ij} = 0$.

Numerical example: Let $X = \{x_1, x_2, x_3, x_4, x_5\}$ be a set of alternatives. Consider the fuzzy preference relation:

$$X = \begin{vmatrix} 0 & 0.06 & 0.42 & 0.84 & 0.50 \\ 0.20 & 0 & 0.85 & 0.02 & 0.70 \\ 0.60 & 0.45 & 0 & 0.68 & 0.43 \\ 0.27 & 0.93 & 0.20 & 0 & 0.30 \\ 0.20 & 0.47 & 0.67 & 0.83 & 0 \end{vmatrix} \qquad (42)$$

If the saturation function is defined by

$$S(x,y)=\frac{x+y}{2},$$

it results:

$$t_{ij}=\left(a_{ij}-a_{ji}\right)_+ \quad (43)$$

$$c_{ij}=\left(a_{ij}+a_{ji}-1\right)_+ \quad (44)$$

$$u_{ij}=\left(1-a_{ij}-a_{ji}\right)_+ \quad (45)$$

$$f_{ij}=\left(a_{ji}-a_{ij}\right)_+ \quad (46)$$

$$i_{ij}=1-|a_{ij}-a_{ji}|-|a_{ij}+a_{ji}-1| \quad (47)$$

Using the presented algorithm one obtains:

$R_1=1.36$, $R_2=1.26$, $R_3=1.41$, $R_4=1.24$, $R_5=1.46$

It results $x_{optim}=x_5$.

Conclusions

In this paper, we propose a different functional approach to model the bipolar information. The new approach is based on two new information concepts: saturation function and ignorance function. Saturation function can be seen as way of generalizing t-conorms dropping out associativity. We must underline that the associativity is not crucial for the construction of five-valued representation. More than that, in our framework, the saturation function has only two arguments: the degree of truth and degree of falsity. Finally, we are dealing with a class of functions different from that of the t-conorms.

The saturation function measures the excess of information, while, the ignorance function measures the lack of information that an estimator suffers when trying to determine if a given sentence is true or false.

The third concept, bi-fuzziness function can be understood as an extension from fuzzy sets to bipolar fuzzy sets of the concept of fuzziness defined by Zadeh. In addition, the index of bi-fuzziness can be understood as a measure of partial uncertainty of bipolar information. Both saturation function and ignorance function are related. Each of them can be recovered in a functional way from the other.

If suitable saturation or ignorance functions are known that fit well for a given problem, they can be used to build a five-valued knowledge representation. In this way, we are able to provide a theoretical framework which is different from the usual one to represent truth, falsity, incompleteness, inconsistency and bi-fuzziness. In this framework, a new five-valued logic was presented based on five logical values: true, false, incomplete, inconsistent and fuzzy. It was identified two components for certainty and three components for uncertainty. Based on this logic, new union and intersection operators were defined for the existing five-valued structure of information.

We also propose an application in preferences under a novel score function. The using of the proposed five fundamental attitudes provides a new perspective in decision making and it offers a simple way to produce a comprehensive judgment.

References

Atanassov, K. 1986. Intuitionistic Fuzzy sets. Fuzzy Sets and Systems 20, 87-96.

Belnap, N. 1977. A Useful Four-valued Logic, Modern Uses of Multiple-valued Logics (D. Reidel, Ed), Dordrecht-Boston, 8-37.

Benferhat, S.; Dubois, D.; Kaci, S. and Prade, H. 2006. Bipolar possibility theory in preference modeling: Representation, fusion and optimal solutions, Information Fusion, Vol 7: 135-150.

Chiclana, F.; Herrera, F. and Herrera-Viedma, E. 1998. Integrating Three Representation Models in Fuzzy Multipurpose Decision Making Based on Fuzzy Preference Relations, Fuzzy Sets and Systems 97, 33-48.

Cornelis, C.; Deschrijver, G. and Kerre, E. E. 2003. Square and Triangle: Reflections on Two Proeminent Mathematical Structures for Representation of Imprecision, Notes on Intuitionistic Fuzzy Sets 9(3): 11-21.

Dubois, D.; Kaci, S. and Prade, H. 2004. Bipolarity in reasoning and decision - An introduction. The case of the possibility theory framework. Proc. of the 10th Inter. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'04), Perugia, Italy.

Fodor, J. and Roubens, M. 1994. Fuzzy Preference Modelling and Multicriteria Decision Support (Kluwer, Dordrecht).

Kaufmann, A. 1975. Introduction to the Theory of Fuzzy Subsets, academic Press, New York.

De Luca, A.; Termini, S. 1972. A definition of nonprobabilistic entropy in the setting of fuzzy theory. Information and Control 20, 301–312.

Patrascu, V. 2007a. Penta-Valued Fuzzy Set, The IEEE International Conference on Fuzzy Systems, (FUZZY-IEEE 2007), London, U.K., 137-140.

Patrascu, V. 2007b. Rough Sets on Four-Valued Fuzzy Approximation Space, The IEEE International Conference on Fuzzy Systems, (FUZZY-IEEE 2007), London, U.K., 133-136.

Patrascu, V. 2008. A New Penta-valued Logic Based Knowledge Representation, Proc. of the 12th Inter. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'08), Malaga, Spain.

Patrascu, V. 2010. Cardinality and entropy for bifuzzy set, Proc. of the Inter. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'10), Dortmund, Germany.

Tanino, T. 1988. Fuzzy Preference Relations in Group Decision Making. In: J. Kacprzyk, M. Roubens (Eds.), Non-Conventional Preference Relations in Decision Making (Springer-Verlag, Berlin, 54-71.

Zadeh, L. A. 1965. Fuzzy sets, Information and Control, 8: 338-353.

Automated Code Generation Using Case-Based Reasoning, Routine Design and Template-Based Programming

Yuri Danilchenko and Richard Fox

Department of Computer Science
Northern Kentucky University
Nunn Drive
Highland Heights, KY 41099
{danilcheny1, foxr}@nku.edu

Abstract

Automated code generation is the process whereby a computer program takes user specifications in some form and produces a program as output. Automated code generation can be the process undertaken by a compiler, which generates an executable program from a source program, but it also applies to the situation where the input is a task described at some level of abstraction and the output is a program that can perform that task. Several different approaches have been utilized to varying degrees of success to automate code generation, including Case-Based Reasoning, formal methods and evolutionary algorithms. In this paper, a system is introduced which combines Case-Based Reasoning, Routine Design and Template-Based Programming to generate programs that handle straight-forward database operations. This paper presents the approach taken and offers some brief examples.

Automated code generation (ACG) is the process whereby a computer program takes user specifications in some form and produces a working program as output. When the user input is some abstract description of a task, as opposed to source code in some high level language, ACG presents both a challenging problem and an opportunity to reduce cost. Automating the programming task, which normally requires a great deal of expertise involves employing techniques that comprise design or planning, logic and programming knowledge. The benefits of ACG include reducing or eliminating expenses involved in software development and maintenance, which studies have indicated could cost corporations as much as 10% of their yearly expenses (Jones 2010).

One form of ACG is the compiler. The user provides source code as input and the compiler generates an executable program. Compilers have been in regular use since the late 1950s when the first high level languages were developed. Although initially many programmers scoffed at the idea that ACG could produce efficient and correct code, very few programmers today would write code in a low level language, favoring the high level languages and compiler technologies. However, the compiler requires too detailed an input as the programmer must still produce the algorithm in a proper syntactic form.

In Artificial Intelligence (AI), a variety of approaches have been explored to support software development. Case-Based Reasoning (CBR), for instance, can be used to maintain a library of code routines (e.g., objects, methods), select the code routines that best match user specifications, and present the those options to the software developer. Alternatively, through genetic and evolutionary programming, code can be mutated and tested for improvements. If improved, the new code becomes a base for the next generation of code. Random changes can potentially lead to code that is more concise, more efficient, or more correct.

At present, neither CBR nor evolutionary approaches has yielded an ACG system that can replace a software developer. In this paper, the research focuses on three different but related areas. First, code generation is thought to be a design problem. A solution will be a plan. Plan steps can be specified at a generic level and then refined into more detail. Eventually all plan steps will be filled in with code components from a component library. Once selected, these components are combined and used to fill in a program template.

Although the word "plan" is being used, the plan is a description of a solution, or a design to solve the stated problem. The plan steps provide goals to be fulfilled. Code components are selected to fulfill each of these goals. Thus, the problem is one of designing a solution through a code component library.

The plan itself is retrieved from a library of plans, based on user specification. Additionally, if the selected plan does not precisely match the user specifications, alterations can be made. The refined plan can be stored for future retrieval. Thus, an ACG system can be built as a combination of Routine Design (RD), Template-Based Programming (TBP) and Case-Based Reasoning (CBR).

In this paper, the Automated Coder using Artificial Intelligence (ACAI) system is presented. The paper is laid out as follows. Section 2 identifies related work and some background into both RD and CBR. Section 3 presents the ACAI system. Section 4 contains a brief example and description of the system in action. Section 5 offers some conclusions and future work.

Related Work

The earliest instance of CBR is found in the system CHEF (Hammond 1986), a program to generate Szechwan cuisine recipes based on user goals. CHEF utilized a library of previous dishes for cases. Cases included such pieces of information as ingredients, textures, and preparation instructions. The CHEF system would retrieve a closely matching recipe based on the user specification, compare the goals of the matched recipe to the user's specifications, identify goals that were not met, or constraints that would not be met, and attempt to repair the selected recipe. The new recipe would be stored in the library so that the system could *learn* over time. This initial CBR system demonstrated the utility of the approach: solving new problems through previous solutions. A CBR system would perform four primary tasks: case retrieval, case reuse, case revision, caseretention.

The Kritik system (Goel, Bhatta and Stroulia 1997) developed in the late 1980s applied CBR to the physical design problem. Cases would represent component parts and Kritik would propose a design for a physical artifact by selecting components. Unlike CHEF where cases were represented by goals, Kritik represented its cases by their structure, function and behavior. The components' structures would be used to ensure that the components did not violate constraints, components' functions would be used to match goals, and components' behaviors could be used in simulation to ensure that the device functioned as expected. CHEF and Kritik are noted for their contribution to CBR although neither addressed ACG.

The Individual Code Reuse Tool, or ICRT, applies CBR to software reuse (Hsieh and Tempero 2006). A library of software components comprises the cases for the system. In ICRT, the software components are represented by both complete code segments and incomplete or partial code segments, the latter of which may be syntactically invalid as is. Cases are stored in a flat structure and indexed using attribute-value pairs. Indexes are assigned by the software developers using the system. Components are selected using a nearest-neighbor algorithm and brought to the developer's attention. It is up to the developer to utilize the suggested code segment or not. Therefore, while CBR is used, it is not an automated system. Of particular note however is the indexing scheme. Case attributes are represented using functionality cards, describing for each code segment the segment's language, feature, property and description.

In the Software Architecture Materialization Explorer (SAME) system, the goal is to produce object-oriented designs (Vazquez, Pace and Campo 2008). These designs are then presented to the developers who use the designs to produce the final programs. The designs are produced from a case library of various software architectural parts, such as a data access layer. Although the developers modify the case components by hand, SAME monitors any such operations to capture the changes for future uses.

The Case-Based Reasoner for Software Component Selection (Fahmi and Choi 2009) is currently only a conceptual design of a CBR system for software component selection. As with the previous two systems, this system automates only the selection of case components from a library of reusable software components. Cases include function, associated components, component justification and case justification in support of providing rationale for why a component might be used.

While the previous systems automated only a portion of the process, the Case-Based Software Reuse System, or CAESAR, (Fouqut and Matwin 1993) offers an example of a complete ACG. CAESAR applies a variant of CBR called compositional software reuse to perform code generation in the domain of linear algebra. Cases are reusable mathematical routines written in C. Code segments are retrieved based on user specifications and partial matching, along with plan decomposition. Inductive logic is used to capture frequently occurring instances of code segments so that these can be stored for future use. Such groupings are called slices.

Finally, Menu Browser Using Case Based Reasoning (MESCA), applies CBR to the problem of generating a user interface based on reusable software components (Joshi and McMillan 1996). Here, the reusable components are menus and the system will adapt menus to fit specified functions, application types, user-tailored fields and graphical design.

Aside from a great number of CBR efforts, the ACAI system highlighted in this paper draws from both RD and TBP. RD (Chandrasekaran and Josephson 2000) is a class of design problem in which the overall design strategy is well known and can be represented through plan decomposition. That is, solving an instance of the design problem is handled by decomposing the problem into subproblems or components. Each component itself might be further decomposed.

In RD, at the lowest level, specific design steps are available as component descriptions. A component description defines in English, through code, or mathematically how a given component is constructed and placed into the overall design. Commonly, there are multiple component descriptions available for any component. Therefore, the best component description is selected using some form of matching knowledge based on user specifications, constraining factors, and demands imposed by other components. RD has been applied to numerous problems from physical design (air cylinders) to abstract planning (air force mission planning) and abstract design (nutritional meal design) (Brown and Chandrasekaran 1989, Brown 1996, Fox and Cox 2000).

Template-based programming (TBP) originated in the 1960s but came into use primarily in the 1990s. The idea is to represent program logic in a generic form that can be filled in later by another program. For instance, a loop might be represented generically only to have its details filled in at a later time when those details become known. TBP has been applied to a number of problems ranging from the numeric subroutines to web site generation (Fernandez et al 1993, Jiang and Dong 2008).

An Automated Coding System: ACAI

The Automated Coder using Artificial Intelligence (ACAI) system is a first pass at a purely automated code generation system (Danilchenko 2011). Code generation systems cited in the previous section either required human involvement in the processing loop or were restricted to domains that may not be amenable to a general case, such as creation of menus and linear algebra. It is envisioned that the approach taken by ACAI can extend to a great number of applications and domains, although currently ACAI only solves database-type problems. Specifically, the initial implementation of ACAI was constructed to tackle the queries listed below. These queries were identified by data analyst at a hospital, citing that software which could solve such tasks would greatly reduce their workload.

- Average, maximum, minimum patient length of stay, by diagnosis, age, department
- Average amount of time patients waited between arrival and first procedure, first lab test, first physician visit, first triage
- Search for all patients who meet a given mode of arrival (ambulance, car, walk-in, air-transport) sorted by arrival time
- Average, maximum, minimum time to get lab results over all patients and lab requests
- Average, total, mean number of patients with/without insurance by day, week, month, year
- Most common diagnoses by time of day, weekday, month or season
- Number of patients by doctor, unit, nurse, diagnosis, location, age
- Average, mean amount of time between preliminary finding and final lab result

The restriction to the medical domain was made because of the interest in the topic. The limitation to handling database-type operations was made to ensure that a prototype system could be constructed. See section 5 for comments on future work.

ACAI accepts two forms of user input, the goal (i.e., the query or queries to be answered) and specifications for how to achieve the goal (e.g., computational complexity, memory and disk usage, form of input, form of output). The output of ACAI is a working Java program.

Given user input, the first step that ACAI undertakes is similar to that of CBR. A case must be retrieved from the library of cases. In ACAI, cases are *plans*, described using XML.

ACAI selects a plan through simple matching of user's stated goal for the program. ACAI contains plans for such activities as sorting, filtering, computation, and reasoning over event durations. As each plan is generic in nature, the queries listed above can be solved by just a few plans. Even so, the user's goals may match multiple plans, in which case ACAI uses a combination of matching plans rather than selecting a single plan.

A plan comprises several sections. First, the plan has a name and a description. Next, a plan has a number of steps broken down in three distinct types: input, operation, and output. Input steps describe from where the program will obtain its input. Operation steps describe the individual, executable portions that must make up the program to solve the given problem. Operation steps include a variety of types of computations such as summation, average, or maximum. Finally, output steps describe where the program will send its output. Notice that input and output steps describe the "where" while the operation steps describe the "how". Each step of a plan is described in terms of goals to be fulfilled. The goals are a list of attributes that describe the code that should be used to implement the given plan step.

Figure 1 provides an example of the input portion of a plan. This section contains two types of inputs. First are the generation inputs. This input allows ACAI to query the user who is generating a program, not the end user. Such input might, for instance, obtain information about the functionality of the intended program. For example, the user might input a specific type of aggregate function such as average or maximum. This input helps specialize a plan step, for instance altering the goal [Utilities – Aggregate – Property] into [Utilities – Aggregate – Maximum] or [Utilities – Aggregate – Maximum - String]. The second type of inputs is the running inputs. This input consists of actual prompting messages that will appear in the generated program so that, when run, the program will be able to ask the end user for additional details. One example might be a pathname and filename for the input file of the program.

```
<UserInputs>
    <GenerationInputs>
        <Input RefineGoal="[Utilities -
               Aggregate - Property]"
               Prompt="Which aggregate function
               (Max, Min, Avg, Total)?"/>
        <Input RefineGoal="[IO - Out]"
               Prompt="Where to output (Console,
               File)?"/>
   </GenerationInputs>
   <RunningInputs>
        <Input Name="AggregateUserInput1"
               Prompt="What is your data file?"/>
        <Input Name="AggregateUserInput2"
               Prompt="What is the name of the
               property you would like to
               aggregate?"/>
        </RunningInputs>
</UserInputs>
```

Figure 1: Example Input Portion of a Plan

The heart of a plan is the list of plan steps. Figure 2 illustrates two plan steps of an aggregate plan. The first plan step is used to declare a variable. In this case, the variable is a collection of maps. The second plan step performs an aggregate computation operation on a declared collection. Notice how the type of operation is not specified. This piece of information is required before a specific piece of code can be generated, and the type of operation is obtained via the user specification.

```
<Step Name="records" StepType="Input">
        <Description>
                Declare a collection.
        </Description>
        <Goals>
        [Variables - Declaration - Declare -
                Collection - Of Maps]
        </Goals>
</Step>

<Step Collection="records"
            PropertyName="AggregateUserInput2">
        <Description>
                Apply an aggregate to a
                collection.
        </Description>
        <Goals>
                [Utilities - Aggregate - Property]
        </Goals>
</Step>
```

Figure 2: Two Sample Plan Steps

Now that ACAI has a plan, with its steps, ACAI must locate code segments to fulfill each of the plan step goals. ACAI contains a library of Java code components. The code components come in two different forms. First are fully written methods, each available to handle a type of goal or situation (e.g., an input routine, a sort routine, a search routine). Second are inline or partial pieces of code. These include, for instance, variable declarations, method calls, control statements and assignment statements. All code components are indexed in a similar strategy to ICRT's attributes. In this case, code indexes are described by:

- Type: variables, collections, I/O, control flow, utilities
- Function: declaration (for variables), filter, aggregate operation, event, input/output, assignment statement
- Operation: initialization, criteria for filtering or sorting, type of loop, duration of event, location of input or output
- Data type operated upon

As noted above, every step of a plan is described by a list of goals. Every goal is a generic version of information that can be found in the component library. For instance, a goal might be to declare a collection type of variable. The goal might be expressed as [Variables - Declaration - Declare - ArrayList]. Code components are selected based on how well they match the goal list of the plan step. Additionally, user specifications that include, for instance, whether speed or space is more critical, help select between matching code segments.

Three example code components are listed here. First is an inline statement that declares a collection and initializes it. Notice the use of ^^ symbols. When surrounding an item, these symbols represent a placeholder to be filled in later.

- Component index: [Variables – Declaration – Initialize and Declare – ArrayList]
- Component: Java.Util.ArrayList ^^Name^^ = new ArrayList( );
- Type of component: inline declaration

Second is another inline statement, in this case a loop. Notice the use of placeholders to flesh out the portions of the for-loop that depend on user specifications, such as data type, or an already generated identifier name that replaced a previous placeholder. Replacing placeholders is described below.

- Component index: [Utilities – Iteration – Collection - Map]
- Component: for (java.util.Map <String,String> ^^CurrentItem^^ : ^^Collection^^) { ^^Body^^ }
- Type of component: inline code

Third is a method to compute event duration. Only the header is shown here.

- Component index: [Utilities – Event –Duration – Find Even Duration – int]
- Component: int findEvenDuration (^^StartTimeStamp^^ ^^Name^^);
- Type of component: method

Now, ACAI replaces the component placeholders to construct final component code. In some cases, placeholders represent data types. The selected data type then is used for all matching placeholders. In other cases, names must be generated. For instance, parameter names for methods and variable names replace placeholders. Similarly, method names and method calls must match. ACAI fills in the placeholders and adds the names to complete the component code.

Once ACAI has complete component code, the next step is to fill in the program template. The template comes with the necessary code to make up a Java program. For instance, the template contains proper import statements, a main method, try and catch blocks, as well as additional placeholders.

Another step, which will not be described in detail here, occurs when multiple plans were initially selected. Recall that ACAI contains only a few basic plans. For a simple problem, only one plan would be retrieved. For instance, if the user requires a program to simply sort a collection of patient records by age, only the sorting plan will be required. However, a more complicated problem might involve first filtering records to find patients that meet a particular criterion (e.g., a diagnosis or arrival time), an aggregate computation involving length of stay between events, and finally a sort. Such a problem would require three different plans. In such a case, ACAI would have to combine the three selected plans together. To date, ACAI has only performed modest forms of plan combination.

The code generation process carried out by ACAI results in a program that fits the user specifications to solve the selected problem. Aside from the generated program, if plan combination was performed, the new plan is indexed and stored for future use.

In summary, ACAI uses CBR to retrieve a solution plan. The system uses RD to select appropriate code components and generate the concrete plan steps required to solve the problem. ACAI uses TBP in that it uses a template of a Java program, filling in the details and replacing the placeholders. The overall architecture for ACAI is shown in figure 3.

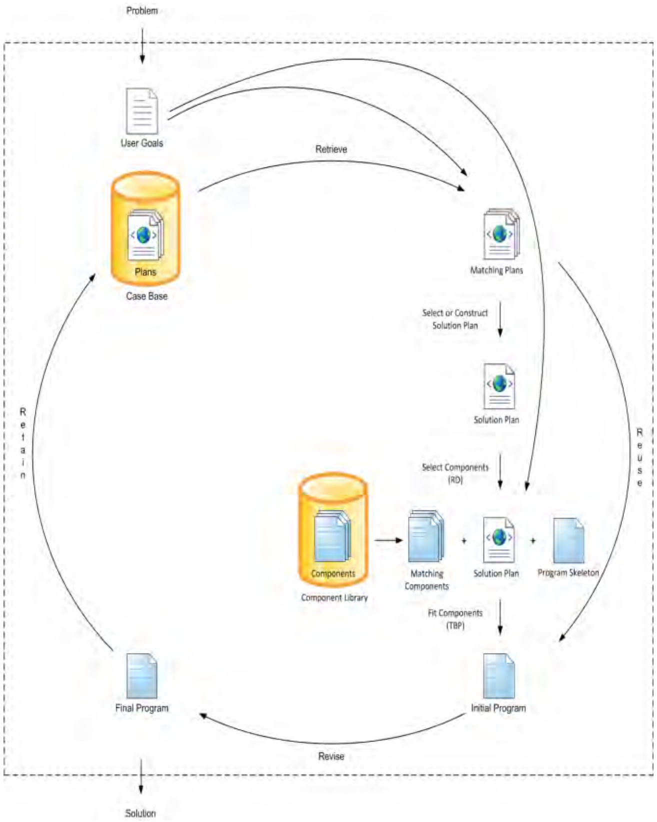


Figure 3: ACAI Architecture

A Brief Example

Here, a brief example is presented to demonstrate how ACAI carries out its code generation task. The user has specified a goal of sorting over integer data and requested the output to be sent directly to the console. Further, the user specifies that speed is of a greater concern than memory space usage.

Based on the input, ACAI retrieves the sort plan. The sort plan contains generation inputs and running inputs. These help specialize some of the goals in the plan steps and provide end user with prompts. The plan steps consist of a declaration of the input collection, an assignment statement to assign a variable to the source of input, a declaration of the sort operation collection variable, a sort routine, and an output step. The goals of these steps are listed here:

- Declare Input Collection: [Variables – Declaration – Declare – Collection – Of Maps – ArrayList]
- Store Input: [Variables – Assignment]
- Obtain Input: [IO – In – File – ArrayList]
- Declare Sorted Data Collection: [Variables – Declare – Declare – ArrayList]
- Store Sorted Data: [Variables – Assignment]
- Sort Data: [Utilities – Sort – ArrayList]
- Output Sorted Data: [IO – Out]

Now, ACAI must identify code components for each of the steps listed above and insert them into appropriate locations of the program template. The template is shown in figure 4.

```
package edu.nku.informatics.thesis.acai;

^^Program Comments^^

public class ProgramSkeleton
{
        public static void main ( String [] args
)
        {
                ^^User Inputs^^

                ^^User Prompts^^

                getUserInputs(userInputs,
                userPrompts);

                ^^Inline Code^^
        }


        // Get the inputs from the user

        ^^Method Code^^
}
```

Figure 4: The Java Program Template

The first code component sought is that of the declaration of input. ACAI selects the following inline statement:

java.util.ArrayList <java.util.Map <String,
String>> ^^Name^^;

Here, ^^Name^^ is a placeholder. ACAI now specializes the instruction to the given program by replacing the placeholder with an actual identifier:

java.util.ArrayList <java.util.Map <String,
String>> records;

The inline code above is inserted into the template under the ^^Inline Code^^ placeholder. As ACAI continues to find code components to fulfill the given plan step goals, the inline code (whether declaration, assignment or method call) are inserted in order based on the original list of plan steps.

With the identifier *records* in place in the program, ACAI will continue to use this name whenever it must replace other placeholders that reference this same datum.

For instance, the first assignment statement step is handled by the inline code:

^^Variable^^ = ^^Body^^;

which becomes

records = ^^Body^^;

The placeholder ^^Body^^ will be replaced by a method call which will obtain the input and return it as an ArrayList to be stored in records. In this case, the selected method is named readCSVFileIntoArrayList, which contains the code to read data from a file and return it as an ArrayList. This method call is used to replace ^^Body^^.

In many cases, the choice of code component to fulfill a plan step goal is a one-to-one mapping. That is, at least presently, there are few options because of the limited domain that ACAI is working in. However, there are some component options. For instance, there are several different sort routines available. For ACAI to select the best code component for the given goal, user specifications may come into play.

The sort step of this example could be fulfilled by any of six different sort methods. The sort code breaks down into two dimensions: the data type to be sorted and the sorting algorithm. Data types are restricted to numeric, date and string. Since numeric types can be handled generically in Java, Float, Integer, and Double are all sorted by the same routine. As a different type of operation is required to compare two Date objects or two String objects, there is a need for three distinct sorting methods. There are currently two sorting algorithms used in ACAI, Quick Sort and Selection Sort. As Quick Sort uses more space but is guaranteed to be as fast as or faster than Selection Sort, the user specification of speed over memory space causes ACAI to select Quick Sort in this example. The result is that the plan step goal is fulfilled by the following method call:

quickSortNumbers(^^Source^^, ^^Criteria^^);

The ^^Source^^ placeholder is replaced by the aforementioned records variable. The ^^Criteria^^ placeholder references the need for the program to obtain from the end user the criteria by which the sort should operate. This will be the type of data to be compared (e.g., a test result, patient's age, number of visits). The placeholder is replaced by code generated based on the running input. The following is the method call inserted into the program.

quickSortNumbers(records,
userInputs.get("SortUserInput2"));

The program's methods must also be inserted into the template. Methods are largely self-contained and require little change. However, they also contain placeholders, such as variable types, identifier names, and other method calls. The example from this section called for output to console. Assume instead that the output was to be sent to a disk file. Figure 5 contains the stored method selected by ACAI for such an output plan step. Recall that the plan step has the generic goal of [IO – Out]. This must be specialized to fit the user specifications, output to disk file. The ^^Data^^ placeholder in the method call must be replaced with the proper value. In this case, ^^Data^^ becomes list.

Once methods are put into place, the program is complete. ACAI now provides the program as output. An end user can now run the program to solve the desired problem. Running inputs are used to obtain the run-time information required for the program to fulfill the given task.

```
printDataToFile ( ^^Data^^ );


private static void printDataToFile ( Object
objData )
{
    try
    {
        // Declare variables
        java.io.BufferedWriter out = new
        java.io.BufferedWriter ( new
        java.io.FileWriter( "Data.txt"));
        // Write the specified string to the file
        out.write ( objData.toString() );
        // Flushes and closes the stream
        out.close ( );
        System.out.print("Result is stored in: "
           +  System.getProperty("user.dir"));
    }
    catch ( java.io.IOException e )
    {
        e.printStackTrace ( );
    }
}
```

Figure 5: Sample Method Call and Method for Output

Conclusions

ACAI, Automated Coder using Artificial Intelligence, combines the case base, case selection and case storage of CBR with plan decomposition of RD to fill in a template program using TBP. In this case, ACAI succeeds in automated code generation (ACG). Unlike other attempts at ACG, ACAI operates without human intervention other than high level input specification.

In ACAI, plans represent generic solutions to given database type problems. Each plan describes its solution through plan steps. A plan step describes the action required in terms of a goal. Goals provide such information as the type of operation, specific criteria for the operation, and data types.

Given a plan with plan steps, ACAI then selects specific code components from a separate code library. Code components are themselves indexed using attribute lists which match or overlap the goals from plan steps. These code components combine both inline Java code and Java methods. The code components are inserted into a Java program template. Placeholders in the code are replaced by specific identifiers, types, method calls and other programming units as needed.

In order to provide variability, each plan tackles a specific type of operation, such as sort or search. In complex problems, multiple plans are selected and refined into a single solution plan. Plan merging, although not discussed here, provides a seamless transition from one plan to another. The result is a new, more complex plan, which is stored back into the case base for future use.

ACAI has successfully generated programs to solve a number of medical database domain queries and subqueries from the list given in Section 3. ACAI is currently limited to the domain of medical record queries. Although this overly restricts ACAI's abilities, it is felt that the approach is amenable to a wide variety of problems.

It is important to note that the advantage of using ACAI, as oppose to solving the same medical record queries using SQL, is that ACAI's architecture is not restricted to any specific programming language. The ACAI system can be used to tackle a much wider range of problems that would be difficult or inappropriate to address with SQL. Additionally, ACAI allows end users with no programming knowledge to obtain desired results, while SQL would require learning the SQL language as well as having knowledge of programming concepts to accomplish the same task.

Due to ACAI's expandable architecture, theoretically, the only limitation of applying the system in other domains is the availability of associated plans and code components. All that is required to expand ACAI is a greater variety of plans and code components that can implement any new plan steps. Expanding ACAI is a direction for future research along with an examination of additional forms of plan step merging and case reusability. Another direction for future research is increasing the number of criteria that a user might specify for code selection beyond the speed versus space tradeoff mentioned here.

References

Brown, D. C, and Chandrasekaran, B. 1989. *Design Problem Solving: Knowledge Structures and Control Strategies*. Research Notes in Artificial Intelligence Series, Morgan Kaufmann Publishers, Inc.

Brown, D. C. 1996. Knowledge Compilation in Routine Design Problem-solving Systems, *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing,* p. 137-138, Cambridge University Press.

Chandrasekaran, B, and Josephson, J. R. 2000. Function in device Representation, *Engineering with Computers*, 162-177, Springer.

Danilchenko, Y. (2011). Automated Code Generation Using Artificial Intelligence. M.S. thesis, Dept. of Computer Science, Northern Kentucky University, Highland Heights, KY.

Fahmi, S. A. and Choi H. 2009. A Study on Software Component Selection Methods, in *Proceedings of the 11th international conference on Advanced Communication Technology,* p. 288-292, Gangwon-Do, South Korea.

Fernandez, M. F., Kernighan, B. W., and Schryer, N. L. 1993. Template-driven Interfaces for Numerical Subroutines, in *ACM Transactions on Mathematical Software (TOMS) TOMS p.* 265-287.

Fouqut, G., and Matwin, S. 1993. Compositional Software Reuse with Case-based Reasoning, in *9th Conference on Artificial Intelligence for Applications.* P. 128-134, IEEE Computer Society Press.

Fox, R. and Cox, M. 2000. Routine Decision Making Applied to Nutritional Meal Planning, in the *Proceedings of the International Conference on Artificial Intelligence, IC-AI'2000*, Volume II, p. 987-993, H. R. Arabnia editor, CSREA Press.

Goel, A., Bhatta, S., and Stroulia, E. 1997. Kritik: An Early Case-based Design System, in *Issues and Applications of Case-Based Reasoning in Design*, by M Maher and P Pu, 87-132. Mahwah, NJ: Erlbaum.

Hammond, K. J. 1986. CHEF: A Model of Case-based Planning, in *Proceedings of the Fifth National Conference on Artificial Intelligence,* p 267-271, AAAI.

Hsieh, M., and Tempero, E. 2006. Supporting Software Reuse by the Individual Programmer, in *Proceedings of the 29th Australasian Computer Science Conference*, p 25-33, Australian Computer Society, Inc.

Jiang, Y., and Dong, H. 2008. A Template-based E-commence Website Builder for SMEs, in *Proceedings of the 2008 Second International Conference on Future Generation Communication and Networking Symposia - Volume 01.* IEEE Computer Society.

Jones, C. 2010. *Software Engineering Best Practices.* The McGraw-Hill Companies.

Joshi, S. R., and McMillan, W. W. 1996. Case Based Reasoning Approach to Creating User Interface Components, in *Proceedings CHI '96 Conference companion on Human factors in computing systems: common ground*, p. 81-82.

Vazquez, G., Pace, J., and Campo M. 2008. A Case-based Reasoning Approach for Materializing Software Architectures onto Object-oriented Designs, in *Proceeding SAC '08 Proceedings of the 2008 ACM symposium on Applied Computing,* p 842-843, ACM.

Pixel Consistency, K-Tournament Selection, and Darwinian-Based Feature Extraction

Joseph Shelton, Melissa Venable, Sabra Neal, Joshua Adams, Aniesha Alford, Gerry Dozier
[#]*Computer Science Department, North Carolina Agricultural and Technical State University*
1601 East Market St. Greensboro, NC, 27411, United States of America
jashelt1@ncat.edu, mdvenabl@ncat.edu, saneal@ncat.edu,
jcadams2@ncat.edu, aalford@ncat.edu, gvdozier@ncat.edu

Abstract

In this paper, we present a two-stage process for developing feature extractors (FEs) for facial recognition. In this process, a genetic algorithm is used to evolve a number of local binary patterns (LBP) based FEs with each FE consisting of a number of (possibly) overlapping patches from which features are extracted from an image. These FEs are then overlaid to form what is referred to as a hyper FE.

The hyper FE is then used to create a probability distribution function (PDF). The PDF is a two dimensional matrix that records the number of patches within the hyper FE that a particular pixel is contained within. Thus, the PDF matrix records the consistency of pixels contained within patches of the hyper FE.

Darwinian-based FEs (DFEs) are then constructed by sampling the PDF via k-tournament selection to determine which pixels of a set of images will be used in extract features from. Our results show that DFEs have a higher recognition rate as well as a lower computational complexity than other LBP-based feature extractors.

Introduction

Genetic & Evolutionary Biometrics (GEB) is the field of study devoted towards the development, analysis, and application of Genetic & Evolutionary Computation (GEC) to the area of biometrics (Ramadan and Abdel-kader 2009; Galbaby et al. 2007; Alford et al. 2012; Shelton et al. 2012c). Over the past few years there has been a growing interest in GEB. To date, GEB has been applied to the area of biometrics in the form of feature extraction (Shelton et al. 2011a; Adams et al. 2010), feature selection (Kumar, Kumar and Rai 2009; Dozier et al. 2011), feature weighting (Popplewell et al. 2011; Alford et al. 2011) as well as cyber security (Shelton et al. 2012a; Shelton et al. 2012b).

$GEFE_{ML}$ (Genetic and Evolutionary Feature Extraction – Machine Learning) (Shelton et al. 2012c) is a GEB method that uses GECs to evolve feature extractors (FEs) that have high recognition accuracy while using a small subset of pixels from a biometric image. The results of Shelton et al. (2012c) show that FEs evolved via $GEFE_{ML}$ outperform the FEs developed via the traditional Local Binary Pattern (LBP) (Ojala and Pietikainen 2002) approach.

In this paper, we present a two-stage process for facial recognition (Tsekeridou and Pitas 1998; Zhao et al. 2003) known as Darwinian-based feature extraction (DFEs). The first stage takes a set of FEs evolved by $GEFE_{ML}$ and superimposes each to create a hyper FE. From this hyper FE, a probability distribution function (PDF) is created. The PDF is represented as a two-dimensional matrix where each position in the matrix corresponds to a pixel within a set of images. Each value within the PDF represents the number of patches an associated pixel is contained within it.

In the second stage of the process, a Darwinian feature extractor (dFE) is developed by sampling the PDF via k-tournament selection (Miller and Goldberg 1996). The selected pixels are then grouped into *c* different clusters by randomly selecting α pixels to serve as centers. Our results show that the computational cost of DFE (in terms of the total number of pixels being processed) via dFEs is far less expensive than the FEs evolved via $GEFE_{ML}$. The dFEs also outperform $GEFE_{ML}$ evolved FEs in terms of recognition accuracy.

The remainder of this paper is as follows. Section 2 provides an overview of the LBP feature extraction method, GECs, and $GEFE_{ML}$. Section 3 provides a description of the two-stage process for developing dFEs. Sections 4 and 5 present our experiment setup and our results respectively. Finally, in Section 6, we present our conclusions and future work.

Background

Local Binary Pattern Method

The LBP method (Ojala and Pietikainen 2002; Ahonen, Hadid and Pietikinen 2006) extracts texture patterns from images in an effort to build a feature vector (FV). It does this by segmenting an image into rectangular regions, referred to as patches, and comparing the grey-scale intensity values of each pixel with the intensity values of a pixel's nearest neighbors. After pixels are compared with their nearest neighbors, a pattern is extracted. This pattern is represented by a binary string. A histogram is built using the frequency of occurring patterns for a patch. The histograms for every patch are concatenated to form a FV.

In the LBP method, images are traditionally partitioned into uniform sized, non-overlapping patches. Within each patch, pixels are sought out that have *d* neighboring pixels on all sides and that are a distance of *r* pixels away from a center pixel. Each of these pixels can be referred to as a center pixel, c_p, due to it being surrounded by a neighborhood of pixels. A texture pattern can be extracted using Equations 1 and 2, where *N* is the set of pixel intensity values for each of the neighboring pixels. In Equation 1, the difference between a neighboring pixel and c_p is calculated and sent to Equation 2. The value returned will either be a 1 or a 0, depending on the difference. The *d* bits returned will be concatenated to form a texture pattern.

$$LBP(N,c_p) = \sum_{t=0}^{d} \mathrm{M}(n_t - c_p) \qquad (1)$$

$$\mathrm{M}(x) = \begin{cases} 1, x >= 0, \\ 0, x < 0. \end{cases} \qquad (2)$$

Each patch has a histogram that stores the frequency of certain texture patterns extracted. The histograms for all patches of an image are concatenated together to create a FV for an image. This FV can be compared to another FV of an image using a distance measure such as the Manhattan Distance measure or the Euclidean distance measure.

GECs

$\mathrm{GEFE}_{\mathrm{ML}}$ uses GECs to evolve FEs (Shelton et al. 2012c). The resulting FEs have been shown to have high recognition rates. A GEC uses artificial evolution to evolve a population of candidate solutions (CSs) to a particular problem. Initially, a population of CSs is randomly generated. Each CS in the population is then assigned a fitness based on a user specified evaluation function. Parent CSs are then selected based on their fitness and allowed to create offspring using a number of recombination and mutation techniques (Spears and DeJong 1991). After the offspring are created, they are evaluated and typically replace the weaker members of the previous population. The process of selecting parents, creating offspring, and replacing weaker CSs is repeated until a user specified stopping condition is met.

$\mathrm{GEFE}_{\mathrm{ML}}$

$\mathrm{GEFE}_{\mathrm{ML}}$ evolves LBP-based FEs using some GEC, so FEs must be represented as a CS. $\mathrm{GEFE}_{\mathrm{ML}}$ represents an FE, fe_i, as a six-tuple, $<X_i,Y_i,W_i,H_i,M_i,f_i>$. The set $X_i = \{x_{i,0}, x_{i,1},\ldots, x_{i,n-1}\}$ represents the x-coordinates of the center pixel of *n* possible patches and $Y_i = \{y_{i,0}, y_{i,1}, \ldots , y_{i,n-1}\}$ represents the y-coordinates of center pixel of *n* possible patches. The widths and heights of the *n* patches are represented by $W_i = \{w_{i,0}, w_{i,1}, \ldots , w_{i,n-1}\}$ and $H_i = \{h_{i,0}, h_{i,1},\ldots, h_{i,n-1}\}$. Because the patches are uniform, $W_k = \{w_{k,0}, w_{k,1}, \ldots , w_{k,n-1}\}$ is equivalent to, $w_{k,0} = w_{k,1},\ldots, w_{k,n-2} = w_{k,n-1}$, and $H_k = \{h_{k,0}, h_{k,1}, \ldots , h_{k,n-1}\}$ is equivalent to, $h_{k,0} = h_{k,1},\ldots, h_{k,n-2} = h_{k,n-1}$, meaning that the widths and heights of every patch are the same. Uniform sized patches are used because uniform sized patches outperformed non-uniform sized patches in (Shelton et al. 2011b). $M_i = \{m_{i,0}, m_{i,1},\ldots, m_{i,n-1}\}$ represents the masking values for each patch and f_i represents the fitness of fe_i . The masking value determines whether a patch is activated or deactivated. If a patch is deactivated, by setting $m_{i,j} = 0$, then the sub-histogram will not be considered in the distance measure, and the number of features to be used in comparisons is reduced. Otherwise, the patch is activated, with $m_{i,j} = 1$.

The fitness f_i is determined by how many incorrect matches it makes on a training dataset *D* and how much of the image is processed by fe_i. The dataset *D* is composed of multiple snapshots of subjects and is divided into two subsets, a probe and a gallery set. The fe_i is applied on both the probe set and gallery set to create FVs for each set. A distance measure is used to compare FVs in the probe to FVs in the gallery and the smallest distances are considered a match. If the FV of an individual in the probe is incorrectly matched with the FV of another individual in the gallery, then that is considered an error. The fitness, shown in Equation 3, is the number of errors multiplied by 10 plus the percentage of image space being processed.

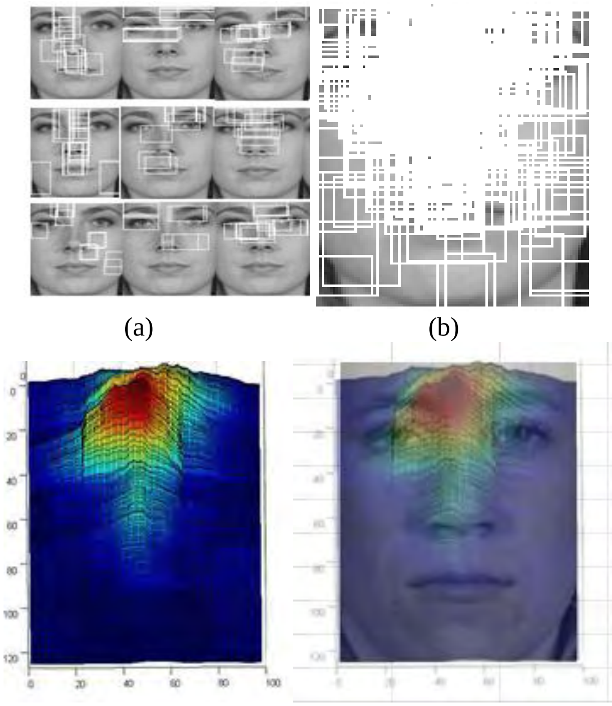
$$f_i = 10\varepsilon(D) + \gamma(fe_i) \qquad (3)$$

To prevent overfitting FEs on a training set during the evolutionary process, cross-validation is used to determine the FEs that generalize well to a dataset of unseen subjects. While offspring are applied to the training dataset to be evaluated, they were also applied to a mutually exclusive validation dataset which does not affect the evolutionary process. The offspring with the best performance on the validation dataset is recorded regardless of its performance on the training set.

The Two-stage Process for Developing a Hyper FE and a PDF

Stage I: Hyper FE/PDF

The hyper FE is constructed by taking a set of FEs from $\mathrm{GEFE}_{\mathrm{ML}}$ and overlaying them. Figure 1a shows a set of sample FEs while Figure 1b shows a sample hyper FE. After the hyper FE is constructed, a PDF, in the form of a matrix, is created. Each position in the matrix contains the number of patches a pixel was contained in. When patches in an FE overlapped on a position multiple times, the overlap is considered in the count. So if the hyper FE had *n* patches, and used κ FEs, the greatest number of times a pixel was contained in a patch would be *n* * κ. Figure 1c shows a 3D plot of a PDF, while Figure 1d shows the 3D plot laid over a facial image.



(a) (b)

(c) (d)

Figure 1: (a) Set of FEs (b) hyper FE (c) 3D plot of PDF and (d) overlay of 3D plot on a facial image

<u>Stage II</u>: Developing dFEs

A dFE can be defined by the number of clusters it has, α, the selection pressure of tournament selection, *μ*, and the patch resolution, *ρ*. The variables *μ* and *ρ* are represented as a percentage, or a value between 0 and 1. Assume that *β* represents the number of pixels a user would want for a cluster, there are α **ρ**β* positions that will be selected to be clustered. Tournament selection selects *μ**σ pixels to compete for clustering, where σ represents the total number of positions in the PDF that have been processed at least once. When performing tournament selection, the position with the greatest consistency will be the winner. If there is a tie, then the first selected position is the winner. Winning pixels are selected without replacement.

After α **ρ* β* pixels have been selected via tournament selection, α random centers for clusters are chosen to be placed within the PDF. The distance between each of the selected positions for clustering will be compared to the center positions, and the pixel will be clustered towards the closest one. After pixels have been assigned to clusters, those pixels undergo LBP feature extraction to extract texture patterns for a cluster. Due to the random placement of clusters, it is possible for different clusters to have different numbers of pixels clustered to it.

The clusters are similar to patches, therefore histograms are associated with each, and the patterns are used to build the histogram and ultimately create FVs for images.

Experiments

Two hyper FEs were used in this experiment: (a) a hyper FE composed of a set of FEs that performed well on the training set, HFE_{trn} and (b) a hyper FE composed of a set of the best performing FEs on the validation set, HFE_{val}. The FEs were evolved using the experimental setup in Shelton et al. (2012c), which used GEFE_{ML}. GEFE_{ML} was run 30 times using increments of 1000, 2000, 3000 and 4000 evaluations. An EDA instance (Larranga and Loranzo 2002) of GEFE_{ML} was used with a population of 20 FEs and an elite of 1, meaning every generation starting from the second contained the single best performing FE of the previous generation. On each run, GEFE_{ML} returned the best performing FE on the training set and the best performing FE with respect to the validation set.

The FEs were trained and validated on two mutually exclusive sets, and they were then applied to a test set. The datasets were composed of subjects from the Facial Recognition Grand Challenge database (FRGC) (Phillips et al. 2005). The training set was composed of 100 subjects ($\mathrm{FRGC\text{-}100}_{trn}$), the validation set was composed of 109 subjects (FRGC-109), and the test set was composed of 100 subjects ($\mathrm{FRGC\text{-}100}_{tst}$). The average number of patches used from the set of generalizing FEs as well as the average number of pixels processed in a patch were calculated in order to set a starting point for this experiment. On average, 12 patches were activated and 504 pixels were processed by each patch using GEFE_{ML}.

In this experiment, instances of 16, 12, 8 and 4 clusters were tested. Different patch resolutions, or the amount of pixels that could belong to a cluster, were also used. In this experiment, σ = 504. This was the average number of pixels in patches of the set of FEs from GEFE_{ML}. Instances of DFE with patch resolutions of 1.0, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2 and 0.1 were run. Each resolution used selection pressures from 0.0 (where number of pixels to be compared in tournament selection is actually 2) to 1.0 and every tenth percentage in between. A dFE is defined to be a cluster, patch resolution, then selection pressure, giving a total of 880 dFEs (4 clusters * 10 patch resolutions * 11 selection pressures * 2 hyper FEs), and each DFE instance was ran 30 times. For each run, a dFE was applied to FRGC-100$_{tst}$.

Results

The results were obtained by running each dFE listed in Section 4 on FRGC-100$_{tst}$.

To compare the effectiveness of each method, we compare the results of different selection pressures within a certain resolution and patch. The results of the best selection pressure for a resolution are compared to the best selection pressures of every other resolution within the cluster group, and this is done for results in every cluster. After the best performing FEs are obtained from each cluster, they are compared to each other as well as the results of GEFE$_{ML}$. Results are compared using an ANOVA test and a t-test on the recognition accuracies for a cluster-resolution-selection pressure instance.

Table I shows the results of this experiment. The first column shows the methods used. The method DFE$_{val}$ represents dFEs that sampled the HFE$_{val}$, while the method DFE$_{trn}$ represents dFEs that sampled the HFE$_{trn}$. The two methods are compared to the original GEFE$_{ML}$ method, shown as GEFE$_{ML}$. The second column, Feature Extractor, shows the number of clusters used, the resolution and the selected pressure for a dFE. The third and fourth columns show the computational complexity (CC) and the average recognition accuracy (Acc) respectively for each method. The computational complexity is the number of pixels processed, or extracted, by each method. Though 880 dFEs were tested, the only ones shown are ones that produced superior results to GEFE$_{ML}$.

For DFE$_{val}$, each dFE showed in Table I outperformed GEFE$_{ML}$ in terms of recognition accuracy. For DFE$_{trn}$, the dFE <12,0.5,0.2> was statistically equivalent to FEs evolved using GEFE$_{ML}$. Though we compare results based on recognition accuracy, we also considered computational complexity.

The results show that the <12,0.5,0.2> dFE (of DFE$_{trn}$) outperforms GEFE$_{ML}$ in terms of computational complexity, and that the <12,0.9,0.1> instance of DFE$_{val}$ outperformed DFE$_{trn}$ and GEFE$_{ML}$in terms of recognition accuracy as well as computational complexity. These results are promising in terms of recognition and feature reduction of DFE.

Table I: Results of DFE$_{val}$, DFE$_{trn}$ and GEFE$_{ML}$

Method	Feature Extractor	CC	Acc
DFE$_{val}$	<16,1.0,0.8>	8064.0	99.82%
	<16,0.9,0.5>	7257.6	99.69%
	<16,0.8,0.4>	6451.2	99.85%
	<16,0.7,0.5>	5644.8	99.60%
	<12,1.0,0.2>	6048.0	99.66%
	<12,0.9,1.0>	**5443.2**	**99.39%**
DFE$_{trn}$	<12,0.5,0.2>	3024.0	98.70%
GEFE$_{ML}$	----	6048.0	99.10%

Conclusion and Future Work

The results of the experiment suggest that the HFE$_{val}$ produces dFEs that generalize well to unseen subjects. The dFEs resulting from the HFE$_{trn}$ also generalized well, but were not as effective as when using dFEs resulting from the HFE$_{val}$. Using both hyper FEs performed better than the set of generalized FEs from GEFE$_{ML}$. Future work will be devoted towards using additional GECs for the DFE.

Acknowledgment

This research was funded by the Office of the Director of National Intelligence (ODNI), Center for Academic Excellence (CAE) for the multi-university Center for Advanced Studies in Identity Sciences (CASIS), NSF SSTEM, Lockheed Martin and the National Science Foundation (NSF) Science & Technology Center: Bio/computational Evolution in Action CONsortium (BEACON). The authors would like to thank the ODNI, the NSF, and Lockheed Martin for their support of this research.

References

Adams, J.; Woodard, D.L.; Dozier, G.; Miller, P.; Bryant, K.; Glenn, G., "Genetic-Based Type II Feature Extraction for Periocular Biometric Recognition: Less is More," *Pattern Recognition (ICPR), 20th International Conference on* , vol., no., pp.205-208, 23-26 Aug. 2010

Ahonen, T., Hadid, A., Pietikinen, M.; "Face Description with Local Binary Patterns: Application to Face Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 12, pp. 2037-2041, Dec. 2006

Alford, A., Popplewell, K., Dozier, G., Bryant, K., Kelly, J., Adams, J., Abegaz, T., and Shelton, J."GEFeWS: A Hybrid Genetic-Based Feature Weighting and Selection Algorithm for Multi-BiometricRecognition," Proceedings of the 2011 Midwest Artificial Intelligence and Cognitive Science Conference (MAICS), April 16-17, Cincinnati, OH., 2011

Alford, A., Steed, C., Jeffrey, M., Sweet, D., Shelton, J., Small, L., Leflore, D., Dozier, G., Bryant, K., Abegaz, T., Kelly, J.C., Ricanek, K. (2012) "Genetic & Evolutionary Biometrics: Hybrid Feature Selection and Weighting for a Multi-Modal Biometric System", IEEE SoutheastCon 2012.

Dozier, G., Purrington, K., Popplewell, K., Shelton, J., Bryant, K., Adams, J., Woodard, D. L., and Miller, P. "GEFeS: Genetic & Evolutionary Feature Selection for Periocular Biometric Recognition," *Proceedings of the 2011 IEEE Workshop on Computational Intelligence in Biometrics and Identity Management (CIBIM-2011)*, April 11-15, Paris, France., 2011

Galbally, J.; Fierrez, J.; Freire, M.R.; Ortega-Garcia, J.; "Feature Selection Based on Genetic Algorithms for On-Line Signature Verification," *Automatic Identification Advanced Technologies, 2007 IEEE Workshop on* , vol., no., pp.198-203, 7-8 June 2007

Kumar, D.; Kumar ,S. and Rai, C.S.; "Feature selection for face recognition: a memetic algorithmic approach." *Journal of Zhejanga University Science A*, Vol. 10, no. 8, pp. 1140-1152, 2009.

Larranaga, P.and Lozano, J. A.; "Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation", Kluwer Academic Publishers, 2002.

Miller B. L. and Goldberg, D. E.; "Genetic algorithms, selection schemes, and the varying effects of noise". *Evol. Comput.* 4, 2, 113-131, June 1996.

Ojala,T., Pietikainen, M.; "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns", IEEE Trans. Pattern Analysis and Machine Intelligence; 971-987; 2002

Phillips, P.J., Flynn, P. J., Scruggs, T., Bowyer, K. W., Chang, J., Hoff, K., Marques, J., Min, J. and Worek, W; "Overview of face recognition grand challenge," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

Popplewell, K., Dozier, G., Bryant, K., Alford, A., Adams, A., Abegaz, T., Purrington, K., and Shelton, J. (2011). "A Comparison of Genetic Feature Selection and Weighting Techniques for Multi-Biometric Recognition," *Proceedings of the 2011 ACM Southeast Conference*, March 24-26, Kennesaw, GA., 2011

Ramadan. R. M. and Abdel-Kader, R. F. ; "Face Recognition Using Particle Swarm Optimization-Based Selected Features," In *International Journal of Signal Processing, Image Processing and Pattern Recognition*, Vol. 2, No. 2, June 2009.

Shelton, J., Dozier, G., Bryant, K., Smalls, L., Adams, J., Popplewell, K., Abegaz, T., Woodard, D., and Ricanek, K. "Comparison of Genetic-based Feature Extraction Methods for Facial Recognition," *Proceedings of the 2011 Midwest Artificial Intelligence and Cognitive Science Conference (MAICS-2011)*, April 16-17, Cincinnati., 2011a

Shelton, J., Dozier, G., Bryant, K., Smalls, L., Adams, J., Popplewell, K., Abegaz, T., Woodard, D., and Ricanek, K. "Genetic and Evolutionary Feature Extraction via X-TOOLSS" in The 8th annual International Conference on Genetic and Evolutionary Methods (GEM), 2011b.

Shelton, J., Bryant, K, Abrams, S., Small, L., Adams, J., Leflore, D., Alford, A., Ricanek, K, and Dozier, G.; "Genetic & Evolutionary Biometric Security: Disposable Feature Extractors for Mitigating Biometric Replay Attacks" proceedings of The 10th Annual Conference on Systems Engineering Research (CSER), 2012a

Shelton, J., Dozier, G., Adams, J., Alford, A.; "Permutation-Based Biometric Authentication Protocols for Mitigating Replay Attacks" will appear in the Proceedings of the 2012 *IEEE World Congress on Computational Intelligence (WCCI 2012)*, 2012b

Shelton, J., Alford, A., Abegaz, T., Small, L., Leflore, D., Williams, J., Adams, J., Dozier, G., Bryant, K.; "Genetic & Evolutionary Biometrics: Feature Extraction from a Machine Learning Perspective", proceedings of *IEEE SoutheastCon 2012c*.

Spears, W. M. and DeJong, K. A. "An analysis of multipoint crossover," in the Proc. 1990 Workshop of the Foundations of Genetic Algorithms, pp. 301-315, 1991

Tsekeridou , S. and Pitas, I. ;"Facial feature extraction in frontal views using biometric analogies" in Proc. of the IX European Signal Processing Conference, I:315–318, 1998

Zhao, W., Chellappa, R., Phillips, P. J. and Rosenfeld, A.; "Face recognition: A literature survey". *ACM Comput. Surv.* 35, 4 (December 2003), 399-458., 2003

Quid Scio? On Cognition & Knowledge

Sahon Bhattacharyya
Department of Computer Science
St. Xavier’s College (Autonomous)
Kolkata, West Bengal, India 700016
E-mail: sahon.dgro@acm.org

Abstract

This paper argues for and attempts to establish three major hypotheses: first, that some of the ideas behind today’s knowledge bases in use in intelligent agents are misleading and this is because most studies tend to overlook the critical differences between the concepts of information and of knowledge, of perception and of truth; second, that knowledge acquisition (or knowledge production, as we term it) refers to the assimilation and association of disparate information according to prior knowledge and third, that all knowledge requires a context of discourse and that discourse is ultimately the source of all knowledge. In addition to arguing for these hypotheses, we also propose and make use of a rudimentary model of cognition alongside our arguments to demonstrate their validity. The afore-mentioned model is based strongly on the theories of cognition and of mind in the Sāmkhya-Yoga and Nyāya-Vaisheshika traditions of Hindu philosophy as are the hypotheses ultimately grounded in. In the course of our arguments for our third thesis, we shall conclude with definitions of thought (as being self-discourse) and sentience (as being the ability of an agent to engage in discourse with other similar agents) – terms known for being confusing and controversial to define – in terms of our idea of discourse. In general conclusion, it emerges from this ‘*discursive theory of knowledge*’ that realistic models of human memory and of knowledge representation can only be designed if and when the elusive line of distinction between information and knowledge is drawn correctly and that the role of discourse in cognition is more significant than what is held in current lines of inquiry.

I. Introduction

“*Quid Scio?”* – Latin for “What do I know?” – reflects one of the most fundamental questions in the cognitive sciences. What does one know? And how does one know that which one knows? In this line of inquiry, the study of cognition bears strong ties with the domains of epistemology and ontology. Cognitive science is the science of the mind – the science of perceiving and knowing – where the human mind is viewed as a complex system, which receives, stores, retrieves, transforms and transmits information (Stillings et. al., 1995, pp. 1). In this objective of understanding the mind and its cognitive processes and modelling them, one first needs, in our opinion, to know first what one is expected to cognize – the nature of the objects of cognition, that is. The problem of knowledge and its representation is one central to cognitive science and artificial intelligence – the analysis of the concept of knowledge and the nature of the justification of belief (Stillings et. al., 1995, pp. 368). Knowledge-based agents - that have been more or less the general trend up till now – consist primarily of the representation of knowledge and the reasoning processes that bring knowledge to life (Russell & Norvig, 2003, pp. 222). In other words, knowledge representation schemes have two parts – a knowledge base and an interpreter that manipulates it (Stillings et. al., 1995). But it emerges from our comparative analysis of the underlying philosophical theories that strongly coupling the two modules of knowledge representation and reasoning lead to certain problems in cognitive modelling. Logic – the study of the principles of valid inference and correct reasoning and of arguments, valid forms and fallacies – can be shown to be often an ineffective form of knowledge representation (Stanford Encyclopaedia of Philosophy). With the integral role of first-order logic in the domains of computer science and artificial intelligence to analyse and represent *truth*, with the Platonic conception of knowledge as ‘*justified true belief*’ and with the Platonic and Aristotelian foundations at the heart of all assumptions upon which knowledge-based systems have been built, the study of logic has overshadowed certain critical issues of epistemology in many areas of knowledge representation. The overt use of logic has resulted in the focus of knowledge-based systems shifting to concepts of truth, of validity, etc. But it is essential to realize that when we are talking of cognition, we should not be concerned with whether what we cognize is valid or invalid or true or false. Russell & Norvig (2003) identify this problem correctly and avoid the issue altogether by the use of the term “*logical agents*” to describe such systems. But it must be realized that one of the fundamental differences between the conception of knowledge in artificial intelligence and in cognitive science is that the former, as we said before, concerns itself with truth, validity and justification, borrowing heavily from classical philosophy, while the latter ought to be more concerned with just knowledge and not its explicit validation. In other words, the study of cognition should distinguish between knowing and validating what is known because validation would require meta-knowledge whose existence we find no reason to presuppose. In our approach we have attempted to deal with this problem not by arguing against the issue within the tradition but by stepping outside the framework and attacking the problem from a different perspective, namely, those of the Sāmkhya-Yoga and Nyāya-Vaisheshikā.

In Hindu philosophy, the issue of epistemology and logic is dealt jointly by the Vaisheshikā and Nyāya traditions which had distinct origins but were later merged as one single school referred to as the Nyāya-Vaisheshikā school. In the Nyāya-Vaisheshikā traditions, a distinction is made between the different kinds of knowledge (*jyāna*), for instance, cognized knowledge and validated knowledge (*pramā*, i.e., knowledge that is validated by *pramāna* or reasoning). The Sāmkhya and Yoga traditions, on the other hand, deal predominantly with psychological evolution with the former drawing a parallel with cosmic evolution. Although they are closely related and their names are hyphenated together, they are not technically merged in the same way as the previous two traditions are. They merely bear a strong relationship where the theories of the latter tradition hold under the overarching philosophical framework of the former. (Radhakrishnan, 2008, orig. 1923) These theories are examined in more detail in Sections III and IV.

We make use of certain epistemological concepts from these traditions and construct a generic model of cognition and a theory of knowledge which we refer to respectively as the discursive model of cognition and the discursive theory of knowledge. This paper is a shorter version of an extended work of this model currently in progress (See Bhattacharyya, 2012). The hypotheses that we shall be arguing for in this paper in support of our model are listed below. It is to be noted that in our arguments some of the theses pre-suppose the validity of the others. In other words, these are cohesively coupled in nature. Also, the exact definitions of data, information, knowledge and discourse are explained in detail later on so as to avoid ambiguity and inconsistency.

Thesis 1: All knowledge requires a ‘knowing’ subject whereas data and information do not.

Thesis 2: All knowledge is interrelated and associated. Information may be disparate and disconnected but knowledge is the assimilation and association of information according to the agent’s prior knowledge.

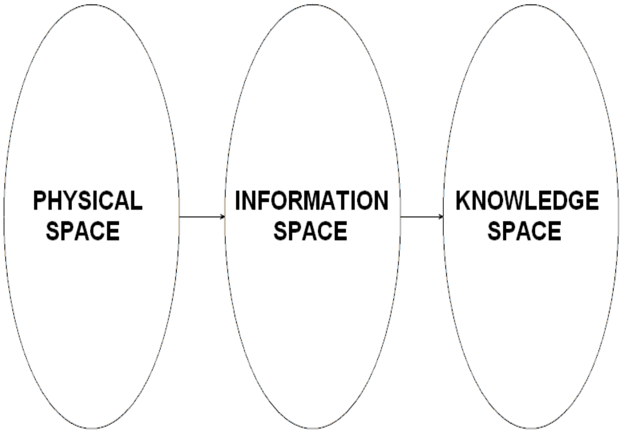
Thesis 3 (a): All knowledge production requires a context of discourse. In other words, discourse is ultimately the source of all knowledge.

Thesis 3 (b): Thought is an agent’s discourse with itself – self-discourse, that is.

Thesis 3 (c): Sentience is the ability of an agent to engage in discourse with other similar agents.

In Section II, we discuss an ontological thought experiment by considering the existence of three different spaces (the Physical space, the Information space and the Knowledge space). In Sections III and IV, the Sāmkhya and Nyāya-Vaisheshikā traditions are explained in more detail. It must be noted that since a full explanation of these traditions is well beyond the scope of this paper, we elucidate only the relevant portions – that is, only those which our model of cognition is based upon. Yoga therefore is not discussed at all because despite this model being based on it, the hypotheses in this paper do not require its exposition for their proof. Section V demonstrates our generic model and explains the various faculties and processes involved in the process of cognition and knowledge production drawing parallels from the original theories and tracking our derivations and deviations. Section VI explains in detail how objects of perception are cognized and how knowledge is produced, the role and significance of discourse in our model and explain why we ascribe to it so much importance so as to name our model after it. In Section VIII, we argue for Theses 3 (b) and 3 (c) and attempt to define *thought* and *sentience* in terms of our conception of discourse. Finally in conclusion, we explore open-ended questions and problems in our model and discuss how they can be resolved.

II. The Three Spaces



There are, we hypothesize, three spaces – the Physical space, the Information space and the Knowledge space. The Physical space is the world of physical objects governed by the laws of Physics – the world of atoms, compounds, sound, light, heat, etc. The Information space is the world of representation and the residence of discourse. And subtly intertwined though distinct in essence from both of these is the Knowledge space – the world of all that we know. (See Figure 1)

Entities in the Information space represent an object in the Physical space. In other words, there exists a mapping between the Physical space and the Information space. But there are physical objects we have not perceived and not having perceived them we have never represented them; so they do not occur in the Information space. And there exists composite and complex information that we have perceived and assimilated which does not exist in the natural physical world. For instance – the symbols on this document (the letters, the words, etc) are all physical entities (either light from a screen or ink on paper) but that they convey information to you that is not inherent in them in anyway implies that all entities in the Information space do not necessarily have an equivalent in the Physical space. What occurs in the Information space is laid bare before us – the knowing subjects - to know. When we perceive an object of the Physical space – directly or indirectly – a mapping function is first invoked. In direct perception, this is done by the mind which collects all sensory information and assimilates them into a coherent representation. In indirect perception, this may be through linguistic statements *referring* to physical objects. That which is represented is not the same as that which is known. That which is known cannot be represented. Therefore, we see from this thought experiment that firstly, it is impossible to represent all that we know because any attempt to describe an entity in the Knowledge space would be to project it onto the Information space; and secondly, entities in the Physical space are ultimately unknowable in their essential form because all attempts at perception would require at first a projection onto the Information and Knowledge spaces. This means that I could very well conjecture that the apple I am aware in front of me is not the physical apple that is present in front of me which in reality is perhaps unknowable. The image of the apple in my eyes creates a representation in the form of neurotransmitter signals of which I then come to 'know' of. And that which I know, I would have to first transform into representation by means of language or diagrams or any vehicle of discourse and then convey it by writing it down or saying it, that is, projecting it onto the Physical space. Therefore entities in the Information space are merely bearers of meaning and referrers of entities in the Physical space.

We compare this loosely with the concepts of data, information and knowledge – three distinct terms often confused and widely misunderstood. From data comes information. From information comes knowledge. The transformation of data to information requires syntax and the transformation of information to knowledge requires semantics and pragmatics. In this data-information-knowledge continuum, data is the absolute end and knowledge is the relative end with information serving as a necessary medium in between. What does absolute and relative signify here? From a certain data based on certain syntax, all agents can derive the same information. But from the same information, all agents may not necessarily derive the same knowledge. To use philosophical terms, knowledge, as may be etymologically obvious, requires a knowing subject. It is possible to replicate syntax but it is not possible – indeed an invalid idea of a reductio ad absurdum nature – to replicate semantics, pragmatics and/or discourse. Let us demonstrate with an example – I give you a set of data {1, 2, 1, 1, 2, 1, 2, -1, 1, -1, 1, -4, -1, -4, -1, -1, -2, -1, -2, 1, -1, 1, -1, 2}. For purposes of convenience, this is in decimal format – it could also have been in binary, it does not matter. Now you may be wondering what this set of data means. Which brings me to my second argument: *data does not carry meaning*. Of course, you can find meaning in this data – you may imagine this as a mathematical progression. But you have no way of knowing what I meant. Which brings me to my third argument: there are two meanings – *intended meaning* and *perceived meaning*. In other words, meaning presupposes the existence of two agents – one which creates the data and one which perceives the data. Therefore it must be realized that in this world of communication and information transmission, data by itself is meaningless unless it is transformed at some stage to information and knowledge. Now, if I tell you what these mean – if I give you the rules governing its correct perception, that is – I can expect reasonably that you will obtain what I intended to convey. For example, say, I tell you that these are the additive components of a number. You add them up and get -4. So that was the message apparently. But notice that if I asked your parrot to perform the same feat (assuming it can do math or even a calculator instead of the poor parrot), I'm confident it'd get -4 too. So now it is obvious what I meant when I said that data is the absolute end of the data-information-knowledge continuum. Now if I tell you that the decimal numbers are to be grouped into groups of two and are to be interpreted as points on a two-dimensional Cartesian plane, I'm guessing you'd like to have a look at the data again. Now what do you see? Now you see a set of points. In all probability, you've already plotted it in your mind. So I hardly need to mention that it's a polygon. So what do you see? You see a cruciform polygon. So that is the message. Therefore not how by different syntax and by different ways of the examination of the data, we obtain different information. But I'd like to point out that it's still information – not knowledge. So how does this become knowledge? It might be amusing to note that it has already made the transformation inside *your* mind; you just aren't aware of it yet – the process is so fast and involuntary. Every representation (symbolic in this context) has two meanings – a *denotation* and a *connotation*. The denotation of the cruciform polygon is, for all intents and purpose, a cross-like figure – nothing else. The connotation of this polygon, on the other hand, is not fixed. For those well-versed in history, it can mean the cross as a form of physical torture; it can refer to the Holy Cross and, by extension, the sufferings of Christ on the cross. For those who aren't, it can mean other things – which we are not in a position to control. This connotation is the aspect which controls the storage and classification of a perceived entity. (After all, if a cross symbolized 'victory' in a given culture, then the agent would have reacted differently to the message.) And it becomes evident that one cannot transform information into knowledge without one's prior knowledge. In other words, knowing something new requires the knowledge (and subsequent recall) of previously acquired knowledge. As data-to-information transformation requires syntactical rules, information-to-knowledge requires prior knowledge. Now it may be asked why one would bother with such associated meanings. The problem is in our too-simple example because a cruciform polygon does not constitute a semantically valid message. An English sentence, say, "I don't know!" is a slightly better example. The data is a string of letters {I, d, o, n, ', t, k, n, o, w, !}. The information is constructed by the rules of English grammar and we get a syntactically valid *sentence*. But do we get its meaning? No we don't. We don't know what the speaker is talking about. *Unless* we are told what the context is – prior knowledge, that is.

III. The Sāmkhya-Yoga tradition

The Sāmkhya tradition is one of the oldest systems of Indian philosophy. The word '*sāmkhya'* derived from the Sanskrit word '*samkhyā'* refers to right knowledge as well as number. As Radhakrishnan (2008, orig. 1923) remarks, this system represents 'a notable departure in thought from what may be called the formalistic habit of mind'. It rejects the rigidity of the Nyāya-Vaisheshikā categories as inadequate instruments for describing the complex and fluid universe. Instead it views the world as a creative evolution not as an act of a supernatural being. At the heart of this tradition is the theory that the effect pre-exists in the cause. From a series of deductive arguments following this predicament (which we do not describe for the sake of brevity), the Sāmkhya arrives at a duality of two unrelated entities, Purusha and Prakrti – the former being the witnessing consciousness or the subjective knower and the latter being used to describe the ultimate unmanifest basis of the empirical universe. As an influence of the Purusha, the vast universe unfolds as a cosmic and psychological evolution of Prakrti. The first evolute that emerges is called *mahat* or *buddhi* (the intellect or the discriminating awareness). Second to arise is *ahamkara* (ego-sense), or the principle of individuation. Other evolutes include the *manas* (the lower mind), the *jyānendriyas* (five organs of cognition) and the *karmendriyas* (five organs of conation). Also produced are the *bhutādi* (five subtle elements), from which emerges the five gross elements. (Radhakrishnan, 2008, orig. 1923)

The Yoga tradition – traditionally having originated from Patanjali's foundational and highly aphoristic Yoga-Sutras – is a psychological and spiritual treatise that discusses cognition. It accepts the psychology and metaphysics of the Sāmkhya tradition and so all that we explained above also holds here. The Yoga, like the Sāmkhya, speaks of the five states of the mind (*vrttis*, that is) – *pramāna* (right knowledge, obtained from sense perception), *viparjaya* (error, stemming from false knowledge and incorrect apprehension), *vikalpa* (imagination or metaphor, where the usage of words is devoid of an actual object), *nidra* (the state of sleep where there is no content) and *smrti* (memory). Yoga uses the term *samskāra* to refer to sense-imprints and memories – the former when the object is present and the latter when the object is absent. (For all purposes, these can be thought of as recorded impressions though it is entirely possible that they are not stored as-is.) It is held by Yoga that mind (*mānas* or *citta*) is reflective in nature – tending to reflect within itself whatever it perceives thereby making it available to the ego-sense (*ahamkāra* or *asmita*). A thing – which, according to Yoga, always consists of the three *gunas* – is known by the mind only when the latter notices the former and the former is said to exist independent of its being noticed by the latter. When the mind consciously focuses on a certain place or environment, it is said to be in a state of *dhārana* (concentration or, technically, attentional control); when it focuses upon a singular object unwaveringly, the state is *dhyāna* (meditation) and when the 'I-ness' or 'ego-sense' is not cognized anymore and the act of cognition itself becomes unconscious, the state is *samādhi* – when the knower merges with the known. And it is through this *samādhi* that insight (or *prajnā*) arises. It is interesting to note that Yoga draws a line of distinction between the act of perceiving an object and the act of ascribing the instrumentality of the act to the ego-sense. The aim of the spiritual aspect of the Yoga therefore is to achieve this sublime state by the 'restraint of the sense-impressions' which disturb the mind 'as waves rippling on the surface of still water'. (Radhakrishnan, 2008, orig. 1923; Sharma, 1987; Bryant, 2009) Although any discussion on Yoga is philosophically incomplete without a description of the practices which lead to this state, we do not discuss them as they are beyond the scope of this paper.

IV. The Nyaya-Vaisheshika tradition

Since it is difficult to describe in short the traditions of the Nyāya and Vaisheshikā which represent the analytical traditions of Indian philosophy, we discuss only the relevant sections. Nyāya – sometimes called *hetuvidyā* (the science of reason on which the validity of an inferential argument depends) – literally means that by which the mind is led to a conclusion. It is hailed as *pramānshāstra* – the science of correct knowledge. According to this theory, all knowledge implies four conditions: the subject, the object, the state of cognition and the means of knowledge (Radhakrishnan, 2008, orig. 1923). 'Every cognitive act, valid or invalid, has the three factors of a cognising subject, a content or a what of which the subject is aware, and a relation of knowledge between the two, which are distinguishable though not separable. The nature of knowledge, as valid or invalid, depends upon the fourth factor of pramāna. It is the operative cause of valid knowledge in normal circumstances.' (*Nyāyavārttika*, i. 1. 1, trans. Radhakrishnan, 2008) Radhakrishnan (2008, orig. 1923, pp. 31) points out that Western treatises on logic do not generally treat of perception, but the Nyāya in contrast regards it as one of the important sources of knowledge. The Nyāya system considers two different kinds of perception: determinate (when you perceive and recognize it) and indeterminate (when you perceive but do not cognize the object). Sharma (1987) remarks that these are not 'kinds' of perception; they are merely stages in the complex process of perception, a view that we agree with.

The Nyāya system considers the indeterminate perception as the starting-point of knowledge production although it is not always held as being synonymous to knowledge – an aspect we shall later imbibe in our discursive model of cognition. It is the stage when the distinction of true or false does not apply and the logical issue does not arise (Radhakrishnan, 2008, orig. 1923). (We do not discuss inference here as it is beyond the scope of this paper.) The third kind of cognition, *upamāna* or comparison, has been defined as the knowledge of the relationship between a word and its denotation. For example – when we identity an object even if we've never seen it before but because it has been described to us before. The fourth kind of cognition, *shabda* or verbal testimony, refers loosely as a trustworthy statement spoken as a meaning collection of words, a sentence that is, taking into consideration the relevant semantics and pragmatics, etc.

The Vaisheshikā philosophy is a pluralistic realism which emphasizes that diversity is the defining nature of the universe. The term Vaisheshikā is derived from vishesha which refers to the particularity of all objects. The Vaisheshikā tradition provides the necessary ontological framework for the Nyāya tradition as the latter lends its epistemology to the former. Sharma (1987) considers the Vaisheshikā categories – distinct from the Aristotelian, the Kantian and the Hegelian categories – as being a metaphysical classification of all knowable objects or of all reals. These categories consist of substance (*dravya*), quality (*guna*), action (*karma*), generality (*sāmānya*), particularity (*vishesha*), inherence (*samavāya*) and non-being (*abhāva*). Of these we focus on generality or *sāmānya* as it is quite relevant in the current context. This generality is defined as a class-concept or a universal – the common character of the things which fall under the same class. It stands, not for the class, but for the common characteristic of certain individuals (Sharma, 1987). In other words, we construct our own ontologies bottom-up by finding common aspects of discrete objects rather than having them defined top-down right at the onset. The mere fact that philosophers and computer scientists struggle with the definitions of upper ontologies but are not overwhelmed by particularity strongly hints at the fact that our minds are more acquainted with the lower details of ontology than the upper levels, if at all.

V. A generic discursive model of cognition

We present below a generic discursive model of cognition below based loosely on the four philosophical traditions discussed above and our conception of the three spaces. This is a rudimentary model that leaves much space for further addition. Moreover, the affective and executive aspects of an agent which we believe are integral to cognition are not shown here. This section describes the various faculties and how the cognitive processes work in an abstract manner. (See Figure 2)

Our basic framework derives heavily from the Samkhya architecture of the human cognitive system. We refrain from calling this a model of the mind because the Western conception of the mind and the Indian conception of the mind are different. In the latter, the mind is only a part of the entire system. The *discriminant* and the *ego-sense* are not technically parts of the mind; they are separate but are strongly coupled. *Manas* (or the Lower Mind, as is often called) manages the five cognitive senses – that of vision, hearing, smell, taste and touch. It is often held to be a sense organ in its own right. Here the sensory data is assimilated and a discrete object is perceived. (The *manas* also co-ordinates the executive faculties – those of movement, speech, etc. – but they are not depicted here for the sake of simplicity.) The *lower mind* receives as input from the cognitive senses individual discrete objects. This is where attention comes in. A unique feature of this model is that when an agent first comes into contact with an environment, it uses attention to perceive the environment in a discrete manner. It does not view the world as a video footage (sort of) or as a sequence of moving frames or images. Rather, attention causes the lower mind to affix itself to any one of the cognitive senses at any given point of time and the perception of the object in the sense organ is relayed back to the lower mind. In this way, attention causes rapid iterations between the lower mind and the sense organs thereby constructing a set of discrete perceptions. Attention is induced or diverted – we are attracted or distracted, that is – because of the ego-sense. The *ego-sense* is the only autonomous faculty within the cognitive system. We borrow from our basic philosophical traditions when we attribute to it the tendency to be continuously swayed by sensory input and be drawn to it by means of attention. It is, as shown in the model, the centre of autonomy, the seat of will or intention and the directing faculty of awareness. Attention is drawn to the cognitive senses by their 'attention directors', as shown in the model. When we are reading a document, say, the *ego-sense* and the *manas* are affixed to the organ of vision. It does not mean that sensory input from the other organs is absent. It merely signifies that we are focused on one single thing. Now if, say, we hear a very loud noise from outside the window, we will obviously be distracted and the *ego-sense* will detach itself from the organ of sight to the organ of hearing. And even if our eyes are still on that document, we won't be actually reading it anymore for a split-second as 'our attention will be elsewhere'. This distraction can be instigated by all the five senses: vision by the intensity of light, hearing by the volume, say, etc. The *discriminant*, as the name implies, refers to the power of discrimination. This faculty is the store-house of sense-impressions (again, not an as-is video footage) and memory, and its function is to discriminate between perceived objects.

So how do these fit together? When my attention is drawn by a particular cognitive sense organ, my *ego-sense* directs my *lower mind* to affix itself to the relevant organ and perceive the object. The *lower mind* obliges and produces a perception of the object through my eyes, say. This is relayed onto the *omniverse of discourse* (which can be thought of for now as the context) – a region accessible to the *ego-sense*, the *discriminant* and the *lower mind*. But this is an indeterminate perception – there's something there, but I don't know what it is. This is the first stage in our perception process. As soon as this is perceived, the *ego-sense* claims subjectivity – 'I' am 'aware' of something (although I do not know what it is). This is the second stage of perception. This is where the *discriminant* comes in and matches it with its store of sense-imprints and comes up with an identifying name: 'pencil'. This is determinate perception and so now I am aware of a pencil which is the third (and presumably final) stage in perception and the *discriminant* records my subjective experience of being aware of a pencil. But this is NOT to say that I am aware of the fact that I am seeing a pencil. There is a subtle and critical difference between the two! Up till now, I have not 'thought' anything nor have I expressed something. So technically I cannot describe what processes my mind has undergone. In order to express the fact that I am aware of something, I require the use of language (or any representation scheme). I need to realize how I became aware of the pencil – my organ of vision. It is only after the *ego-sense* consults the discriminant a number of times before I am able to assert that 'I can see a pencil'. The acknowledgement of the process of perception is a conscious act which is the fourth stage of perception.

So the steps are something like this:

1. A particular sense organ requests attention from the *ego-sense* (from me, i.e.).
2. The *ego-sense* (I, i.e.) directs the *lower mind* to affix itself onto that particular sense organ.
3. The *lower mind* perceives a discrete object (indeterminate perception).
4. The *ego-sense* assumes subjectivity and the subjective experience of being aware of something arises. ('I am aware' of something.)
5. The *discriminant* 'recognizes' the perceived object, say, as an apple.
6. Current state of my mind → 'I am aware of an apple.'

It is not that attention is compulsory for perception nor is it that the perceived object is determined only after the *ego-sense* assumes subjectivity. For instance, when in a car speeding along a highway in Japan (say!) and staring out absent-mindedly at the signposts, it is a common observation that our eyes will glaze over the strange symbols but notice the English ones even when we're not explicitly looking for them. This is because even when the ego-sense is not explicitly requesting the *lower mind* to perceive objects, the latter still keeps on receiving sense-impressions from the world (as long as the channels are available, i.e.) and submitting them to the *discriminant* for identification which then takes the liberty of discarding unknown symbols and raising an alarm (figuratively!) when it recognizes familiar symbols and causes the *ego-sense* to direct its attention to the object in question. So in this case we see how the perception of an object and the determination of the perceived object are accomplished even before we are aware of it.

However, whatever we have discussed till now is the perception and awareness of objects in our environment; we have not explored the foundations of thought processes. So this, we hypothesize, is the limit of perception, in that it causes a subjective awareness of perceived objects only. In order to reflect on these perceived objects, to relate or associate them, one is required to engage in discourse – an issue we address in the following section.

VI. A discursive theory of cognition and knowledge

What is discourse? We define discourse as a generic form of communication whose conception supersedes those of dialogues, dialectics, conversations or even written forms of communication involving the use of a certain representation scheme held by and with agents who may be spatially or temporally separated. We extend this definition to include not only the communication but along with it the circumstances which present themselves as demanding some manner of communication either in between two or more agents or even between an agent and itself. Neither is there a word that we read or write nor a word that we hear or we speak that finds itself lacking of membership in a discourse.

Therefore when we wish to convey some knowledge to other agents, we use information – the bearer of knowledge and the discursive vehicle – codified often as linguistic statements, which is then interpreted by those agents. All entities in the Information space therefore are elements of some discourse; all that is known therefore is through discourse itself. The cognitive process described in the previous section enables us to be aware of physical objects only. But with the discursive process, these discrete perceptions within the Knowledge space are projected back onto the Information space to form entities (in the form of linguistic assertions), for which there exists no equivalent in the Physical space. What do we mean by this? It is by the process of perception that I cognize (through *manas*) and recognize (through the *discriminant*) two discrete objects or entities: 'a horse' and 'an apple'. But it is only through discourse that I assert: 'Horses love apples'. Horses and apples both exist in the Physical space. But the assertion 'Horses love apples' does not exist in the Physical space although it exists as an element of discourse in the Information space. The construction of the above sentence requires the knowledge of how sentences are built (grammar, i.e.), the fact that the terms 'horse' and 'horses' refer to the same class of objects differing only in number (likewise for 'apple' and 'apples') and the semantic meaning of 'love'. These are not innate; no human is ever born with this knowledge. Therefore it follows that the ability to engage in discourse is more fundamental and primary than knowledge acquisition and language acquisition. So the problems of defining exhaustive ontologies and of embedding innate grammar in intelligent agents are actually one and the same. If an agent can engage in discourse, it can resolve both these issues by itself. That is where true sentience can be hypothesized to reside. But here we have referred only to the contents of discourse; in the next paragraph we explore its situational aspect.

We hypothesize that there exist innumerable numbers of discourses in the world, nested in a manner not unlike hierarchical tree-like data structures in computer science. That I am engaged in intense thought about the processes of cognition employing the rules of inference can be thought of as my discourse with myself. That I am typing out my thoughts in the form of sentences on my laptop is a larger discourse that encompasses the previous one. This, in turn, is part of an even larger discourse, namely, that of my intellectual pursuit along this particular line of inquiry in the disciplines of cognitive science. And this in turn is part of my academic endeavors in order to advance my career – an even larger discourse, say discourse D1. A sub-discourse of this D1 can be that I am associated with my institution engaged in higher studies and required to interact with other academics. Sub-discourses of this may include each of the projects that I am involved in with other academics. But this is only one way of imagining it. Say, engaged in my immediate discourse (that of writing this paper), I suddenly am asked by someone about a football match. In such situations, we 'switch contexts' or, as we put it, enter a different discourse. And having dealt with it, we return. The mind, it therefore can be conjectured, is not a Turing machine; it does not work on an endless sequence of input on an infinitely long tape. Nor is it a video analyzer which breaks all perceptions of the world into frames and arranges them along the temporal co-ordinate and then try to find patterns for recognition. All knowledge is centered on and arranged according to discourses.

A discourse, therefore, is a cognitive phenomenon that has as its focal point an object of attention and perception common to the agents engaged in it. In the discourses discussed above, the paper I'm writing now is an object of perception and subsequently, of discourse; likewise, the football match is another event-object that I and my friend have perceived, be it directly or indirectly, which forms the focal point of our discourse. So what generally confuses us as the 'train of thought' is actually the attention of the *ego-sense* flitting from object to object – our focus shifting from one discourse to another. (So we should maybe refer to it as the 'train of discourse' instead!)

Having said the above, we are now in a reasonable position to return to our model and explain what we meant by the term '*omniverse of discourse*'. The omniverse of discourse refers therefore to all the 'universes of discourse' that an agent has engaged in till now. It is hierarchically arranged and created bottom up (not top-down!). We mentioned before that the *lower mind* was a sense organ too and by dint of its being a sense organ it can therefore attract the *ego-sense's* attention onto itself. But by 'itself', we mean the omniverse of discourse itself, or more specifically, a particular discourse. These discourses, in our model, are not stored anywhere else; they are always actively awaiting the *ego-sense's* attention, connected to the *discriminant*, the *ego-sense* and the *lower mind*.

Our memories, it therefore can be inferred, are also arranged according to the discourses we engage ourselves in. Discrete perceptions (objects or words) serve as pointers to discourses. The perception of an object or a word, as we highlighted before, consists of a denotation and a connotation. The connotation acts as a *selector* to a particular discourse in the omniverse of discourse, not quite unlike the *chip-select* signal in a multiplexer. And discourses in turn serve as pointers to our recollections, not quite unlike indexes in databases. (Compare the psychology of the adage, "Out of sight, out of mind.")

The selection of a discourse therefore we have seen depends on the perception of an object. But the thoughts that ensue require linguistic expression. One cannot think, one cannot reason, one cannot express if one does not know how to. And on this issue, although we are not in a position to experimentally verify our claim, we can conjecture that the principle of linguistic relativity holds (if at all) only for the 'train of thought', once we have entered a particular discourse; but not for the 'train of discourse' which is guided by perception and devoid of the influences of language.

Ideas are not all that central to knowledge nor are concepts; the central and defining aspect and substratum of all thought and all knowledge therefore is *discourse*.

VII. Sentience and Thought

We have already discussed how thought is initiated from a discourse in the previous two sections. But how does one think? What sequence, if any, does one follow? Thought involves perceived objects; thought associates them, manipulates them, etc. The laws of logic and the rules of inference are not things which are innate. Some are implicitly acquired; some are explicitly acquired. But an association of discrete objects and the application of rules of logic and inference which are conceived, expressed and learnt through language require the same linguistic capability. Thought, as explained in detail before, is nothing but self-discourse – where the *ego-sense* is engaged in discourse with itself. The only difference is that in discourses involving multiple agents there is a possibility of the connotation distracting the agent's current domain of discourse or leading it to misinterpret the information being conveyed. In self-discourse, these two situations do not arise. We know exactly what we are thinking and we are fully focused on our thought. In our model described and discussed above, thought is a state of mind when the attention is on the *lower mind* (in effect, the omniverse of discourse) and not on the sensory input.

Sentience is a term often used in the contexts of artificial intelligent agents and also of science fiction. What constitutes the sentience of such an agent is often debated. Is an agent considered intelligent or sentient if it passes the

Turing test (Turing, 1950)? Or would the Chinese interpreter in Searle's famous thought experiment oppose the idea? (Searle, 1980) Would the chat-bots on the internet qualify as sentient? Would an agent be called sentient if it fooled a human into thinking it is human as well? Having described a discursive model of cognition, we would prefer to define sentience as the ability to engage in discourse. This definition, in our opinion, clearly distinguishes human beings and all manner of intelligent agents conceived, designed or built to this day.

VIII. Conclusion

"The prospect of representing everything in the world is daunting." (Russell & Norvig, 2003, pp. 348) Of the myriad problems arising from the epistemological issues confronting cognitive science, the knowledge representation problem is 'both formidable and central to the enterprise' (Stillings et. al., 1995). Through this work, an attempt has been made to pave the way for a better understanding of knowledge and its representation. It is hoped that this discursive hypothesis will aid in our understanding our knowledge representation and acquisition. It may be, as Stillings et. al. (1995) point out, that our hypothesis turns out to be wrong but we hope it aids in the formation of other more-close-to-true theories based on earlier attempts, if that be the case.

In the present work, we have explored the nature of cognition, the processes by which we perceive various objects and how knowledge may be acquired. In the course of our inquiry, we have found a striking significance of discourse in the process of cognition which has led us to conclude that it is one of the fundamental factors in knowledge acquisition and representation. As mentioned before, this work is a shorter version of an extended work currently in progress. This discursive model of cognition is part of a larger framework that links cognition, affection and conation together as a seamless whole. We hope to be able to discuss this larger framework in our future works.

If one looks closely about the world, one is sure to notice the influences of discourse on other spheres of life and other disciplines of the world. Books also fall into our category of discourses – a discourse between the author and all the readers. Discourses find their ways in religious texts often setting examples of faith, belief, tradition, ethical standards, etc. The world of logic, of argumentation and of dialectics is grounded in the world of discourse. In the age of the internet and the World Wide Web – where social networking, online forums, communication and interactions, etc. are widespread – one hardly needs to stress the universal nature of discourse. Discourses are everywhere.

Lastly, we draw an interesting (although slightly irrelevant) parallel from the Rig Veda – an ancient text of Indian origin – which frequently lauds and offers prayers to Vak, the ancient Vedic goddess of speech and utterance widely identified with the later Hindu deity Saraswati, the goddess of knowledge and creation. In spite of this highly fascinating note, in reality, whether or not discourse bears such a close relationship with knowledge and whether or not the ancient Indian seers were aware of such a connection or even its possibility, one can only wonder.

References

Stillins, N.,Weisler, S., Chase, C., Feinstein, M., Garfield, J. and Rissland, E., 1995. Cognitive Science: An Introduction. MIT Press.

Bechtel, W. and Graham, G., 1999. A Companion to Cognitive Science. Blackwell Publishing.

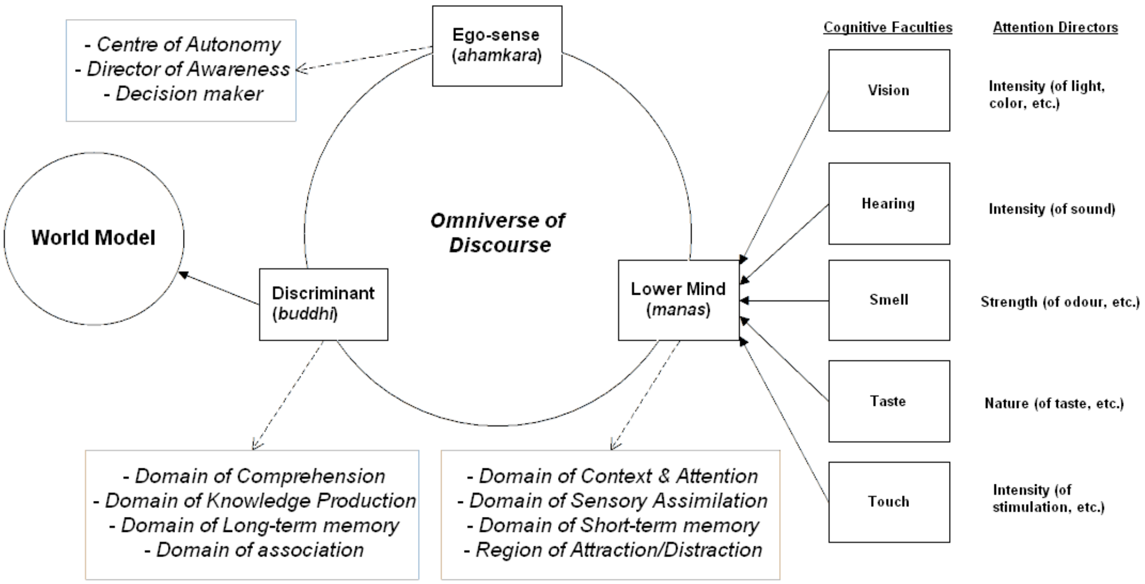
Radhakrishnan, S., 2008, orig. 1923. Indian Philosophy (Volumes I and II). Oxford University Press.

Sharma, C., 1987. A Critical Survey of Indian Philosophy. Motilal Banarsidass Publishers.

Bryant, E., 2009. The Yoga-Sutras of Patanjali. North Point Press.

Illustrations and Figures

A Generic Discursive Model of Cognition (Bhattacharyya, 2012)


- Centre of Autonomy
- Director of Awareness
- Decision maker
Ego-sense
(ahamkara)
Cognitive Faculties
Attention Directors
Vision
Intensity (of light, color, etc.)
World Model
Omniverse of Discourse
Hearing
Intensity (of sound)
Discriminant
(buddhi)
Lower Mind
(manas)
Smell
Strength (of odour, etc.)
Taste
Nature (of taste, etc.)
- Domain of Comprehension
- Domain of Knowledge Production
- Domain of Long-term memory
- Domain of association
- Domain of Context & Attention
- Domain of Sensory Assimilation
- Domain of Short-term memory
- Region of Attraction/Distraction
Touch
Intensity (of stimulation, etc.)

Comparison of Optimization Methods for L1-regularized Logistic Regression

Aleksandar Jovanovich
Department of Computer Science and Information Systems
Youngstown State University
Youngstown, OH 44555
aleksjovanovich@gmail.com

Alina Lazar
Department of Computer Science and Information Systems
Youngstown State University
Youngstown, OH 44555
alazar@ysu.edu

Abstract

Logistic regression with L1-regularization has been recognized as a prominent method for feature extraction in linear classification problems. Various optimization methods for L1 logistic regression have been proposed in recent years. However there have been few studies conducted to compare such methods. This paper reviews existing methods for optimization and then tests the methods over a binary dataset. Results are recorded and comparisons are made. After analyzing the results, the conclusion is that the GLMNET method is the best in terms of time efficiency.

Introduction

Digital information is growing at an extreme rate. Emerging technologies have created an environment that is information driven. From social media to medical records, data is collected in all forms from around the world. Current trends suggest a jump in information gathered and collected over the next decade and beyond. Never before has there been an abundance of data and information as we see today.

As the amount of data collected continues to grow so does the challenge of processing and gathering information. The data is growing wide, and the amount of attributes and features that can be derived sometimes outnumber the sample size. Now, more and more binary large objects are appearing in databases which require a different approach to identifying and extracting information.

Researchers have turned to regularized general linear models to form relationships about the binary data. Regularization is required to avoid over-fitting when there are a large number of parameters. In particular, L1-regularized regression is often used for feature selection, and has been shown to generate sparse models (Yuan, Chang, and Lin 2010).

Recently, there has been a large amount of research conducted to related regularization methods. Each method is differentiated by various aspects including: convergence speed, implementation, and practicability. Therefore, there is significance in conducting a thorough comparison and evaluation (Yuan, Chang, and Lin 2010). In this paper, we review prevailing methods for L1-regularized logistic regression and give a detailed comparison.

Background

Logistic regression is used for prediction of the probability of occurrence of an event by fitting data to a function. It is a generalized linear model used for binomial regression. Like other forms of regression analysis, it makes use of one or more predictor variables that may be either numerical or categorical. The logistic regression problem is an optimization problem, and can be solved by a wide variety of methods; such as gradient descent, steepest descent, and Newton. Once optimization is complete and maximum likelihood values are found, a prediction on the probability of the two possible outcomes can be made (Koh, Kim, and Boyd 2007).

The logistic model has the form:

$$\mathrm{Prob}(b|x) = \frac{\exp(b(w^T x + v))}{1 + \exp(b(w^T x + v))} \qquad [1]$$

Where $b \in$ (-1, +1) denotes the associated binary output and where Prob(b|x) is the conditional probability of b.

L1-regularized logistic regression has recently received attention. The main motivation is that L1-regularized logistic regression yields a sparse vector and has relatively few nonzero coefficients (Koh et al. 2007). A logistic model with sparse vectors is simpler and more efficient when dealing with data having a smaller number of observations than features. When compared to L2-regularized logistic regression, L1-regularized logistic regression outperforms L2-regularized logistic regression (Wainwright, Ravikumar, and Lafferty 2007).

The L1-regularized logistic regression problem minimizes the following equation:

$$l\text{avg}(v,w)+\text{l}||w||1=(1=m)\sum_{1}^{m} f(w^{T}ai+vbi)+\text{l}\sum_{1}^{m} ||\text{w}|| \quad [2]$$

Where λ > 0 is the regularization parameter. A solution must exist, but it need not be exclusive. The objective function in the L1-regularized Logistic regression problem is not differentiable so solving the problem is a computational challenge (Koh, Kim, and Boyd 2007).

A regularization path is the set of solutions obtained from L1-regularized linear regression problems while solving for λ. In many cases, the entire regularization path needs to be computed, in order to determine an appropriate value of λ. The regularization path in a smaller L1-regularized linear regression problem can be computed efficiently (Friedman, Hastie, and Tibshirani 2010). Hastie et al. describe an algorithm for computing the entire regularization path for general linear models including logistic regression models. Path-following methods can be slow for large-scale problems, where the number of observations is very large.

Optimization

Each method uses a type of optimization approach to find the regularization path as well as λ. The general model used in each method consists of iterations of the descent, where a chosen subset of variables is deemed the working set and all other variables become fixed. With every step the resulting sub-problem contains fewer variables and therefore solved easier.

Coordinate Descent Method

Typically, a coordinate descent method sequentially goes through all variables and then repeats the same process. By solving the regression problem along an entire path of values, this method efficiently calculates the regularization parameters (Friedman, Hastie, and Tibshirani 2010).

Generalized Linear Model with Elastic Net

GLMNET applies a shrinking technique to solve smaller optimization problems. GLMNET conducts feature-wise normalization before solving the optimization problem. Then, GLMNET measures the relative step change in the successive coordinate descent iterations (Yuan, Chang, and Lin 2010).

Continuous Generalized Gradient Descent

An effective regularization strategy in generalized regression is using validation methods to choose a suitable point in a trajectory or a family. Due to the use of gradient information, the number of iterations is less than cyclic coordinate descent methods. However, the cost per iteration is higher (Zhang 2007).

Least Angle Regression

LARS relates to the classic model-selection method known as Forward Selection (described in Efron, Hastie, Johnstone and Tibshirani 2004). Given a collection of possible predictors, a selection is made based on the largest absolute correlation with the response y. Thereafter simple linear regression is performed on the response y. This leaves a residual vector that can be considered the response. Projection is made over the other predictors orthogonally to the response. The selection process is then repeated. After n steps this results in a set of predictors that are then used to construct a n-parameter linear model.

Relaxed Lasso

Relaxo is a generalization of the Lasso shrinkage technique for linear regression. Both variable selection and parameter estimation is achieved by regular Lasso, yet both steps do not necessarily use the same penalty parameter. The results include all Lasso solutions but allow for sparser models while having similar predictive performance if many predictor variables are present. The package is based on the LARS package (Meinshausen 2007).

Datasets

All the experiments were done using the *Leukemia* dataset, a gene-expression data. This dataset was first mentioned in (Golub et al. 1999). The pre-processed dataset using methods from (Dettling, 2004) was used. The datasets consists of 72 genes that are part of two classes 0 and 1. There are 47 genes are from class 0 and 25 are from class 1.

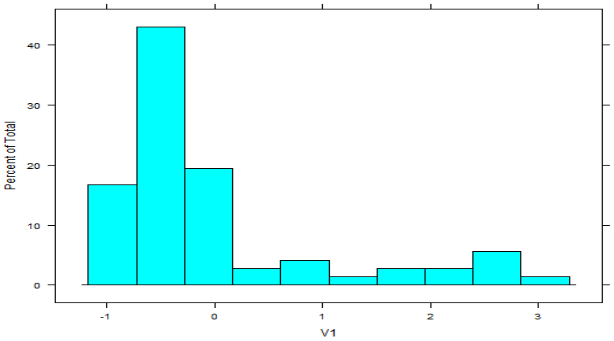


Figure 1. Histogram for Predictor Variable1

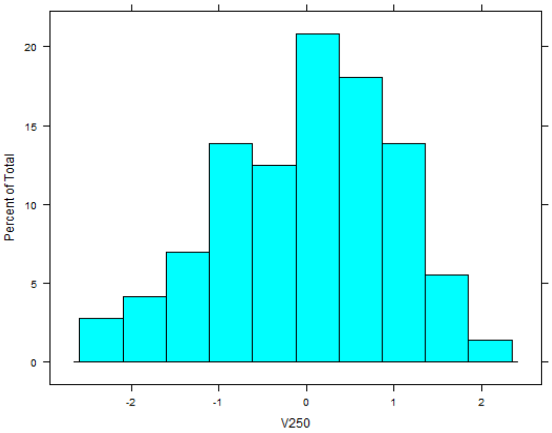


Figure 2. Histogram for Predictor Variable 250

There are 3,571 predictor variables that have numeric values in the interval [-10, 10] with most of the values close to 0.

The two figures above represent the histograms for two of the variables, the first one and the 250th one. More than 75% of the values of variable 1 are in the [-1, 0] interval. The values of variable 250 are normally distributed in the [-2.5, 2.5] interval.

Experiments

So far, we have described several large-scale optimization methods for solving L1-regularized logistic regression problems. In this section, we conduct experiments to investigate their individual and group performances. First we describe the experimental settings. Then the optimization methods are compared in terms of accuracy and time.

To be able to provide good predictions using the GLMNET algorithm, the regularized parameter λ has to be found first. That can be done in R using a grid search and functions from the caret package (Kuhn, 2012). First, the trainControl function is used to set the training parameters. Bootstrap sampling is done 25 times to increase the chance of getting high accuracy results.

```
model <- train(FL,data=trainset,method='glmnet',
          metric = "ROC",
          tuneGrid = expand.grid(.alpha=c(0,1),
          .lambda=seq(0.02,.4,length=20)),
          trControl=MyTrainControl)
```

The model is obtained by using the caret's train function. The search interval for λ is [0.02, .4] with a step of 0.02. Parameter α can take 2 values 0 or 1. For α = 0 and all λ values the AUC (area under the curve) is maximum at 0.992. These results are shown in Figure 3.

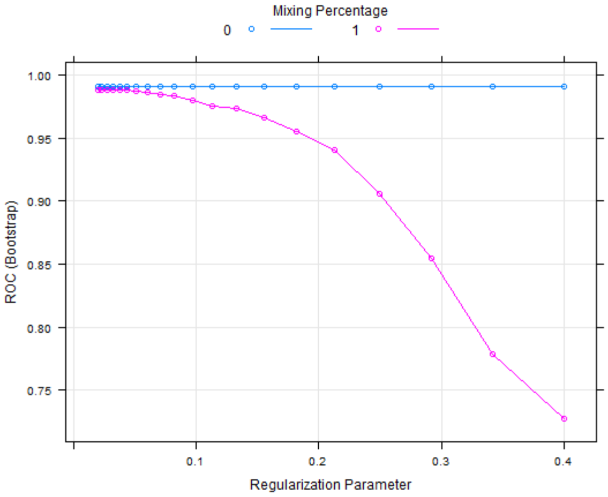


Figure 3.Glmnet ROC curve for the grids search

To run the experiments we used the GLMNET, CGGD, Relaxo, and LARS package in R. The LARS and Relaxo packages fit lasso model paths, while the GLMNET package fits lasso and elastic-net model paths for logistic and multinomial regression using coordinate descent. The algorithms are extremely fast, because they exploit sparsity in the data matrix. The CGGD is used for performing regressions while continuously varying regularization. The method returns the models fit along the continuous paths of parameter modification.

The coefficients from step 1 to 100 were recorded and their profile is plotted in figures 4, 5 and 6. Unfortunately we were unable to plot the coefficients of the Relaxo package.

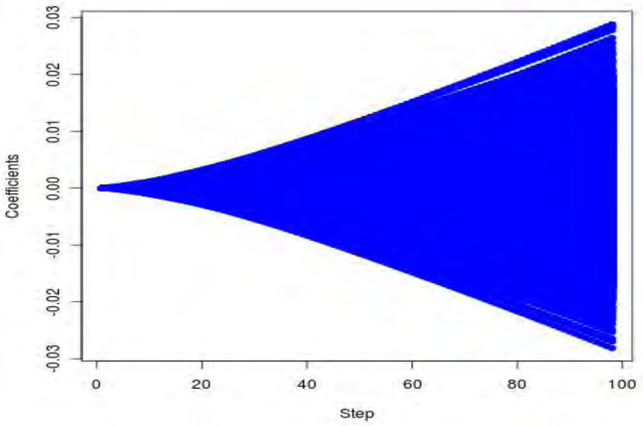


Figure 4: Profile of estimated coefficients for GLMNET method

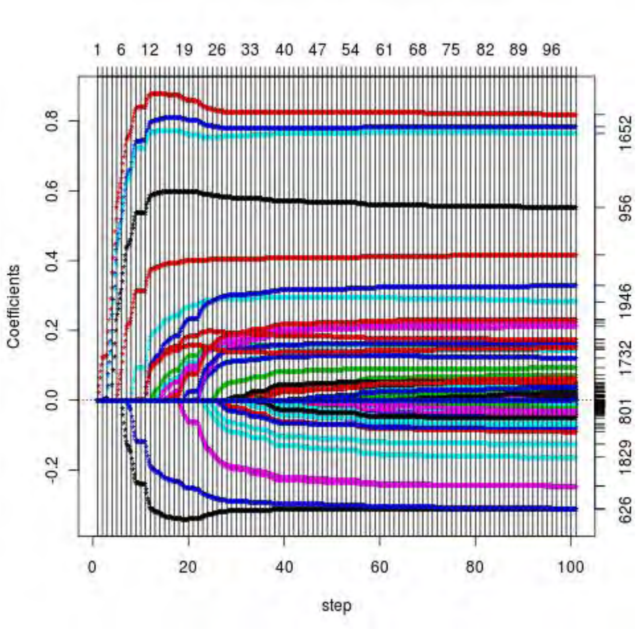


Figure 5: Profile of estimated coefficients for CGGD method

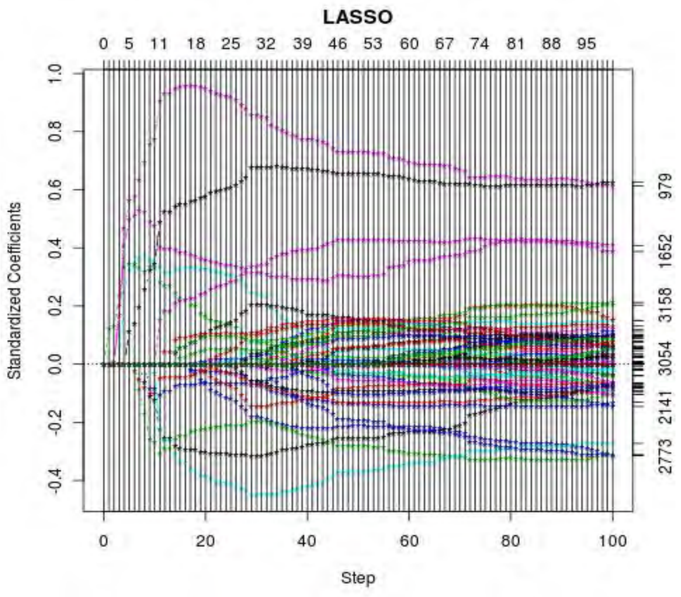


Figure 6: Profile of estimated coefficients for LARS method

10 Fold cross validation was used, and timings were recorded. Timing in seconds for GLMNET, CGGD, Relaxo, and LARS over Leukemia data is presented. The timings were performed on one HP TX2000 series laptop.

Optimization 100 Steps	
GLMNET	0.036s
Relaxo	0.064s
LARS	0.116s
CGGD	1.280s

Cross-validation 100 Steps	
GLMNET	0.420s
CGGD	1.38s
LARS	1.932s
Relaxo	4.076s

Conclusions

When compared, GLMNET is the more efficient algorithm. By the 100th step the predicted coefficients for GLMNET are stronger than both CGGD and LARS. When comparing the timings, GLMNET is almost 4 times as quick as CGGD in both optimization and cross validation. Relaxo is the almost twice as slow as GLMNET when comparing optimization and almost 10 times as slow when cross validating. We can conclude that the most efficient method for L1-regularized logistic regression is GLMNET. The Leukemia dataset has a larger number of features compare to the number of instances. Linear models work well with datasets with such characteristics. The data while large however contained a small number of samples. Testing over a dataset with a large sample and small feature should be further investigated.

References

Dettling M. 2004. BagBoosting for Tumor Classification with Gene Expression Data. *Bioinformatics*, 20, 3583-3593.

Efron, B.; Hastie, T.; Johnstone, I.; and Tibshirani, R. 2004. Least angle regression. *Annals of Statistics*, 32:407-499.

Friedman, J.; Hastie, T.; and Tibshirani, R. 2010. Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software* 33(1):1-22

Golub T.; Slonim D.K.; Tamayo P.; Huard C.; Gaasenbeek M.; Mesirov J.P.; Coller H.; Loh M.L.; Downing J.R.; Caligiuri M.A.; Bloomfield C.D.; Lander E.S. 1999. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science*, 286, 531-536.

Hastie, T.; Rosset, S.; Tibshirani, R.; and Zhu, J. 2004. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, *5*:1391–1415.

Koh, K.; Kim, S. J.; and Boyd, M. 2007. An Interior-Point Method for Large-Scale L1-Regularized Logistic Regression. *Journal of Machine Learning Research* 8:1519-1555.

Kuhn M. 2012. Package ‘caret’ http://cran.r-project.org/web/packages/caret/caret.pdf

Meinshausen N. 2007. Relaxed Lasso. *Computational Statistics and Data Analysis* 52(1), 374-393

Wainwright, M.; Ravikumar, P.; and Lafferty, J. 2007. High-dimensional graphical model selection using L1-regularized logistic regressionn. *Advances in Neural Information Processing Systems (NIPS) 19*.

Yuan, G. X.; Chang, K. W.; and Lin, C. J. 2010. A Comparison of Optimization Methods and Software for Large-scale L1-regularized Linear Classification. *Journal of Machine Learning Research* 11: 3183-3234.

Yuan, G. X.; Ho, C. H.; and Lin, C. J. 2011. . An Improved GLMNET for L1-regularized Logistic Regression. *In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Zhang, C. H. 2007. Continuous Generalized Gradient Descent. *Journal of Computational and Graphical Statistics.*

Fuzzy Logic and Systems

Chair: Anca Ralescu

Expert System for Security Audit Using Fuzzy Logic

K. Kozhakhmet, G. Bortsova, A. Inoue, L. Atymtayeva

Kazakh-British Technical University
Tole bi st., 59
Almaty, Kazakhstan
kanik85@gmail.com, gerdabortsova@gmail.com, inoueatsushij@gmail.com, l.atymtayeva@gmail.com

Abstract

Information security auditing plays key role in providing any organization's good security level. Because of the high cost, time and human resource intensiveness, audit expenses optimization becomes an actual issue. One of the solutions could be development of software that allows reducing cost, speeding up and facilitating Information Security audit process. We suggest that fuzzy expert systems techniques can offer significant benefits when applied to this area. This paper presents a novel expert systems application, an Expert System in Information Security Audit (ESISA).

Today organizations, facing with a wide range of potential threats to their information security (IS), are increasingly interested in high level of it. One of the best ways to estimate, achieve and maintain security of information is an Information Security auditing. Audit of security (broadly-scoped) is a complex, many-stage and labor-intensive process involving high-qualified specialists (experts) in IS, what makes it a quite expensive service. There are many types of audit, including certain security standards (e.g. ISO 27K) compliance audits.

Typically, information security audit is conducted in the following steps [1]:

1. **Scoping and pre-audit survey**: determining the main area of focus; establishing audit objectives.
2. **Planning and preparation**: usually generating an audit workplan/checklist.
3. **Fieldwork**: gathering evidence by interviewing staff and managers, reviewing documents, printouts and data, observing processes in action, etc.
4. **Analysis**: sorting out, reviewing and examining of the accumulated evidence in relation to the objectives.
5. **Reporting**: reviewing all previous stages, finding relations in the collected information and composing a written report.
6. **Closure.**

Each of the stages is accompanied with a large amount of information, which needs to be recorded, organized and, finally, analyzed.

One of the efforts taken in reducing expenses and facilitating audit is using of helping tools for identifying the gaps that exist between certain security standard and an organization's security practices, like checklists and questionnaires. For example, ISO 17799 Checklist ([2]) provides number of audit questions (like "Whether responsibilities for the protection of individual assets and for carrying out specific security processes were clearly defined."), each corresponding to particular section of the standard (4.1.3 for the previous example). ISO IEC 27002 2005 (17799) Information Security Audit Tool, described in [3], offers several hundred audit questions, stated in yes-no form (e.g. "Have you reduced the risk of theft, fraud, or misuse of facilities by making sure that all prospective employees understand their responsibilities before you hire them?"), pointing to security practices that need to be implemented and actions that should be taken (in case of "no" answer to question). So, the auditing may be viewed as a process of asking questions and analyzing answers to produce recommendations.

Of course, these tools are very useful to auditors and security related staff. But the questionnaires don't give an overall impression of organization IS level, entries of the checklists are too general (not concrete, not related to particular organization's actual policies, procedures, etc.). Such kind of disadvantages doesn't allow them to be used independently, without any additional security measurements.

Another step forward developing effective tools for audit is a knowledge base for Chief Information Security Officers (CISOs) assisting them in justifying their information security management decisions, presented in [4]. Key components of the base are: "Asset", "Source" (standard), "Vulnerability", "Step" (a refinement of part of a "Guideline" in particular standard) and others. Every "Step" is linked with an asset it protects, type of vulnerability it is against and also cross-references to other stored guidelines. The proposed tool provides a search in the knowledge base for guidelines in standards using their components.

That is, a sort of meta-model of security standards' recommendations could be constructed.

We think that considerable expenses accompany regular security audit of companies could be significantly reduced by intellectual software, capable of substituting human specialists in performing IS audit. This is a good field for application of artificial intelligence techniques, like expert systems.

Expert System in Information Security Audit

An expert system (ES) is a computer system that emulates the decision-making ability of a human expert. (Jackson 1998)

The knowledge in expert systems, commonly represented in form of IF-THEN type-rules, may be either expertise or knowledge that is generally available from written sources. [5] We think that in IS field, along with human knowledge, security standards' (ISO/IEC, COBIT and ITIL, in particular) recommendations can also serve as a source of expertise and may be translated into rules.

We consider implementing question-answer interaction between user and system, similar to checklist and questionnaire principle: ES will take user's answers on auditing questions, analyze them, and output a result in form of recommendations.

A little more detailed procedure of audit, performed by the expert system:

1. Company information acquisition: defining assets to be protected (equipment, data, etc.). Depending on this, the system will prepare some general questions to start from.
2. Process of obtaining information by the system from personnel by asking appropriate (possible in particular situation of the organization, described in stage 1) questions.
3. Expert system's logical inference.
4. The system produces the output as a list of recommendations.

In comparison with audit process described in previous section, looks much easier. I.e., our idea is to automate some stages of the audit. In our opinion, expert systems technique has much to offer in information security.

Some of advantages of the use of expert systems (according to [5]), particularly in IS field are:

- **Reduced cost.** Development of an expert system is relatively inexpensive. Taking into consideration an opportunity of repeated use by multiple organizations, the cost of the service per client is greatly lowered.
- **Increased availability.** Expert knowledge becomes available using any suitable device at any time of the day. Web-based expert systems open up ability to access expertise from any Internet connected device. In some sense, "expert system is the mass production of expertise." (Giarratano & Riley 1998)
- **Multiple expertise.** Using knowledge from multiple sources increases total level of expertise of the system. In case of IS, a combination of number of security standards' recommendations and knowledge of several independent specialists could be used.
- **Time saving.** IS auditing is a time consuming process. Expert systems at some phases of audit (analysis of gathered evidence, reporting) can save days (or weeks) by faster responding (in comparison with a human expert) and reducing amount of paper work.
- **Steady, unemotional, and complete response at all times.** By the use of programs, human factor influence decreases.

We believe that developing web-based Expert System in Information Security Audit (ESISA), from the first, practical side, will save time and money of companies-clients, and, from the second, theoretical side, it will be a good fundamental experience for further development of methodologies for applying Artificial Intelligence techniques in IS field.

Previously expert systems approach in security area was applied in computer security auditing. An Expert System in Security Audit (AudES), designed for automating some audit procedures, like identifying potential security violations by scrutinizing system logs, described in [6]. But the field of expert systems methodology application in information security audit in its broader sense, i.e. not only IT, (what actually we would like to implement) remains largely untouched.

Information security usually divides on administrative, physical and computer security. We're planning to involve each of those types in our system. If to be based on ISO 27K, some of issues, those will be included, are: asset management (corresponding chapter 7 of ISO), human resource security (8), communications and operations management (10), access control (11), incident management (13), etc.

But we decided to go further in increasing of our ES human thinking pattern emulation accuracy by adding uncertainty management ability, i.e. developing *fuzzy* expert system (expert systems using fuzzy sets and logic for inference) in IS auditing. The exploitation of the tolerance for uncertainty underlies the remarkable human ability to make rational decisions in an environment of imprecision. [7]

Handling Uncertainties

We think that the task of developing ES in broad scale audit requires methods, more sophisticated than classical expert systems. They don't capture all the aspects of complex procedures, such as security estimation, which involves so many factors.

In real life, people do not often think about problems in terms of crisp names and numbers, they constantly deal with wide range of uncertainties. This is also applicable to professionals, when they are solve problems. [8] The subjective judgment of experts produces better results than objective manipulation of inexact data [9].

Experts in information security usually operate with fuzzy terms, such as "sensitiveness" (e.g. when applied to information), "completeness" (job applicant's CV) and so on. To handle uncertainties like this we consider applying another Artificial Intelligence technique – fuzzy sets and logic – those are effective tools for approximate reasoning, in comparison with traditional methods.

Fuzzy inference method was already used in risk assessment field (it is described in [10] and will be discussed further), which itself contains great value of uncertainties. We can use it for information security risk management, which is necessary in audit.

In information security, likely in every field, where humans are involved, things like perception take place. For example, auditor asks from user: “How frequently do you change your password?” He doesn’t expect answers like “often”, “rarely”, because usually people’s perceptions differ; furthermore, user may have distorted concept about information security. Here auditor perception is more adequate than auditees’ perception. A numerical value (e.g. password changes per month) would be absolute, independent and, therefore, more sufficient answer. Fuzzyfication is performed on expert’s side (he decide, if it is often, rare, etc.).

Of course, fuzzy logic and sets approach is advantageous here. The need of fuzzy logic is going to be proved in this paper.

System Modeling (Framework)

The following terms play a key role in organization’s information security assessment [1, 4, 11]:

- **vulnerabilities**: any weaknesses in the system of controls that might be exploited by threats;
- **threats**: generally people, things or situations that could potentially cause loss;
- **impacts**: what would be the (worst case) effects if some of those threats actually materialized.

In order to perform qualified security estimation, an auditor should think carefully about each of those things.

We decided to follow their thinking pattern and define assets of the organization, vulnerabilities that may exist, threats, particular harm could be inflicted to them, and also consider the impacts of those threats.

On the scheme below (Figure 1) you may see these categories organized in 3 layers, several samples for each are given (they are going to be discussed further).

There are two asset types: physical assets (for example, computers, servers, etc.) and information (e.g. employees’, clients’ data stored in databases), which are to be protected. (ISO/IEC 27002, 7.1.1, “Inventory of assets”)

Each of the assets matches one or more vulnerability it may have, each of the vulnerabilities is influenced by several factors (white boxes in 2nd layer), and may cause particular threat(s) materialization with some possibility. For example, physical security weakness, like poor physical entry controls (ISO/IEC 27002, 9.1.2, “Physical entry controls”) depends on proper use of authentication controls and good monitoring, monitoring in turn depends on turnover rate on guard’s position and their background checks; this weakness may become a cause of physical assets damage, or sensible data theft, or both.

Because the possibility of something to happen, especially an IS event, is very hard to evaluate precisely, it should be represented as fuzzy term. In order to calculate overall possibility, all factor’s impacts should be taken into account.

Impact of vulnerability on the particular threat is reflected in rules, which have the following pattern:

```
IF vulnerability is very serious, THEN threat execution possibility is low/moderate/high (fuzzy value).
```

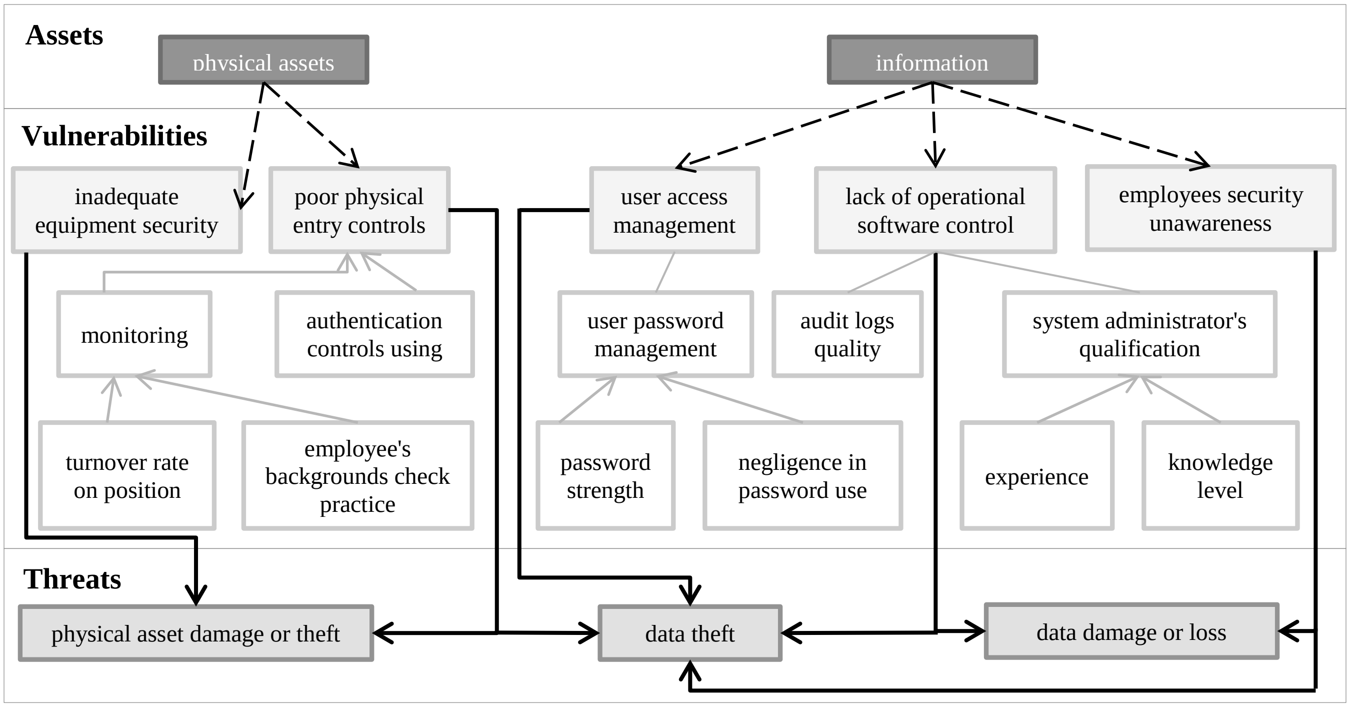

Assets
physical assets
information
Vulnerabilities
inadequate equipment security
poor physical entry controls
user access management
lack of operational software control
employees security unawareness
monitoring
authentication controls using
user password management
audit logs quality
system administrator's qualification
turnover rate on position
employee's backgrounds check practice
password strength
negligence in password use
experience
knowledge level
Threats
physical asset damage or theft
data theft
data damage or loss

Figure 1: Audit scheme

For example:

```
IF an equipment security is *inadequate*, THEN physical asset damage or theft *slightly* increases.
IF a physical entry control is *poor*, THEN physical asset damage or theft *greatly* increases.
```

(ISO/IEC 27002, 9.2, “Equipment security”, 9.1, “Secure areas”)

According to this principle, there will be as many rules with same consequence (differentiating by degree of severity), as many vulnerabilities influence this threat. In order to produce one value, corresponding fuzzy numbers are summed and divided by maximal numbers * quantity (e.g. *high* fuzzy number).

We can also consider impacts of materializing of these threats in money equivalent, in order to perform some risk assessment.

There is no such thing as an “exact” value of risk. Risk assessment is based on imprecisely defined inputs: the likelihood of the threat being exercised against the vulnerability and the resulting impact from a successful compromise. For example, in [10], Security Management System robustness (with values inadequate, good and excellent) and severity of consequence (category of health harm from 1 up to 5) of incident on the industry are taken as the input, the value of the risk (negligible, low, moderate, high and unacceptable) is the output.

In our system risk could be calculated in the same way, as a function of likelihood of the threat, found as summation of vulnerabilities impact rates, and size of possible impact in money equivalent. According to these risks, factors, those lower security, may be sorted and recommendations are given labeled with a requirement level.

According to Brander [12], we can use keywords in our recommendation reports of expert system like "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY" and "OPTIONAL", which could be implemented to fuzzy variables. These keywords can deeply and clearly show recommendation’s priority and notification manner.

Once common methodology is defined, one of the issues that arise is how standards’ recommendations will be translated to the rules and what kind of inputs the system will gather.

Knowledge Acquisition and Inputs

The simplest case for inputs is numerical values: this is, for example, turnover rate (per cent of employees substituted on particular position during a year), employee’s experience (years). Let’s look at example.

According to ISO 27002 “Control of operational software” (12.4.1) “a” step, “the updating of the operational software, applications, and program libraries should only be performed by trained administrators upon appropriate management authorization”.

It cannot be directly figured out if employee is *trained* or not trained enough. Admin’s qualification depends on his/her experience and knowledge. Experience may be retrieved as simply a numerical value. Of course, knowledge of human, even in restricted field, cannot be assessed by asking one question.

Not only direct values asking need may appear. For example, some test could be provided:

1. When setting permissions in NTFS for an individual's network drive, which option(s) of the following levels do you give a default user?

Answers: Full Control / Modify / Read & Execute / Read / Write.

2. What do administrative shared folder names always end with?

Answers: # / $ / @ / % / ~

3. Which one of the following is equal to 1 kilobyte (KB)?

Answers: 512 bytes / 1000 bytes / 1024 bytes / 1028 bytes / 2048 bytes.

4. etc.

The score, expressed in %, is also a fuzzy variable.

Sample rule, displaying system administrator’s experience, knowledge level and qualification:

```
IF employee is sufficiently experienced AND score is very high THEN employee is well qualified.
```

Use of tests exists in many aspects, like User security awareness (8.2.2, “Information security awareness, education, and training”) estimation. Each member of particular user group may be offered to answer some questions like (multiple choice test, one answer is correct):

1. What is true?

- Leave terminal logged in is a bad security practice; (correct)
- Frequent logging in and logging out leads to computer’s hardware faster deprecation;
- Logging out when leaving a work place is a good corporate culture indicator;
- Constantly logging in and out is time consuming.

2. Do you use your personal laptop at work? If no, do you want to?

- No, I think it’s reasonable expense; (correct)
- No, I don’t want to buy my own;
- Yes, it is convenient;
- Yes, personal laptop is a secure decision.

All scores of the group could be combined in one value (average score), expressed either in %, or a number from 0 to 1.

But some of variables that not explicitly expressed in numbers could be still obtained using 1 question. It refers to a situation when a particular quantity consists of several simple (true-false valued) weighted components, it could be calculated as a checklist.

Let's consider one of the aspects of user access management issue, password managing, as an example.

ISO/IEC 27002 "Password use" (11.3.1):

"All users should be advised to:

a) keep passwords confidential;
b) avoid keeping a record (e.g. paper, software file or hand-held device) of passwords, unless this can be stored securely and the method of storing has been approved;
c) change passwords whenever there is an y indication of possible system or password compromise;
d) select quality passwords with sufficient minimum length which are:
e) easy to remember;
f) not based on anything somebody else could easily guess or obtain using person related information, e.g. names, telephone numbers, and dates of birth etc.;
g) not vulnerable to dictionary attacks (i.e. do not consist of words included in dictionaries);
h) free of consecutive identical, all-numeric or all-alphabetic characters;
i) change passwords at regular intervals or based on the number of accesses (passwords for privileged accounts should be changed more frequently than normal passwords), and avoid re-using or cycling old passwords;
j) change temporary passwords at the first log-on;
k) not include passwords in any automated log-on process, e.g. stored in a macro or function key;
l) not share individual user passwords;
m) not use the same password for business and non-business purposes."

These guidelines could be clearly divided into two parts: concerning user's negligence in password managing and password strength. Password security variable (which is going to be computed) represents a possibility of password to be stolen (number from 0 to 1), which can take values, say, high, moderate, or low and depends on two parameters, mentioned in previous sentence.

At first, we will try to compose some questions for user about how he/she manages his/her passwords: one question for one variable to be retrieved.

Question 1. Mark points you think are true for you:

- My colleagues/family members/friends or somebody else know my password. :0.2
- I consider writing down my logins and passwords on paper, storing them in files, or let my browser remember them very convenient way not to forget my passwords. :0.15
- If something suspicious happens, I don't think it is necessary to immediately change my password. :0.25
- I don't change my password without any serious reason, my memory is not so good to remember all this stuff.
- I use a default password, I think it is strong enough. :0.25
- I advocate a use of same password in multiple services. :0.15

The exact value of negligence level in password using is computed as a sum of coefficients for all points that were matched as true (value from 0 to 1).

Question 2. My password normally:

- is difficult to remember
- is a default password, like password, default, admin, guest, etc. :0.2
- contains dictionary words, like chameleon, RedSox, sandbags, bunnyhop!, IntenseCrabtree, etc. :0.1
- consists of words with numbers appended: password1, deer2000, john1234, etc. :0.15
- is one of common sequences from a keyboard row: qwerty, 12345, asdfgh, fred, etc. :0.3
- contains personal information, like name, birthday, phone number or address. :0.15
- contains symbols such as (mark each):
  - Lowercase letters (26)
  - Uppercase letters (26)
  - Numbers (10)
  - Punctuation marks (5)
- has average length: (specify number of characters)
- (not an option: using 2 previous options number of possible combinations of characters is calculated as (summary number of symbols)^(length); coefficient for this question is 0.1 if combination is bigger than 10^12, and combinations number / 10^12 * 0.1 else)

Value for the question is calculated as sum of coefficients of all entries.

These two values (value of negligence level and for the password strength) are subjects to fuzzyfication into fuzzy subsets like weak, good, strong for password's strength and low, moderate, high for negligence (a sample of fuzzy sets you can see at Figure 2).

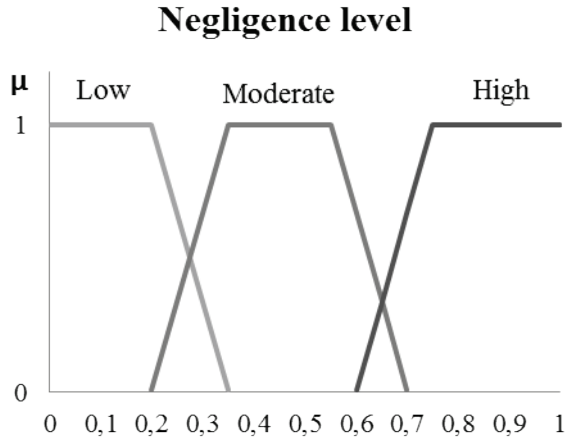


Figure 2: Negligence level fuzzy set sample

The password security also could be high (H), low (L) and moderate (M). We outline it in table on the next page.

Negligence \ Strength	weak	good	strong
low	M	H	H
moderate	L	M	H
high	L	L	M

Table 1: User account's break likelihood

On this table basis fuzzy rules could be composed as following:

```
IF negligence IS low AND password IS strong, THEN password security IS high.
IF negligence IS low AND password IS good, THEN password security IS high.
IF negligence IS low AND password IS weak, THEN password security IS moderate.
IF negligence IS moderate AND password IS strong, THEN password security IS high.
IF negligence IS moderate AND password IS good, THEN password security IS moderate.
IF negligence IS moderate AND password IS weak, THEN password security IS low.
IF negligence IS low AND password IS strong, THEN password security IS moderate.
IF negligence IS low AND password IS good, THEN password security IS low.
IF negligence IS low AND password IS weak, THEN password security IS low.
```

We performed a baseline audit [13] before the password policy changes and two follow-up password audits in the course of implementation. The results are following: The base line audit cracked results were 91%, and after recommendations it is decreased to 57%.

To summarize, data could be retrieved from user in various ways, including directly asking for an exact value, using "checklists" and test. We think that in some cases the use of fuzzy sets as the input would be also efficient.

Size of the possible loss of an organization in case of materializing of particular threat may serve as a good example. In qualitative risk analysis, the impact in money equivalent is usually treated as low, moderate and high. It could be retrieved as user constructed fuzzy set, like *about 10000$* (i.e. more precise than those 3 levels, but less exact than numeric value).

Conclusion

Fuzzy logic methodology provides a way to characterize the imprecisely defined variables, define relationships between variables based on expert human knowledge and use them to compute results. Fuzzy expert system applied to information security field is sufficient technique for emulating specialist's decision-making ability.

Also, one of the advantages of fuzzy expert systems is that the rules can be written in natural language, rather than in computer jargon. As a consequence, communication between domain expert and knowledge engineer is greatly simplified.

In conclusion, we claim that there are enough unexplored areas and bright intersections in implementing expert systems in security auditing, development of fuzzy based knowledge base for expert systems, integration of fuzzy coefficients in development of recommendations for security auditing, and etc. Theoretical significance of researches above has been presented in authors' publications [14]. This paper is actually a part of whole scientific research, touched approaches and several issues of implementing fuzzy logic in problems of information security auditing and development of fuzzy expert systems. It is obvious that this kind of research directions could be a good scientific fundamental in artificial intelligence area.

References

1. Hinson, G. 2008. Frequently Avoided Questions about IT Auditing - http://www.isect.com/html/ca_faq.html

2. Val Thiagarajan, B.E. 2002. BS 7799 Audit Checklist. - www.sans.org/score/checklists/ISO_17799_checklist.pdf

3. ISO IEC 27002 2005 Information Security Audit Tool - http://www.praxiom.com/iso-17799-audit.htm

4. Stepanova, D., Parkin, S. and Moorsel, A. 2009. A knowledge Base For Justified Information Security Decision-Making. In 4th International Conference on Software and Data Technologies (ICSOFT 2009), 326–311.

5. Giarratano, J., and Riley, G. eds. 2002. Expert Systems: Principles and Programming. Reading, Mass.: PWS Publishing Company.

6. Tsudik, G. and Summers, R. 1990. AudES - an Expert System for Security Auditing. IBM Los Angeles Scientific Center.

7. Zadeh, L. 1994. Fuzzy Logic, Neural Networks, and Soft Computing.

8. Siler, W., Buckley, J. eds. 2005. Fuzzy Expert Systems and Fuzzy Reasoning. Reading, Mass.: Wiley-interscience.

9. Borjadziev, G., Borjadziev, M. eds. 1997. Fuzzy Logic for Business, Finance, and Management. Reading, Mass: World Scientific.

10. Mahant, N. 2004. Risk Assessment is Fuzzy Business—Fuzzy Logic Provides the Way to Assess Off-site Risk from Industrial Installations. Bechtel Corporation.

11. Elky, S. 2006. An Introduction to Information Security Risk Management. SANS Institute.

12. Bradner, S. 1997. Key words for use in RFCs to Indicate Requirement Levels. IETF RFC Repository. http://www.ietf.org/rfc/rfc2119.txt

13. Williams, M. 2003. Adventures in implementing a strong password policy. SANS Institute.

14. Atymtayeva, L., Akzhalova, A., Kozhakhmet, K., Naizabayeva, L. 2011. Development of Intelligent Systems for Information Security Auditing and Management: Review and Assumptions Analysis. In Proceedings of the 5th International Conference on Application of Information and Communication Technologies, Baku, Azerbaijan, pp.87-91.

Fuzzy Logic-based Robust Control of a Flexible two-mass System (1990 ACC Benchmark Problem)

Adekunle C. Adediji and John Essegbey

School of Electronics and Computing Systems
University of Cincinnati
497 Rhodes Hall
Cincinnati, Ohio 45221-0030

Abstract

In intuitive design steps, a fuzzy logic-based robust controller is designed to address the first 1990-1992 American Control Conference benchmark problem. Using a conceptual transformation of the original flexible body into a perpetual rigid body mode, a final design which succeeds in stabilizing the system after a unit impulse disturbance is developed. The simulation results are shown to achieve and exceed the required design specifications of the benchmark problem, as well as those of other fuzzy logic-based solutions.

Introduction

As the complexity of engineered systems increased, it became imperative that the American Controls Conference (ACC) adopt a set of control design problems as robust control benchmark problems. This has led to several attempts by authorities in the field to come up with the best possible solutions, serving as a good basis for comparing the various heuristics and methodologies in designing for robust control. One of these problems, referred to by (Wie and Bernstein 1992) as ACC benchmark problem 1, was concerned with vibration control of a two-mass system with an uncertain spring constant (Figure 1). The flexible tow-mass system addresses, primarily, a disturbance rejection control problem in the presence of parametric uncertainty. This problem has been addressed in over 30 papers, including papers in special issues of the Journal of Guidance, Control and Dynamics and the International Journal of Robust and Nonlinear Control (Linder and Shafai 1999).

Probably due to the linearity of this problem, most published solutions have appropriated linear controllers of some sort, from H-infinity to game theory. (Niemann et al 1997) applied the μ-synthesis method for mixed perturbation sets using a modified D-K iteration approach, while (Wie and Liu 1992) proposed a solution using the H_∞ controller design methodology. In addition, (Farag and Werner 2002) compared the performance of his robust H_2 design with a collection of existing controllers such as Pole Placement, and Minmax Linear Quadratic Gaussian (LQG). (Hughes and Wu 1996) also presented an observer-based extension of a passive controller design, due to the fact that strictly passive feedback could no longer guarantee stability for the given problem.

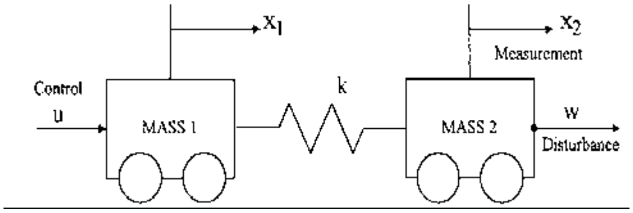


Figure 1: The ACC benchmark problem consisting of a dual mass, single spring system.

Some recent solutions, however, make use of qualitative approaches capitalizing on fuzzy reasoning, which have been shown to perform just as good as or even better than the existing quantitative methods (Cohen and Ben Asher 2001). It is worth noting that the presence of design constraints, and plant, as well as parameter uncertainties, drastically increases the complexity of modeling plant behavior, and makes the application of non-linear solutions worthwhile.

In this paper, we build on a solution using fuzzy logic. We start by generating a detailed model of the system and highlight the required design objectives for the controller. Next we obtain a reduced or simplified model of the system in the rigid-body mode, where spring oscillations have been effectively damped out using fuzzy logic heuristics (Linder and Shafai 1999). Finally, an additional fuzzy controller produces a superimposition of stability and tracking behaviors to ensure the achievement of stated design objectives.

Problem Description and Modeling

The benchmark plant shown in Figure 1 consists of two masses connected via a spring, with the following characteristics.

1) The system has a non-collocated sensor and actuator; the sensor senses the position of m_2 while the actuator accelerates m_1. This introduces extra phase lag into the system, making control of the plant difficult (Cohen and Ben Asher 2001).
2) The system is characterized by uncertainties in the temporal plant (spring constant that varies within a very wide range)
3) The system exists in both the flexible body mode (due to the spring) and rigid-body mode (when relative movements due to the spring are damped out).

For the above system, consider a simplification, where $m_1 = m_2 = 1$ and $k = 1$ with the appropriate units. A control force acts on body 1 (m_1), and x_2, which is the position of body 2 (m_2), is instead measured thus resulting in a non-collocated control problem. The state space representation of the system is given as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -k/m_1 & k/m_1 & 0 & 0 \\ k/m_2 & -k/m_2 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1/m_1 \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/m_2 \end{bmatrix} w$$

$$y = x_2$$

where x_1 and x_2 are the positions of body 1 and body 2, respectively; $x_3$and x_4 are the velocities of body 1 and body 2, respectively; u is the control input acting on body 1; y is the sensor measurement, w is the disturbance acting on body 2, and k is the spring constant. The transfer function representation is

$$T_{uy} = \frac{(k/m_1 m_2)}{s^2[s^2 + k(m_1 + m_2)/m_1 m_2]}$$

and the corresponding transfer function between a disturbance to $m_2$and plant output is

$$T_{wy} = \frac{(1/m_2)(s^2 + k/m_1)}{s^2[s^2 + k(m_1 + m_2)/m_1 m_2]}$$

This paper considers only problems 1 and 2 as described by (Wie and Bernstein 1992) and ignores the effect of sensor noise (full state feedback) and disturbance acting on body 1. The constant-gain linear feedback controller design requirements are stated as

1. The closed-loop system is stable for $m_1 = m_2 = 1$ and $0.5 < k < 2.0$.
2. The disturbance w(t)=unit impulse at t=0 and y has a settling time of 15sec for the nominal plant parameters $m_1 = m_2 = 1$ and $k = 1$.
3. Reasonable performance/stability robustness and reasonable gain/phase margins are achieved with reasonable bandwidth.
4. Reasonable control effort is used.
5. Reasonable controller complexity is needed.
6. Settling is achieved when y is bounded by ± 0.1 units.

This problem addresses, primarily, a disturbance rejection control problem in the presence of parametric uncertainty. The plant has eigenvalues at $(\pm j \sqrt{(k(m_1 + m_2)/(m_1 m_2))}, 0{,}0)$, and a single-input/single-output (SISO) controller must close its loop around T_{uy}, which has a pole-zero surplus of four (Stengel and Marrison 1992).

Robust Design Solution using Fuzzy Logic

Fuzzy logic controller design was first started by (King and Mamdani 1977) on the basis of the fuzzy logic system generalized from the fuzzy set theory of (Zadeh 1965). It has gained wide practical acceptance providing a simple, intuitive, and qualitative methodology for control (Jamshidi, Vadiee, and Ross 1993), (Yen, Langari, and Zadeh 1992), (Zadeh 1994). In a typical implementation, a fuzzy controller consists of a set of if-then rules, where the controller output is the combined output of all the rules evaluated in parallel from the antecedents of the inputs. The inference engine, of a fuzzy logic controller, plays the role of a kernel that explores the fuzzy rules pre-constructed by experts to accomplish inferences.

Since the rules specify the implication relationships between the input variables and output variables characterized by their corresponding membership functions, the choice of the rules along with the membership functions makes significant impacts on the final performance of the controller and therefore becomes the major control strategy in Fuzzy Logic Controller design.

Common classifications of fuzzy controllers include fuzzy Proportional Integral Differential (PID) controllers, fuzzy sliding-mode controllers and fuzzy gain scheduling controllers (Driankov, Hellendoom, and Reinfrank 1996), (Jang and Sun 1995). Even though all three categories realize closed-loop control action and are based on quantitative control techniques, the first and second are implementations of the linear quantitative PID controller and a nonlinear, quantitative sliding-mode controller. The last category, however, utilizes Sugeno fuzzy rules to interpolate between several control strategies, and are suitable for plants with time varying or piecewise linear parameters (Jang and Sun).

A. Fuzzy Logic for Benchmark

For the robust control problem described above, plant stabilization is required first before performance objectives. Ensuring stability, however, entails the dampening of vibrations after an external disturbance is applied. (Linder and Shafai 1999) described an approach using Qualitative Robust Control (QRC) methodology, where stability and tracking behaviors are separately developed, and the superimposition of these behaviors achieves the final control objective. These behaviors exploit the rigid body mode of the plant, where the plant behaves as if the masses are rigidly connected. The stability behavior is derived from the heuristic that a control action is more effective in suppressing plant vibration if it is applied when the spring is neutral, and the control action opposes the motion of the spring.

Using fuzzy logic, a process model of the spring, needed to provide the qualitative state information that dampens plant vibrations and achieve stability, is achieved by abstracting the system to a state that indicates whether the spring is at its neutral length and whether the spring is in the process of compressing or elongating. In modeling the spring, the length of the spring and its rate of stretching or contraction are used as input parameters and the output, its state. The process utilizes a qualitative spring state that is specified by a qualitative partition of the spring length $L = x_2 - x_1$ and the spring length velocity $\dot{L} = \dot{x}_2 - \dot{x}_1$. These parameters are partitioned using five membership functions as shown in Figure 2. A Mamdani Fuzzy Inference System (FIS) applies 25 rules, shown in the Fuzzy Association Memory (FAM) of Figure 3, to infer the qualitative spring state from inputs L and $\dot{L}$. The fuzzy controller is developed using the minimum operator to represent the “and” in the premise, and the Center of Gravity (COG) defuzzification as the implication.

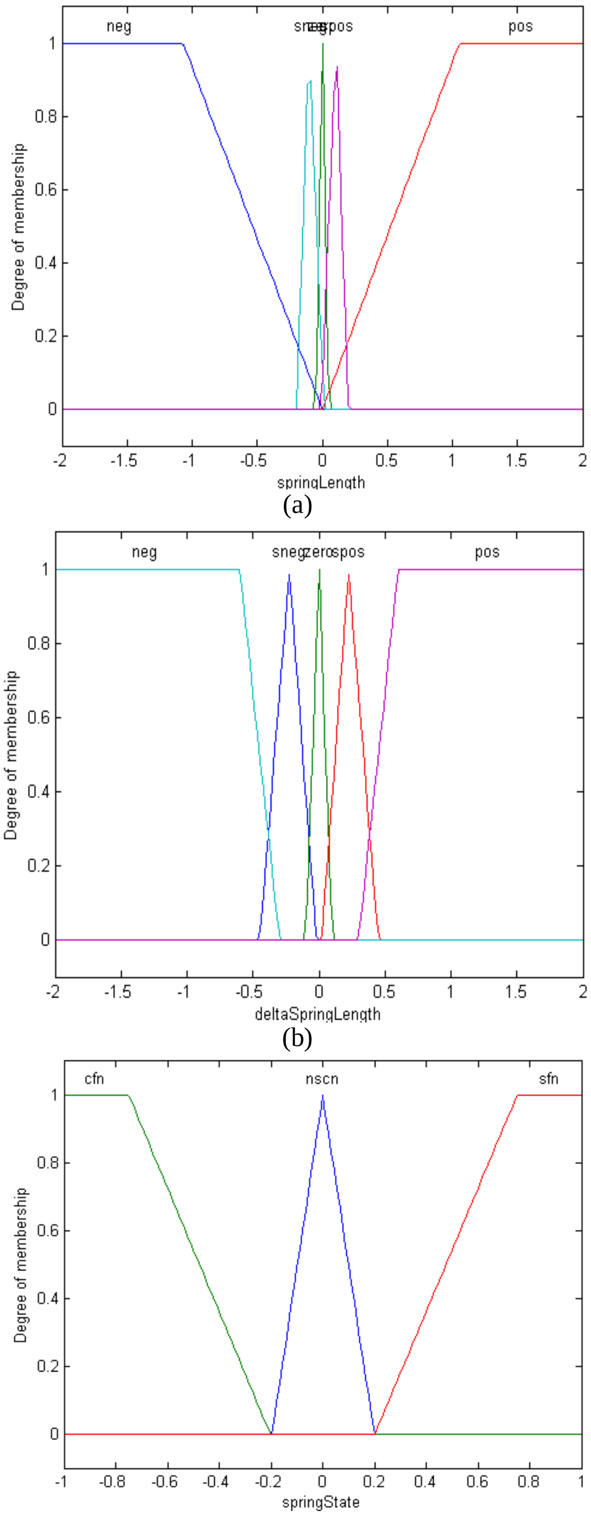
The qualitative behavior of the spring is based on a sense of direction and rate. Thus the parameters are defined on a bivalent range or universe of [-1, 1], and the outputs are described as follows;

NSCN: Not Stretching or Compressing with Neutral spring
CFN: Compressing Fast with Neutral spring
SFN: Stretching Fast with Neutral spring

The decision surface of Figure 4 is such that a vibration is observed when L is Small_positive or Small_negative, and $\dot{L}$ is Negative_large or Positive_large. A similar situation occurs when L is Zero and $\dot{L}$ is Small_negative or Small_positive.

B. State Observers

The above model is possible only if the states of the masses can be observed or correctly estimated. Due to the



(a)

(b)

(c)

Figure 2: Fig. 2(a) Membership functions of Spring Length $L = x_2 - x_1$ Fig. 2(b) Membership functions of the velocity of spring contraction or stretching $\dot{L} = \dot{x}_2 - \dot{x}_1$ Fig. 2(c) Membership functions of the output, spring state.

springLength \ deltaSpringLength	neg	sneg	zero	spos	pos
neg	nscn	nscn	nscn	nscn	nscn
sneg	cfn	nscn	nscn	nscn	sfn
zero	cfn	cfn	nscn	sfn	sfn
spos	cfn	nscn	nscn	nscn	sfn
pos	nscn	nscn	nscn	nscn	nscn

Figure 3: Fuzzy association memory of the spring model.

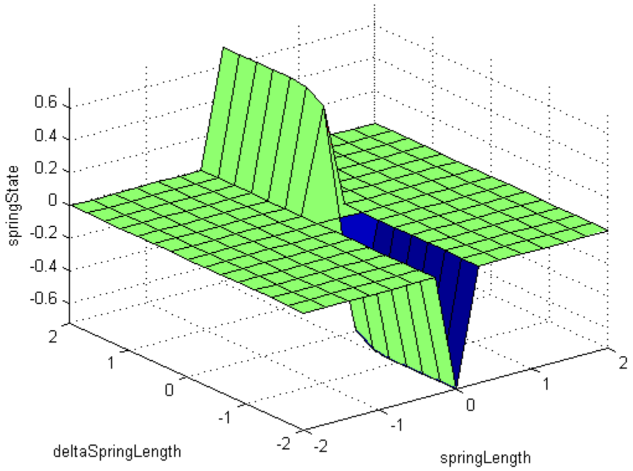


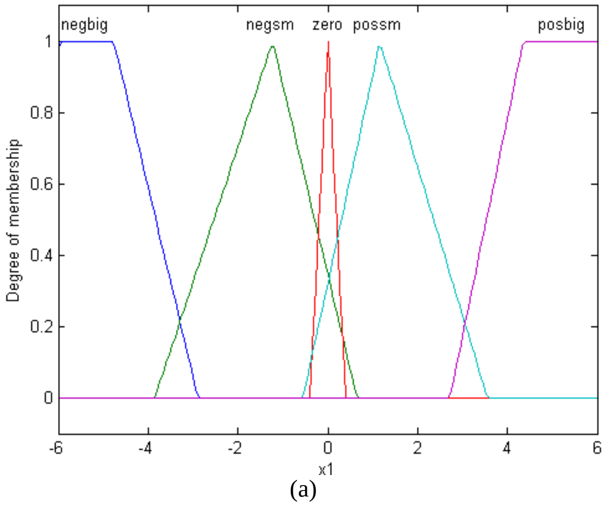
Figure 4: Output surface of the spring fuzzy process model.

non-collocated nature of this problem, designing for robust disturbance rejection requires the use of state observers to model disturbances and other uncertainties, such as position of the masses. In the deterministic case, when no random noise is present, the Luenberger observer and its extension may be used for time-invariant systems with known parameters. When parameters of the system are unknown or time varying, an adaptive observer is preferred. The corresponding optimum observer for a stochastic system with additive white noise processes, with known parameters, is the Kalman filter. As indicated earlier, this project assumes full state feedback of masses 1& 2.

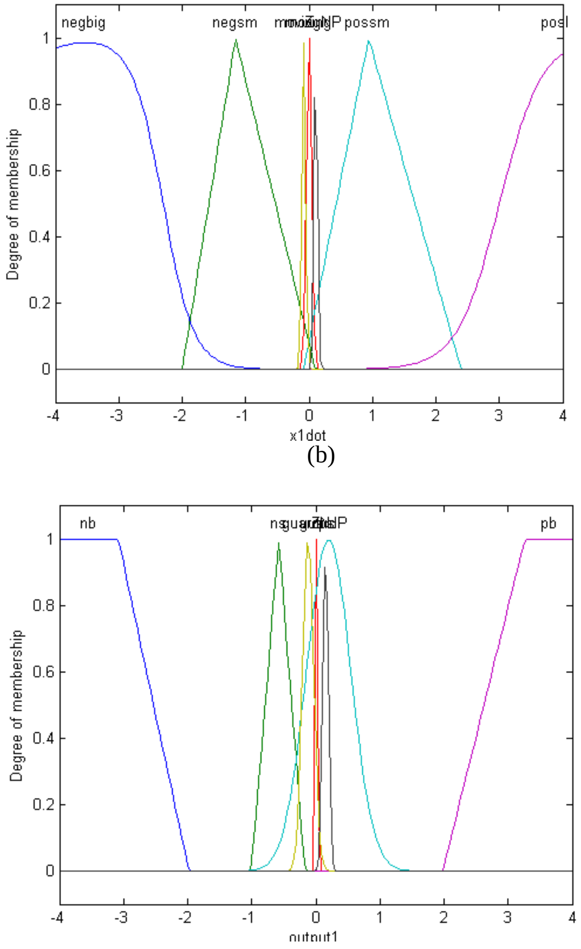
C. Robust Tracking and Stability

With the system in a rigid-body mode, due to the damping effects on the interconnecting spring, it is evident that the position and velocity of body 2, x_2 and $\dot{x}_2$, are fixed relative to body 1. Hence, measuring $\dot{x}_1$ gives us $\dot{x}_2$, while the displacement of x_1 from its initial position at rest is equivalent to the displacement of x_2 from its own initial position. Essentially, the problem has been reduced to one that can be solved with a collocated controller on body 1. In robust control, collocation guarantees the asymptotic stability of a wide range of SISO control systems, even if the system parameters are subject to large perturbations, while also enabling the achievement of desired performance objectives.

We also use an additional Mamdani fuzzy controller which receives the position and velocity of body 1, x_1 and $\dot{x}_1$, as inputs and outputs an appropriate control action. This output is superimposed directly on the output of the spring controller to obtain the final control action on the system. The controller utilizes a qualitative partitioning of x_1 and $\dot{x}_1$ using five membership functions as shown in Figure 5. The input partitions of negbig (Negative), negsm (Negative_small), Zero, possm (Positive_small) and posbig (Positive) produce output partitions of nb (Negative), ns (Negative_small), Zero, ps (Positive_small) and pb (Positive), which represent the control force on body 1.



(a)



(b)

(c)

Figure 5: Fig. 5 (a) Membership functions of position of body 1 x_1 Fig. 5 (b) Membership functions of the velocity of body 1 $\dot{x}_1$ Fig. 5 (c) Membership functions of output.

The observed decision surface of Figure 6 shows that the corresponding output produced, for a given set of inputs, has a somewhat inverse linear relationship to those inputs. Two special membership functions, movingN and movingP, with output membership functions of guardP and guardN respectively, were also added to $\dot{x}_1$ (velocity of body 1) to ensure full stability.

Simulation Results

The performance of our fuzzy controller was investigated using computer simulations in Simulink® and MATLAB®. Figure 7 shows the response to a unit impulse disturbance to m_2, w(t) at t=0, for the nominal plant parameters m_1 = m_2 = 1 and k = 1. The controller shows excellent vibration suppression properties as the position initially increases from 0 to 1.068 units before returning and staying bounded within the required ± 0.1 units of the final value in 4.8s. System stability was obtained as required in the design specifications, and a reasonable maximum value of u was obtained to be 1.262 units as shown in Figure 8.

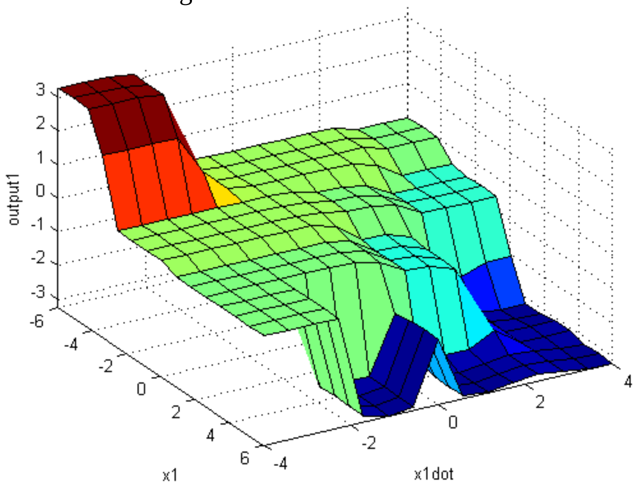


Figure 6: Output surface of the controller.

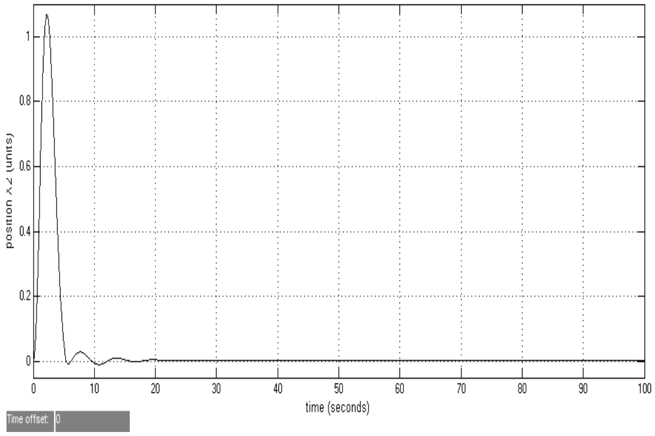


Figure 7: Time series of position of body 2, x_2, after a unit impulse disturbance on m_2 for nominal plant parameters m_1 = m_2 = 1 and k = 1. Settling time (Ts) = 4.8 seconds, Peak time (Tp) = 2.2 seconds and Peak Value (Pv) = 1.068 units.

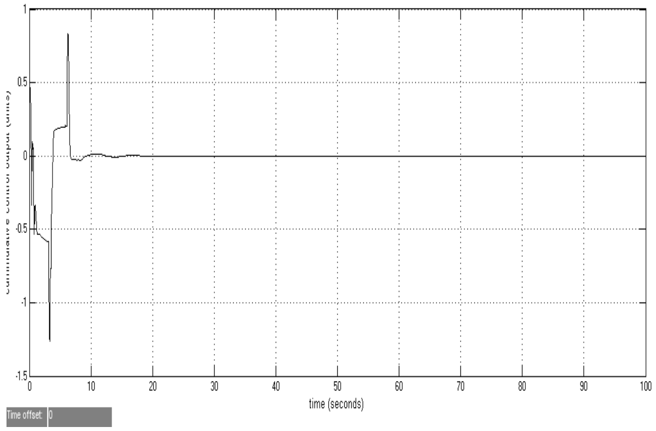
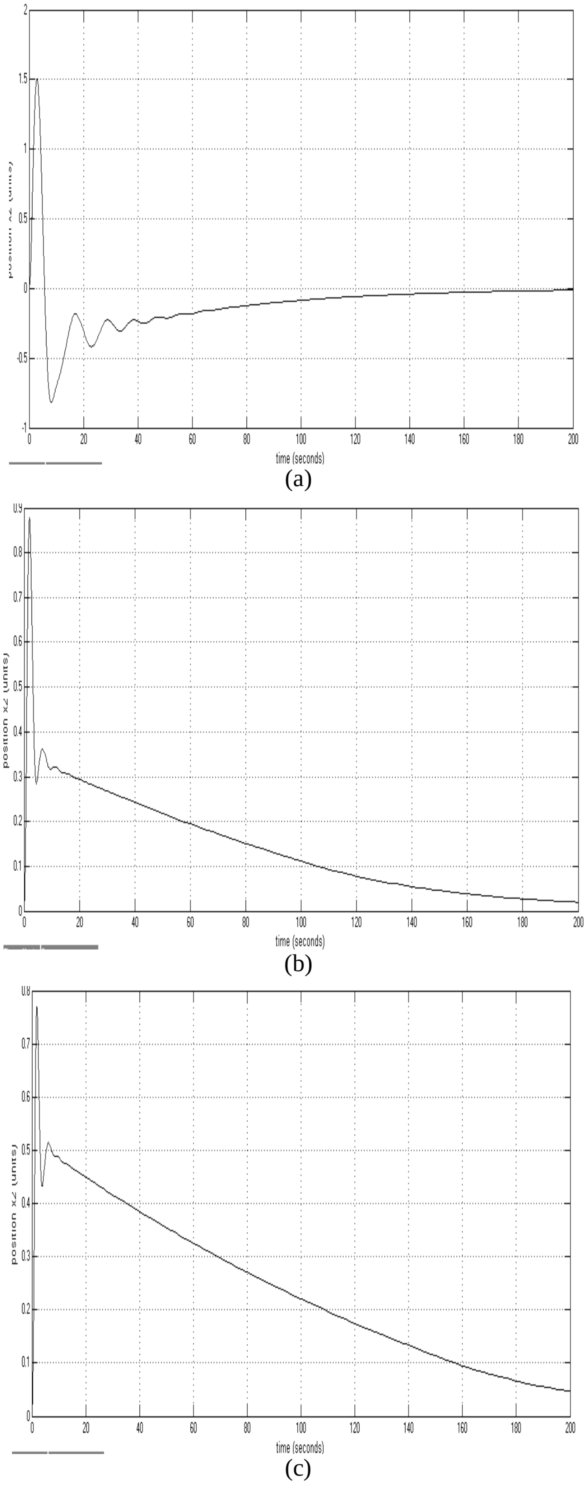


Figure 8: Time series of cumulative controller output u after a unit impulse disturbance on m_2 for nominal plant parameters m_1 = m_2 = 1 and k = 1. Maximum value of u = 1.262 units.

Figure 9 shows the stability of the system to varying spring constants in the range 0.5<k<2.0, while Table 1 summarizes the performances of the controller as compared to design objectives.



(a)

(b)

(c)

Figure 9: Time series of position of body 2, x_2 after a unit impulse disturbance on m_2 for nominal plant parameters $m_1 = m_2 = 1$. Fig. 9(a) k=0.5 Fig. 9(b) k=1.5 Fig. 9(c) k=2.0

Table 1. Controller performances for nominal plant parameters $m_1 = m_2 = 1$ and , in comparison with other fuzzy-logic based solutions to the benchmark problem (base on Linder and Shafai 1999, and Cohen and Ben Asher 2001)

	Our Controller	Linder and Shafai A	Linder and Shafai B	Cohen and Ben Asher	Design objectives
Settling time (Ts)	4.8	15.0	8.0	8.8	15.0
Max controller output *u*	1.262	-	-	0.53	--

It is evident that the fuzzy logic-based controller solves the first two of the benchmark problems. It, however, achieves better settling time performance over other fuzzy logic solutions, while staying within the requirements of reasonable controller output. This is due to unique fuzzy membership function placements and tunings, especially for stability and robust tracking.

Also, as the value of the spring constant k is increased the peak time and peak value decreases simultaneously. This is due to the fact that an increase in the spring constant allows the system to exhibit more inherent natural dampness that ensures less oscillations or more rigidity. This, however, increases the settling time significantly as the controller has less "control" over the system. The designed controller has been optimized for the case where k=1. This can be repeated for other values of spring constants in order to achieve better performances.

Conclusion

This paper uses a superimposition of qualitative stability and tracking behaviors instantiated with fuzzy rules which have clear linguistic interpretations. The impressive performance of the fuzzy logic controller on the ACC robust control benchmark shows its suitability for designing and developing controllers for stability and performance robustness in view of plant uncertainties, and sensitivity to actuator/sensor noncollocation. Of significant interest is the fact that the developed control strategy leads to robust near time-optimal control while requiring a relatively small amount of control effort.

Further studies can be pursued to test and improve the controller presented herein for the vibration suppression of structures, such as beams, plates, shells, and those possessing very high modal densities at lower frequencies. Also, the effects of high frequency sensor noise can be modeled in to the system, and a stochastic robustness analysis, using Monte Carlo simulations, can be used to obtain performance metrics, as estimated probabilities of stability/performance.

References

Wie, B., and Bernstein, D. S. 1992. Benchmark Problems for Robust Control Design. *Journal of Guidance, Control, and Dynamics*. Vol. 15, No.5: 1057-1059.

Linder, S. P., and Shafai, B. 1999. Qualitative Robust Fuzzy Control with Applications to 1992 ACC Benchmark. *IEEE Transactions on Fuzzy Systems*. Vol. 7, No. 4: 409-421.

Niemann, H. H., Stoustrup, J., Tofnner-Clausen, S., and Andersen, P. 1997. Synthesis for the Coupled Mass Benchmark Problem. In Proceedings of the American Control Conference, 2611-2615. Technical University of Denmark, Denmark.

Wie, B., and Liu, Q. 1992. Robust Control Design for Benchmark Problem #2. Arizona State University, Tempe.

Farag, A., and Werner, H. 2002. Robust Controller Design and Tuning for the ACC Benchmark Problem and a Real-time Application. IFAC 15th Triennial World Congress, Barcelona, Spain.

Hughes, D., and Wu, J. T. 1996. Passivity Motivated Controller Design for Flexible Structures. *Journal of Guidance, Control, and Dynamics*. Vol. 19, No. 3: 726-729.

Cohen, K., and Ben Asher, J. Z. 2001. Control of Linear Second-Order Systems by Fuzzy Logic-Based Algorithm. *Journal of Guidance, Control, and Dynamics*. Vol. 24, No. 3: 494-501.

Stengel, R. F., and Marrison, C. I. 1992. Robustness of Solutions to a Benchmark Control Problem. *Journal of Guidance, Control, and Dynamics*. Vol. 15, No. 5: 1060-1067.

King, P. J., and Mamdani, E. H. 1977. The Application of Fuzzy Control Systems to Industrial Processes. *Automatica*. Vol. 13, No. 3: 235-242.

Zadeh, L. A. 1965. Fuzzy Sets. Inform. Contr., Vol. 8: 338-353.

Jamshidi, M., Vadiee, N., and Ross, T. 1993. Fuzzy Logic and Control: Software and Hardware Applications. Englewood Cliffs, NJ: Prentice-Hall.

Yen, J., Langari, R., and Zadeh, L. 1992. In Proceedings of 2nd Int. Workshop Indust. Fuzzy Contr. Intell. Syst. College Station, Texas.

Zadeh, L. A. 1994. Fuzzy Logic, Neural Networks, and Soft Computing. Communications Association of Computing Machinery. Vol. 37: 77-84.

Driankov, D., Hellendoorn, H., and Reinfrank, M. 1996. An Introduction to Fuzzy Control, 2nd ed. New York: Springer-Verlag.

Jang, J.-S. R., and Sun, C.-T. 1995. Neuro-Fuzzy Modeling and Control. In Proceedings of IEEE. Vol. 83: 378-406.

Learning Fuzzy Cognitive Maps by a Hybrid Method Using Nonlinear Hebbian Learning and Extended Great Deluge Algorithm

Zhaowei Ren

School of Electronic and Computing Systems
University of Cincinnati
2600 Clifton Ave., Cincinnati, OH 45220

Abstract

Fuzzy Cognitive Maps (FCM) is a technique to represent models of causal inference networks. Data driven FCM learning approach is a good way to model FCM. We present a hybrid FCM learning method that combines Nonlinear Hebbian Learning (NHL) and Extended Great Deluge Algorithm (EGDA), which has the efficiency of NHL and global optimization ability of EGDA. We propose using NHL to train FCM at first, in order to get close to optimization, and then using EGDA to make model more accurate. We propose an experiment to test the accuracy and running time of our methods.

Introduction:

Fuzzy Cognitive Maps (FCM) (1) is a modeling methodology that represents graph causal relations of different variables in a system. One way of developing the inferences is by a matrix computation. FCM is a cognitive map with fuzzy logic (2).FCM allows loops in its network, and it can model feedback and discover hidden relations between concepts (3). Another advantage is that Neuron network techniques are used in FCM, e.g. Hebbian learning(4), Genetic Algorithm (GA) (5), Simulated Anealling (SA) (6).

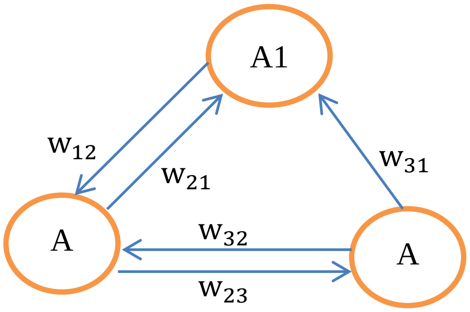


Figure 1 An example of FCM

The structure of FCM is similar to an artificial neuron network, e.g. Figure 1. There are two elements in FCM, concepts and relations. Concepts reflect attributes, qualities and states of system. The value of concepts ranges from 0 to 1. Concepts can reflect both Boolean and quantitative value. For example, a concept can reflect either the state of light (while 0 means off and 1 means on), or water level of a tank. If it reflects a quantitative value, equation [1] can be used for normalization.

$$\text{normalized value} = \frac{A - A_{min}}{A_{max} - A_{min}} \qquad [1]$$

where A is the concept value before normalization, and A_{max} and A_{min} are the possible maximum and minimum value of A. Relations reflect causal inference from one concept to another. Relations have direction and weight value. w_{ij} is denoted as the weight value from concept A_i to conceptA_j. For a couple of nodes, there may be two, one or none relations between them. There are three possible types of causal relations:

- $0 < w_{ij} < 1$ the relation from concept A_i to concept A_j is positive. When concept A_i increases (decreases), concept A_j also increases (decreases).
- $-1 < w_{ij} < 0$ the relation from concept A_i to concept A_j is negative. When concept A_i increases (decreases), on the contrary, concept A_j decreases (increases).
- $w_{ij} = 0$ there is no relations between A_i and A_j

When initial state of FCM is given, FCM will converge to a steady state through iteration process. One concept value is computed by the sum of weighted sum of all concepts that may be related to it. In each iteration, concept value is calculated by equation [2].

$$A_i^{k+1} = f\left(A_i^k + \sum_{\substack{j\neq i\\ j=1}}^{N} A_j^k \cdot w_{ji}\right) \qquad [2]$$

where A_i^{k+1} is the value of conceptin iteration $k+1$, A_i^k is the value of concept A_i in iteration, and w_{ji}^k is the weight value of the edge from concept A_i to concept A_j. And $f(x) = \frac{1}{1-e^{-\lambda x}}$, which is a transfer function to normalize weight value to [-1,1]. λ is a parameter that determines its steepness.

For example, figure 2 is It is a problem an industrial process control problem (8). There is a tank with two valves where liquids flow into the tank. These two liquid had reaction in this tank. There is another valve which empties the fluid in the tank. There is also a sensor to gauge the gravity of produced liquids (proportional to the rate of reaction) in tank. As described in the figure below

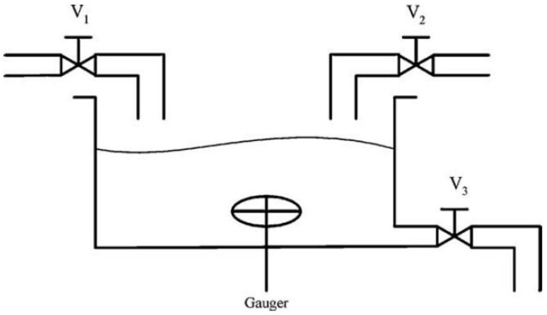


Figure 2 An industrial control problem

There are two constraints of this problem. The first one is to maintain value of G in a particular range, and the second one is to keep height of liquids (T) in a range. Parsopoulos et al. (8) proposes that there should be five concepts: (a) height of liquid in the tank, (b) the state of valve 1, (c) the state of valve 2, (d) the state of valve 3, and (e) the gravity of produced liquid in the tank. Our aim is to find out the causal inference value from one concept to another one.

There are mainly two strategies to learn an FCM. One is to exploit expert domain knowledge and formulate a specific application's FCM (7), this can be used when there is no good automated or semi-automated methods to build this model. If there are multiple domain experts available, each expert choose a value (e.g. very weak, weak, medium, strong, very strong) for the causal effect from one concept to another one; then the values are quantified and combined together into one value between -1 and 1. This strategy has its own shortage: when the problem is complex and need a large number of concepts to describe a system, the cost of expert strategy is very high; moreover, it is difficult to discover new hidden relations by this strategy. Another strategy is to develop a data driven learning method. Input, output and state of a system are recorded when it is running, and these records are used as a neuron network training dataset.

Background

One branch of Fuzzy Cognitive map (FCM) learning is Hebbian learning. Different Hebbian learning has been proposed, for example, Differential Hebbian Learning (DHL)(4), and its modified version Banlanced Differential Hebbian Learning (BDHL)(9). DHL changes weight matrix by the difference of two records, but it did not consider the scenario that multiple concepts have effect on one mutually. BDHL covers this situation, but it is costly owe to lack of optimization. The two Hebbian learning methods that have been used in real world is Active Hebbian learning (AHL) (10) and Nonlinear Hebbian Learning(NHL) (11), and both of them require expert knowledge before computation. AHL explores a method to determine the sequence of active concepts. For each concept, only concepts that may affect it are activated. AHL is fast but requires expert intervention. Experts should determine the desired set of concepts and initial structure of FCM. NHL is a nonlinear extension of DHL. In NHL, before iteration starts,experts have to indicate an initial structure and sign of each non-zero weight. Weight values are updated synchronously, and only with concepts that experts indicate.

Another branch of learning FCM structure is population-based method. Koulouriotis (12)proposes evolution strategies to train fuzzy cognitive maps. In 2007 Ghazanfari et al. (6) proposes using Simulated Annealing (SA) to learn FCM, and he compared Genetic Algorithm (GA)(5) and SA. They concluded that when there are more concepts in the network, SA has a better performance than genetic algorithm. In 2011, Baykasoglu and Adil (13) proposed an algorithm called extended great deluge algorithm (EGDA) to train FCM. EGDA is quite similar to SA, but it demands smaller number of parameters than SA. Population-based method is capable to reach global optimization even if the initial weight matrix is not good, but it is usually computationally costly, especially when the initial weight matrix is far from optimal position. Moreover, population-based methods have many parameters that have to be set before processing. The parameters are set usually by experiences, and then duplicated experiments with different parameters should be made to get better performance. Hebbian learning methods are relatively fast, but their performance depends on initial weight matrix and predefined FCM structure very much. Expert intervention is usually essential. Experts need to indicate a structure before experiments.

The third branch is hybrid method, which takes both the effectiveness of Hebbian learning and global search capability of population-based methods. Papageorgiou and Groumpos (14) proposed a hybrid learning method that combines NHL and Differential Evolution algorithm (DE). First, NHL is used to learn FCM, and then its result is feed to DE algorithm. This method makes uses of both the effectiveness of Hebbian learning and the global search ability of population-based method. The three experiments they did show this hybrid method is capable to train FCM effectively. Zhu et.al(15) proposes another hybrid method which combines NHL and Real-coded Genetic Algorithm (RCGA)

Here I suggest a hybrid method combing NHL and EGDA. EGDA has global search ability and relatively less demand of parameters. If its initial weight matrix is close to optimal condition, it will save much computing expense. Here we use NHL to train FCM roughly first, and then feed its result to EGDA. NHL is picked because it is simple and fast, and it can deal with continuous range of value of concepts

Hybrid Method Using NHL and EGDA

This hybrid method is processed by two stages.

Stage 1 use nonlinear Hebbian learning (NHL) (11)to train FCM

Step 1: Initialize weight matrix W^0 with help of experts and read input concept A^0. We feed the initial weight matrix to feed W^0

Step2:

Calculate A^1 (concept value in iteration 1.Initial values can be denoted as values in iteration 0) by the equation [3]

$$A_i^{k+1} = f(A_i^k + \sum_{j \neq i}^{N} A_j^k \cdot w_{ji}^k) \quad [3]$$
$$j=1$$

where $f(x) = \frac{1}{1-e^{-\lambda x}}$. λ is a parameter that determines increasing rate of curve. It is a transfer function. When x changes from - ∞ to ∞, f(x) changes from 0 to 1. Therefore, final result of concept value is still from zero to one.

Step 3:

Use equation [4] to update weights,

$$w_{ji}^k = w_{ji}^{k-1} + \eta_k A_j \ (A_i^k - A_j \ w_{ji}^{k-1}) \quad [4]$$

where η_k is learning rate function, and it decreases as k increases.

Step 4: At the end of each updating, the error function is computed as equation [5]

$$J = \sum_{i=1}^{N} (A_i^{k+1} - A_i^k)^2 \quad [5]$$

where k is the iteration number. There are two termination conditions. One is that value of error function [3] is below a threshold, and the other is there are enough times of iterations. If one of the termination conditions is reached, the iteration ends. Otherwise, go on the next iteration.

For example, now we have time series data of each concept value as Table 1

A1	A2	A3
0.5	0.5	0.1
0.6	0.4	0.2
0.5	0.3	0.3

Table 1 Concept value record

Each tuple is a record of three concept value.

Initial weight matrix is predicted by experts or generated randomly. Here it is as Table 2

W	1	2	3
1	N/A	0.3	0
2	0.7	N/A	0.2
3	-0.6	-0.3	N/A

Table 2 Initial weight matrix

w_{ij} (the weight from concept I to concept j) is the value in line I and column j. For example, $w_{21} = 0.7$

For example, we want to update w_{21} and w_{31} using the first tuple of data. First, we use equation [3] to calculate A_1^1. λ is set to 1 here.

$$A_1^1 = f(A_1^0 + w_{21}^0 A_2^0 + w_{31}^0 A_3^0)$$
$$= f(0.5 + 0.7 * 0.5 + (-0.6) * 0.1)$$
$$= f(0.74)$$
$$= 0.677$$

Then we use equation [2] to update w_{21} and w_{31}.Here the learning rate $\eta = 0.5$

$$w_{21}^1 = w_{21}^0 + \eta * (A_1^0 - A_2 \, w_{21}^0)$$
$$= 0.7 + 0.5 * (0.5 - 0.5 * 0.7)$$
$$= 0.707$$
$$w_{31}^1 = w_{31}^0 + \eta * (A_1^0 - A_3 w_{31}^0)$$
$$= -0.6 + 0.5 * \left(0.5 - 0.1 * (-0.6)\right)$$
$$= -0.32$$

Other weights are updated as above. Then we got the new weight matrix as below

W	1	2	3
1	N/A	0.475	0
2	0.707	N/A	0.317
3	-0.32	0.565	N/A

And A1=0.677, A2= 0.65, A3=0.55

$$J = (0.677 - 0.5)^2 + (0.65 - 0.5)^2 + (0.55 - 0.1)^2$$
$$= 0.2563$$

If J is larger than termination threshold, then go to step 2. Otherwise, terminate this algorithm and got to stage 2.

Stage 2: use extended great deluge algorithm (EGDA)(13) to train FCM.

Step 1: Initialize the weight matrix with the suggested value from stage 1. The output of step 1 is feed to this step. Assume the weight matrix we got from last step is as Table 3

W	1	2	3
1	N/A	0.3	0
2	0.6	N/A	0.1
3	-0.4	-0.3	N/A

Table 3 Weight matrix after stage 1

Step 2: find a new neighbor of the current weight matrix. For each non-zero weight (because the edge with zero weight does not exist by expert prediction)in the matrix, use the equation below to generate their neighbor.

$$w_{ij}^* = w_{ij} + (2 * \mathrm{random}(\quad) - 1) * \mathrm{step}_k \quad [6]$$

where random( ) is a function to generate random value from 0 to 1, and then $2 * \mathrm{random}(\quad) - 1$ is a function to generate random value from -1 to 1. step_k is a step size of moving. It is gradually decreased so this algorithm can have a more detailed search during the end of the search.

Step 3: Use equation [1] and new weight matrix to calculate estimated concept value. Then calculate fitness function to determine if new configuration is better than current one. Here we use the total error to be fitness function. The equation is as below

$$\text{Total error} = \frac{1}{KN} \sum_{i=1}^{N} |\mathrm{output}_i^{\mathrm{estimated}} - \mathrm{output}_i^{\mathrm{real}} \quad [7]$$

where *K* is the number of iteration, and *N* is the number of concepts.

Step 4: If the fitness function value of the neighbor configuration is better than tolerance, it is picked as current configuration. And then go to step 5, otherwise, go to step 2. Then reduce the tolerance.

Step 5: If the value of fitness function is better than best condition, update best condition.

For example: First we find a new neighbor for this. step_1 is set as 0.2.

$$
\begin{aligned}
w_{12}^{*} &= 0.3 + (2 * \mathrm{random}(\quad) - 1) * 0.2 \\
&= 0.3 + (2 * 0.5478 - 1) * 0.2 \\
&= 0.3191
\end{aligned}
$$

(random()=0.5478, generated randomly by matlab)

Using the same equation, we could get new weight matrix

W	1	2	3
1		0.3197	0
2	0.5482		0.2945
3	-0.2453	-0.4583	

Then calculate the concept value $A_1^{\mathrm{estimated}}, A_2^{\mathrm{estimated}}, A_3^{\mathrm{estimated}}$. In this example we use the record below

A1	A2	A3
0.5	0.5	0.1

$$
\begin{aligned}
A_1^{\mathrm{estimated}} &= f\big(0.5 + 0.5 * 0.5482 + 0.1 * (-0.2453)\big) \\
&= 0.6981 \\
A_2^{\mathrm{estimated}} &= f\big(0.5 + 0.5 * 0.3197 + 0.1 * (-0.4583)\big) \\
&= 0.5985 \\
A_3^{\mathrm{estimatd}} &= f(0.1 + 0.5 * 0 + 0.5 * 0.2945) = 0.6351
\end{aligned}
$$

$$
\text{Total error} = \frac{1}{1 * 3}\left(|0.6981 - 0.5| + |0.5985 - 0.5| + |0.6351 - 0.1|\right) = 0.2772
$$

If this value is above a tolerance, it is denoted as current configuration, and then it is compared with best configuration to see if it is the best so far. If the new neighbor is not below tolerance, find another neighbor near current one. Reduce tolerance after each search. If the total error is below a threshold or there is enough number of iteration, then this algorithm terminates.

Experiment Design:

There are two steps of our experiment. First we are going to test our method by simulated data, and try to find out the scenario that our method can be most efficient and accurate. On the second step, we will use our method in a real application.

In this experiment, data is generated by a random process. First the number of concepts and density of relations are set. We can try different number of concepts, from small to large, in order to test the performance of this method in network with different complexity. Density represents how many percent of edges exist in a network. It is defined as equation [8].

$$
\begin{aligned}
\mathit{density} &= \frac{\mathit{edges\ that\ exist}}{\mathit{all\ edges\ that\ may\ connects\ two\ concepts}} \\
&= \frac{\mathit{edges}}{\mathit{number\ of\ nodes}(\mathit{number\ of\ nodes} - 1)} \qquad [8]
\end{aligned}
$$

For example, if we set number of concepts as 5, and density as 0.4, number of edges is computed as below

$$
\mathit{number\ of\ edges} = 5 \times (5 - 1) \times 0.4 = 8
$$

then there would be eight edges in this network.

After number of concepts and edges are set, a model can be generated with random weight, and we name it original model. Then random data is generated, and they are fed to equation [1] iteratively, until it reaches a steady state (the error in equation [7] is lower than threshold). The steady state would be a record for simulated data. After a certain time of iteration, if it still cannot reach steady state, a new tuple of data would be generated randomly and fed to equation [1]. After hundreds of times, we will have a series of data as training set. This data is used to learn FCM by our method. The weight matrix we get would be compared with the original model. The error is calculated as equation [9]

$$
\mathrm{error} = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{\substack{j=1 \\ i \neq j}}^{N} \left(w_{ij} - \overline{w_{ij}}\right)^2 \qquad [9]
$$

where *N* is the number of concepts in this model.

Some other methods (NHL, EGDA, SA) can also be programmed, and compared with this method. These methods will be compared in accuracy and running time, in several conditions.

After simulated experiment, based on the best conditions for our method, we will apply it on a real practical problem.

Conclusion

We propose a hybrid method to learn FCM. Our method has taken advantages of fast speed of NHL and global search ability of EGDA. Moreover, we propose an experiment to test our algorithm, and try to apply it into practice.

Reference

1. Bart and Kosko. Fuzzy cognitive maps. International Journal of Man-Machine Studies 1986; 24: 65.
2. Kosko B. Fuzzy engineering. Upper Saddle River, NJ, USA: Prentice-Hall, Inc. 1997: .
3. Papageorgiou EI, Stylios C, and Groumpos PP. Unsupervised learning techniques for fine-tuning fuzzy cognitive map causal links. International Journal of Human-Computer Studies 2006; 64: 727.
4. Dickerson JA, Kosko B. Virtual worlds as fuzzy cognitive maps. Virtual Reality Annual International Symposium, 1993 , 1993 IEEE 1993; 471-477.
5. Stach W, Kurgan L, Pedrycz W, and Reformat M. Genetic learning of fuzzy cognitive maps. Fuzzy Sets Syst 2005; 153: 371-401.
6. Ghazanfari M, Alizadeh S, Fathian M, and Koulouriotis DE. Comparing simulated annealing and genetic algorithm in learning FCM. Applied Mathematics and Computation 2007; 192: 56.

7. Khan MS, Quaddus M. Group decision support using fuzzy cognitive maps for causal reasoning. Group Decis Negotiation 2004; 13: 463-480.

8. Parsopoulos KE, Papageorgiou EI, Groumpos PP, and Vrahatis MN. A first study of fuzzy cognitive maps learning using particle swarm optimization. 2003; 2: 1440.

9. Huerga AV. A balanced differential learning algorithm in fuzzy cognitive maps. 2002; .

10. Papageorgiou EI, Stylios CD, and Groumpos PP. Active hebbian learning algorithm to train fuzzy cognitive maps. International Journal of Approximate Reasoning 2004; 37: 219.

11. Papageorgiou E, Stylios C, and Groumpos P. Fuzzy Cognitive Map Learning Based on Nonlinear Hebbian Rule. In: Gedeon T and Fung L eds. AI 2003: Advances in Artificial Intelligence. Springer Berlin / Heidelberg, 2003: 256-268.

12. Koulouriotis DE, Diakoulakis IE, and Emiris DM. Learning fuzzy cognitive maps using evolution strategies: A novel schema for modeling and simulating high-level behavior. 2001; 1: 364.

13. Baykasoglu A, Durmusoglu ZDU, and Kaplanoglu V. Training fuzzy cognitive maps via extended great deluge algorithm with applications. Comput Ind 2011; 62: 187.

14. Papageorgiou EI, Groumpos PP. A new hybrid method using evolutionary algorithms to train fuzzy cognitive maps. Applied Soft Computing 2005; 5: 409.

15. Yanchun Z, Wei Z. An integrated framework for learning fuzzy cognitive map using RCGA and NHL algorithm. 2008; 1.

This page is intentionally left blank

Learning and Classifications

Chair: Suranga Hettiarachchi

Multi-K Machine Learning Ensembles

Matthew Whitehead
Colorado College
Mathematics and Computer Science
14 E. Cache La Poudre St.
Colorado Springs, CO 80903
matthew.whitehead@coloradocollege.edu

Larry S. Yaeger
Indiana University
School of Informatics and Computing
919 E. 10th St.
Bloomington, IN 47408
larryy@indiana.edu

Abstract

Ensemble machine learning models often surpass single models in classification accuracy at the expense of higher computational requirements during training and execution. In this paper we present a novel ensemble algorithm called Multi-K which uses unsupervised clustering as a form of dataset preprocessing to create component models that lead to effective and efficient ensembles. We also present a modification of Multi-K that we call Multi-KX that incorporates a metalearner to help with ensemble classifications. We compare our algorithms to several existing algorithms in terms of classification accuracy and computational speed.

Introduction

Groups of machine learning models, called ensembles, can help increase classification accuracy over single models. The use of multiple component models allows each to specialize on a particular subset of the problem space, essentially becoming an expert on part of the problem. The component models are trained as separate, independent classifiers using different subsets of the original training dataset or using different learning algorithms or algorithm parameters. The components can then be combined to form an ensemble that has a higher overall classification accuracy than a comparably trained single model. Ensembles often increase classification accuracy, but do so at the cost of increasing computational requirements during the learning and classification stages. For many large-scale tasks, these costs can be prohibitive. To build better ensembles we must increase final classification accuracy or reduce the computational requirements while maintaining the same accuracy level.

In this paper, we discuss a novel ensemble algorithm called Multi-K that achieves a high-level of classification accuracy with a relatively small ensemble size and corresponding computational requirements. The Multi-K algorithm works by adding a training dataset preprocessing step that lets training subset selection produce effective ensembles. The preprocessing step involves repeatedly clustering the training dataset using the K-Means algorithm at different levels of granularity. The resulting clusters are then used as training datasets for individual component classifiers. The repeated clustering helps the component classifiers obtain different levels of classification specialization, ultimately leading to effective ensembles that rarely overfit. We also discuss a variation on the Multi-K algorithm called Multi-KX that includes a gating model in the final ensemble to help merge the component classifications in an effective way. This setup is similar to a mixture-of-experts system. Finally, we show the classification accuracy and computational efficiency of our algorithms on a variety of publicly available datasets. We also compare our algorithms with well-known existing ensemble algorithms to show that they are competitive.

Related Work

One simple existing ensemble algorithm is called bootstrap aggregating, or *bagging* (Breiman 1996). In bagging, component models are given different training subsets by randomly sampling the original, full training dataset. The random selection is done with replacement, so some data points can be repeated in a training subset. Random selection creates a modest diversity among the component models.

Bagging ensembles typically improve upon the classification accuracies of single models and have been shown to be quite accurate (Breiman 1996). Bagging ensembles usually require a large number of component models to achieve higher accuracies and these larger ensemble sizes lead to high computational costs.

The term *boosting* describes a whole family of ensemble algorithms (Schapire 2002), perhaps the most famous of which is called Adaboost (Domingo & Watanabe 2000), (Demiriz & Bennett 2001). Boosting algorithms do away with random training subset selection and instead have component models focus on those training data points that previously trained components had difficulty classifying. This makes each successive component classifier able to improve the final ensemble by helping to correct errors made by other components.

Boosting has been shown to create ensembles that have very high classification accuracies for certain datasets (Freund & Schapire 1997), but the algorithm can also lead to model overfitting, especially for noisy datasets (Jiang 2004).

Another form of random training subset selection is called *random subspace* (Ho 1998). This method includes all training data points in each training subset, but the included data point features are selected randomly with replacement. Adding in this kind of randomization allows components to focus on certain features while ignoring others. Perhaps predictably, we found that random subspace performed better on datasets with a large number of features than on those with few features.

An ensemble algorithm called *mixture-of-experts* uses a gating model to combine component classifications to produce the ensemble's final result (Jacobs *et al.* 1991), (Nowlan & Hinton 1991). The gating model is an extra machine learning model that is trained using the outputs of the ensemble's component models. The gating model can help produce accurate classifications, but overfitting can also be a problem, especially with smaller datasets.

The work most similar to ours is the layered, cluster-based approach of (Rahman & Verma 2011). This work appears to have taken place concurrently with our earlier work in (Whitehead 2010). Both approaches use repeated clusterings to build component classifiers, but there are two key differences between the methods. First, Rahman and Verma use multiple clusterings with varying starting seed values at each level of the ensemble to create a greater level of training data overlap and classifier diversity. Our work focuses more on reducing ensemble size to improve computational efficiency, so we propose a single clustering per level. Second, Rahman and Verma combine component classifications using majority vote, but our Multi-KX algorithm extends this idea by placing a gating model outside of the ensemble's components. This gating model is able to learn how to weight the various components based on past performance, much like a mixture-of-experts ensemble.

Multi-K Algorithm

We propose a new ensemble algorithm that we call *Multi-K*, here formulated for binary classification tasks, but straightforwardly extensible to multidimensional classification tasks. Multi-K attempts to create small ensembles with low computational requirements that have a high classification accuracy.

To get accurate ensembles with fewer components, we employ a training dataset preprocessing step during ensemble creation. For preprocessing, we repeatedly cluster the training dataset using the K-Means algorithm with different values of K, the number of clusters being formed. We have found that this technique produces training subsets that are helpful in building components that have a good mix of generalization and specialization abilities.

During the preprocessing step the value for the number of clusters being formed, k, varies from K_{start} to K_{end}. K_{start} and K_{end} were fixed to two and eight for all our experiments as these values provided good results during pretesting. Each new value of k then yields a new clustering of the original training dataset. The reason that k is varied is to produce components with different levels of classification specialization ability.

With small values of k, the training data points form larger clusters. The subsequent components trained on those subsets typically have the ability to make general classifications well: they are less susceptible to overfitting, but are not experts on any particular region of the problem space. Figure 1 shows the limiting case of $k = 1$, for which a single classifier is trained on all of the training data. Figure 2 shows that as the value of k increases, classifiers are trained on smaller subsets of the original data.

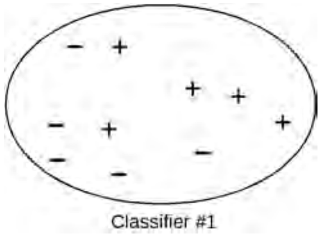


Figure 1: For $k = 1$, a single classifier is trained on the entire training set.

Larger values of k allow the formation of smaller, more specialized clusters. The components trained on these training subsets become highly specialized experts at classifying data points that lie nearby. These components can overfit when there are very few data points nearby, so it is important to choose a value for K_{end} that is not too large. This type of repeated clustering using varying values of k forms a pseudo-hierarchy of the training dataset.

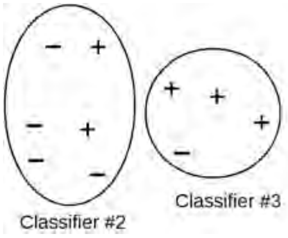


Figure 2: For $k = 2$, classifiers are trained on two disjoint subsets of the data, that overlap the $k = 1$ training set.

Figure 3 shows that as k is further increased, the clustered training datasets decrease in size allowing classifiers to become even more highly specialized. In this particular example, we see that classifier 6 will be trained on the same subset of training data as classifier 3 was above. In this way, tightly grouped training data points will be focused on since they may be more difficult to discriminate between. The clustering algorithm partitions the data in such a way as to foster effective specialization of the higher-k classifiers, thus maximizing those classifiers' ability to discriminate.

Following the clustering preprocessing step, each component model is trained using its assigned training data subset. When all the components are trained, then the ensemble is ready to make new classifications. For each new data point to classify, some components in the ensemble make classification contributions, but others do not. For a given new data point to classify, p, for each level of clustering, only those components with training data subset centroids nearest to p influence the final ensemble classification. Included component classifications are inversely weighted according to their centroid's distance from p.

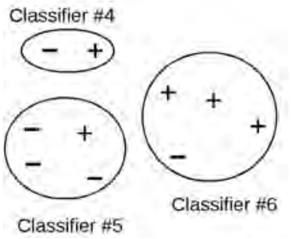


Figure 3: For $k = 3$, classifiers will be even more specialized because of their localized learning areas. If resulting clusters are identical, as is the case with clusters 3 and 6 in this example, only one cluster need be used to keep the final ensemble as small as possible.

Figure 4 shows an example where classifier #2 contributes to the final ensemble classification, since the data point to be classified (denoted by a '?') is nearest to its cluster centroid. Classifier #3 makes no contribution for this particular sample.

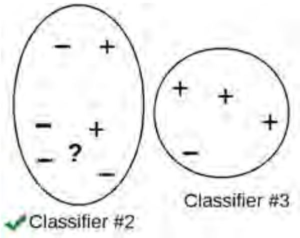


Figure 4: Classifier #2 contributes to the ensemble's final classification, but classifier #3 does not.

Figure 5 continues this example of component selection for the next level of clustering. At this level, classifier #5 is selected to contribute to the final classification and the other classifiers are ignored. Since classifier #5 was trained on a smaller training data subset than classifier #2 from Figure 4, it will be more specialized for the given problem. Its cluster centroid is also closer to the data point to be classified, so its classification weight will be greater.

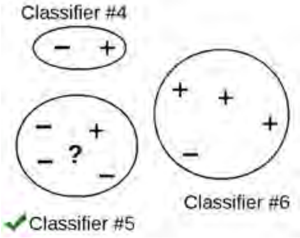


Figure 5: Classifier #5 is selected for ensemble contribution and is given a larger weight than classifier #2 from Figure 4.

The final classification is the weighted average of the n ensemble components' classifications in the current ensemble formation:

$$\frac{\sum_{i=0}^{n} w_i \cdot C_i(p)}{\sum_{i=0}^{n} w_i}$$

where $C_i(p)$ is classifier i's classification of target data point p and each w_i is an inverse distance of the form:

$$w_i = \frac{1}{dist(p, centroid_i)}$$

Pseudocode listing 1 shows the algorithm for clustering and training the component classifiers in Multi-K. Once the clustering and component classifier training is complete, then the ensemble is ready to classify new data points. Pseudocode listing 2 shows the algorithm for choosing the appropriate component classifiers given each new target data point to classify.

Given

D : training dataset
K_{start}: The number of clusters in the first level of clustering.
K_{end}: The number of clusters in the last level of clustering.

Ensemble Training Pseudocode

```
for $k$ from $K_{start}$ to $K_{end}$:
  cluster $D$ into $k$ clusters, $D_{ki}$, $i \in [1, k]$
  for $i$ from $1$ to $k$:
    train classifier $C_{ki}$ on data $D_{ki}$
```

Pseudocode 1: Multi-K ensemble training.

Ensemble Formation Pseudocode

```
Given, $p$, a data point to classify

For each clustering ($k$):
  Find the cluster, $D_{ki}$, with centroid
  $< D_{ki} >$ nearest to $p$
  Add $C_{ki}$, trained on $D_{ki}$, to ensemble
  Compute weight of $C_{ki}$ with distance from
  $< D_{ki} >$ to $p$
```

Pseudocode 2: Multi-K ensemble formation.

Finally, once the appropriate component classifiers have been selected, then the final ensemble classification can be calculated. Pseudocode listing 3 shows the algorithm for calculating the final classification based on a weighted sum of the outputs of the selected components.

Multi-KX Algorithm

The Multi-K algorithm used training dataset preprocessing to form effective component models. Final classifications were then performed by the entire ensemble by combining component classifications together based on the distance between $x_{predict}$ and the centroids of each of the component

Ensemble Classification Pseudocode

```
sum = 0
sum_weights = 0
for each classifier, $C$, in ensemble:
  weight$_C$ = $\frac{1}{dist(C,p)}$
  sum += weight$_C$ * $C's$ classification of $p$
  sum_weights += weight$_C$

final_classification = sum / sum_weights
```

Pseudocode 3: Multi-K ensemble classification.

training datasets. This technique works well, but we also thought that there may be non-linear interactions between component classifications and a higher accuracy could be gained by using a more complex method of combining components.

With this in mind, we propose a variation to Multi-K, called *Multi-KX*. Multi-KX is identical to Multi-K except in the way that component classifications are combined. Instead of using a simple distance-scaled weight for each component, Multi-KX uses a slightly more complex method that attempts to combine component outputs in intelligent ways. This intelligent combination method is achieved by the use of a gating metanetwork. This type of metanetwork is used in standard mixture-of-experts ensembles. This metanetwork's job is to learn to take component classifications and produce the best possible final ensemble classification. Figure 6 shows the basic setup of the ensemble.

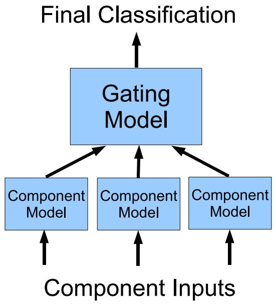


Figure 6: Ensemble classification using a gating model metanetwork.

The metanetwork can then learn the best way to combine the ensemble's components. This can be done by weighting certain components higher for certain types of new problems and ignoring or reducing weights for other components that are unhelpful for the current problem.

Building a Metapattern For the metanetwork to effectively combine component classifications, it must be trained using a labeled set of training data points. This labeled training set is similar to any other supervised learning problem: it maps complex inputs to a limited set of outputs. In this case, the metanetwork's training input patterns are made up of two different kinds of values. First, each classification value is included from all the ensemble's components. Then the distance between $x_{predict}$ and each cluster's centroid is also included. Figure 7 shows the general layout for a metapattern.

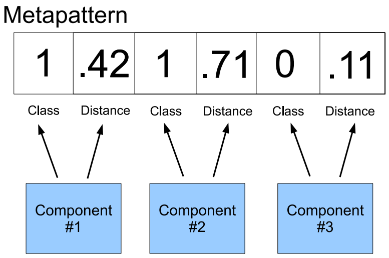


Figure 7: A Multi-KX metapattern. Inputs are grouped in pairs. The first number of each pair is the normalized component class prediction value (1 or 0 for this example). The second number is a measure of the distance between $x_{predict}$ and a component's training cluster centroid, normalized to the range (0, 1) where a value of 1 is used for the maximum distance centroid and a value of 0 is used for the minimum distance centroid. The metanetwork tries to learn the mapping from these values to the actual correct data point classification.

Experimental Results

To test the performance of our algorithms, we performed several experiments. First, we tested the classification accuracy of our algorithms against existing algorithms. Second, we measured the diversity of our ensembles compared to existing algorithms. Finally, we performed an accuracy vs. computational time test to see how each algorithm performs given a certain amount of computational time for ensemble setup and learning.

Accuracy Tests

Ensembles need to be accurate in order to be useful. We performed a number of tests to measure the classification accuracy of our proposed algorithms and we compared these results with other commonly-used ensemble techniques. We tested classification accuracy on a series of datasets from the UCI Machine Learning Repository (Asuncion & Newman 2007) along with a sentiment mining dataset from (Whitehead & Yaeger 2009). We performed a K-fold cross validation (with K=25) test using each algorithm on each dataset and we repeated each test ten times to ensure that the results were statistically stable. Each reported accuracy value is the mean of the resulting 250 test runs.

All accuracy tests were performed using support vector machines (Chang & Lin 2001) with linear kernels as the component classifiers, except we also compare our accuracies with boosting ensembles of decision stumps since the boosting algorithm is known to suffer less from overfitting with these component classifiers. For these tests, ensembles created with commonly used algorithms each had 50 component classifiers, as in (Breiman 1996).

Table 1 shows the classification accuracies for each tested algorithm and dataset. For each tested dataset, the most accurate result is shown in bold face. These results show that the proposed algorithms are competitive with existing ensemble algorithms and are able to outperform all of those algorithms for some datasets. The telescope dataset in particular yielded encouraging results with more than a 3% increase in classification accuracy (a 16% reduction in error) obtained by the Multi-KX algorithm. Performance was also good on the ionosphere dataset with an almost 2% higher accuracy (a 17% reduction in error) than other ensemble algorithms.

The dataset that Multi-K performed most poorly on was the restaurant sentiment mining dataset, where it was more than 2% behind the subspace ensemble. Since that dataset uses an N-gram data representation model, the data is considerably more sparse than the other tested datasets. We hypothesize that the sparsity made the clustering and the resulting component classifiers less effective. None of the ensemble algorithms were able to outperform a single SVM on the income dataset. This again suggests that the nature of the dataset will occasionally determine which algorithms do well and which do poorly.

Diversity Measurements

To form an effective ensemble, a certain amount of diversity among component classifiers is required. We measured the diversity of the ensembles formed by Multi-K using four different pairwise diversity metrics from (Kuncheva & Whitaker 2003):

- Q statistic - the odds ratio of correct classifications between the two classifiers scaled to the range -1 to 1.
- ρ - the correlation coefficient between two binary classifiers.
- Disagreement measure - proportion of the cases where the two classifiers disagree.
- Double-fault measure - proportion of the cases misclassified by both classifiers.

Figure 8 shows that the diversity of Multi-K ensembles generally falls in between algorithms that rely on random subsampling (bagging and subspace) and the one tested algorithm that particularly emphasizes diversity by focusing on previously misclassified training points (boosting). For example, values for the Q statistic and ρ were higher for the random methods and lower for boosting. The disagreement measure again shows Multi-K in the middle of the range.

Double fault values were nearly identical across all algorithms, suggesting that double fault rate is a poor metric for measuring the kind of diversity that is important to create ensembles with a good mix of generalization and specialization.

Combining Accuracy and Computational Efficiency

Since our main goal was to provide an algorithm that yielded high classification accuracies with the minimal amount of computational overhead, we performed a final combined accuracy and complexity test to show the relationship between the two for various ensemble algorithms. To do this, we ran existing ensemble algorithms with a wide variety of parameters that affect accuracy and training time. We then plotted a number of these training time/classification accuracy points, hoping to provide a simple, but informative way to compare the two across ensemble algorithms. We also ran each of the Multi-* algorithms and plotted each result as a single point on the graph because they have no free parameters to change. Each plot line is a best-logarithmic-fit for each existing algorithm to help see general trends. Figure 9 shows the results and has been normalized against the classification accuracy and computational time of a single SVM. Averaging across all tested datasets, Multi-K provided higher accuracy than other algorithms using anything less than about three times the compute time, and Multi-KX provided the highest accuracy of all, out to at least twice its computational costs.

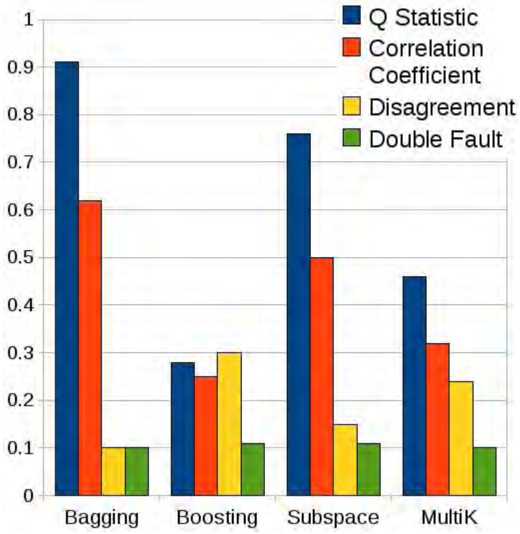


Figure 8: Measures of diversity for Multi-K ensembles on the heart dataset

Conclusions and Future Directions

We found that ensembles created using the Multi-* algorithms showed a good amount of diversity and had strong classification performance. We attribute this performance to a good mix of components with varying levels of generalization and specialization ability. Some components are effective over a large number of data points and thus exhibit the ability to generalize well. Other components are highly specialized at making classifications in a relatively small region of the problem space. The mix of both these kinds of components seems to work well when forming ensembles.

In the future, we hope to extend our method beyond binary to multi-class classification. In addition, we speculate that including additional characterizations of datasets and models as inputs to the gating network may further improve the accuracy of Multi-KX. We also are continuing work investigating alternative ways of preprocessing training datasets.

Ensemble	heart	bre	dia	iono	spam	tele	sent	inc
Single SVM	80.8	**96.7**	77.0	88.2	92.3	79.5	84.0	**85.2**
Bag	81.6	**96.7**	77.1	89.0	92.3	79.5	85.3	**85.2**
Boost SVM	81.6	96.6	77.1	88.9	**92.4**	79.6	84.0	84.5
Boost Stump	82.2	94.7	76.0	86.9	82.3	78.5	73.5	84.8
Subspace	**83.4**	96.7	76.4	88.2	91.7	77.8	**86.6**	84.6
Multi-K	83.1	96.5	77.2	88.6	92.1	79.8	84.3	85.1
Multi-KX	80.7	96.5	**77.4**	**90.9**	92.1	**82.8**	84.8	84.7

Table 1: Accuracy of algorithms with K-fold (K=25) tests - heart disease, breast cancer, diabetes, ionosphere, spam detection, telescope, restaurant sentiment mining, and income. For each tested dataset, the highest accuracy is shown in bold font.

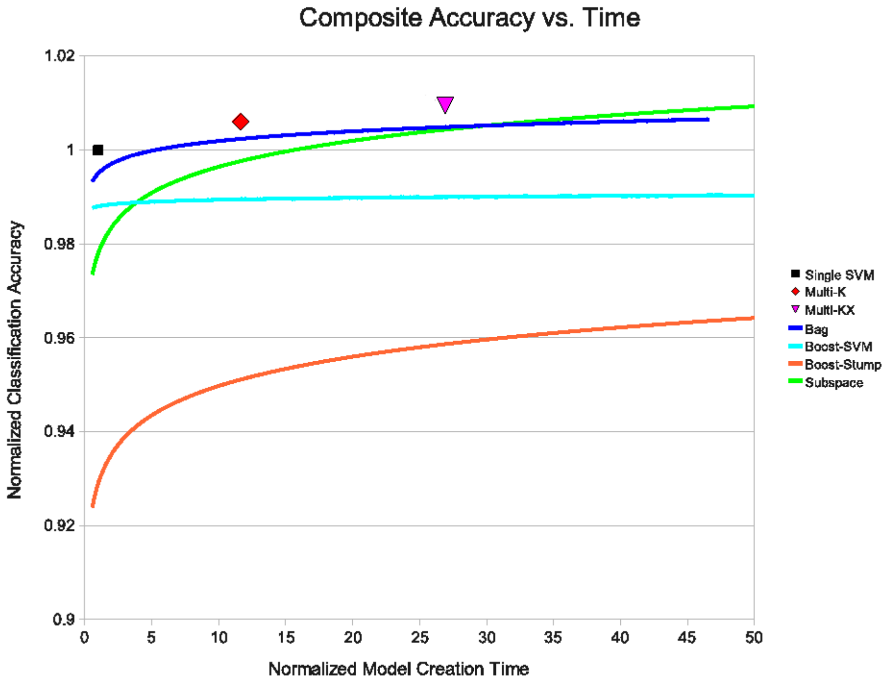


Figure 9: Normalized composite algorithm performance.

References

Asuncion, A., and Newman, D. 2007. UCI machine learning repository.

Breiman, L. 1996. Bagging predictors. *Machine Learning* 24(2):123–140.

Chang, C. C., and Lin, C. J. 2001. *LIBSVM: a library for support vector machines.* http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

Demiriz, A., and Bennett, K. P. 2001. Linear programming boosting via column generation.

Domingo, and Watanabe. 2000. Madaboost: A modification of adaboost. In *COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*.

Freund, and Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *JCSS: Journal of Computer and System Sciences* 55.

Ho, T. K. 1998. The random subspace method for constructing decision forests. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* Volume 20(Issue 8):832–844.

Jacobs, R.; Jordan, M.; Nowlan, S. J.; and Hinton, G. E. 1991. Adaptive mixtures of local experts. *Neural Computation* 3:79–87.

Jiang, W. 2004. Boosting with noisy data: Some views from statistical theory. *Neural Computation* 16(4):789–810.

Kuncheva, L. I., and Whitaker, C. J. 2003. Measures of diversity in classifier ensembles. *Machine Learning* 51:181–207.

Nowlan, S. J., and Hinton, G. E. 1991. Evaluation of adaptive mixtures of competing experts. *Advances in Neural Information Processing Systems*.

Rahman, A., and Verma, B. 2011. Novel layered clustering-based approach for generating ensemble of classifiers. In *IEEE Transactions on Neural Networks*, 781–792. IEEE.

Schapire, R. E. 2002. The boosting approach to machine learning: An overview.

Whitehead, M., and Yaeger, L. 2009. Building a general purpose cross-domain sentiment mining model. In *Proceedings of the 2009 CSIE World Congress on Computer Science and Information Engineering*. IEEE Computer Society.

Whitehead, M. 2010. Creating fast and efficient machine learning ensembles through training dataset preprocessing. In *Ph.D. Dissertation, School of Informatics and Computing Indiana University*.

Knowledge Elicitation to Prototype the Value of Information

Timothy Hanratty, Eric Heilman and John Dumer
Computational Information Science Directorate
US Army Research Laboratory
Aberdeen Proving Ground, MD, USA
timothy.hanratty@us.army.mil

Robert J. Hammell II
Department of Computer & Information Sciences
Towson University
Towson, MD, USA
rhammell@towson.edu

Abstract

From Wall Street to the streets of Baghdad, information drives action. Confounding this edict for the military is not only the unprecedented increase in the types and amount of information available, but the ability to separate the important information from the routine. Termed the value of information (VOI), the modern military commander and his staff require improved methodologies for assessing the applicability and relevance of information to a particular operation. This paper presents the approach used to elicit the knowledge necessary to value information for military analysis and enable the construction of a fuzzy-based prototype system for automating this valuation.

Introduction

Today's military operations require information from an unprecedented number of sources which results in an overload of information. With the requirement that relevant information be consistently available to troops as they conduct operations, a primary challenge for military commanders and their staff is separating the important information from the routine (FM 6-0 2003; DoD 2010). Calculating information importance, termed the value of information (VOI) metric, is a daunting task that is highly dependent upon its application to dynamic situations and human judgment (Alberts et al. 2001).

Currently the VOI assigned a piece of information is ascertained via a multiple step process requiring intelligence collectors and analysts to judge its value within a host of differing operational situations. For example, the types and immediacy of mission information needs will influence the amount of data reviewed and the value that an analyst will ascribe. While there is doctrine that describes a process of assigning value, it is sufficiently vague to allow multiple interpretations. As such, the cognitive processes behind these conclusions resist codification with exact precision and offer an excellent opportunity to leverage a computational intelligent solution using fuzzy inference.

This paper presents the approach used for gathering parameters necessary to value information for military analysis. Section 2 reviews the background information on the military domain with respect to VOI. Section 3 is an overview of knowledge elicitation techniques and the knowledge elicitation process utilized to capture values for fuzzy VOI rules. Section 4 presents a brief overview of the resulting prototype system. The conclusions and next steps are presented in Section 5.

Background

Understanding the Domain Challenge

On today's battlefield, information drives action. Commanders must know details about important persons, places and events within their area of operations to address issues ranging from kinetic fights to adjudicating legal disputes to revitalizing a depleted economy. From sophisticated unmanned ground acoustic sensors to open-source RSS news feeds, military commanders are inundated with an unprecedented opportunity for information. Table 1 depicts military information volume. As unit echelon increases, the scope of military operations and number of information reports grows tremendously. Intelligence analysts examine this information to determine the impact of trends, important human networks, and threat tactics, techniques, and procedures on current and future plans.

As shown in Figure 1, accurate VOI estimation is essential to the intelligence analysis process, promoting improved situational understanding and effective decision-making. The entire process is designed to produce and

Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-11-2-0092. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

Echelon	Planning time	Execution time	Reports per hour	Area of Operation
Division	Week	Week/Days	~Millions	Province
Brigade	Days	Days	170K	Province /district
Battalion	Days/hours	Day	56K	District
Company	Hours	Hours	18K	Village
Platoon	Hour/Min	Hour/Min	6K	Village/Hamlet

Table 1: Military Echelons with typical Operational Times / Areas (James 2010)

make available relevant intelligence information. For all military data, intelligence collectors are responsible for the initial estimation of information value. While there are guidelines for VOI determination, these are subject to collector/analyst interpretation. In point of fact, a recent US Army Intelligence Center of Excellence study considered “Information Validation (Data Pedigree, Corroboration and Cross Validation) and Stance Analysis (Elimination of Bias and Use of Multiple Analysis Perspectives)” as major issues (Moskal, Sudit, and Sambhoos 2010).

Proper VOI is integral to battlefield success. VOI is essential in the collect-assess portion of the intelligence process. At higher echelons, VOI is a metric useful in determining the degree of situational estimate accuracy amidst the uncertainty of combat. Additionally, VOI is a focusing element as a searchable criterion, enabling analysts to find relevant information quickly. At lower echelons, analysts can use VOI to create an optimum course of action for immediate mission execution.

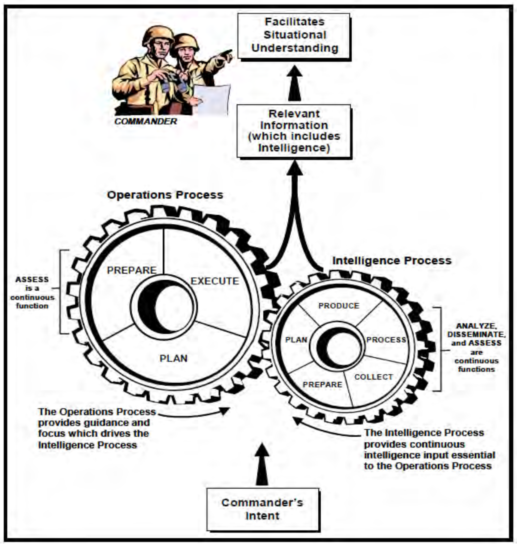

Facilitates Situational Understanding
COMMANDER
Relevant Information (which includes Intelligence)
Operations Process
Intelligence Process
ASSESS is a continuous function
PREPARE
EXECUTE
PLAN
PRODUCE
PLAN
PROCESS
PREPARE
COLLECT
ANALYZE, DISSEMINATE, and ASSESS are continuous functions
The Operations Process provides guidance and focus which drives the Intelligence Process
The Intelligence Process provides continuous intelligence input essential to the Operations Process
Commander's Intent

Figure 1: Military Information Process (FM 2-22.3 2006)

VOI Guidelines

The procedure for alphanumerically rating the “confidence” or “applicability” assigned a piece of information is essentially described in the annex to NATO STANAG (Standard Agreement) 2022 as well as in Appendix B of US Army FM-2-22.3 (FM 2-22.3 2006; NATO 1997). The NATO standard further dictates that, where possible, “an evaluation of each separate item of information included in an intelligence report, and not merely the report as a whole” should be made. The weight given each piece of information is based on the combined assessment of the *reliability of the source* of the information with the assessment of its *information credibility or content*.

As depicted in Table 2 and Table 3, respectively, the alphabetic *Reliability* scale ranges from A (Completely Reliable) to E (Unreliable) while the numeric *Content* scale ranges from 1 (Confirmed by other sources) to 5 (Improbable) (FM 2-22.3 2006; NATO 1997). Both scales account for the information that cannot be judged for source reliability or content with ratings F and 6.

So as an example, a piece of information that was received by a source that has in the past provided *valid* information would be scored a *Reliability Rating* of either B or C; depending on the degree of doubt in authenticity. That same piece of information, if not confirmed, but seeming logical, would receive a *Content Rating* of either 2 or 3; again depending on the degree the information was consistent with other information. It quickly becomes obvious the subjective nature of the ratings (B2 vs. C3) can quickly lead to ambiguity.

A	Reliable	**No doubt** of authenticity, trustworthiness, or competency; has a history of complete reliability
B	Usually Reliable	**Minor doubt** about authenticity, trustworthiness, or competency; has a history of valid information most of the time
C	Fairly Reliable	**Doubt** of authenticity, trustworthiness, or competency but has provided valid information in the past
D	Not Usually Reliable	**Significant doubt** about authenticity, trustworthiness, or competency but has provided valid information in the past
E	Unreliable	**Lacking** in authenticity, trustworthiness, and competency; history of invalid information
F	Cannot Judge	**No basis** exists for evaluating the reliability of the source

Table 2: Source Reliability (NATO 1997)

1	Confirmed	**Confirmed** by other independent sources; **logical** in itself; **Consistent** with other information on the subject
2	Probably True	Not confirmed; **logical** in itself; **consistent** with other information on the subject
3	Possibly True	Not confirmed; **reasonably logical** in itself; **agrees with some** other information on the subject
4	Doubtfully True	Not confirmed; possible but **not logical; no other information** on the subject
5	Improbable	Not confirmed; **not logical** in itself; **contradicted** by other information on the subject
6	Cannot Judge	**No basis** exists for evaluating the validity of the information

Table 3: Information Content (NATO 1997)

In an attempt to guide the application of composite ratings (i.e., B2 vs. C3) to varied operational situations, organizations have generalized the usefulness of data by developing charts similar to the one shown in Figure 2 (Hanratty et al. 2011). Positioned along the x-axis are the possible ratings for source reliability while the y-axis reflects those possible for information content. Combined, these ratings form a composite that in general reflects the generic value of a piece of information to analysis efforts; that is, a value within a general context. As shown in Figure 2, a piece of information can have three distinct value states, namely black is good, grey is questionable, and white is not useable. This rudimentary attempt to form a composite value shows progress, but the three states encompass several combined categories resulting in a blurred understanding of VOI. Capturing the complexity of analyst's intuitive knowledge through elicitation methods required an increased specificity of VOI states.

	A	B	C	D	E	F
1	■	■	▨	▨		
2	■	■	▨	▨		
3	▨	▨	▨	▨		
4	▨	▨	▨	▨		
5						
6						

Figure 2: Example Information Source / Reliability Matrix (Hanratty et al. 2011)

Knowledge Elicitation

Overview of Knowledge Elicitation

Knowledge elicitation is generally the first step in the construction of a system that seeks to use expert knowledge to solve a problem. In the process of building such a system, the knowledge engineer must interact with one or more Subject Matter Experts (SMEs) to gather, organize, and codify the appropriate domain specific problem-solving expertise (Martin and Oxman 1988).

While knowledge *elicitation* and knowledge *acquisition* are occasionally used interchangeably in the literature, most researchers draw a clear distinction between the two; additionally, knowledge *engineering* is a third concept that appears in the literature (Addis 1987; Cooke 1999; Daintith 2012; Hoffman et al. 1995; Sagheb-Tehrani 2009). Though some slight differences exist in how the three terms are defined and described, the following categorizations are used in this research and generally capture the essence of the distinctions. *Knowledge engineering* is the over-arching process of building knowledge based systems which includes elicitation, representation, and implementation. *Knowledge acquisition* is a subset of knowledge engineering, and consists of the gathering of all forms of domain knowledge using any methods. Finally, *knowledge elicitation* is a subset of knowledge acquisition and encompasses the extraction of domain knowledge from human experts. While all the steps involved with knowledge engineering must be performed to construct a usable knowledge-based system, herein we only seek to describe our knowledge *elicitation* efforts.

The knowledge elicitation process is much more complex than just arranging a meeting or meetings with SMEs. One important but perhaps subtle aspect of the process is the need to choose experienced and available experts that have excellent communication skills as well as at least some commitment to the project at hand (Liou 1992). Additionally, it is important that the knowledge engineer have at least a working knowledge of the domain, including the terminology and basic concepts regarding the problem and the problem-solving process in the specific environment (Waterman 1983). Finally, it is also important that the appropriate knowledge elicitation method or methods are chosen (Liou 1992).

Knowledge Elicitation Methods

There are a myriad of assessments of knowledge elicitation methods and numerous representations for how to classify them. For our purposes here, we will present and briefly describe the four categories of knowledge elicitation methods identified by Cooke (1999).

Observation. This process consists of watching an SME perform the task or tasks in question. Typically, great care should be taken to avoid disrupting the SME during the reasoning process. The observations are recorded somehow (video, photographs, audio, notes, and the like). This method can be particularly useful as a beginning technique to allow the knowledge engineer to understand enough to develop more structured knowledge elicitation sessions.

Interviews. Interviews are used to simply ask SMEs what they know. The interviews may be structured, unstructured, or a combination. Unstructured interviews are free-form and use open-ended questions; they may be useful in the beginning knowledge elicitation efforts to get a preliminary understanding of the domain. Structured interviews set up an artificial scenario to impose constraints on the SME's responses. Interview methods are often specifically tailored to the particular domain or problem so that some precise type of knowledge may be obtained. The Critical Decision Method falls into this category of techniques.

Process Tracing. This method is used for gathering information that is procedural in nature; it looks at behavioral events that are sequential in form. It is useful to ascertain conditional rules or note the order in which cues are used by the decision maker. The "think-aloud" technique is included in this category.

Conceptual Methods. This process attempts to gather conceptual structures present within the domain that are derived as concepts and interrelations. Steps include: 1) discovering relevant concepts, perhaps through interviews; 2) gathering opinions from one or more SMEs as to how the concepts relate; 3) representing the relationships; and 4) interpreting the result. One method of obtaining the SME's beliefs as to how the concepts relate is by using a grid approach. In this method, concepts are rated across a set of dimensions, and then the similarity among concepts can be determined in some way.

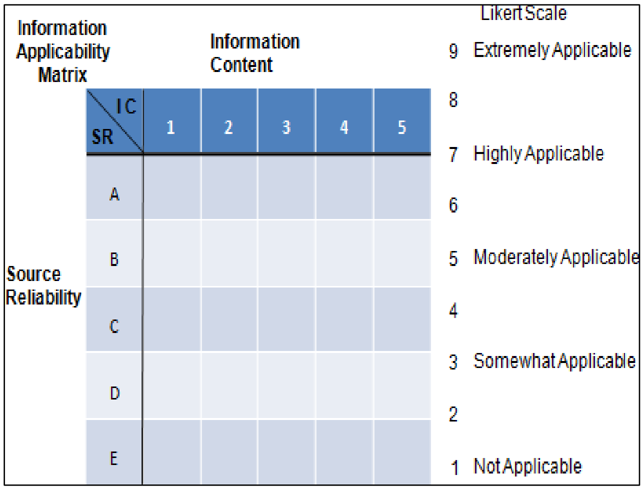
Knowledge Elicitation Within the VOI Domain

In general, military operations are defined by their associated *operation tempo*; that is, the time it takes to plan, prepare and execute an exercise. High-tempo operations typically require the decision cycle to be measured in minutes to hours. Slower tempo operations will generally allow the decision cycle to be measured in months or longer. Absent from the model presented in Section III is the application of the *information applicability* rating to a specific operation type. Without the specific framework of a given operation type the associated impact of information latency (or information timeliness) requirements are lost. Restated, the true VOI is dependent upon the type of military operation to which the information is being applied. For instance, in a high-tempo operation, where decisions are made in short timeframes, added emphasis is assigned to information that has high applicability and was more recently received than others.

In order to capture the cognitive requirements necessary to refine our model and build the fuzzy association rules, the team applied the Conceptual Method posed by Cooke. A review of the military intelligence process revealed several relevant concepts such as operational tempo mentioned above. The team and the SMEs then discussed the relationships between data age, operational tempo, and information applicability. These relationships were developed into a two-part Likert survey instrument and the final product presented to the SMEs to gather specific values; the process of using the two surveys is detailed further in the rest of this section. The initial interpretation of the results led to the fuzzy rules that were codified in the prototype. Of course, any sort of "validation" of the system actually implies that the SMEs must corroborate the fuzzy rules, which basically requires other iterations of knowledge elicitation to ensure that the resulting system is accurate and precisely reflects the meaning and relationships the SMEs intended to convey. These efforts are briefly discussed in later sections.

The first survey was used to capture the generic *information applicability* rating from the doctrinal model described in Section III; that is, how to define the potential importance of a piece of information given a specific type of operation. The second survey was used to calculate the actual VOI based on the temporal latency of the information and a particular operational tempo. In this case the temporal latency was defined as either: recent, somewhat recent or old. It is particularly noteworthy that the cognitive concept of temporal latency was purposefully left as a subjective construct for the SME. In general, the surveys provided contextual structure for the structured interview. Additionally, the matrices proved useful in physically recording SME VOI determination responses to the questions of information applicability and the value within military mission execution context.

For the first part of the survey, a Likert instrument was developed that incorporated the military doctrinal information rating system. This system features a combination of information content and source reliability. Information content is rated on a scale of one thru five with one (best case) being termed as, "Confirmed by other independent sources" and five (worst case) being termed as, "Not confirmed." Likewise, source reliability is also rated on a scale of one thru five; with one being termed as, "No doubt of authenticity, trustworthiness, or competency", and five being termed as, "Lacking in authenticity, trustworthiness, an competency" (FM 2-22.3 2006). The authors have coined the combination of these two ratings as a general "*information applicability*" rating for a given piece of information. The composite rating is expressed on a Likert scale of one through nine with nine being extremely applicable and one being least applicable to military missions. The instrument, shown in Figure 3, is the matrix used to capture SME ratings reflecting applicability.



Information Applicability Matrix

Information Content

Source Reliability

IC SR	1	2	3	4	5
A					
B					
C					
D					
E					

Likert Scale

9	Extremely Applicable
8	
7	Highly Applicable
6	
5	Moderately Applicable
4	
3	Somewhat Applicable
2	
1	Not Applicable

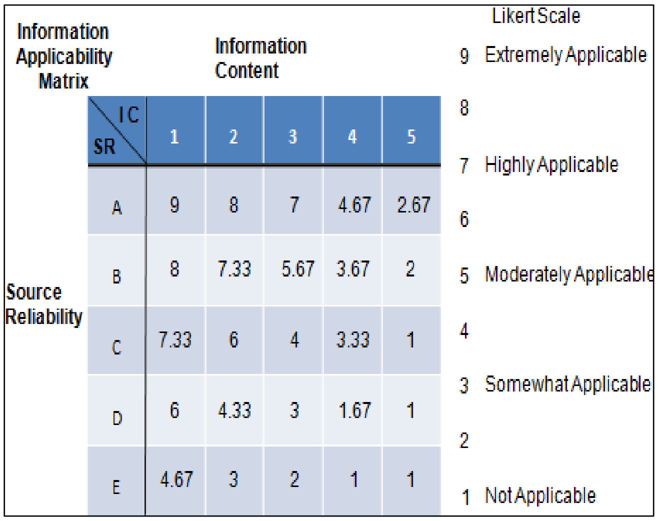
Figure 3: Likert Survey for Refined Information Applicability

During the pilot session, three intelligence analysts rendered their opinions on the generic applicability of data with ratings reflected within each cell of the matrix. For example, an applicability rating of “A1” that reflects the most applicable data would lend itself to the *Extremely Applicable* rating of 9. The averaged information applicability ratings for the three analysts are shown in Figure 4.

With the generic *information applicability* ratings completed, the second step involved applying those ratings against the *aspects* associated with a specific mission type. While many different *aspect* possibilities exist, the focus of this pilot survey was on the two primary military aspects of operational tempo and the temporal latency of the information. In this case the operational tempo was defined as either ‘tactical’, ‘operational’ or ‘strategic’, where the differences between the operational tempos is defined by the immediacy of the mission and is measured in the amount of time it takes to plan, prepare and execute a mission. The temporal latency of the information, on the other hand, was measured as a degree to which the information was either recently collected, somewhat recently collected or old.

The resulting VOI matrix that would be used for one of the specific operational tempos is shown in Figure 5. Here the composite VOI rating is expressed on a Likert scale of zero thru ten with ten being extremely valuable and zero equally no value to the mission.

The SMEs used three individual surveys to gauge the VOI for military mission immediacy of data use, namely one for use within a short time, one for use within a moderate time and one for use within a long time. The VOI results gained for data use in a short amount of time are shown in Figure 6.


Information Applicability Matrix

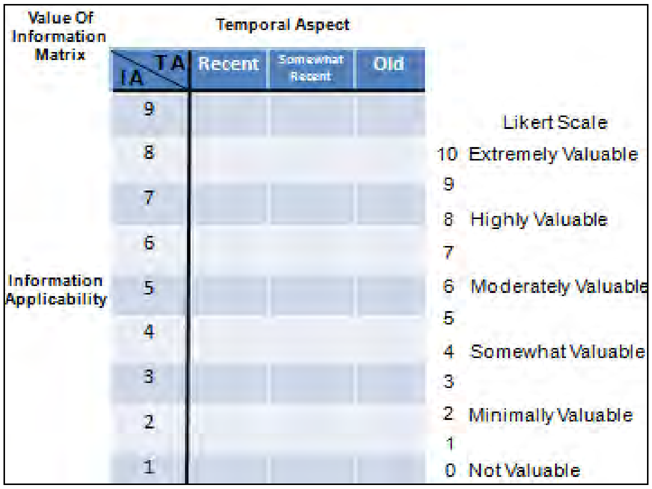
Information Content

Source Reliability

SR \ IC	1	2	3	4	5
A	9	8	7	4.67	2.67
B	8	7.33	5.67	3.67	2
C	7.33	6	4	3.33	1
D	6	4.33	3	1.67	1
E	4.67	3	2	1	1

Likert Scale
9 Extremely Applicable
8
7 Highly Applicable
6
5 Moderately Applicable
4
3 Somewhat Applicable
2
1 Not Applicable

Figure 4: Averaged SME Information Applicability Ratings


Value Of Information Matrix

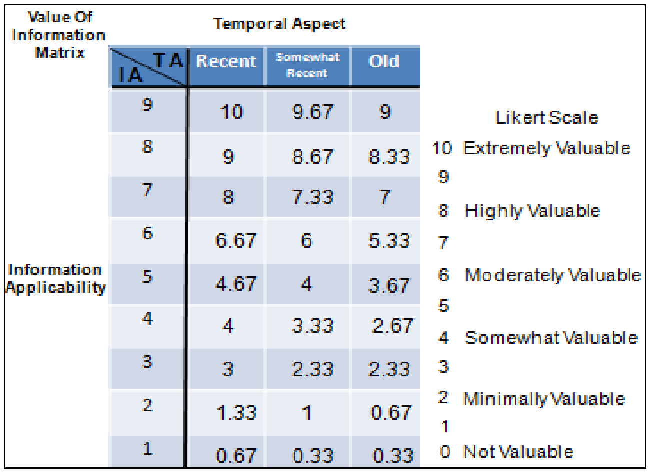
Temporal Aspect

Information Applicability

IA \ TA	Recent	Somewhat Recent	Old
9			
8			
7			
6			
5			
4			
3			
2			
1			

Likert Scale
10 Extremely Valuable
9
8 Highly Valuable
7
6 Moderately Valuable
5
4 Somewhat Valuable
3
2 Minimally Valuable
1
0 Not Valuable

Figure 5: VOI Likert Survey with Temporal Aspect


Value Of Information Matrix

Temporal Aspect

Information Applicability

IA \ TA	Recent	Somewhat Recent	Old
9	10	9.67	9
8	9	8.67	8.33
7	8	7.33	7
6	6.67	6	5.33
5	4.67	4	3.67
4	4	3.33	2.67
3	3	2.33	2.33
2	1.33	1	0.67
1	0.67	0.33	0.33

Likert Scale
10 Extremely Valuable
9
8 Highly Valuable
7
6 Moderately Valuable
5
4 Somewhat Valuable
3
2 Minimally Valuable
1
0 Not Valuable

Figure 6: VOI SME Results Fast Op Tempo

The Likert scales were easy to explain and provided a readily understood scale for rating data values. Data collected via the pilot session with the military analyst SMEs reveals the use of similar trends in analysis and is readily adaptable to the project mathematical model. Further, the inherent flexibility in the collection process seems applicable to additional military contexts that can become the subject of future model applications.

Initial Prototype

The Fuzzy Associative Memory (FAM) model was chosen to construct the prototype fuzzy system. A FAM is a *k*-dimensional table where each dimension corresponds to one of the input universes of the rules. The *i*th dimension of the table is indexed by the fuzzy sets that compromise the decomposition of the *i*th input domain. For the prototype system, three inputs are used to make the VOI decision (source reliability, information content, and timeliness); with three input domains, a 3-dimensional

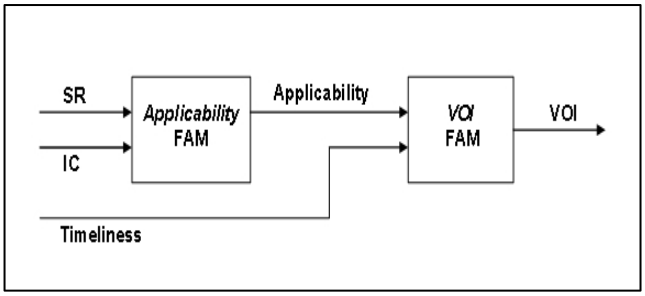


Figure 7: Prototype System Architecture

FAM could be used. However, the decision was made to use two, 2-dimensional FAMs connected "in series" to produce the overall VOI result for several pragmatic reasons (Hammell, Hanratty, and Heilman 2012).

The overall architecture of the prototype fuzzy system is shown in Figure 7. Two inputs feed into the *Applicability* FAM: source reliability and information content; the output of the FAM is the information applicability decision. Likewise, two inputs feed into the *VOI* FAM: one of these (information applicability) is the output of the first FAM; the other input is the information timeliness rating. The output of the second FAM, and the overall system output, is the VOI metric.

The rules elicited from the SMEs are represented in the appropriate FAMs and form the fuzzy rule bases. The number of fuzzy sets, and thus the "language" of the rules, was defined in the knowledge elicitation phase using the two surveys described above. That is, the decomposition of the domains is as shown in Figures 3 and 5. The two inputs to the *Applicability* FAM are divided into five fuzzy sets; the output domain is divided into nine fuzzy sets. Likewise, for the *VOI* FAM, the input domain for information applicability is divided into nine fuzzy sets (as just mentioned), the timeliness input domain three fuzzy sets, and the output domain eleven fuzzy sets.

Figures 4 and 6 actually represent the fuzzy rules bases for the two FAMs resulting from the knowledge elicitation process. For example, Figure 4 demonstrates that one rule in the *Applicability* FAM is "If *source reliability* is reliable (A) and *information content* is possibly true (3), then *information applicability* is highly applicable (7)". Note that the *VOI* FAM shown in Figure 6 applies only to the fast operational tempo (tactical) mission context, while the *Applicability* FAM is constant across all three mission contexts.

Triangular membership functions are used within the system, wherein the triangles are isosceles with evenly spaced midpoints. The output from each FAM is determined by the standard centroid defuzzification strategy. More detailed description of the FAMs, the fuzzy rule bases, the domain decompositions, and other implementation aspects of the prototype system can be found in (Hammell, Hanratty, and Heilman 2012).

The prototype system has been exercised across numerous scenarios (that is, various combinations of input values) to produce VOI determinations. These preliminary system results have been demonstrated to the SMEs and the system performance has been validated in principal and concept. That is, the system output has been judged to be consistent with what the SMEs would expect, and the prototype has demonstrated the feasibility to both elicit rules from experts in this domain as well as to use the extracted knowledge in a meaningful way.

Note that there is no current system against which the results can be compared. As such, the system has not been tested comprehensively due to the human-centric, context-based nature of the problem and usage of the system. Thus, the system performance will need to be validated by providing the SMEs with various scenario-based VOI results for their examination and feedback. In some cases the output of the system is an exact application of the rules provided by the SMEs which should permit easy judgment; in other instances, the system output is less clear and will require more detailed examination.

Conclusion and Future Work

Information drives action and for the military that is facing an unprecedented increase in the types and amount of information available, the ability to separate the important information from the routine is paramount. This paper presented an approach used for gathering the parameters to calculate the VOI for military analysis and allow the subsequent development of a fuzzy-based prototype system.

The obvious next step for this effort is to seek validation of the system from the SMEs by producing a comprehensive, well-designed set of scenario-based VOI results for their examination and feedback. It is entirely possible that the concepts and relationships captured through the conceptual method of knowledge elicitation would require modification. If so, further iterations of the knowledge elicitation process will occur. As the program matures, the capability to accommodate inconsistent or contradictory information will be investigated. For the military, the ability to efficiently and effectively calculate VOI and separate the wheat from the chaff is paramount. This program is an important step towards that goal.

References

Addis, T.R., "A Framework for Knowledge Elicitation", University of Reading, Department of Computer Science, 1987.

Alberts, David S., John J. Garstka, Richard E. Hayes, and David T.Signori. *Understanding Information Age Warfare.* Washington, DC: CCRP, 2001.

Cooke, N.J., “Knowledge Elicitation”, in *Handbook of Applied Cognition*, F. Durso, ed., Wiley, pp. 479-509, 1999.

Daintith, J, “Knowledge Acquisition”, A Dictionary of Computing, *Encyclopedia.com*, 20 Feb 2012, <http//www.encyclopedia.com>, 2004.

DoD (Department of Defense), “Quadrennial Defense Review”, January 2010.

FM 2-22.3 (FM 34-52) Human Intelligence Collector Operations, Headquarters, Department of the Army, Sept 2006.

FM 6-0 (US Army Field Manual 6-0), Mission Command: Command and Control of Army Forces, US Army, August 2003.

Hammell, R.J. II, T. Hanratty, and E. Heilman, “Capturing the Value of Information in Complex Military Environments: A Fuzzy-based Approach”, *Proceedings of the IEEE International Conference on Fuzzy Systems 2012 (FUZZ-IEEE 2012)*, 10-15 June 2012, Brisbane, Australia, accepted.

Hanratty, T. P., et. al., “Counter-Improvised Explosive Device (IED) Operations Integration Center (COIC) Data Fusion Operations and Capabilities: An Initial Assessment”, US Army Technical Report, December 2011.

Hoffman, R.R., N.R. Shadbolt, A.M. Burton, and G. Klein, “Eliciting Knowledge from Experts: A Methodological Analysis”, *Organizational Behavior and Human Decision Processes*, vol. 62, no. 2, pp. 129-158, May, 1995.

James, John, “Military Data”, presentation, Network Science Center, West Point, Oct 2010.

Liou, Y.I., “Knowledge Acquisition: Issues, Technologies and Methodology”, *ACM SIGMIS Database*, vol 23, no. 1, pp. 59-64, Winter 1992.

Martin, J. and S. Oxman, *Building Expert Systems: A Tutorial*. Englewood Cliffs, New Jersey: Prentice Hall, 1988.

Moskal, Michael D., Dr. Moises Sudit, Dr. Kedar Sambhoos, “Providing Analytical Rigor in Intelligence Analysis Processes Utilizing Information Fusion Based Methods”, CUBRC/University at Buffalo, Sep 2010.

North Atlantic Treaty Organizaiton (NATO) Standard Agreement 2022 (Edition 8) Annex, 1997.

Sagheb-Tehrani, M., “A Conceptual Model of Knowledge Elicitation”, *Proceedings of Conference on Information Systems Applied Research (CONISAR) 2009*, §1542, pp. 1-7, 5-8 Nov, 2009, Washington, D.C.

Waterman, D.A., *Building Expert Systems*. Reading, MA: Addison Wesley, 1983.

A Dynamic Programming Approach for Heart Segmentation in Chest Radiographs

Aarti Raheja, Jason Knapp, Chunsheng Fang
{araheja,jknapp,vfang}@riverainmedical.com

Riverain Technologies
3020 South Tech Boulevard
Miamisburg, OH 45342-4860

Abstract

Chest radiographs are the most routinely acquired exams, which makes their use for diagnosis cost effective. In this paper we present a dynamic programming approach for automated heart segmentation on posterior-anterior (PA) chest radiographs. The goal of the proposed algorithm is to provide an accurate and reproducible method for heart segmentation, which can then be used to detect certain cardiac abnormalities. Our method has several advantages over previous methods, and provides superior performance to previously published results.

Introduction

Heart segmentation in chest radiographs is a challenging task. One major difficulty in segmenting the heart is the low contrast found in the mediastinum and diaphragmatic regions. These areas are difficult to visualize even by radiologists. Other aspects that make the problem challenging include: the significant variation in heart size across patients, the presence of disease in the lungs, and poor breadth holds by patients (leading to lower contrast on the heart boundary). Despite the challenges, development of an automated method for heart segmentation could provide significant clinical value [1].

Several methods have been proposed [1][2][3] for segmenting the heart. Nakamori et al [1] discuss a method to segment the heart by detecting points along the heart boundary, which are then fitted using a Fourier shape model. This method was used in [2] to automatically compute the cardiothoracic ratio (CTR) in 400 radiographs. Out of the 400 radiographs, 20% required manual intervention. It was also shown in [3] that the heart boundaries outlined by four experienced radiologists had a high degree of variability, which is an important result when considering how to assess automatic methods.

Van Ginneken et al [4] discuss several approaches to heart segmentation: active appearance model (AAM), active shape model (ASM) and pixel classification. The individual methods performed comparably well, though significantly better performance was obtained when a hybrid voting scheme was used to combine the three methods. Shape models, such as the ASM, have the drawback that their fitting routine can get caught in local optima [5]. This effect can become quite pronounced when applied to images that differ significantly from those used to build the model. This point is particularly important in our application as abnormal hearts are precisely what we're trying to detect. For this reason, we opted for a different approach.

One important use of heart segmentation is the measurement of the cardiothoracic ratio. The CTR is an important measurement that can imply cardiomegaly (abnormally large heart) [1]. The CTR is defined as the maximum transverse diameter of the heart, divided by the maximum internal diameter of the thoracic cage [6] Research in to methods for automatic CTR extraction has a long history [6]. Later in the paper we show how the CTR can be used for assessing the quality of a heart segmentation. Although the CTR can be computed without segmenting the heart, segmentation is useful as it can help radiologists validate the result. Figure 1 illustrates the idea.

We use an algorithm based on dynamic programming (DP) to segment the heart. DP, an important algorithm in Artificial Intelligence [7], is used in applications such as finding the shortest path within a graph. DP decomposes a complicated problem into simpler sub problems; and, based on Bellman's "Principle of Optimality", the optimal solution to the original problem can be obtained by combining the solutions to each sub problem.

In the proposed algorithm we formulate the DP sub problem in an innovative way. The cost matrix is generated using image information assigning minimum cost to the pixels having heart edge characteristics. The cost matrix is generated in the polar domain since the heart shape is mostly circular. By using this method we allow the shape to vary in regions where enough information is present, but force the shape to be circular in regions of uncertainty.

In the next sections we describe our algorithm based on dynamic programming in detail, followed by a presentation of extensive experimental results and a conclusion.

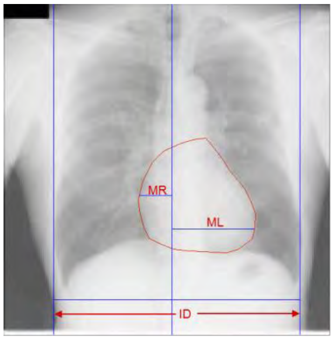


Figure 1: Chest Radiograph with heart outline showing the maximum internal diameter of the thorax (ID) and maximum transverse diameter of the heart that is the sum of maximum right heart width (MR) and maximum left heart width (ML).

Materials and Methods

We used the 247 chest radiographs from the JSRT database to test the method. The JSRT database is available publicly and consists of screen-film images digitized with a 0.175mm pixel size and 2048×2048 image size [8]. The heart annotations for this dataset [9] are available and were used to evaluate our method.

In Figure 2 a flowchart of the method is shown.

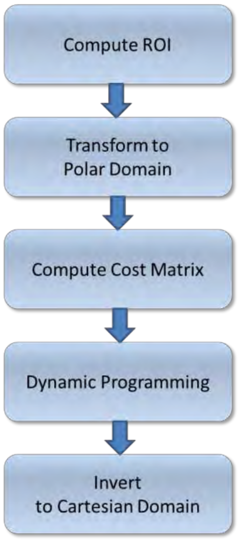


Figure 2: Algorithm Flowchart

Region of Interest around the Heart

We first obtain the ribcage mask and the segmented lung masks from the chest radiographs. This is done using a method developed by Riverain Technologies. The lung masks are then used to detect locations where the air, heart, and diaphragm intersect as shown in Figure 3. These locations are computed based on a curvature detection method as discussed in [10]. The average of these two locations, as shown in Figure 3, is used as the end row value to define an approximate bounding box around the heart region.

The top row of the bounding box, as shown in Figure 3, is selected as the location where the heart and the left lung first meet. The bounding box column locations, as shown in Figure 4, are the locations along each lung mask that are at a maximum distance from the central column.

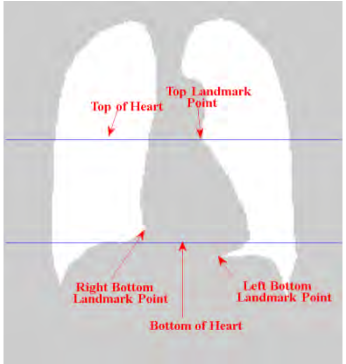


Figure 3: Landmark points computed using curvature information on the lung masks

This bounding box is used to define a center and a radius around the approximate heart region. The center is selected as the midpoint of the bounding box and the radius is selected as half of the distance between the end column locations.

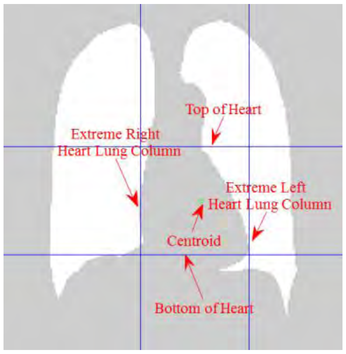

Top of Heart
Extreme Right Heart Lung Column
Extreme Left Heart Lung Column
Centroid
Bottom of Heart

Figure 4: The heart region determined using the lung mask

Polar Transform

The border of the heart is roughly circular. For this reason we apply a polar transform defined in equations (1)-(3) to the approximate heart region.

$$I(x,y) \rightarrow J(r,\theta) \tag{1}$$

$$r = \sqrt{x^2 + y^2} \tag{2}$$

$$\theta = \operatorname{atan}(y,x) \tag{3}$$

where $I(x,y)$ is the image in the Cartesian coordinate system and $J(r,\theta)$ is the image in the polar coordinate system.

The polar transform is applied to the image as shown in Figure 5(a) using the center and radius as defined in the previous section. To ensure all of the heart is included, the radius is multiplied by a factor α. In this paper we selected a α value of 1.5. The polar domain image as shown in Figure 5(b) is used to compute a cost matrix for the purpose of dynamic programming.

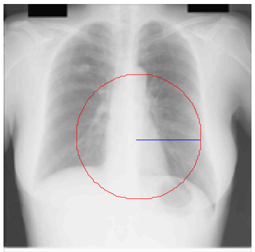


Figure 5.(a): Cartesian system image with center and radius marked for conversion in the polar domain

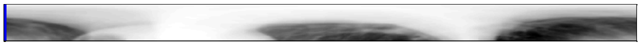


Figure 5.(b): Polar co-ordinate image expressed in terms of radial distance.

Dynamic Programming

In dynamic programming, the most important part is constructing the cost matrix. Each pixel in the cost matrix is assigned a local cost, where we use low cost values for pixels that have characteristics typical of the heart boundary. The local cost is defined as a linear combination of individual cost images:

$$local\_cost = w_{grad} * C_{grad} + w_{gsc} * C_{gsc} \qquad (4)$$

where C_{grad} is the cost based on the gradient magnitude, w_{grad} is the weight assigned to C_{grad}, C_{gsc} is the cost based on a smoothed gray scale image and w_{gsc} is the weight assigned to C_{gsc}. The gradient is calculated by computing the derivative along each column (derivative in the radial direction). The gray scale cost term is defined by first computing a nominal value for the heart-lung border. This is done for each column within a smoothed image. These nominal values are then used to measure each pixel's deviation from the expected border value. Each local cost term is scaled to the unit interval prior to combining.

Given the local cost matrix, the next step is to compute the cumulative cost. The cumulative cost accounts for both the local and transitional costs. The transitional term weights the cost of going from one pixel to the next. The transitional cost we use increases with pixel distance, thus enforcing a smoother result. The total cumulative cost matrix is defined as follows:

$$C(i,1) = local\_cost(i,1) \qquad (5)$$

$$C(i,j+1) = \min_{-k\le s\le k}\{C(i+s,j) + local\_cost(i,j+1) + T(s)\} \qquad (6)$$

where T represents the transition cost. The value "s" is the offset between pixels when going from one column to the next. The value of this offset is not allowed to be larger than a specified value, "k", depending upon the desired path smoothness. The value of k for our experiments was set to 3 pixels.

Pixels outside the lung mask, or those having cost values above a maximum acceptable threshold, are set to the maximum cost value as shown in Figure 6. This causes a straight line to be the optimum path for these regions (circular arc in Cartesian domain).

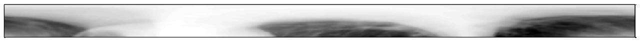


Figure 6.(a): Original Cost Image

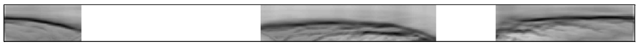


Figure 6.(b): Cost Image with non-air pixels suppressed

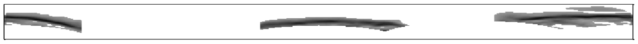
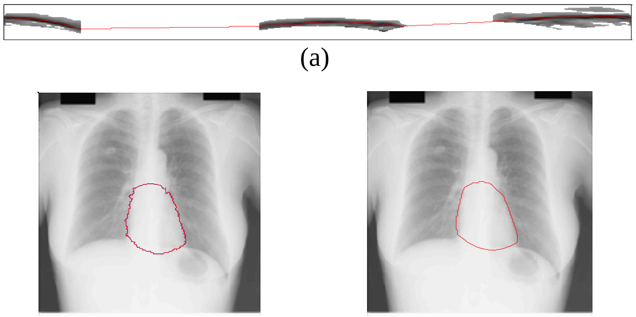


Figure 6.(c): Cost image with pixels having cost values above a maximum acceptable threshold value set to the maximum cost value

Once we obtain the optimal DP solution path, the heart segmentation is obtained by transforming the path to the Cartesian domain, as shown in Figure 7.



(a)

(b) (c)

Figure 7: (a) The optimum path obtained from dynamic programing solution is converted into (b) Cartesian co-ordinate system to obtain the heart segmentation with (c) some post processing.

Some morphological post processing is applied to make the heart shape smooth and convex, see Figure 7 for an example.

Experiments

We carried out two experiments to validate the proposed method. First, the algorithm output is compared to the manual outlines to evaluate the accuracy of the heart segmentation. In a second experiment, we compared CTR values extracted from the algorithm against those extracted from manual outlines. The specific aim of this experiment was to evaluate if a reliable CTR estimate can be obtained even with a low overlap score.

The overlap score, **Ω**, between the manually outlined heart boundary and the output of our method is defined in equation (7).

$$\Omega = \frac{TP}{TP+FP+FN} \qquad (7)$$

where TP is the true positive area, FP is the false positive area, and FN is the false negative area.

Figure 8 illustrates a summary of the overlap scores obtained by our method.

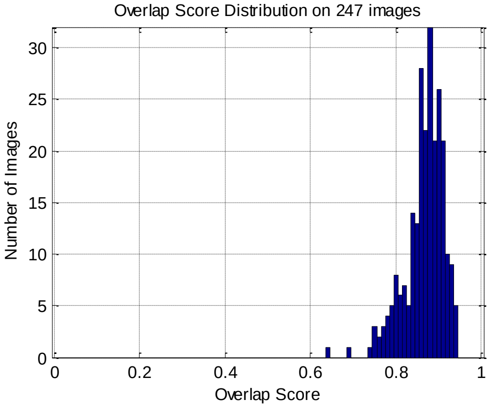


Figure 8: Histogram of overlap scores on 247 chest X ray images. Most of the overlap scores concentrate around 0.87, indicating the high accuracy of our method.

The CTR values are computed by detecting the internal diameter (ID) of the thorax and the transverse diameter of the heart (TD = MR+ML, Figure 1).

$$CTR = \frac{ID}{TD} * 100 \qquad (8)$$

The ID value was derived from the ribcage mask. The TD values were computed using the heart mask derived from the algorithm output and the manual outlines.

A relative difference between the CTR values was computed using the above TD and ID values. Figure 9 shows a scatterplot comparing the overlap score with the relative CTR measure. From this plot we can deduce that a good CTR estimate can be obtained even with a low overlap score. An example of such a case is shown in Figure 10. The reason this can occur is that the source of low overlap is generally from the mediastinal and sub diaphragmatic regions, which do not influence the transverse diameter of the heart.

Figure 11 shows the only case with a low overlap score that was not due to the mediastinal or sub diaphragmatic regions. The difficulty here is the fusion of the left lung and colon. This leads to an inaccurate estimate of the left-lower landmark intersection location, which results in significant under segmentation. Fortunately, such an occurrence is rare and is left as an area for future improvement.

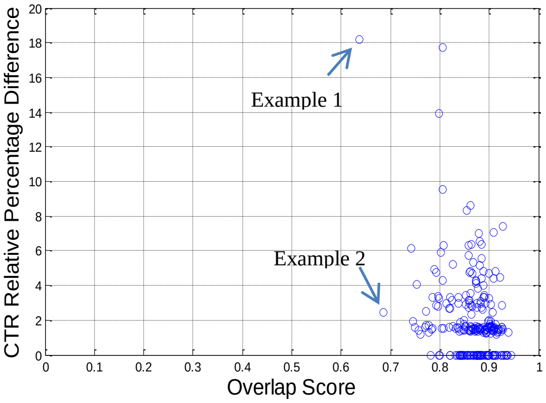
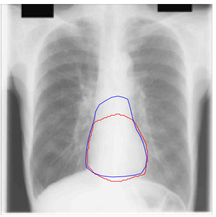


Figure 9: Scatterplot comparing the overlap score with the relative CTR measure



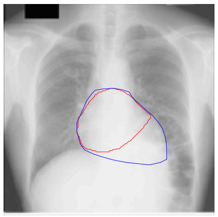


Figure 10: Example 1 with the lowest overlap score of 0.63

Figure 11: Example 2 having low overlap score, but a more accurate CTR value

Some typical output segmentations are presented in Figure 12. As can be seen, our proposed method captures the actual heart contour fairly accurately in most of the cases.

Discussion

An average overlap score of 0.867 ± 0.046 was obtained from the 247 JSRT images. We find that our method produces outputs that are close to the human observer, while comparing favorably to the other methods discussed in the survey paper [4]. The overlap scores in Table 1 are for the three hybrid methods discussed in [4]. These hybrid methods make use of multiple methods making them computationally intensive. In addition, these methods are supervised approaches whose outputs might not extend to more atypical cases. By comparison, our method is far less complex and has the advantage of making very few assumptions about the shape of the heart.

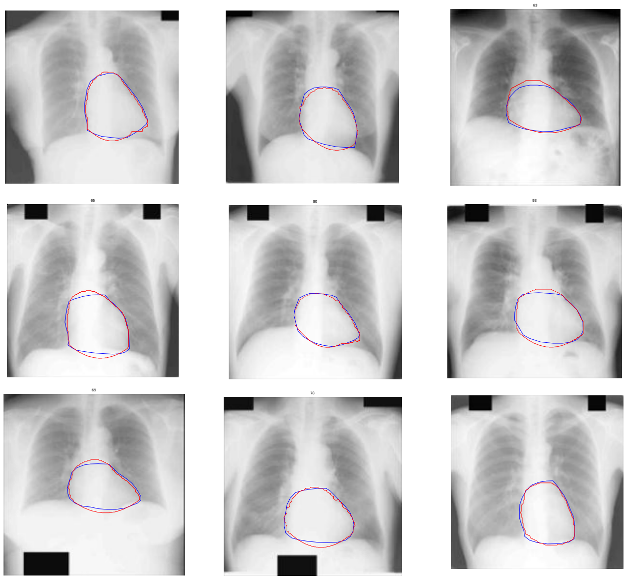


Figure 12: Heart Segmentation. Blue represents user annotation and red represents output of the current method.

Conclusion

We presented an algorithm for segmenting the heart region using dynamic programming. The proposed algorithm provided an accurate and reproducible method for heart segmentation. The presented method makes few assumptions about the heart shape, has a simple implementation, and provides superior performance to previously published results.

Future work will involve the collection of more data, which is needed for further evaluation and the development of strategies for handling outlier cases. Also, additional image features for improving the local cost term will be explored.

References

[1] N. Nakamori, K. Doi, V. Sabeti, and H. MacMohan, “Image feature analysis and computer-aided diagnosis in digital radiography: Automated analysis of sizes of heart and lung in chest images,” Med Phys., vol.17: pp. 342-350, 1990.

[2]N. Nakamori, K. Doi, H. MacMohan, Y.Sasaki, and S. Montner, “Effect of heart-size parameters computed from digital chest radiographs on detection of cardiomegaly: Potential usefulness for computer-aided diagnosis,” Investigat. Radiol., vol. 26: pp. 546-550, 1991

[3] R. Kruger, J. Townes, D. Hall, S. Dwyer, S. Lodwick. “Automated radiographic diagnosis via feature extraction and classification of cardiac size and shape descriptors,” IEEE transaction on Biomedical Engineering., vol. BME-19:pp. 174-186, 1972

[4] B. Van Ginneken, M. Stegmann, M. Loog. “Segmentation of anatomical structures in chest radiographs using supervised methods: a comparative study on a public database”, 2004.

[5] T. F. Cootes, C. J. Taylor, D. Cooper, and J. Graham. “Active shape models – their training and application,” Computer Vision and Image Understanding., vol. 61(1):pp. 38–59, 1995.

[6] H. Becker, W. Nettleton, P. Meyers, J. Sweeney, Jr CM Nice. “Digital computer determination of a medical diagnostic index directly from chest X-ray images,” IEEE Transaction on Biomedical Engineering., vol. BME-11:pp. 67-72, 1964.

[7] Stuart Russell, Artificial Intelligence: A Modern Approach .

[8] J. Shiraishi, S. Katsuragawa, J. Ikezoe, T. Matsumoto, T. Kobayashi, K. Komats, M. Matsui, H. Fujita, Y. Kodera, and K. Doi. “Development of a digital image database for chest radiographs with and without a lung nodule: Receiver operating characteristic analysis of radiologists' detection of pulmonary nodules”. AJR., vol. 174:pp. 71-74, 2000.

[9] Image Sciences Institute Research Databases. http://www.isi.uu.nl/Research/Databases/.

[10] S. Muhammad, M Asif and M. R. Asin. “A new approach to corner detection,” Computer imaging and vision, vol. 32:pp. 528-533, 2006.

Table 1: Overlap score results compared to a human observer and various methods discussed in [4].

Heart	μ±σ	min	Q1	median	Q3	max
Human Observer	0.878±0.054	0.571	0.843	0.888	0.916	0.965
Dynamic Programming	**0.867±0.0460**	**0.636**	**0.846**	**0.875**	**0.898**	**0.944**
Hybrid Voting	0.860±0.056	0.651	0.833	0.870	0.900	0.959
Hybrid ASM/PC	0.836±0.082	0.430	0.804	0.855	0.889	0.948
Hybrid AAM/PC	0.827±0.084	0.499	0.791	0.846	0.888	0.957

Investigating a Bayesian Hierarchical Framework for Feature-Space Modeling of Criminal Site-Selection Problems

Jon Fox, Samuel H. Huddleston, Matthew Gerber, Donald E. Brown
Department of Systems and Information Engineering, University of Virginia
151 Engineer's Way
Charlottesville, Virginia 22904–4747
jmf3a@virginia.edu

Abstract

A significant amount of academic research in criminology focuses on spatial and temporal event analysis. Although several efforts have integrated spatial and temporal analyses, most previous work focuses on the space-time interaction and space-time clustering of criminal events. This research expands previous work in geostatistics and disease clustering by using a Bayesian hierarchical framework to model criminals' spatial-temporal preferences for site-selection across a continuous time horizon. The development of this Bayesian hierarchical feature-space model (BHFSM) offers law enforcement personnel a method for accurate crime event forecasting while improving insight into criminal site-selection at the strategic level. We compare the BHFSM to other feature-space modeling techniques using both a long range and short range criminal event dataset collected from police reporting. While the BHFSM remains sufficiently accurate for event prediction, current computational requirements limit the applicability for "just-in-time" crime modeling.

Introduction

Although much theoretical and practical work has been done on the use of Bayesian hierarchical modeling for geostatistics and disease clustering, applications within the criminal site-selection problem have been limited. This article merges the feature-space model of Liu and Brown (1998) with the Markov random field construct of Zhu, Huang, and Wu (2006) to model the criminal's preference for initiating a crime within a specific spatial-temporal zone. By adapting theoretical and computational work from disease mapping and environmental studies, we develop a Bayesian hierarchical feature-space model (BHFSM) for the criminal event prediction problem in order to examine both parameter estimation and predictive inference. The remainder of this section provides a quick review of applicable crime theory and feature-space modeling for criminal site-selection problems. The subsequent sections discuss the Bayesian hierarchical framework, introduce the dataset used for this article, and review the performance of the BHFSM against the dataset for both a long term and a short term temporal study horizon. In the final section, we review conclusions from this initial research and propose paths for future research.

Crime Theory

Much of the work in crime studies proceeds from a frame of reference built upon the location of the crime (Townsley, Homel, and Chaseling 2000; Groff and LaVigne 2002). This frame of reference is conditioned upon Tobler's first law of geography: "everything is related to everything else, but near things are more related than distant things" (Tobler 1970). For the crime analyst, this means that if a crime happened yesterday at the corner of Main Street and Broadway, then the most likely location for a crime tomorrow is the corner of Main and Broadway. Hotspotting and crime clustering are built upon the assumption that future crimes are likely to occur at the same location as past crimes (Ratcliffe 2004; Cohen, Gorr, and Olligschlaeger 2007).

Rational criminal theory assumes that individuals have specific reasons for committing a crime at a certain time and a certain location (Clark 1980). By examining the historical criminal activity data within a spatial region, we can discover patterns that might indicate criminals' preferences for executing crimes at certain locations (Brantingham and Brantingham 1984). Spatial choice models offer analysts a methodology for identifying a criminal's preference for one site over another within a spatial region.

Spatial choice models assume an actor will select a site (e.g., for migration, retail establishment, or criminal event) based on the perceived utility, or worth, of that site from a set of alternatives (Ewing 1976; McFadden 1986). The use of spatial choice models nests well within the rational criminal theory since it assumes that spatial point processes involving actors are a result of the actors' mental processes and perceptions (Burnett 1976). This article expands on the spatial choice problem to examine the impact of both geographic and temporal features on the criminal's site-selection process.

Feature-Space and Criminal Site-Selection

This work is inspired by the following question: What if, instead of focusing on where the crime happened on the ground, we focus on where the crime initiation took place in the mind of the criminal? The idea of spatial choice presents a framework of decision processes for a rational actor to choose a location based on the perceived value of that location. Consider a criminal who wants to steal a car. Will he choose a parking garage at the center of town with restrictive traffic flows or will he choose the mall parking lot near a major freeway on the outskirts of town? Previous work has shown that the car thief will take the car from the mall since features surrounding a location are as critical as the location itself (Rengert 1997). Brown, Liu, and Xue (2001) showed that data mining previous criminal events provides insight to what spatial features might be considered by a criminal in selecting a location to commit a crime. We define this set of spatial considerations to be the feature-space. Several investigations have shown that feature-space modeling performs as well, or better, than density based methods (Brown, Dalton, and Holye 2004; Smith and Brown 2004).

Criminal site-selection is the process by which a criminal selects the time and space to execute an event based on their feature-space preferences (Porter 2006). Rather than using a latitude and longitude to describe each location in a study region, we use spatial distances to environmental features — such as schools, streets, or stadiums — and spatial representations of social demographics — such as population, percent rental properties, and household income — to examine which locations are preferred by criminals for certain types of crimes (Bannatyne and Edwards 2003; Liu and Brown 2003; Huddleston and Brown 2009).

Bayesian Hierarchical Modeling

Hierarchical models allow us to deconstruct complex problems into a series of smaller tractable problems. Using the methodology developed by Wickle (2003), we formulate the criminal site-selection problem into three basic stages: a data model, a process model, and a parameter model. Our data model accounts for our knowledge of the spatial-temporal patterns of crime within the study region. The process model provides insight to the criminal site-selection process while accounting for spatial and temporal effects. Finally, our parameter model accounts for the uncertainty in both the data and process models (Wickle 2003).

Formulation

The goal of this article is to develop a Bayesian hierarchical model that uses the feature-space methodology to accurately predict crime events across an irregular lattice while providing insight into the criminal site-selection process. To estimate the criminal's spatial preferences, our ***data model*** represents the criminal's site-selection process as a binary random variable where $Y_{s,t} \in 0, 1$ is the observation of the presences, or absence, of crime at location s at time t given a set of features $\boldsymbol{X}$.

$$Y_{s,t}|\boldsymbol{X} \sim \; Bern(\mu_{s,t}) \tag{1}$$

For our least complex model, we assume that the probability $\mu_{s,t}$ is a function of the criminal's preferences for certain features and a random effects term. Mathematically, we represent the ***process model*** as:

$$\begin{gathered}\mu_{s,t} = \operatorname{logit}^{-1}\left(\beta_0 + \beta_1 X_{s1} + \ldots + \beta_k X_{sk} + \theta_{s,t}\right), \\ \text{for } s = 1, ..., S \text{ and} \\ \text{for } t = 1, ..., T\ .\end{gathered} \tag{2}$$

Equation 2 uses a set of features $\boldsymbol{X}$ as a vector of length k for each location s combined with the estimated $\boldsymbol{\beta}$ values from the parameter model to estimate the probability $\mu_{s,t}$. For this article, we use a set of demographic variables to represent a portion of the feature-space considered by the criminal in their site-selection process. Analyzing previous criminal event data gives us a method to account for the criminal's site-selection process. By modeling the relationship between the features and the probability of crime, we estimate the preferences criminals have for locations with a specific set of features. However, just as the criminal's preferences for certain locations might change depending on proximity to freeways or vacant houses, the criminal site-selection process can also change depending on the time of day or other seasonal events (Rossmo, Laverty, and Moore 2005; Gorr 2009a). The variable $\theta_{s,t}$ provides a method for including other random effects.

The first random effect considered is the temporal component. We consider a temporal effect $g_t \sim N(g_{t-1}, \tau_g)$. Based on previous research (Gorr, Olligschlaeger, and Thompson 2003; Eck et al. 2005), we believe that criminal activity often preceeds criminal activity. Using this temporal component allows us to account for periods of criminal activity that match the routine activities and population dynamics of the study region. We will discuss the inital conditions for the variance estimates in the parameter model.

We use a Markov random field (MRF) construct as the second random effect by assuming that the likelihood of a crime at a specific location is dependent only on its neighbors and its previous temporal state (Zhu, Huang, and Wu 2006). Recent work on point processes uses MRFs as a secondary structure that results from an aggregation process of event counts. For our crime data, we construct the MRF along an irregular lattice structure defined by political and cultural boundaries using the construct provided by Illian et al. (2008). We consider a MRF effect that accounts for the past value at the location s and the second-order neighbors such that $\omega_s \sim N(\omega_{j-1}, \tau_o)$. The index j accounts for the second-order neighbors of location s. The inclusion of the neighborhood spatial effects gives us a method to include criminal repeat information into the feature-space model. Studies on criminal repeats have shown that for short temporal intervals, locations that have experienced crime have an increased likelihood for repeat victimization (Townsley, Homel, and Chaseling 2000).

The third random effect considered for this article is an interaction term. We consider an interaction term $\psi_{s,t} \sim N(0, \tau_p)$. The interaction term is uncorrelated but can identify potential spatial-temporal interactions within the data that are not accounted for in the base feature-space model (Lawson 2009). The final random effect is an uncorrelated error term $v_s \sim N(0, tau_v)$ that accounts for any uncorrelated spatial components of the criminal site-selection process. The research design section outlines the four primary models considered for this article using different combinations of these random effects.

Finally, we specify the ***parameter models*** by establishing the initial distributions for the parameters. As seen in Figure 1, the $\boldsymbol{\beta}$ vector appears in the process model. However, we provide initial estimates for the individual βs within the parameter model. Estimating the $\boldsymbol{\beta}$ values increases the complexity of the parameter model, since for both the long term and short term data study, we initially estimate each β for each feature during the model fitting phase. In order to reduce the computational requirements, we substitute a feature-space prior calculated from linear model regression (Lunn et al. 2000). The initial assumptions for the parameter model follow:

$$\begin{gathered}\beta \sim N(\hat{\beta}, \tau_b)\\ \tau_b \sim N(0, svb), svb \sim U(0,10)\\ \tau_u \sim N(0, svu), svu \sim U(0,10)\\ \tau_g \sim N(0, svg), svg \sim U(0,10)\\ \tau_o \sim N(0, svo), svo \sim U(0,10)\\ \tau_p \sim N(0, svp), svp \sim U(0,10)\end{gathered} \tag{3}$$

The parameter model sets the initial conditions for the simulation methods used to estimate the process and data model and completes the model hierarchy (Wickle 2003). More details on the simulation methods can be found in (Lawson 2009; Kery 2010).

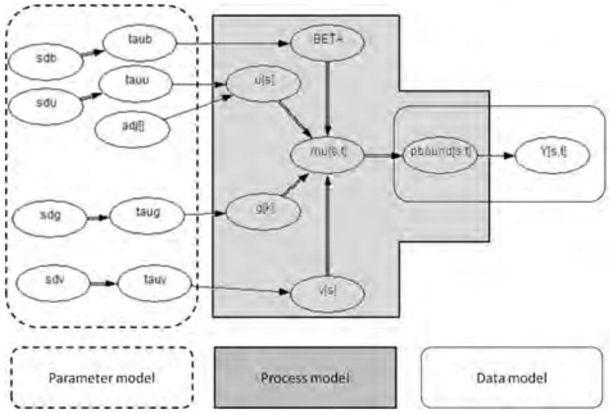


Figure 1: Directed acylic graph for Bayesian hierarchical feature-space model.

Bayesian methods provide a means to calculate the posterior distribution from our three stage-hierarchical model. Using the example from Wickle (2003), our posterior distribution is proportional to our data model conditioned upon the process and parameter models times the process model conditioned upon the parameters:

$$\begin{gathered}[process, parameters|data] \propto\\ [data|process, parameters] \times\\ [process|parametersl][parameters]\end{gathered} \tag{4}$$

Since our goals in modeling criminal site-selection problems include both predictive inference and parameter understanding, we desire to solve for the left hand side of Equation 4. However, the complexity of the posterior distribution makes obtaining a closed form solution almost, if not completely, unobtainable. Using simulation methods, built upon empirical knowledge from the data and expert knowledge on the prior distributions, we obtain samples that provide estimates of our target variables (Lawson 2009).

Research design

The Bayesian hierarchical feature-space model (BHFSM) is a limited feature-space logistic regression model with an auto-regression on the state of the neighboring locations across an irregular lattice at discrete temporal intervals. Following work from disease mapping and geostatistics, we examine four models of random effects for our variable $\theta_{s,t}$. The models considered provide several methods for including other random effects (Lawson 2009). The four models considered for random effects include:

- A time-varying trend g_t plus an uncorrelated error v_s
- A Markov random field ω_s accounting for the sum of the neighboring effects at a previous time plus v_s
- g_t plus ω_s plus v_s
- g_t plus ω_s plus v_s and an interaction term $\psi_{s,t}$

Figure 1 displays a graphical representation of the third model developed for this article without an interaction term.

Study dataset

The primary source of data for this article is an incident database for the city of Charlottesville, Virginia. We sample the complete dataset to develop a subset that contains a time horizon spanning four years with over 2,000 incidents. We restrict the crime types analyzed for this article to assaults, both simple and aggravated. We drape an irregular lattice over the study area and aggregate the criminal incidents at the daily level. Although the aggregation introduces some level of discreteness, we treat the temporal intervals as continuous points along the temporal horizon. The irregular lattice structure is based on the thirty-seven US Census block-groups for the city. Using the this lattice structure facilities inclusion of demographic information at the block-group level. We use the census information as proxies for complex factors that actually affect criminals. We are not claiming that a criminal actually considers the percent of houses in area that are rentals when deciding to execute a crime. However, the percentage of rental houses in an area might correlate with other factors that are part of the criminal site-selection process. Figure 2 depicts the study region draped with the irregular lattice and shows spatial-temporal patterns of assaults over four distinct temporal intervals. The analysis that follows uses a second-order neighbor structure over the irregular lattice depicted in Figure 2.

We set $Y_{i,t} = 1$ if a criminal assault occurs within the specified block-group $i = 1, ..., 37$ during one of the days $t = 1, ..., 1095$ of the study horizon. The block-group and daily aggregation results in a 37×365 matrix for a total of 40,515 observations in space-time. Figure 3 depicts a one year snapshot of criminal events across the entire spatial region.

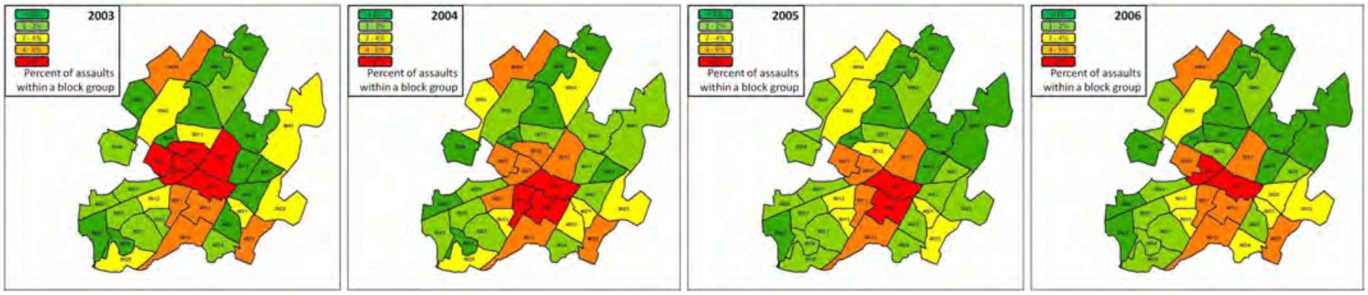

2003
2 - 4%
4 - 5%
Percent of assaults within a block group
2004
Percent of assaults within a block group
2005
Percent of assaults within a block group
2006
Percent of assaults within a block group

Figure 2: Evolution of spatial patterns over continuous temporal horizon. Since we identify changes in the map over time, we hypothesize that we have spatial and temporal effects within the criminal site-selection process.

Model comparison

For this article, we compare each model's predictive performance against a test set from the dataset. For the long term study, we use a 365 day temporal window for model fitting and then evaluate against a ninety day test. For the short term study, we use a thirty day temporal window surrounding special events in Charlottesville for model fitting and then evaluate against the thirty day temporal window surrounding the same special event in the following year.

Prior to comparing predictive performance, we use a goodness of fit measure to evaluate each model. Borrowing from conventional generalized linear modeling, we use deviance as a measure of how well the model fits the data. In the software used for this article, we can expect the deviance to decrease by 1 for each predictor added to the model (Gelman and Hill 2007).

As an additional method for comparing goodness of fit, we use the mean squared predictive error (MSPE). Given our known spatial-temporal dataset from the test period, $\boldsymbol{Y}$, our estimated spatial-temporal dataset, $\hat{\boldsymbol{Y}}$, and a number of observations m from a simulation sample of G, we use Lawson's (2009) formulation such that:

$$MSPE = \frac{|\boldsymbol{Y} - \hat{\boldsymbol{Y}}|^2}{(G \times m)} \tag{5}$$

One of the challenges for spatial-temporal data is selecting an appropriate statistical measure for examining model performance. Originally used to assess radar performance in World War II, the receiver operating characteristic (ROC) curve are particularly useful for evaluating the ability of a model to predict the occurrence of an event accurately while minimizing the number of false positive predictions (Bradley 1997; Swets, Dawes, and Monahan 2000). Similiar to the ROC curve, the surveillance plot provides a method for evaluating model performance in spatial-temporal classification problems. The surveillance plot gives the analyst a method for monitoring the amount of area within the study region that needed to be observed in order to identify the highest percentage of crimes (Huddleston and Brown 2009; Kewley and Evangelista 2007). Using a contingency table, or decision matrix, similar to Table 1, we record the possible outcomes of prediction estimated with the model being considered against the true conditions observed in the test set.

Table 1: **Contingency Table**

	True Condition		
Test Result	Positive	Negative	Measures
Positive	TP	FP	TP + FP
Negative	FN	TN	FN + TN
Measures	TP + FN	FP + TN	

We build the surveillance plot by plotting the rate of accurate crime predictions against the rate of crime incidents predicted where crimes did not occur. Although the surveillance plot provides a measure for comparing model performance visually, translating the surveillance plot into a numerical measure provides a method for comparing the performance of multiple models against a common test set. A model with high accuracy — predicting all the crime locations perfectly — would have a ratio of all true positives versus zero false positives while a model with an equal ratio of true positives and false positives is basically guessing (Bradley 1997; Swets, Dawes, and Monahan 2000).

$$PLR = \frac{n \times TP}{(TP + FN) \times (TP + FP)} \tag{6}$$

The performance limit ratio (PLR) measures the model's trade-off in accuracy and precision by focusing on the model's better-than-chance ratio (Gorr 2009b) of correctly predicting crimes within a test set of size n. A model that is more accurate in predicting crimes across the space-time surface will have a higher PLR. Rather than focusing on the entire area under the curve, we reduce the focus to the first 20% of the space-time surface observed while discounting the area under the curve that accounts for random guessing.

Long Term Study Results

For the long term study, the block-group and daily aggregation results in a 37×365 matrix for a total of 13,505 observations in space-time. We use a second-order neighbor model

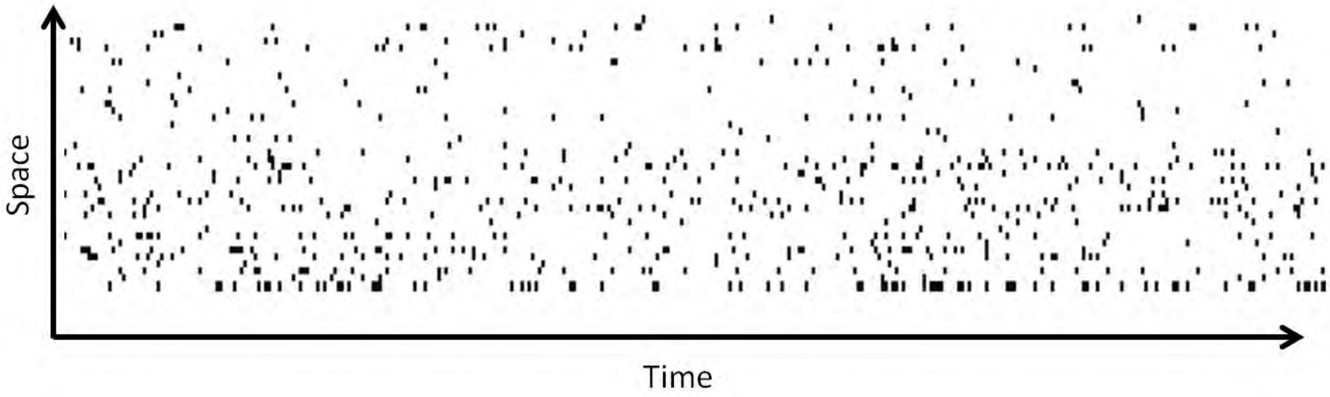


Figure 3: Space-time lattice for study domain. The arrangement of spatial regions along the *y* axis might falsely identify spatial clusters, however, the temporal horizon along the *x* axis does allow for visual identification of temporal clusters for assaults within the study region. When using the surveillance plot for measuring model performance, we iteratively evaluate all spatial locations within each temporal interval.

to account for all the criminal activity in all the surrounding census blocks. Table 2 outlines the specific models examined using both the demographic features and a feature-space prior obtained from a generalized linear regression similar to the work of (Liu and Brown 2003). As discussed in the research design section, we consider four alternatives to model the random effects using our variable $\theta_{s,t}$ in the process model: 1) a time-varying trend; 2) a Markov random field accounting for the sum of the neighboring effects at a previous time; 3) a time-varying trend with a Markov random field; 4) a time-varying trend with a Markov random field and an interaction term. For every alternative we include a space-time independent noise term. For the first four alternatives, we attempt to account for the criminal site-selection preference by modeling $\boldsymbol{\beta}$ as seen in Figure 1. After model fitting, we evaluate performance using the MSPE discussed above.

Although the predictive performance of the BHFSM is not significantly better than the base feature-space model, we were expecting to see significant lift in the parameter estimation related to identifying criminal preferences for certain spatial features. In fact, even with all four models converging, the only feature-space variable with significantly better estimation was the preference for areas with high percent vacancy. However, (Lawson 2009) shows that the combination of spatially-referenced explanatory variables within a Markov random field construct often yields poor estimates of the regression coefficients and produces computational challenges related to multi-collinearity. Both of our approaches to reduce the impact of correlation created additional challenges. First, removing the features that are spatially dependent limits our insight into the criminal site-selection process for identifying feature-space preferences. Second, introducing new variables that have a stationary spatial attribute but are non-stationary temporally limits our ability to identify how the criminal's feature-space preferences evolve over time. Overall, the Bayesian approach offers promise for reducing uncertainty in the predictive surfaces. However, as discussed in (Withers 2002; Zhu et al. 2008), the computational time required for sampling from the posterior distribution for Bayesian inference for criminal site-selection problems is a major drawback. We discuss an alternative approach in the conclusion that offers computational advantages while remaining sufficiently accurate for prediction. In the next section, we scale down the horizon of the study period as an additional step in examining the BHFSM.

Short Term Study Results

Although applying the Bayesian framework to the long term study data did not result in significant gains in predictive performance, the initial disappointment was not entirely unexpected. Previous research shows that spatial-temporal analysis focused on criminal site-selection requires focused efforts on periods of temporal transition and local knowledge of the environment (Kerchner 2000; Bernasco and Block 2009). A more appropriate methodology for including temporal information into the BHFSM reduces the scope of the temporal horizon to those intervals with the greatest variance in crime rates. Research has also shown that spatial regions experience great variance in crime rates for certain locations depending on the temporal proximity to special events (Cohen, Gorr, and Olligschlaeger 2007). Reducing the temporal horizon to a smaller scale — such as a thirty day window before and after large spikes in crime rates — makes it easier to examine the impact of these special events on the criminal site-selection process. More importantly, including additional data from local law enforcement personnel takes advantage of their local knowledge of the temporal environment (Cressie and Wikle 2011).

As with the long term study, we consider all four alternatives to model the random effects using our variable $\theta_{s,t}$ in the process model. Table 3 outlines the specific models ex-

Table 2: Bayesian Hierarchical Feature-Space Model Development for Long Term Data Study

Model	Predictors	Time	Deviance	MSPE
Spatial Choice and Trend	35	3049	4632	0.0430
Spatial Choice and MRF	35	2587	4622	0.0429
Spatial Choice and MRF and Trend	36	4694	4625	0.0429
Spatial Choice and MRF and Trend and Interaction	43	19481	4609	0.0428
Feature-Space Prior and Trend	31	2273	4785	0.0435
Feature-Space Prior and MRF	30	2314	4632	0.0430
Feature-Space Prior and MRF and Trend	38	7219	4614	0.0429
Feature-Space Prior and MRF and Trend and Interaction	40	9515	4619	0.0429

Table 3: Bayesian Hierarchical Feature-Space Model Development for Short Term Data Study

Model	Predictors	Time	Deviance	MSPE	PLR
Feature-Space Model	7	5	423	0.0479	0.46
Spatial Choice and Trend	19	182	423	0.0479	0.52
Spatial Choice and MRF	19	303	423	0.0479	0.53
Spatial Choice and MRF and Trend	22	332	422	0.0478	0.53
Spatial Choice and MRF and Trend and Interaction	23	900	421	0.0477	0.53
Feature-Space Prior and Trend	8	180	420	0.0477	0.49
Feature-Space Prior and MRF	8	138	420	0.0477	0.47
Feature-Space Prior and MRF and Trend	10	371	419	0.0476	0.48
Feature-Space Prior and MRF and Trend and Interaction	12	1022	417	0.0475	0.50

amined using both the demographic features and a feature-space prior obtained from a generalized linear regression similar to the work of (Liu and Brown 2003) and as seen in our visual graph from Figure 1. Again, the only feature-space variable with significantly better estimation was the preference for areas with high percent vacancy. After model fitting, we evaluate performance using the PLR discussed above. While each BHFSM performs better than the base feature-space model, the computational time required for sampling from the posterior distribution for Bayesian inference is still several orders of magnitude greater than the time required for using generalized linear regression on the base feature-space model.

Conclusions

For city-wide, or regional-level, crime monitoring, the BHFSM offers a methodology for modeling criminal activity across continuous time. For this article, we added a Bayesian framework to the base feature-space model to include variables that account for both spatial *and* temporal patterns within the criminal site-selection process. We applied this methodology to both a long term and short term data study for criminal events in a small US city. Using data aggregated at the census block-group level for a medium temporal resolution, the BHFSM allowed us to model an actor's spatial-temporal preferences within a limited temporal period. Incorporating elements of the feature-space methodology into the Bayesian construct allowed us to blend the benefits gained from understanding multiple covariates within the actor's spatial-temporal decision process with the basic elements of geographic recency and spatial dependence found in hotspot modeling. Although the overall predictive performance is not significantly improved, by reducing the variance on estimates for a criminal's feature-space preferences, we gain understanding into the temporal variations of the criminal site-selection process. Enhanced understanding of the criminal site-selection process allows law enforcement personnel to adjust resource allocation strategies to better mitigate short term changes in the criminal site-selection process.

Several challenges remain for further consideration of the Bayesian framework for feature-space modeling of the criminal's site-selection process. The methodology examined in this article is computationally intensive. Although the BHFSM did provide improvement in predictive performance over the base feature-space model for the short term data study, the increased computational requirements hinder the application of the BHFSM for "just-in-time" crime modeling. Extending the Bayesian framework for modeling data at either a finer temporal or spatial resolution would increase the computational complexity since the size of the spatial-temporal event matrix is a multiple of the temporal intervals and the spatial dimensions. Future work will attempt to reduce this computational complexity by adding temporal and neighborhood indicator functions to the base feature-space model (Diggle, Tawn, and Moyeed 1998). Using indicator functions allows for faster sampling from the data while still accounting for temporal preferences in the criminal's site-selection process.

Structural vector autoregressive models (SVARs) show promise for forecasting employment rates given spatially based economic indicators (Rickman, Miller, and McKenzie 2009). Using an SVAR construct for modeling criminal site-selection might improve predictive ability if temporal changes in other features affect a criminal's temporal considerations for certain sites. However, the computational requirements for SVARs, like the Bayesian construct, are still rather demanding (Petris, Petrone, and Campagnoli 2009).

The social sciences offer another approach for reducing the computational demands of criminal site-selection modeling. Spatial-temporal designs for environmental research often include panel methods for monitoring and detecting temporal patterns and spatial relationships (Dobbie, Henderson, and Stevens 2008). We are not designing a method for collecting criminal event data, but rather examining historical collections of crime data. And as mentioned above, studies at fine temporal and spatial resolutions require a large spatial-temporal event matrix. Using a variation of stratified sampling (Gilbert 1987; Dobbie, Henderson, and Stevens 2008) on the spatial-temporal event matrix might reduce the computational time while retaining comparable predictive performance.

References

Bannatyne, J. C., and Edwards, H. P. 2003. A Bayesian explorations of the relationship between crime and unemployment in New Zealand for the time period: 1986-2002. In *International Workshop on Bayesian Data Analysis*.

Bernasco, W., and Block, R. 2009. Where offenders choose to attack: A discrete choice model of robberies in chicago. *Criminology* 93–130.

Bradley, A. P. 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition* 30:1145–1159. Retrieved July 2010.

Brantingham, P. J., and Brantingham, P. L. 1984. *Patterns in Crime*. New York: Macmillan Publishing Company.

Brown, D. E.; Dalton, J.; and Holye, H. 2004. Spatial forecast methods for terrorist events in urban environments. In *Proceedings of the Second NSF/NIJ Symposium on Intelligence and Security Informatics*. Heidelberg: Springer-Verlag.

Brown, D. E.; Liu, H.; and Xue, Y. 2001. Mining preferences from spatial-temporal data. In *Proceedings of the SIAM Conference*. Chicago: Society for Industrial and Applied Mathematics.

Burnett, P. 1976. Behavioral geography and the philosophy of mind. In Golledge, R., and Rushton, G., eds., *Spatial Choice and Spatial Behavior*. Columbus: Ohio State University Press. pp. 23–50.

Clark, R. V. 1980. Situational crime prevention: Theory and practice. *British Journal of Criminology* 136–147.

Cohen, J.; Gorr, W.; and Olligschlaeger, A. 2007. Leading indicators and spatial interactions: a crime forecasting model for proactive police deployment. *Geographical Analysis* 39:105–127.

Cressie, N. A., and Wikle, C. K. 2011. *Statistics for Spatial-Temporal Data*. Hoboken, New Jersey: John Wiley and Sons.

Diggle, P. J.; Tawn, A. J.; and Moyeed, R. A. 1998. Model-based geostatistics. *Applied Statistics* 299–350.

Dobbie, M. J.; Henderson, B. L.; and Stevens, D. L. 2008. Sparse sampling: Spatial design for monitoring stream networks. *Statistics Surveys* 2:113–153.

Eck, J. E.; Chainey, S.; Cameron, J. G.; Leitner, M.; and Wilson, R. E. 2005. Mapping crime: Understanding hot spots. Technical report, National Institute of Justice.

Ewing, G. O. 1976. Environmental and spatial preferences of interstate migrants in the United States. In Golledge, R. G., and Rushton, G., eds., *Spatial Choice and Spatial Behavior*. Columbus: Ohio State University Press. pp. 250–270.

Gelman, A., and Hill, J. 2007. *Data Analysis using Regression and Multilevel / Hierarchical Models*. Cambridge: Cambridge University Press.

Gilbert, R. O. 1987. *Statistical Methods for Environmental Pollution Monitoring*. New York: John Wiley and Sons.

Gorr, W.; Olligschlaeger, A.; and Thompson, Y. 2003. Short term forecasting of crime. *International Journal of Forecasting* 19:579–594.

Gorr, W. L. 2009a. Cloudy with a chance of theft. *Wired*.

Gorr, W. L. 2009b. Forecast accuracy measures for exception reporting using receiver operating characteristic curves. *International Journal of Forecasting* 25(1):48–61.

Groff, E. R., and LaVigne, N. G. 2002. Forecasting the future of predictive crime mapping. In Tilley, N., ed., *Analysis for Crime Prevention*, volume 13 of *Crime Prevention Series*. Monsey, NY: Lynne Rienner Publishers. 29–57.

Huddleston, S., and Brown, D. E. 2009. A statistical threat assessment. *Systems, Man, and Cybernetics, Part A.* 39:1307–1315.

Illian, J.; Penttinen, A.; Stoyan, H.; and Stoyan, D. 2008. *Statistical Analysis and Modeling of Spatial Point Patterns*. West Sussex: John Wiley and Sons Ltd.

Kerchner, S. H. 2000. Spatial-temporal event prediction. Master's thesis, University of Virginia, Charlottesville.

Kery, M. 2010. *Introduction to WinBUGS for Ecologists*. Amsterdam: Elsevier.

Kewley, R. H., and Evangelista, P. 2007. Evaluating machine learning methods for geospatial prediction problems. being prepared for submission to IEEE.

Lawson, A. B. 2009. *Bayesian Disease Mapping*. Boca Raton: CRC Press.

Liu, H., and Brown, D. E. 1998. Spatial-temporal event prediction: A new model. In *Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics*, volume 3, 2933–2937. San Diego: IEEE.

Liu, H., and Brown, D. E. 2003. Criminal incident prediction using a point-pattern based density model. *International Journal of Forecasting* 19:603–622.

Lunn, D.; Thomas, A.; Best, N.; and Spiegelhalter, D. 2000. Winbugs – a Bayesian modelling framework: concepts, structure, and extensibility. *Statistics and Computing* 10:325–337.

McFadden, D. 1986. The choice theory approach to market research. *Marketing Science* 5:275–297.

Petris, G.; Petrone, S.; and Campagnoli, P. 2009. *Dynamic Linear Models with R*. Dordrecht: Springer.

Porter, M. D. 2006. *Detecting Space Time Anomalies in Point Process Models of Intelligent Site Selection*. Ph.D. Dissertation, University of Virginia, Charlottesville.

Ratcliffe, J. H. 2004. The hotspot matrix: A framework for the spatio-temporal targeting of crime reduction. *Police Practice and Research* 5:5–23.

Rengert, G. F. 1997. Auto theft in central philadelphia. In Homel, R., ed., *Policing for Prevention: Reducing Crime, Public Intoxication and Injury*, volume 7 of *Crime Prevention Series*. Monsey, NY: Lynne Rienner Publishers. 199–219.

Rickman, D. S.; Miller, S. R.; and McKenzie, R. 2009. Spatial and sectoral linkages in regional models: A bayesian vector autoregression forecast evaluation. *Papers in Regional Science* 88(1):29–41.

Rossmo, K.; Laverty, I.; and Moore, B. 2005. Geographic profiling for serial crime investigation. In Wang, F., ed., *Geographic Information Systems and Crime Analysis*. Hershey: Idea Group Publishing. 137–152.

Smith, M. A., and Brown, D. E. 2004. Hierarchical choice modeling of terror attack site selection. *Decision Support Systems*.

Swets, J. A.; Dawes, R. M.; and Monahan, J. 2000. Better decisions through science. *Scientific American* 283(4):82–87.

Tobler, W. 1970. A computer movie simulating urban growth in the detroit region. *Economic Geography* 46(2):234–240.

Townsley, M.; Homel, R.; and Chaseling, J. 2000. Repeat burglary victimisation: Spatial and temporal patterns. *Australian and New Zealand Journal of Criminology* 33(1):37–63.

Wickle, C. K. 2003. Hierarchical bayesian models for predicting the spread of ecological processes. *Ecology* 84(6):1382–1394.

Withers, S. D. 2002. Quantitative methods: Bayesian inference, bayesian thinking. *Progress in Human Geography* 26:553–566.

Zhu, J.; Zheng, Y.; Carroll, A. L.; and Aukema, B. H. 2008. Autologistic regression analysis of spatial-temporal binary data via monte carlo maximum likelihood. *Journal of Agricultural, Biological and Environmental Statistics* 13:84–98.

Zhu, J.; Huang, H. C.; and Wu, J. 2006. Modeling spatial-temporal binary data using markov random fields. Department of Statistics.

Poster Session 2

Undergraduate poster session without articles

Andrew Pope, Cory Scott, Matthew Whitehead, d00dWorld: An Extensible Artificial Life Simulation in Python and SQL

Sam Johnson, Matthew Whitehead, The Automatic Summarization of News Articles Using Natural Language Processing and Heuristics

Abdallah A. Mohamed, Roman V. Yampolskiy, Using Wavelet Eigenfaces for Recognizing Avatar Faces

Authors Index