Vicenç Torra
Yasuo Narukawa
Guillermo Navarro-Arribas
David Megías (Eds.)

LNAI 8234

Modeling Decisions for Artificial Intelligence

10th International Conference, MDAI 2013
Barcelona, Spain, November 2013
Proceedings

Springer

Lecture Notes in Artificial Intelligence 8234

Subseries of Lecture Notes in Computer Science

Vicenç Torra Yasuo Narukawa
Guillermo Navarro-Arribas David Megías (Eds.)

Modeling Decisions for Artificial Intelligence

10th International Conference, MDAI 2013
Barcelona, Spain, November 20-22, 2013
Proceedings

Springer

Volume Editors

Vicenç Torra
IIIA-CSIC, Bellaterra, Catalonia, Spain
E-mail: vtorra@iiia.csic.es

Yasuo Narukawa
Toho Gakuen, Tokyo, Japan
E-mail: nrkwy@ybb.ne.jp

Guillermo Navarro-Arribas
Universitat Autonoma de Barcelona, Bellaterra, Catalonia, Spain
E-mail: guillermo.navarro@uab.cat

David Megías
Universitat Oberta de Catalunya, Barcelona, Catalonia, Spain
E-mail: dmegias@uoc.edu

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-41549-4 e-ISBN 978-3-642-41550-0
DOI 10.1007/978-3-642-41550-0
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013950267

CR Subject Classification (1998): I.2.3, I.2.6, I.5.2-3, H.3.4-5, G.1.6, G.1.9, F.2.1, H.2.8

LNCS Sublibrary: SL 7 – Artificial Intelligence

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This volume contains papers presented at the 10th International Conference on Modeling Decisions for Artificial Intelligence (MDAI 2013), held in Barcelona, Catalonia, Spain, November 20–22. This conference followed MDAI 2004 (Barcelona, Catalonia, Spain), MDAI 2005 (Tsukuba, Japan), MDAI 2006 (Tarragona, Catalonia, Spain), MDAI 2007 (Kitakyushu, Japan), MDAI 2008 (Sabadell, Catalonia, Spain), MDAI 2009 (Awaji Island, Japan), MDAI 2010 (Perpinyà, Catalonia, France), MDAI 2011 (Changsha, China), and MDAI 2012 (Girona, Catalonia, Spain) with proceedings also published in the LNAI series (Vols. 3131, 3558, 3885, 4617, 5285, 5861, 6408, 6820, and 7647).

The aim of this conference was to provide a forum for researchers to discuss theory and tools for modeling decisions, as well as applications that encompass decision making processes and information fusion techniques.

The organizers received 40 papers from 17 different countries, from Europe, Asia, and America, 24 of which are published in this volume. Each submission received at least two reviews from the Program Committee and a few external reviewers. We would like to express our gratitude to them for their work. The plenary talks presented at the conference are also included in this volume.

The conference was supported by the Universitat Oberta de Catalunya, Universitat Autònoma de Barcelona, the Catalan Association for Artificial Intelligence (ACIA), the European Society for Fuzzy Logic and Technology (EUSFLAT), the Japan Society for Fuzzy Theory and Intelligent Informatics (SOFT), the UNESCO Chair in Data Privacy, the Spanish MINECO (TIN2011-15580-E), and the Spanish MEC (ARES - CONSOLIDER INGENIO 2010 CSD2007-00004).

September 2013

Vicenç Torra
Yasuo Narukawa
Guillermo Navarro-Arribas
David Megías

Modeling Decisions for Artificial Intelligence – MDAI 2013

General Chairs

Guillermo Navarro-Arribas	Universitat Autònoma de Barcelona, Spain
David Megías	Universitat Oberta de Catalunya, Spain

Program Chairs

Vicenç Torra	IIIA-CSIC, Catalonia, Spain
Yasuo Narukawa	Toho Gakuen, Japan

Advisory Board

Bernadette Bouchon-Meunier	Computer Science Laboratory of the University Paris 6 (LiP6), CNRS, France
Didier Dubois	Institut de Recherche en Informatique de Toulouse (IRIT), CNRS, France
Lluis Godo	IIIA-CSIC, Spain
Kaoru Hirota	Tokyo Institute of Technology, Japan
Janusz Kacprzyk	Systems Research Institute, Polish Academy of Sciences, Poland
Sadaaki Miyamoto	University of Tsukuba, Japan
Michio Sugeno	European Centre for Soft Computing, Spain
Ronald R. Yager	Machine Intelligence Institute, Iona Collegue, USA

Program Committee

Gleb Beliakov	Deakin University, Australia
Gloria Bordogna	Consiglio Nazionale delle Ricerche, Italy
Tomasa Calvo	Universidad Alcala de Henares, Spain
Susana Díaz	Universidad de Oviedo, Spain
Josep Domingo-Ferrer	Universitat Rovira i Virgili, Spain
Jozo Dujmovic	San Francisco State University, USA
Katsushige Fujimoto	Fukushima University, Japan
Michel Grabisch	Université Paris I Panthéon-Sorbonne, France
Enrique Herrera-Viedma	Universidad de Granada, Spain

Supporting Institutions

Universitat Oberta de Catalunya
Universitat Autònoma de Barcelona
The Catalan Association for Artificial Intelligence (ACIA)
The European Society for Fuzzy Logic and Technology (EUSFLAT)
The Japan Society for Fuzzy Theory and Intelligent Informatics (SOFT)
The UNESCO Chair in Data Privacy
The Spanish MEC (ARES - CONSOLIDER INGENIO 2010 CSD2007-00004)
The Spanish MINECO (TIN2011-15580-E)

Invited Talks
(Abstracts)

A Way to Choquet Calculus

Michio Sugeno

European Centre for Soft Computing,
33600 Mieres, Asturias, Spain
michio.sugeno@gmail.com

Abstract. The aim of this talk is to suggest Choquet Calculus as a new research paradigm. We deal with Choquet integrals on the non-negative real line. The Choquet integral is a non-additive integral of a function with respect to the fuzzy measure (or capacity) derived from the Choquet functional. Most of the previous studies on Choquet integrals have been devoted to a discrete case. The calculation of discrete Choquet integrals is quite easy. It is, however, not the case for continuous Choquet integrals. We begin with a representation theorem of the Choquet integral of a non-negative, continuous and increasing function with respect to a general fuzzy measure. Then, restricting fuzzy measures to a class of distorted Lebesgue measures, we consider Choquet integral equations. A distorted Lebesgue measure is a fuzzy measure generated by a monotone transformation of the Lebesgue measure with an increasing function called 'generator'. For the distorted Lebesgue measure, it is shown that the Choquet integral equation is formulated as the Volterra integral equation of the first kind. For a case where the integrand is not increasing, we suggest a method of 'increasing arrangement' by which a non-increasing function can be transformed to an increasing function equivalent up to the Choquet integral.

Concerning the Choquet integral equation with respect to the distorted Lebesgue measure, we pose three problems: (i) calculating Choquet integrals, (ii) solving Choquet integral equations and (iii) identifying fuzzy measures. It is found that all the problems can be solved by applying the Laplace transformation. Through the processes solving these problems, we present a way to Choquet calculus. First, a concept of the derivatives of functions with respect to fuzzy measures is introduced to solve Choquet integral equations where the differentiability of functions with respect to fuzzy measures is discussed. We show basic properties of the derivative in contrast with those of the Choquet integral. Then, we consider differential equations with respect to distorted Lebesgue measures. In order to solve them, we introduce a concept 'm-exponential function' as an extension of the ordinary exponential function where m means a generator for a distorted Lebesgue measure, and show some examples to solve differential equations: nonhomogeneous first order and homogeneous second order.

Next we discuss a relation of the Choquet integral equation with the Abel integral equation, and also show a relation of Choquet calculus with fractional calculus which is one of the recent advanced topics in engineering. Fractional calculus is, however, not a new research field and in fact

we can go back to the end of the 17th Century when Leibniz gave a comment on the possibility of semi derivative. Since then many distinguished mathematicians like Laplace, Fourier, Abel, Liouville, Cauchy, Riemann and Laurant contributed to fractional calculus. Yet there have been given no adequate interpretations, in particular, on fractional derivatives. In this talk we point out that fractional calculus is a very particular case of Choquet calculus and give a fuzzy-measure-theoretic interpretation on fractional calculus. For instance the Mittag-Leffler function used to solve fractional differential equations is found to be a special case of our m-exponential function.

In addition we present a Leibniz-like rule for the derivative of a product function. Finally we give the definition of conditional distorted Lebesgue measures and show their basic properties.

Preference-Based Optimization Using Rank-Dependent Aggregation Functions

Patrice Perny

LIP6, UPMC, Paris, France

Abstract. The developments of Decision Theory in the last decades have provided a variety of sophisticated preference models for decision making in complex environments (Multicriteria Decision Making, Social Choice, Uncertainty and Risk). Among them, Rank-Dependent Aggregation Functions such as OWA, WOWA, RDU, and more generally Choquet integrals received much attention due to their descriptive possibilities. However, when the set of feasible solutions has a combinatorial structure and/or is implicitly defined (e.g. by a set of constraints), the optimization of a rank-dependent aggregation function raises new algorithmic issues due to the particular structure of preferences to deal with. The aim of this presentation is to provide an overview of typical problems to overcome when dealing with such decision models and to present solution methods either based on combinatorial algorithms or on mathematical programming. Examples will be chosen in various contexts such as multiobjective/multiagent combinatorial optimization and decision under risk or ambiguity.

Decision Making in an Interval-Valued Setting

H. Bustince

Public University of Navarra
bustince@unavarra.es

Abstract. In a multicriteria decision making problem a set of n alternatives $a_1, \ldots a_n$ and a set of m experts $e_1, \ldots, e_m$ are provided. Each expert expresses his/her preferences by means of a $n \times n$ preference matrix in which entry ij expresses how much alternative a_1 is preferred to alternative a_j. In this way, we have m preference matrices that must be merged into a single one (aggregation phase). This aggregated matrix, which is also of dimension $n \times n$ is called collective preference matrix.

For the exploitation phase of the collective preference matrix there exist several different methods. Many of them aggregate the preferences for each of the alternatives to obtain a single number. Then, the alternatives are ranked by means of this number and the best located one is chosen as the best alternative. In particular, the weighted vote is one of the most widely used methods following this methodology.

However, in some situations problems may arise that make it difficult to arrive at a final decision. For instance, it could happen that it is not possible to distinguish between two of the alternatives since they get the same value in the final ranking. Or, if the values of the preferences of one alternative over another are around 0.5,–which means indifference of one alternative against another–, it may not exist enough information to decide which is the best alternative.

In these situations it could be useful to consider the use of interval-valued preferences. To do so, the value of the actual preference of the expert is considered to be a point inside the preference interval, whereas the length of the interval is a measure of the lack of certainty of the expert when he/she provides the preference value.

However, the use of intervals to express preferences gives raise to a new problem. Namely, contrary to the case of the real numbers in [0,1], it does not exist a natural linear order between intervals. This means that:

1. The definition of interval-valued aggregation is not straightforward.
2. There could exist intervals which are not comparable, which means that the intervals obtained from the exploitation phase may not be suitably ranked.

In order to avoid this problem, in [1] authors propose the so-called admissible orders, which are linear and defined in terms of two aggregation functions. This admissible orders allow to extend the notion of aggregation function to the interval-valued setting, overcoming the difficulties

that arise with fuzzy methods in the above mentioned situations. In particulatr, they allow to define Choquet integrals in a general way, not based just in the consideration separately of the lower and the upper bounds of the intervals.

Reference

[1] H. Bustince, J. Fernandez, A. Kolesárová, R. Mesiar, Generation of linear orders for intervals by means of aggregation functions, Fuzzy Sets and Systems, Volume 220, 1 June 2013, Pages 69-77

Table of Contents

Theory and Applications of Non-additive Measures and Corresponding Integrals

Endre Pap

University of Novi Sad, 21000 Novi Sad, Serbia,
Óbuda University, H-1034 Budapest, Hungary,
Singidunum University, 11000 Belgrade, Serbia
pape@eunet.rs

Abstract. Itis given an short overview of some recent results in the theory of non-additive measures and corresponding integrals. It is presented the universal integral, which include among others, Lebesgue, Choquet, Sugeno, pseudo–additive, Shilkret integrals. Related pseudo-integral a generalization of L^p space is introduced. Many useful applications illustrate the power of non-additive measures and corresponding integrals.

Keywords: Non-additive measure, pseudo-additive measure, universal integral, Choquet integral, Sugeno integral, pseudo-additive integral, decision making, utility theory, nonlinear equation, fuzzy number.

1 Introduction

Introducing non-additive measure, called also fuzzy measure or capacity, and the corresponding integrals, e.g., Choquet, Sugeno, idempotent integrals, see [9,16,19,24,25,32,35], generalized measure theory have been developed. This theory, contrary to classical measure theory, deals with modeling of certain phenomena involving interaction between criteria. The Choquet and Sugeno integrals have wide applications as aggregation functions, see [13,34,35]. Recently, the universal integral, whose special cases are all the mentioned integrals, has been proposed in [18]. In the framework of the pseudo-analysis, we present a generalization of L^p space and related convergences of sequences of measurable functions, see [7,29]. The main tools are the Hölder, Minkowski and Markov inequalities for the pseudo-integral, see [4]. The inequalities for integrals based on non-additive measures, e.g., Choquet, Sugeno, pseudo-integral have been recently given, see for an overview [28]. Specially, in [3,4,27,28] inequalities with respect to pseudo-integrals were considered. The convergence of decision variables were presented in [8]. Several convergence concepts based on the Sugeno and Choquet integrals are observed, see [35,36].

2 Non-additive Measures

Let X be a non-empty set, $\mathcal{A}$ be a σ-algebra of subsets of X. A set function $m : \mathcal{A} \to [0,\infty]$ is said to be *continuous from below*, if $\lim_{n\to\infty} m(A_n) = m(A)$ whenever $A_n \nearrow A$; *continuous from above*, if $\lim_{n\to\infty} m(A_n) = m(A)$ whenever $A_n \searrow A$ and there exists n_0 with $\mu(A_{n_0}) < \infty$; *continuous*, if m is continuous from below and above.

V. Torra et al. (Eds.): MDAI 2013, LNAI 8234, pp. 1–10, 2013.

Definition 1. *A monotone measure on $\mathcal{A}$ is an extended real valued set function $m : \mathcal{A} \to [0,\infty]$ satisfying the following conditions:*
(i) $m(\varnothing) = 0$;
(ii) $m(A) \leqslant m(B)$ *whenever* $A \subset B$ *and* $A, B \in \mathcal{A}$.

When m is a monotone measure, the triple $(X, \mathcal{A}, m)$ is called a monotone measure space. There are many special type of non-additive measures with some additional properties, e.g., null-additive, subadditive, superadditive, submeasure, pseudo-additive, see [9,24,25,32,35].

For simplicity, we use the following notations:

- $\mathcal{F}^{(X,\mathcal{A})}$ denote the set of all $\mathcal{A}$-measurable functions $f : X \to [0,\infty]$;
- For each number $a \in]0,\infty]$, $\mathcal{M}_a^{(X,\mathcal{A})}$ is the set of all monotone measures satisfying $m(X) = a$, and denote by

$$\mathcal{M}^{(X,\mathcal{A})} = \bigcup_{a\in]0,\infty]} \mathcal{M}_a^{(X,\mathcal{A})};$$

- $\mathcal{S}$ is the class of all measurable spaces, and

$$\mathcal{D}_{[0,\infty]} = \bigcup_{(X,\mathcal{A})\in\mathcal{S}} \mathcal{M}^{(X,\mathcal{A})} \times \mathcal{F}^{(X,\mathcal{A})}.$$

An equivalence relation between pairs (m_1, f_1), $(m_2, f_2) \in \mathcal{D}_{[0,\infty]}$ was introduced in [18].

Definition 2. *Two pairs* $(m_1, f_1) \in \mathcal{M}^{(X_1,\,\mathcal{A}_1)} \times \mathcal{F}^{(X_1,\mathcal{A}_1)}$ *and* $(m_2, f_2) \in \mathcal{M}^{(X_2,\mathcal{A}_2)} \times \mathcal{F}^{(X_2,\mathcal{A}_2)}$ *satisfying*

$$m_1(\{x \in X_1 \mid f_1(x) \geq t\}) = m_2(\{x \in X_2 \mid f_2(x) \geq t\})$$

for all $t \in]0,\infty]$, *will be called integral equivalent.*

3 Non-additive Integrals

We have by [18,32].

Definition 3. *A pseudo-multiplication is a function* $\otimes : [0,\infty]^2 \to [0,\infty]$ *with the following properties:*

(i) it is non-decreasing in each component, i.e., for all $a_1, a_2, b_1, b_2 \in [0,\infty]$ *with* $a_1 \leq a_2$ *and* $b_1 \leq b_2$ *we have* $a_1 \otimes b_1 \leq a_2 \otimes b_2$,
(ii) 0 *is an annihilator of, i.e., for all* $a \in [0,\infty]$ *we have* $a \otimes 0 = 0 \otimes a = 0$,
(iii) has a neutral element different from 0, *i.e., there exists an* $e \in]0,\infty]$ *such that, for all* $a \in [0,\infty]$ *we have* $a \otimes e = e \otimes a = a$.

The universal integral is introduced using axioms ([18]).

Definition 4. *A function* $\mathbf{I}:\mathcal{D}_{[0,\infty]}\to[0,\infty]$ *is called a universal integral if the following axioms hold:*

(i) for any measurable space $(X,\mathcal{A})$ *the restriction of the function* $\mathbf{I}$ *to* $\mathcal{M}^{(X,\mathcal{A})}\times\mathcal{F}^{(X,\mathcal{A})}$ *is non-decreasing in each coordinate;*

(ii) there exists a pseudo-multiplication $\otimes:[0,\infty]^2\to[0,\infty]$ *such that for all pairs* $(m,c\cdot 1_A)\in\mathcal{D}_{[0,\infty]}$ *(where* 1_A *is the characteristic function of the set A)*

$$\mathbf{I}(m,c\cdot 1_A)=c\otimes m(A);$$

(iii) for all integral equivalent pairs $(m_1,f_1),(m_2,f_2)\in\mathcal{D}_{[0,\infty]}$ *we have*

$$\mathbf{I}(m_1,f_1)=\mathbf{I}(m_2,f_2).$$

The Choquet, Sugeno and Shilkret integrals are particular cases of the preceding integral. In the following theorem are given the smallest and the greatest universal integral based on $\otimes$ ([18]).

Theorem 1. *Let* $\otimes:[0,\infty]^2\to[0,\infty]$ *be a pseudo-multiplication on* $[0,\infty]$. *Then the smallest universal integral* $\mathbf{I}_\otimes$ *and the greatest universal integral* $\mathbf{I}^\otimes$ *based on* $\otimes$ *are given by*

$$\mathbf{I}_\otimes(m,f)=\sup_{t\in]0,\infty]}\left(t\otimes m(\{x\in X\mid f(x)\geqslant t\})\right),$$

$$\mathbf{I}^\otimes(m,f)=\operatorname*{essup}_m f\otimes\sup_{t\in]0,\infty]}m(\{x\in X\mid f(x)\geqslant t\}),$$

where

$$\operatorname*{essup}_m f=\sup\{t\in[0,\infty]\mid m(\{x\in X\mid f(x)\geqslant t\})>0\}.$$

For further results on non-additive integrals see [9,13,22,24,25,34,35].

4 Pseudo-additive Measures and Corresponding Integrals

Let $[a,b]$ be a closed (in some cases semiclosed) subinterval of $[-\infty,\infty]$. The full order on $[a,b]$ will be denoted by $\preceq$. This can be the usual order of the real line, but it can be another order. The operation $\oplus$ (pseudo-addition) is a commutative, non-decreasing (with respect to $\preceq$), associative function $\oplus:[a,b]\times[a,b]\to[a,b]$ with a zero (neutral) element denoted by $\mathbf{0}$. Denote $[a,b]_+=\{x:x\in[a,b],\mathbf{0}\preceq x\}$. The operation $\odot$ (pseudo-multiplication) is a function $\odot:[a,b]\times[a,b]\to[a,b]$ which is commutative, positively non-decreasing, i.e., $x\preceq y$ implies $x\odot z\preceq y\odot z$, $z\in[a,b]_+$, associative and for which there exist a unit element $\mathbf{1}\in[a,b]$, i.e., for each $x\in[a,b]$, $1\odot x=x$. We assume $\mathbf{0}\odot x=\mathbf{0}$ and that $\odot$ is distributive over $\oplus$. The structure $([a,b],\oplus,\odot)$ is called a semiring (see [24,25]). For $x\in[a,b]_+$ and $p\in]0,\infty[$, the pseudo-power $x_\odot^{(p)}$ is defined in the following way. If $p=n$ is an integer then $x_\odot^{(n)}=\underbrace{x\odot x\odot\ldots\odot x}_{n-\text{times}}$. Moreover,

$x_{\odot}^{(\frac{1}{n})} = \sup\left\{y \mid y_{\odot}^{(n)} \le x\right\}$. Then $x_{\odot}^{(\frac{m}{n})} = x_{\odot}^{(r)}$ is well defined for any rational $r \in]0,\infty[$, independently of representation $r = \frac{m}{n} = \frac{m_1}{n_1}$, m, n, m_1, n_1 being positive integers (the result follows from the continuity and monotonicity of $\odot$). Due to continuity of $\odot$, if p is not rational, then $x_{\odot}^{(p)} = \sup\left\{x_{\odot}^{(r)} \mid r \in]0,p[,\ r \text{ is rational}\right\}$. If $\odot$ is idempotent, then $x_{\odot}^{(p)} = x$ for any $x \in [a,b]$ and $p \in]0,\infty[$.

A set function $m : \mathcal{A} \to [a,b]_+$ is a $\oplus$-*measure* if there hold $m(\varnothing) = 0$, and

$$m(A \cup B) = m(A) \oplus m(B)$$

for $A, B \in \mathcal{A}$ such that $A \cap B = \varnothing$, see [26]. An $\oplus$ - measure m is σ- $\oplus$ - measure if

$$m\left(\bigcup_{i=1}^{\infty} A_i\right) = \bigoplus_{i=1}^{\infty} m(A_i)$$

holds for any sequence (A_i) of pairwise disjoint sets from $\mathcal{A}$.

Pseudo-integral with respect to a σ-$\oplus$-measure m of a simple function s is given by

$$\int_X^{\oplus} s \odot dm = \bigoplus_{i=1}^{n} a_i \odot m(A_i),$$

and of a bounded measurable function $f : X \to [a,b]$, is given by

$$\int_X^{\oplus} f \odot dm = \lim_{n\to\infty} \int_X^{\oplus} s_n \odot dm,$$

where $(s_n)_{i\in\mathbb{N}}$ is sequence of simple functions converging to f. We have by [29].

Definition 5. *Let A be a non-empty set. A function $d_{\oplus} : A \times A \to [a,b]_+$ is a pseudo-metric on A if there hold*

(PM1) $d_{\oplus}(x,y) = \mathbf{0}$ *iff* $x = y$, *for all* $x, y \in A$,
(PM2) $d_{\oplus}(x,y) = d_{\oplus}(y,x)$, *for all* $x, y \in A$
(PM3) there exists $c \in [a,b]_+$ *such that for all* $x, y, z \in A$ *it holds*

$$d_{\oplus}(x,y) \preceq c \odot (d_{\oplus}(x,z) \oplus d_{\oplus}(z,y)).$$

Example 1. (i) Let $([a,b], \oplus, \odot)$ be the semiring with generated pseudo-operations by an increasing and continuous function g. Here we have $x \odot y = g^{-1}(g(x) \cdot g(y))$, and therefore $x_{\odot}^{(p)} = g^{-1}(g^p(x))$. If the function $d_{\oplus} : [a,b] \times [a,b] \to [a,b]$ is defined by

$$d_{\oplus}(x,y) = g^{-1}(|g(x) - g(y)|),$$

then $d_{\oplus}$ is the pseudo-metric on $[a,b]$ and $c = \mathbf{1}$.

(ii) In the semiring $([a,b], \oplus, \odot)$ where $x \oplus y = \sup(x,y)$, $x \odot y = g^{-1}(g(x)g(y))$ and g is an increasing and continuous function. The function $d_{\oplus} : [a,b] \times [a,b] \to [a,b]$ defined also by (1) is the pseudo-metric on $[a,b]$ and $c = g^{-1}(2)$. In [8] the semiring $([-\infty,\infty[, \sup, +)$ are considered. The pseudo-metric is defined by $d_{\oplus}(x,y) = \log|e^x - e^y|$. The condition (PM3) is fulfilled for $c = \log 2$.

Let $([a,b],\oplus,\odot)$ be a semiring and $(X,\mathcal{A})$ be a measurable space, m a σ-$\oplus$-measure with the corresponding pseudo-integral, see [16,17,19,20,23,24,26]. We define for $0 < p < \infty$ and $u,v : X \to [a,b]$ measurable functions

$$D_{p\oplus}(u,v) = \left(\int_X^{\oplus} (d_{\oplus}(u,v))_{\odot}^{(p)} \odot dm \right)_{\odot}^{\left(\frac{1}{p}\right)}.$$

If $p = \infty$, then

$$D_{\infty\oplus}(u,v) = \inf\{\alpha \in [a,b] \mid m(\{x \in X \mid d_{\oplus}(u(x),v(x)) \geqslant \alpha\}) = \mathbf{0}\}.$$

Similarly as in the classical measure theory, we consider the equivalence classes of functions which are equal almost everywhere with respect to σ-$\oplus$-measure m on X. By Minkowski inequality [4] the function $D_{p\oplus}$ is a pseudo-metric on $L^p_{\oplus}$. Due to Hölder inequalities [4] we have the following theorem, obtained in [29].

Theorem 2. *Let $x \oplus y = \sup(x,y)$ and $x \odot y = g^{-1}(g(x)g(y))$. Let m be a σ-$\oplus$-measure and p and q be conjugate exponents with $1 \leqslant p \leqslant \infty$. If $u \in L^p_{\oplus}$ and $v \in L^q_{\oplus}$, and generator $g : [a,b] \to [0,\infty]$ of the pseudo-addition $\oplus$ and pseudo-multiplication $\odot$ is an increasing function, then $u \odot v \in L^1_{\oplus}$.*

We have introduce various notions of convergence related to a σ-$\oplus$-measure and pseudo-integral in the pseudo-L^p space. The relationships among these types of convergence were considered in [29] using the Minkowski type inequality [4].

5 Applications

Very briefly we present few applications of non-additive measures and corresponding integrals.

5.1 Decision Making

Decision making occurs in almost all fields of human activities, and it is devoted to aggregating scores or preferences on a given set of alternatives, the scores or preferences being obtained from several decision makers, voters, experts, etc., see [13,34]. Multicriteria decision making follows the spirit of multiattribute utility theory. Main step in multicriteria decision making is the aggregation, where quantities to be aggregated are most often scores on criteria for a given alternative. The Choquet integral play special role, since the notions of importance index, interaction index, tolerance, veto and favor, etc., can be well handled by monotone measures and the Choquet integral. The situation is similar for multiobjective optimization, although the main difference is related to huge number of alternatives, which make most often an infinite set. The theoretical framework for decision with several persons is social choice theory, e.g., voting procedures as Borda count and the Condorcet rule. Voters give scores to candidates, representing in some sense the intensity of their preference, permits to escape Arrow's theorem, but formally amounts to using methods of multicriteria decision making, up to the difference that most often, voters are anonymous, and thus symmetric aggregation functions have to be chosen.

5.2 Pattern Recognition and Classification

One popular approach to classification of objects and pattern recognition is data or sensor fusion, see [13,31]. To classify a given (unknown) object, one transforms the measurements given by the sensors or the values of the attributes into confidence degrees of belonging to some class, and then an aggregation of the confidence degrees is performed. The Choquet integral, due to its versatility, has often been used with success in practical applications.

Image analysis is a hard task that often sensor (or method) fusion is used. Again, the Choquet integral has been widely used. The second place where aggregation functions appear in image processing is filtering, where Choquet integral filters have been used also for texture recognition.

5.3 Hybrid Utility Function

A generalization of the utility theory of von Neumann-Morgenstern in the paper [10] is obtained. The basic tool was a characterization of the pairs of continuous t-norm T and t-conorm S such that the former is conditionally distributive over the latter (related with a number $a \in [0,1]$), see [16]. Let $\mathcal{P}_{S,a}$ be the set of ordered pairs (α,β) given by

$$\mathcal{P}_{S,a} = \{(\alpha,\beta) \mid (\alpha,\beta) \in]a,1[^2, \alpha+\beta = 1+a\}$$

$$\cup\{(\alpha,\beta) \mid \min(\alpha,\beta) \leqslant a, \max(\alpha,\beta) = 1\}.$$

A hybrid mixture set is a quadruple $(\mathcal{G},M,T,S)$ where $\mathcal{G}$ is a set, (S,T) is a pair of continuous t-conorm and t-norm, respectively, which satisfy the condition conditional distributivity and $M : \mathcal{G}^2 \times \mathcal{P}_{S,a} \to [0,1]$ is a function (hybrid mixture operation) given by

$$M(x,y;\alpha,\beta) = S(T(\alpha,x),T(\beta,y)),$$

where $x,y \in [0,1]$ are utilities and (α,β) is a pair of degrees of plausibility from $\mathcal{P}_{S,a}$. There is completely described in [10] the behavior of the decision maker related to the mixture M.

5.4 Generalization of Portmanteau Theorem

Theorem of Portmanteau type for pseudo-weak convergent sequences of capacity functionals for sequence of random closed sets obtained in [15], see also [6,30]. Random set theory is a part of abstract mathematical analysis and it is applied in many fields as image processing, mathematical morphology, expert system, theoretical statistic, etc. A generalization of the notion of the weak convergence of sequences of probability measures is introduced in [15]: A sequence of capacity functionals $(\mathsf{F}_n)_{n\in\mathbb{N}}$ pseudo-weak converges to capacity functional F if and only if for each continuous, bounded function $f : \mathbb{R} \to [0,\infty]$ we have

$$\lim_{n\to\infty} \int^{\oplus} f \odot d\mathsf{F}_n = \int^{\oplus} f \odot d\mathsf{F}.$$

Among other results it proved in [15] the following theorem.

Theorem 3. *If a sequence of capacity functionals* $(\mathsf{F}_n)_{n\in\mathbb{N}}$ *pseudo-weak converges to capacity functional* F*, then* $\limsup_n \mathsf{F}_n(C) \leqslant \mathsf{F}(C)$ *for all closed sets* $C \subseteq \mathbb{R}$.

5.5 Fuzzy Logics and Fuzzy Numbers

For a t-norm T, see [16], the strong negation c given by $c(x) = 1 - x$, and with the t-conorm S dual to T given by $S(x,y) = c\big(T(c(x),c(y))\big)$, we obtain the basic logic connectives in a $[0,1]$–valued logic: *conjunction*: $x \wedge_T y = T(x,y)$, *disjunction*: $x \vee_T y = S(x,y)$. The implication is introduced in different ways, see [16].

The arithmetical operations with fuzzy numbers is based on Zadeh extension principle. Let T be an arbitrary but fixed t-norm and $*$ a binary operation on $\mathbb{R}$. Then the operation $*$ is extended to fuzzy numbers A and B by $A *_T B(z) = \sup_{x*y=z} T(A(x),B(y))$ for $z \in \mathbb{R}$. Some usual operations with fuzzy numbers are following: Addition is obtained for $* = +$: $A \oplus_T B(z) = \sup_{x+y=z} T(A(x),B(y))$, and multiplication for $* = \cdot$: $A \odot_T B(z) = \sup_{x\cdot y=z} T(A(x),B(y))$. We remark that multiplication and addition are generalized pseudo-convolutions of the second type based on semiring $([0,1],\max,T)$.

5.6 Fractals Dimensions

We present special max-measures in the framework of fractal dimension, see [11]. One of the oldest and most important is the Hausdorff dimension defined for an arbitrary subset of $\mathbb{R}^n$. For a subset F of $\mathbb{R}^n$, nonnegative number s and arbitrary $\delta > 0$ we define first set function

$$\mathcal{H}^s_\delta(F) = \inf\{\sum_{i=1}^{\infty} d(U_i)^s \mid \{U_i\} \text{ is } \delta - \text{cover for } F\},$$

where $d(A) = \sup\{\|xy\| \mid x,y \in A\}$ for $A \subset \mathbb{R}^n$ and $\|x\| = (x_1^2 + \cdots + x_n^2)^{1/2}$, and δ-cover of the set F is a family of sets $\{U_i\}$ with the property $F \subset \cup_{i=1}^{\infty} U_i$ and d $(U_i) < \delta$ for $i \in \mathbb{N}$. Then $\mathcal{H}^s(F) = \lim_{\delta\to 0} \mathcal{H}^s_\delta(F)$ is called s-dimensional Hausdorff measure of the set F. Hausdorff dimension of the set F is given by

$$Dim_{\mathbf{H}} F = \inf\{s \mid \mathcal{H}^s(F) = 0\} = \sup\{s \mid \mathcal{H}^s(F) = \infty\}.$$

Hausdorff dimension is a σ- max-measure on $2^{\mathbb{R}^n}$ (in the theory of fractals this property is called countable stability).

The second type is the so-called box-counting measure or Kolmogorov entropy, for which we give the following definition: The upper and lower box-counting measure of subset F of $\mathbb{R}^n$ is given by

$$\underline{dim}_{\mathbf{B}} F = \underline{\lim}_{\delta\to 0} \frac{\log N_\delta(F)}{-\log\delta}, \qquad \overline{dim}_{\mathbf{B}} F = \overline{\lim}_{\delta\to 0} \frac{\log N_\delta(F)}{-\log\delta},$$

respectively, while the dimensions of the box- counting measure for F is given by

$$Dim_{\mathbf{B}} F = \lim_{\delta\to 0} \frac{\log N_\delta(F)}{-\log\delta}$$

(if there exists the limit), where $N_\delta(F)$ is the minimum number of closed balls of radius δ that cover F. Upper box-counting measure $\overline{dim}_{\mathbf{B}}$ is finitely max-measure, and $\underline{dim}_{\mathbf{B}}$ is not. Box-counting measure in general is not a σ-max-measure. For example, if we take a the set of rational numbers F between 0 and 1, then we have that the box-counting measure of each rational number as singleton set is zero, but the box-counting measure for F is equal 1, since F is a dense subset of the interval $[0,1]$.

5.7 Nonlinear Partial Differential Equations in Control Theory

In the classical linear mathematics all methods of solutions of linear equations are based on linear supperposition principle. Unfortunately, this method can not apply on nonlinear equations. But taking other operations instead of the usual addition and multiplication we can extend the previous superposition principle. We illustrate this on nonlinear partial differential equations.

We have that if u_1 and u_2 are solutions of the following general Hamilton-Jacobi equation

$$\frac{\partial u(x,t)}{\partial t} + H\left(\frac{\partial u}{\partial x}, x, t\right) = 0,$$

where H is a convex function,then $(\lambda_1 \odot u_1) \oplus (\lambda_2 \odot u_2)$ is also a solution of the preceding Hamilton-Jacobi equation, with respect to pseudo-operations $\oplus = \min$ and $\odot = +$ (pseudo linear superposition principle). More details can be found in [19,24]. It is solved by pseudo-weak solution, see [19], avoiding the use of the "viscosity solution" method.

6 Conclusion

We have given a very brief presentation of a part of the theory of non-additive measures and corresponding integrals, as a generalization of the classical measure theory. We illustrate the application in very different fields (decision making, pattern recognition and classification, utility theory, random sets, fuzzy logics, fuzzy sets, fractal dimension, nonlinear partial differential equations), although the range of applications is much wider.

Acknowledgments. The authors are supported in part by the Project MPNRS 174009, and by the project "Mathematical models of intelligent systems and their applications " of Academy of Sciences and Arts of Vojvodina supported by Provincial Secretariat for Science and Technological Development of Vojvodina.

References

1. Agahi, H., Mesiar, R., Ouyang, Y.: General Minkowski type inequalities for Sugeno integrals. Fuzzy Sets and Systems 161, 708–715 (2010)
2. Agahi, H., Mesiar, R., Ouyang, Y.: New general extensions of Chebyshev type inequalities for Sugeno integrals. Int. J. of Approximate Reasoning 51, 135–140 (2009)

3. Agahi, H., Mesiar, R., Ouyang, Y.: Chebyshev type inequalities for pseudo-integrals. Nonlinear Analysis: Theory, Methods and Applications 72, 2737–2743 (2010)
4. Agahi, H., Mesiar, R., Ouyang, Y., Pap, E., Štrboja, M.: Hölder and Minkowski type inequalities for pseudo-integral. Applied Mathematics and Computation 217(21), 8630–8639 (2011)
5. Agahi, H., Mesiar, R., Ouyang, Y., Pap, E., Štrboja, M.: Berwald type inequality for Sugeno integral. Applied Mathematics and Computation 217, 4100–4108 (2010)
6. Akian, M.: Densities of idempotent measures and large deviations. Transactions of the American Mathematical Society 351(11), 4515–4543 (1999)
7. Bede, B., O'Regan, D.: The theory of pseudo-linear operators. Knowledge Based Systems 38, 19–26 (2013)
8. Del Moral, P.: Résolution particulaire des problèmes d'estimation et d'optimisation nonlinéaires. Thesis dissertation, Université Paul Sabatier, Toulouse (1994)
9. Denneberg, D.: Non-additive measure and integral. Kluwer Academic Publishers, Dordrecht (1994)
10. Dubois, D., Pap, E., Prade, H.: Hybrid probabilistic-possibilistic mixtures and utility functions. In: Fodor, J., de Baets, B., Perny, P. (eds.) Preferences and Decisions under Incomplete Knowledge. STUDFUZZ, vol. 51, pp. 51–73. Springer, Heidelberg (2000)
11. Falconer, K.: Fractal Geometry. John Wiley and Sons, Chichester (1990)
12. Flores-Franulič, A., Román-Flores, H., Chalco-Cano, Y.: Markov type inequalities for fuzzy integrals. Applied Mathematics and Computation 207, 242–247 (2009)
13. Grabisch, M., Marichal, J.L., Mesiar, R., Pap, E.: Aggregation Functions. Encyclopedia of Mathematics and Its Applications, vol. 127. Cambridge University Press (2009)
14. Grabisch, M., Murofushi, T., Sugeno, M. (eds.): Fuzzy Measures and Integrals. Theory and applications. Physica-Verlag, Heidelberg (2000)
15. Grbić, T., Pap, E.: Generalization of the Portmanteau theorem with respect to the pseudo-weak convergence of random closed sets (Veroyatnost i Primenen.). SIAM Theory of Probability and Its Applications 54(1), 51–67 (2010)
16. Klement, E.P., Mesiar, R., Pap, E.: Triangular norms. In: Trends in Logic. Studia Logica Library, vol. 8. Kluwer Academic Publishers, Dodrecht (2000)
17. Klement, E.P., Mesiar, R., Pap, E.: Integration with respect to decomposable measures, based on a conditionally distributive semiring on the unit interval. Internat. J. Uncertain. Fuzziness Knowledge-Based Systems 8, 701–717 (2000)
18. Klement, E.P., Mesiar, R., Pap, E.: A Universal Integral as Common Frame for Choquet and Sugeno Integral. IEEE Transactions on Fuzzy Systems 18(1), 178–187 (2000)
19. Kolokoltsov, V.N., Maslov, V.P.: Idempotent Analysis and Its Applications. Kluwer Academic Publishers, Dordrecht (1997)
20. Mesiar, R., Pap, E.: Idempotent integral as limit of g -integrals. Fuzzy Sets and Systems 102, 385–392 (1999)
21. Mesiar, R., Li, J., Pap, E.: The Choquet integral as Lebesgue integral and related inequalities. Kybernetika 46, 1098–1107 (2010)
22. Mesiar, R., Li, J., Pap, E.: Discrete pseudo-integrals. Int. J. of Approximative Reasoning 54, 357–364 (2013)
23. Pap, E.: g-calculus. Univ. u Novom Sadu Zb. Rad. Prirod.-Mat. Fak. Ser. Mat. 23(1), 145–156 (1993)
24. Pap, E.: Null-Additive Set Functions. Kluwer Academic Publishers, Dordrecht (1995)
25. Pap, E. (ed.): Handbook of Measure Theory. Elsevier, Amsterdam (2002)
26. Pap, E.: Pseudo-Additive Measures and Their Aplications. Handbook of Measure Theory. In: Pap, E. (ed.) Handbook of Measure Theory, vol. II, pp. 1403–1465. Elsevier (2002)
27. Pap, E., Štrboja, M.: Generalization of the Jensen inequality for pseudo-integral. Information Sciences 180, 543–548 (2010)

28. Pap, E., Štrboja, M.: Generalizations of Integral Inequalities for Integrals Based on Nonadditive Measures. In: Pap, E. (ed.) Intelligent Systems: Models and Applications. TIEI, vol. 3, pp. 3–22. Springer, Heidelberg (2013)
29. Pap, E., Štrboja, M.: Pseudo-L^p space and convergence. Fuzzy Sets and Systems (in print)
30. Puhalskii, A.: Large deviations and idempotent probability. Chapman, Hall/CRC (2001)
31. Rudas, I.J., Pap, E., Fodor, J.: Information aggregation in intelligent systems: an application oriented approach. Knowledge Based Systems 38, 3–13 (2013)
32. Sugeno, M.: Theory of fuzzy integrals and its applications. Ph.D. Dissertation, Tokyo Institute of Technology (1974)
33. Sugeno, M., Murofushi, T.: Pseudo-additive measures and integrals. J. Math. Anal. Appl. 122, 197–222 (1987)
34. Torra, V., Narukawa, Y.: Modeling Decisions: Information Fusion and Aggregation Operators. Cognitive Technologies. Springer (2007)
35. Wang, Z., Klir, G.J.: Generalized measure theory. Springer, Boston (2009)
36. Wang, R.S.: Some inequalities and convergence theorems for Choquet integrals. J. Appl. Comput. 35(1-2), 305–321 (2011)

Some New Domain Restrictions in Social Choice, and Their Consequences

Salvador Barberà[1], Dolors Berga[2], and Bernardo Moreno[3]

[1] MOVE, Universitat Autònoma de Barcelona, and Barcelona GSE, Departament d'Economia i d'Història Econòmica, Edifici B, 08193 Bellaterra, Spain
salvador.barbera@uab.cat

[2] Departament d'Economia, C/ Universitat de Girona 10, Universitat de Girona, 17071 Girona, Spain
dolors.berga@udg.edu

[3] Departamento de Teoría e Historia Económica, Facultad de Ciencias Económicas y Empresariales, Campus de El Ejido, 29071 Málaga, Spain
bernardo@uma.es

Abstract. Restricting the domains of definition of social choice functions is a classical method to test the robustness of impossibility results and to find conditions under which attractive methods to reach collective decisions can be identified, satisfying different sets of desirable properties. We survey a number of domains that we have recently explored, and exhibit results emerging for functions defined on them. In particular, we have identified a condition called top monotonicity under which the core of voting rules is non-empty, a second one called sequential inclusion where individual and group strategy-proofness become equivalent, and still a third condition called intertwinedness where the strategy-proofness of social choice functions is guaranteed as soon as they satisfy very simple monotonicity and invariance requirements.

Keywords: Strategy-proofness, group strategy-proofness, single-peaked preferences, separable preferences, top monotonicity, sequential inclusion, reshuffling invariance, monotonicity, intertwined domains.

1 Introduction

Social choice theory has a tradition of delivering impossibility results, but these are starting points for further, more constructive work. Indeed, impossibility results usually open the gate toward further understanding of design issues that eventually lead to positive proposals and to the characterization of rich classes of collective decision-making procedures. The crucial difference between positive and negative results is in most cases related to the domain of definition of the social choice rules under consideration. Arrow's impossibility theorem and the Gibbard-Satterthwaite theorem (Gibbard, 1973, and Satterthwaite, 1975) are the two most classical examples of impossibility results, and both are predicated on social choice rules defined for the universal domain of preference profiles: they hold for functions that must take values for any possible combination of agent's

V. Torra et al. (Eds.): MDAI 2013, LNAI 8234, pp. 11–24, 2013.

preferences over alternatives. By contrast, when social choice rules are defined in the restricted domain of single-peaked preferences, then the impossibilities turn into possibilities, and simple majority, among others, emerges as a fully satisfactory social welfare function generating strategy-proof social decisions as well (Black, 1948 and Moulin, 1980).

Single-peakedness is the best known, but not the only domain restriction under which the Arrowian aggregation difficulties can be overcome. Others are, for example, the restriction to single-crossing families of preferences, or to sets of profiles satisfying an intermediateness condition (Gans and Smart, 1996, Grantmond, 1978, Rothstein, 1990, Saporiti, 2009). Likewise, majority voting operates adequately within these domains, but other social choice rules may also be satisfactorily used (Austen-Smith and Banks, 1999). Characterizing the social choice rules that operate properly, in some well defined sense, under given domain restrictions, is a very fruitful approach to examine social welfare issues: Arrow's impossibility paves the way, but then positive characterization results follow. Similarly, the Gibbard-Satterthwaite theorem establishes that one may not expect to find nontrivial strategy-proof social choice rules operating on universal domains, but opens the door to different possibility results. We know the form of all the social choice rules that are strategy-proof when agent's preferences are single-peaked in different contexts (Moulin, 1980; Sprumont, 1991), but also of those that meet this condition under separable preferences (Barberà, Sonnenschein, and Zhou, 1991, Barberà, Gul, and Stacchetti, 1993, Barberà, Massó, and Neme, 1997, 1999, Barberà, Massó, and Serizawa, 1998, Serizawa, 1996), single-plateaued preferences (Berga, 1998), single-dipped (Barberà, Berga, and Moreno, 2012b,c; and Manjunath, 2013), for example.

The purpose of this expositional paper is to introduce the reader to three families of domains that we have identified in recent work as being sufficient (and close to necessary) to guarantee that social choice rules defined on them can satisfy a variety of desirable properties. In the spirit of our previous remarks, we shall present three puzzles in social choice, and show that their solution depends on the domain of preference profiles for which our relevant social choice rules are defined.

Here are the questions we want to address:

1. Is there a common root to the conditions of single-peakedness, single-crossing and intermediateness? It is known that voting equilibria under qualified majority rules can be guaranteed within these three domain restrictions and that, in addition, these equilibria are of similar form. We shall present the domain of top monotonic profiles that contains all three, and for which the existence of voting equilibria with essentially the same traits is still guaranteed.
2. What is the connection between individual and group strategy-proofness? It is known that for some domain restrictions, both conditions become equivalent, while in others individual strategy-proofness is a strictly weaker requirement. We shall present a domain condition called sequential inclusion where equivalence is implied, and that is "almost necessary" for the equivalence to hold.

3. When can strategy-proofness be guaranteed by the sole satisfaction of two simple and natural conditions of monotonicity and invariance? We shall exhibit a "connectedness" condition defining what we call intertwined domains and that ensures strategy-proofness thanks to these two conditions alone, while proving that in other cases it may be necessary to add further and less natural requirements.

Since the main purpose of this paper is expositional, we shall provide the results without their proofs, and refer for those to the original papers. After a brief section with general definitions, we devote one section to each of the puzzles that we just stated. In each one of them we try to motivate our question, describe the classes of domains for which we can provide definite answers and offer an example of alternative domains where social choice rules would fail to meet our requirements.

2 The Setup and Some Definitions

Let A be a set of *alternatives and* $N = \{1, ..., n\}$ be a finite set of *agents.* Let $\mathcal{R}$ be the set of all preorders (complete, reflexive, and transitive binary relations) on A and $\mathcal{R}_i \subseteq \mathcal{R}$ be *the set of admissible preferences for agent* $i \in N$. Denote by $\mathcal{P} \subseteq \mathcal{R}$ the set of all antisymmetric preorders. We denote by $R_i \in \mathcal{R}_i$ an admissible preference relation and let as usual, P_i and I_i be the strict and the indifference part of R_i, respectively. When all the admissible preferences for individual i are strict, we will use the notation $\mathcal{P}_i$, instead of the general expression $\mathcal{R}_i$. A *preference profile,* denoted by $R = (R_1, ..., R_n)$, is an element of $\times_{i\in N}\mathcal{R}_i$. For $C \subseteq N$ we will write the profile $R = (R_C, R_{N\setminus C}) \in \times_{i\in S}\mathcal{R}_i$ when we want to stress the role of coalition C. Then the subprofiles $R_C \in \times_{i\in C}\mathcal{R}_i$ and $R_{N\setminus C} \in \times_{i\in S\setminus C}\mathcal{R}_i$ denote the preferences of agents in C and in $N\setminus C$, respectively.

For any $R_i \in \mathcal{R}_i$ and $x \in A$, define *the lower contour set of* R_i *at* x as $L(R_i, x) = \{y \in A : xR_iy\}$. Similarly, the strict lower contour set at x is $\overline{L}(R_i, x) = \{y \in A : xP_iy\}$.

For any $R_i \in \mathcal{R}_i$ and $B \subseteq A$, we denote by $t(R_i, B)$ *the set of maximal elements of* R_i *on* B. That is, $t(R_i, B) = \{x \in B : xR_iy \text{ for all } y \in B\}$. We call $t(R_i, B)$ the top of i in B or the peak on B when it is a singleton.

For any $x, y \in A$ and $R \in \times_{i\in N}\mathcal{R}_i$, let $P(x, y; R) \equiv \{i \in N : xP_iy\}$ and $R(x, y; R) \equiv \{i \in N : xR_iy\}$. That is, the set of agents who strictly (respectively, weakly) prefer x to y according to their individual preferences in R.

A *social choice function* is a function $f : \times_{i\in N}\mathcal{R}_i \to A$. Let A_f denote the range of the social choice function f. We say that f is *onto* if $A_f = A$.

We are interested in social choice functions that are nonmanipulable, either by a single agent or by a coalition of agents. We first define what we mean by a manipulation and then we introduce the well known concepts of *strategy-proofness* and *group strategy-proofness.*

Definition 1. *A social choice function* f *is group manipulable on* $\times_{i\in N}\mathcal{R}_i$ *at* $R \in \times_{i\in N}\mathcal{R}_i$ *if there exists a coalition* $C \subseteq N$ *and* $R'_C \in \times_{i\in C}\mathcal{R}_i$ $(R'_i \neq R_i$ *for any* $i \in C)$ *such that* $f(R'_C, R_{N\setminus C})P_if(R)$ *for all* $i \in C$. *We say that* f *is*

individually manipulable if there exists a possible manipulation where coalition C is a singleton.

Definition 2. *A social choice function f is group strategy-proof on $\times_{i\in N}\mathcal{R}_i$ if f is not group manipulable for any $R \in \times_{i\in N}\mathcal{R}_i$. Similarly, f is strategy-proof if it is not individually manipulable.*

Next we introduce the notions of a preference aggregation rule, voting rule and voting equilibrium. We follow closely Austen-Smith and Banks (1999) since our results will extend those that they present in Chapter 4 of their book.

Definition 3. *A preference aggregation rule is a map, $F : \times_{i\in N}\mathcal{R}_i \to \mathcal{B}$, where $\mathcal{B}$ denotes the set of all reflexive and complete binary relations on A. We denote by R_F the image of profile R under preference aggregation rule F.*

Definition 4. *Given any two profiles $R, R' \in \times_{i\in N}\mathcal{R}_i$ and $x, y \in A$, a preference aggregation rule F is:*
(1) neutral if and only if $[\forall a, b \in A,\ P(x, y; R) = P(a, b; R')$ and $P(y, x; R) = P(b, a; R')]$ imply xR_Fy if and only if aR_Fb;
(2) monotonic if and only if $[P(x, y; R) \subseteq P(x, y; R')$, $R(x, y; R) \subseteq R(x, y; R')$ and $xP_Fy]$ imply xP'_Fy.

A neutral preference aggregation rule treats all alternatives equally when making pairwise comparisons. Monotonicity implies that if x is socially preferred to y and then some people change their preferences so that the support for x does not decrease, while the support for y does not increase, then x must be still socially preferred at the new profile.

One can always associate to each preference aggregation rule a family of ordered pairs of coalitions that represent the ability of different groups of agents in determining the social preference relation.

Definition 5. *The decisive structure associated with a preference aggregation rule F, denoted by $\mathcal{D}(F)$, is a family of ordered pairs of coalitions $(S, W) \subseteq N\times N$ such that $(S, W) \in \mathcal{D}(F) \Leftrightarrow S \subseteq W$ and $\forall x, y \in A$, $\forall R \in \times_{i\in N}\mathcal{R}_i$, $[xP_iy\ \forall i \in S$ and $xR_iy\ \forall i \in W \to xP_Fy]$.*

We now notice that we could have started to define our preference aggregation rule by first providing a family of ordered pairs of coalitions.

Definition 6. *A set $\mathcal{D} \subset 2^N \times 2^N$ is*
(1) monotonic if $(S, W) \in \mathcal{D}$, $S \subseteq S' \subseteq W'$ and $S \subseteq W \subseteq W'$ imply $(S', W') \in \mathcal{D}$
(2) proper if $(S, W) \in \mathcal{D}$, $S' \subseteq N\backslash W$ and $W' \subseteq N\backslash S$ imply $(S', W') \notin \mathcal{D}$.

Definition 7. *Given a proper set D, the preference aggregation rule induced by $\mathcal{D}$, denoted $F_{\mathcal{D}}$, is defined as $\forall x, y \in A$, $xP_{F_{\mathcal{D}}}y \Leftrightarrow [\exists (S, W) \in \mathcal{D} : xP_iy\ \forall i \in S$ and $xR_iy\ \forall i \in W]$[1].*

[1] Notice that the requirement that $\mathcal{D}$ is proper guarantees that $f_{\mathcal{D}}$ is well defined.

It is useful to state the connections between preference aggregation rules and decisive structures, because one is closer to the language of social choice and the other is closer to that of public economics and political economy. More precisely, one can ask when it is the case that a decisive structure and a preference aggregation rule can be used interchangeably as being the primitives. This will happen when the decisive structure associated with F induces F again. Austen-Smith and Banks (1999) define voting rules as those preference aggregation rules that have this property, and provide a characterization for them.

Definition 8. *A preference aggregation rule F is a voting rule if $F = F_{\mathcal{D}(F)}$.*

Proposition 1. *A preference aggregation rule is a voting rule iff it is neutral and monotonic.*

In this survey we concentrate on the study of voting rules and their equilibria, which we now define.

Definition 9. *Let F be a preference aggregation rule and $R \in \times_{i \in N} \mathcal{R}_i$. The core of F at R, $C_F(R, S)$ is the set of maximal elements in $S \subseteq A$ under the binary relation R_F. Elements in the core of a voting rule will be called voting equilibria.*

3 Top Monotonicity

In this section we provide a condition on preference profiles, that we call top monotonicity. This condition, when satisfied by all profiles in a domain, guarantees that voting functions on that domain generate games with a non empty core. Moreover, these core elements, or voting equilibria, are generalized medians in the distribution of preferences for the agents. Furthermore, the classical domains of single-peaked, single-plateaued, single crossing or intermediate (order restricted) profiles are all included within this larger domain.

Voting rules are included among the methods that will fail to satisfy Arrow's theorem, when defined on the universal domain. When restricted to operate on the classical domains that we mention, they produce voting equilibria that are, in addition, nicely expressed as the medians of the distribution of voter's best elements. Our result unifies these possibility results by showing that, although the classical domains are different from each other, they all share one basic feature: they all satisfy our condition of top monotonicity. Moreover, this fact is sufficient to understand why the equilibria under these different restrictions also share the common structural fact of being closely linked to medians.

This section summarizes results that were first stated and proven in Barberà and Moreno (2011) where the authors propose a new condition on preference profiles over one-dimensional alternatives, called top monotonicity. And where they prove that top monotonicity can be viewed as the common root of a bunch of classical domain restrictions, which had been perceived in the literature as rather different and unrelated to each other.

We additionally assume that individual preferences are continuous binary relations on A. Let us introduce some notation: For each preference profile R, let $A(R)$ be the family of sets containing A itself, and also all triples of distinct alternatives where each alternative is top on A for some agent $k \in N$ according to R.

Now we present top monotonicity.

Definition 10. *A preference profile R is top monotonic iff there exists a linear order $>$ of the set of the alternatives, such that*
(1) $t(R_i, A)$ is a finite union of closed intervals for all $i \in N$, and
(2) For all $S \in A(R)$, for all $i, j \in N$, all $x \in t(R_i, S)$, all $y \in t(R_j, S)$, and any $z \in S$

$$[z < y < x \text{ or } z > y > x] \rightarrow \begin{array}{c} yR_iz \text{ if } z \in t(R_i, S) \cup t(R_j, S) \\ \text{and} \\ yP_iz \text{ if } z \notin t(R_i, S) \cup t(R_j, S). \end{array}$$

When convenient, we'll say that a preference profile is top monotonic relative to $>$.

We can begin by comparing top monotonicity with single-peakedness and single-plateauedness to see that it represents a significant weakening of these conditions. Single-peakedness requires each agent to have a unique maximal element. Moreover, it must be true for any agent that any alternative y to the right (left) of its peak is strictly preferred to any other that is further to the right (left) of it. In particular, this implies that no agent is indifferent between two alternative on the same side of its peak. Hence, indifference classes may consist of at most two alternatives (one to the right and one to the left of the agent's peak).

In contrast, our definition of top monotonicity allows for individual agents to have nontrivial indifference classes, both in and out of the top. In that respect, it allows for many more indifferences than single-plateaued preferences do. Most importantly, top monotonicity relaxes the requirement imposed on the ranking of two alternatives lying on the same side of the agent's top. Under our preference condition, this requirement is only effective for triples where the alternative that is closest to the top of the agent is itself a top element for some other agent. Moreover, the implication is only in weak terms when the alternative involved in the comparison is top for one or for both agents.

A similar, although less direct comparison can be made between top monotonicity and intermediateness or order restriction. The original conditions involve comparisons between pairs of alternatives, regardless of their positions in the ranking of agents. Top monotonicity is also a strict weakening of these requirements, involving the comparison of only a limited number of pairs.

We now state a first result showing that top monotonicity is a common root for a lot of typical preferences restrictions, as it is implied by any of them.

Theorem 1. *(see Theorem 1 in Barberà and Moreno, 2011) If a preference profile is either single-peaked, single-plateaued, single crossing or order restricted, then it also satisfies top monotonicity.*

We now show that top monotonicity guarantees the existence of voting equilibria under any voting rule, and that these will be closely connected to an extended notion of the median voter. Before stating this second result of the paper, we introduce some additional notation, and we propose an extension of the notion of median.

Let $>$ be a linear order of the set of alternatives and R be a preference profile. For any $z \in A$, we define the following three sets

$$N_{\{z\}} = \{j \in N : z \in t_j(A)\},$$

$$N_{\{z\}^-} = \{k \in N : z > x \text{ for all } x \in t_k(A)\},$$

and

$$N_{\{z\}^+} = \{h \in N : z < x \text{ for all } x \in t_h(A)\}.$$

We remark that when R is top monotonic relative to $>$, and z is in the top of some agent i, then $N_{\{z\}} \neq \emptyset$ and the three sets $(N_{\{z\}^-}, N_{\{z\}}, N_{\{z\}^+})$ constitute a partition of the set of voters N. Indeed, $N_{\{z\}}$ contains all voters, including i, for whom z is in the top. $N_{\{z\}^-}$ (resp. $N_{\{z\}^+}$) contains all voters for which all top elements are to the left (resp. to the right) of z. Clearly, then, these three sets are disjoint. To prove that their union contains all elements of N, suppose not. For some agent l, z should not be in $l's$ top, while some alternatives x and y, one to the right and one to the left of z, should belong to the top of l. But then, by top monotonicity we would have zR_lx and also zR_ly. Since x and y are both top for l, so is z, a contradiction[2].

Let n, $n_{\{z\}}$, $n_{\{z\}^-}$, and $n_{\{z\}^+}$ be the cardinalities of N, $N_{\{z\}^-}$, $N_{\{z\}}$ and $N_{\{z\}^+}$, respectively. From the remark above, we know that if z is in the top of some agent, then $n_{\{z\}} + n_{\{z\}^-} + n_{\{z\}^+} = n$. The following definition will allow us to establish an analogue of the classical median voter result for the case of top monotonic profiles.

Definition 11. *Let F be a voting rule. An alternative z is a weak F-median top alternative in a top monotonic profile R relative to an order $>$ of the set of alternatives iff*
(1) z is a top alternative in R for some agent, and
(2) $(N_{\{z\}^-}, N_{\{z\}^-} \cup N_{\{z\}}) \notin \mathcal{D}(F)$ and $(N_{\{z\}^+}, N_{\{z\}} \cup N_{\{z\}^+}) \notin \mathcal{D}(F)$.[3]

[2] Notice that our definition of top monotonicity does not preclude the possibility that an agent's top might be non-connected relative to the order of $>$. Informally, what it demands is that, if an agent has two peaks with a valley in between, then no other agent's peak lies in that valley. In that sense also, our condition is less demanding than that of single plateaued, where we assumed that the tops are connected.

[3] When f is the majority rule we say that an alternative z is a weak median top alternative in a top monotonic profile $\succcurlyeq$ relative to an order $>$ of the set of alternatives iff
(1) z is a top alternative in $\succcurlyeq$ for some agent, and
(2) $n_{\{z\}^-} + n_{\{z\}} \geqslant n_{\{z\}^+}$ and $n_{\{z\}} + n_{\{z\}^+} \geqslant n_{\{z\}^-}$.

We will denote by $WM_F(R)$ the set of weak F-median top alternatives at that profile. We define m^- and m^+ as the lowest and the highest elements in this set according to the order $>$ at that profile.

Definition 12. *An alternative z is an extended weak F-median in a top monotonic profile R relative to an order $>$ of the set of alternatives iff $m^- \geqslant z \geqslant m^+$.*

It is not hard to prove that extended medians in our sense always exist. We will denote by $M_F(R)$ the set of extended weak F-median alternatives at that profile.

We can now state the following result.

Theorem 2. *(see Theorem 1 in Barberà and Moreno, 2011)*
(1) Let F be a voting rule. Whenever a profile of preferences R is top monotonic relative to some order $>$, $C_F(R)$ is not empty and $C_F(R) \subseteq M_F(R)$.
(2) If the profile of preferences R is peak monotonic, $WM_F(R) \subseteq C_F(R)$.

4 Sequential Inclusion

In this section we define a domain condition that we call sequential inclusion. Social choice functions defined on domains that meet this condition will have the property that individual and group strategy-proofness become equivalent. That is, all individual strategy-proof social choice functions on domains satisfying sequential inclusion will also be immune to group manipulation.

Our research was prompted by the observation that this equivalence does not only trivially hold under the universal domain, where only dictatorial social choice functions can be strategy-proof, but also for much more interesting domains, like those of single-peaked preferences. However, both conditions are not equivalent in other contexts, where non trivial strategy-proof social choice functions do exist and yet are subject to manipulation by groups. What we do is to provide an (almost) exact frontier between those cases when the domain of definition of a social choice function does guarantee the equivalence between both properties, and those where it does not.

This section summarizes results that were first stated and proven in Barberà, Berga, and Moreno (2010). Our focus will be on specific cases where it is possible to at least define satisfactory strategy-proof social choice functions. The main question addressed in our paper was what is needed then to hope for the stronger and much more reassuring property of group strategy-proofness to also hold?

We start by defining our condition on preference profiles, called sequential inclusion, and we establish the equivalence between individual and group strategy-proofness for social choice functions defined on domains satisfying that condition. Let us introduce some notation.

Definition 13. *Given a preference profile $R \in \times_{i \in N} \mathcal{R}_i$ and a pair of alternatives $y, z \in A$, we define a binary relation $\succsim (R; y, z)$ on $P(y, z; R)$ as follows:*

$$i \succsim (R; y, z) j \text{ if } L(R_i, z) \subseteq \overline{L}(R_j, y).$$

Note that the binary relation $\succsim$ must be reflexive but not necessarily complete. As usual, we can define the strict and the indifference binary relations associated to $\succsim$. Formally, $i \sim j$ if $L(R_i, z) \subseteq \overline{L}(R_j, y)$ and $L(R_j, z) \subseteq \overline{L}(R_i, y)$. We say that iPj if $L(R_i, z) \subseteq \overline{L}(R_j, y)$ and $\neg[L(R_j, z) \subseteq \overline{L}(R_i, y)]$.

We can now define our main condition.

Definition 14. *A preference profile $R \in \times_{i \in N} \mathcal{R}_i$ satisfies sequential inclusion if for any pair $y, z \in A$ the binary relation $\succsim (R; y, z)$ on $P(y, z; R)$ is complete and acyclic. A domain $\times_{i \in N} \mathcal{R}_i$ satisfies sequential inclusion if any preference profile in this domain satisfies it.*

Since sequential inclusion is a property on preference profiles, it follows that if a domain satisfies sequential inclusion each subdomain inherits the same property. Remarkably, this condition does not require domains to be large in size, contrary to others, like "richness" (see Dasgupta, Hammond, and Maskin [5], Le Breton and Zaporozhets [12]) or our own condition of indirect sequential inclusion defined in Barberà, Berga, and Moreno (2010).

We now present our first main result.

Theorem 3. *(see Theorem 1 in Barberà, Berga, and Moreno, 2010) Let $\times_{i \in N} \mathcal{R}_i$ be a domain satisfying sequential inclusion. Then, any strategy-proof social choice function on $\times_{i \in N} \mathcal{R}_i$ is group strategy-proof.*

A surprising result for the case of three alternatives is that when there are at most three alternatives at stake, any strategy-proof social choice function is group strategy-proof: This is mainly due to the fact that in such framework any preference profile satisfies sequential inclusion (see Corollary 1 in Barberà, Berga, and Moreno, 2010).

In our paper, we also provide another condition, weaker than sequential inclusion and called indirect sequential inclusion, that still guarantees the equivalence between individual and group strategy-proofness (see Theorem 2 in Barberà, Berga, and Moreno, 2010). It is no longer a condition on individual profiles. Rather, it requires that, given a profile within the domain, some other profile, conveniently related to the first one, does indeed satisfy our previous requirement. That is why we say that profiles that meet our new condition satisfy indirect sequential inclusion. The new definition allows us to incorporate new and interesting domains into our list of those guaranteeing equivalence. The interested reader can check the more cumbersome concept of indirect sequential inclusion in Definition 8 in Barberà, Berga, and Moreno (2010).

It is also worth mentioning the partial necessity result we obtain in our work: sequential inclusion is almost necessary to guarantee that individual and group strategy-proofness become equivalent (see Theorem 4 in Barberà, Berga, and Moreno, 2010).

To finish this section we present a simple example that illustrates how our results would fail on alternative domains. We exhibit one domain, that of separable preferences, that violates both direct and indirect sequential inclusion. This is because we can find a social choice function that is strategy-proof but

not group strategy-proof on the mentioned domain. In view of Theorem 4 this proves that indirect sequential inclusion fails (thus, also the direct version).

Example 1. Two candidates a and b may be elected to join a club.[4] Alternatives in this problem are sets of candidates: $A = \{\varnothing, a, b, \{a, b\}\}$.
Given a preference on sets, candidates are called good if they are better than the empty set, when chosen alone, and bad otherwise. Preferences are separable if adding a good candidate to any set makes the union better, and adding a bad one makes the union worse. The set of individual separable preferences is the same for each agent $i \in N$:

R^1	R^2	R^3	R^4	R^5	R^6	R^7	R^8
$\varnothing$	$\varnothing$	a	a	b	b	$\{a,b\}$	$\{a,b\}$
a	b	$\varnothing$	$\{a,b\}$	$\varnothing$	$\{a,b\}$	a	b
b	a	$\{a,b\}$	$\varnothing$	$\{a,b\}$	$\varnothing$	b	a
$\{a,b\}$	$\{a,b\}$	b	b	a	a	$\varnothing$	$\varnothing$

Consider the social choice function, called voting by quota one: each agent declares her best set of objects and any object that is mentioned by some agent is selected.

This social choice function is clearly strategy-proof. Yet notice that for a profile where $R_1 = R^3$, $R_2 = R^5$ and for any other agent $R_i = R^1$ the outcome would be $\{a, b\}$, whereas agents 1 and 2 could vote for $\varnothing$ and get a preferred outcome. Thus, it is not group strategy-proof since any pair of agents can manipulate it.

5 Intertwined Domains

In this section we define and study two conditions on social choice functions that we find especially attractive: reshuffling invariance and monotonicity. Reshuffling invariance and monotonicity are always necessary for strategy-proofness, whatever the domain of definition of the functions, but need not be sufficient. Because of that, we ask ourselves the following question: can we identify domains of preferences having the property that, when functions are defined on these domains, then our conditions are equivalent to strategy-proofness? We answer this question in the positive. For those domains that we call intertwined, and for any possible social choice function defined on them, the equivalence holds.

For this study we consider the set of alternatives to be finite and, for sake of simplicity, we assume that agents' preferences are strict.

This section presents results originally obtained in Barberà, Berga, and Moreno (2012a).

[4] The example easily extends to any set of candidates: just take profiles as we have just defined and extend individual preferences such that the relative ordering among a, b, $\{a, b\}$ and the empty set are like in the above table and any other object is bad.

Definition 15. *A social choice function f satisfies* ***monotonicity*** *on $\times_{i\in N}\mathcal{P}_i$ if and only if for any $P \in \times_{i\in N}\mathcal{P}_i$ such that $f(P) = x$, and for any $P' \in \times_{i\in N}\mathcal{P}_i$ satisfying the following conditions*

(i) for any $i \in N$, for any $y \in A\backslash\{x\}$; $[xP_iy \Rightarrow xP'_iy]$, and
(ii) for any $i \in N$, for any $y, z \in A\backslash\{x\}$; $[yP_iz \Leftrightarrow yP'_iz]$.

then, $f(P') = x$.

In words: If an alternative x is chosen by a social choice function f at profile $(P_i, P_{N\backslash\{i\}})$, and P'_i is a new preference where x has improved its position then f must still choose x.

Definition 16. *Let $P_i \in \mathcal{P}_i$ and $x \in A$. We say that $P'_i \in \mathcal{P}_i$ is a* x-reshuffling of P_i *if $\overline{L}(P_i, x) = \overline{L}(P'_i, x)$.*

In words: P'_i is a x-reshuffling of P_i if it results from keeping all alternatives that were worse than x and no other, as still being worse, though maybe in a different order.

Definition 17. *A social choice function f satisfies* ***reshuffling invariance*** *on $\times_{i\in N}\mathcal{P}_i$ if and only if for any $P \in \times_{i\in N}\mathcal{P}_i$ such that $f(P) = x$, and for any $(P'_i, P_{N\backslash\{i\}}) \in \times_{i\in N}\mathcal{P}_i$ such that P'_i is a x-reshuffling of P_i, then $f(P'_i, P_{N\backslash\{i\}}) = x$.*

In words: If an alternative x is chosen at a profile, x must be chosen at any other profile obtained from an x-reshuffling of agent i's preferences.

We now introduce our notion of intertwined domains. Whether a domain is intertwined or not will turn out to be crucial to determine whether the different conditions we are interested in may or may not be equivalent, when applied to social choice functions defined on such domains.

Before we provide a formal definition, let us describe the condition informally. For any $i \in N$, select any two (strict) preferences P_i and P'_i, and any two alternatives x and y, where xP_iy (the relationship between the two in P' can be any). Suppose that there exists in our domain a third preference $\overline{P}_i$ such that one can transform P_i into $\overline{P}_i$, through a sequence of changes in the positions of alternatives, such that these changes, at each step, simply consist in lifting the position of y, or of reshufflings around y. Suppose that one can also transform P'_i into $\overline{P}_i$ through another sequence of the same type of transformations, this time with liftings of x and reshufflings around x. We will then say that P_i and P'_i are (x, y)-intertwined.

A domain of preferences will be intertwined if and only if any two of the preferences it contains are intertwined for any two alternatives.

Even more informally, we can say that an intertwined domain is one where one can travel from any pair of preferences to some intermediary preference just by lifting and reshuffling alternatives.

We now proceed to our formal definitions.

Definition 18. *Let $P_i, \overline{P}_i \in \mathcal{P}_i$ and $x \in A$. We say that $\overline{P}_i$ is a x-**direct transform of** P_i if either $\overline{P}_i$ is a x-reshuffling of P_i or $\overline{P}_i$ is a x-monotonic transformation of P_i.*

Definition 19. *Let $P_i, \overline{P}_i \in \mathcal{P}_i$ and $x \in A$. We say that $\overline{P}_i$ is a x-**transform of** P_i if there exist a sequence of preferences $P_1, P_2, ..., P_T$ such that $P_1 = P_i$, $P_T = \overline{P}_i$, and for any $t \in (1, T]$, each P_t is a x-direct transform of P_{t-1}.*

Definition 20. *Let $P_i, P_i' \in \mathcal{P}_i$, $x, y \in A$ where xP_iy. We say that P_i is (x, y)-**intertwined with** P_i' if there exists $\overline{P}_i \in \mathcal{P}_i$ such that $\overline{P}_i$ is both a y-transform of P_i and a x-transform of P_i'.*

Definition 21. *A set of individual preferences $\mathcal{P}_i$ is **intertwined** if for any $P_i \in \mathcal{P}_i$, for any $x, y \in A$ such that xP_iy, and any $P_i' \in \mathcal{P}_i$, P_i is (x, y)-intertwined with P_i'.*

Definition 22. *A domain $\times_{i \in N} \mathcal{P}_i$ is intertwined if for any agent i, $\mathcal{P}_i$ is intertwined.*

We are now ready to state our equivalence result.

Theorem 4. *(see Theorem 1 in Barberà and Moreno, 2012a) Any social choice function defined on an intertwined domain is strategy-proof if and only if it satisfies monotonicity and reshuffling invariance.*

As we already proved in Proposition 1 (1) in Barberà, Berga, and Moreno (2012a), let us remark that monotonicity and reshuffling invariance, our two independent conditions, are necessary for any social choice function defined on any domain to be strategy-proof. However, as we show in the following example our conditions are not always sufficient to guarantee strategy-proofness: there exist social choice functions satisfying both of them which are nevertheless manipulable; clearly the domain of preferences is crucial as Theorem 4 states.

Example 2. (borrowed from the proof of Proposition 1 (3) in Barberà, Berga, and Moreno, 2012a)

Consider the framework in Example 1

Our example refers to a social choice function defined on the domain of separable preferences for the case of two candidates, four alternatives and three voters:

Define the social choice function as the Borda count on A with tie breaking. Voters rank the four alternatives, and each alternative gets three points whenever a voter ranks it first, two when ranked second, one when third and none if last. The choice is the alternative with the highest sum of points, if unique. As for possible ties, notice that, in our example, when there is a tie for first position. there may be at most one voter for whom none of the tied alternatives is the best for him. If there is such an individual, the tie is broken in favor of that alternative that he prefers. Otherwise, the tie is broken according to a pre-determined order of alternatives, say $O : \{a, b\}, b, a, \varnothing$.

Notice that the only cases where the antecedents of reshuffling invariance and monotonicity would apply are those where we change a preference to another having the same top. Given that, it is easy to see that both conditions are respected in our example.

Yet, observe that the function is still manipulable. To see that, let $P = (P^1, P^6, P^7)$, $P' = (P^6, P^6, P^7)$. Then, $f(P) = b$ (b and $\{a, b\}$ have the same score and agent 1 breaks the tie) and $f(P') = \{a, b\}$ (b and $\{a, b\}$ have the same score but all agents have b or $\{a, b\}$ as best alternative, so we use O). Thus, agent 1 manipulates f at P' via P^1.

To finish this section two comments are in order. First, with strict preferences and under intertwined domains, our two conditions are not only equivalent to strategy-proofness but also to strong positive association (Muller and Satterthwaite, 1977). However, strong monotonicity (Moulin, 1988) is weaker than monotonicity and reshuffling invariance conditions (see Proposition 5 in Barberà, Berga, and Moreno, 2012a). Second, let us mention that, in general, intertwinedness and (indirect) sequential inclusion are independent. However, intertwinedness implies indirect sequential inclusion when the set of individual preferences are strict and equal for all agents (see Proposition 6 in Barberà, Berga, and Moreno, 2012a).

6 Concluding Remarks

Our main message is that every specific social choice problem deserves a careful analysis of domains on which we need to define the method to be used, since this may open the doors to attractive possibility results. We have exemplified this message by presenting three domains that we have found worth studying and hope that the readers find them useful for their further work.

Ackowledgements. Salvador Barberà gratefully acknowledges support from "Consolidated Group-C" ECO2008-04756 and FEDER, SGR2009-0419, and the Severo Ochoa Programme for Centres of Excellence in R&D, SEV-2011-0075. Dolors Berga acknowledges the support of the Spanish Ministry of Science and Innovation through grant ECO2010-16353, of Generalitat de Catalunya, through grant SGR2009-0189. Bernardo Moreno gratefully acknowledges financial support from Junta de Andalucia through grants SEJ4941 and SEJ-5980 and the Spanish Ministry of Science and Technology through grant ECO2011-29355.

References

Austen-Smith, D., Banks, J.S.: Positive Political Theory I. Collective Preference. The University of Michigan Press (1999)

Barberà, S., Berga, D., Moreno, B.: Individual versus group strategy-proofness: when do they coincide? J. Econ. Theory 145, 1648–1674 (2010)

Barberà, S., Berga, D., Moreno, B.: Two necessary conditions for strategy-proofness: on what domains are they also sufficient? Games Econ. Beh. 75, 490–509 (2012a)

Barberà, S., Berga, D., Moreno, B.: Domains, ranges and strategy-proofness: the case of single-dipped preferences. Soc. Choice Welfare 39(2-3), 335–352 (2012b)
Barberà, S., Berga, D., Moreno, B.: Group strategy-proof social choice functions with binary ranges and arbitrary domains: characterization results. Int. J. Game Theory 41(4), 791–808 (2012c)
Barberà, S., Gul, F., Stacchetti, E.: Generalized Median Voter Schemes and Committees. J. Econ. Theory 61, 262–289 (1993)
Barberà, S., Massó, J., Neme, A.: Voting under Constraints. J. Econ. Theory 76(2), 298–321 (1997)
Barberà, S., Massó, J., Neme, A.: Maximal domains of preferences preserving strategy-proofness for generalized median voter schemes. Soc. Choice Welfare 16(2), 321–336 (1999)
Barberà, S., Massó, J., Serizawa, S.: Strategy-proof Voting on Compact Ranges. Games Econ. Beh. 25(2), 272–291 (1998)
Barberà, S., Moreno, B.: Top monotonicity: a common root for single peakedness, single crossing and the median voter result. Games Econ. Beh. 73, 345–359 (2011)
Barberà, S., Sonnenschein, H., Zhou, L.: Voting by committees. Econometrica 59, 595–609 (1991)
Berga, D.: Strategy-Proofness and Single-Plateaued Preferences. Math. Soc. Sci. 35(2), 105–120 (1998)
Dasgupta, P., Hammond, P., Maskin, E.: The Implementation of Social Choice Rules: Some General Results on Incentive Compatibility. Rev. Econ. Stud. 46, 185–216 (1979)
Gans, J.S., Smart, M.: Majority Voting with Single-Crossing Preferences. J. of Public Econ. 59, 219–237 (1996)
Grandmond, J.-M.: Intermediate Preferences and Majority Rule. Econometrica 46, 317–330 (1978)
Gibbard, A.: Manipulation of Voting Schemes: A General Result. Econometrica 41, 587–601 (1973)
Le Breton, M., Zaporozhets, V.: On the Equivalence of Coalitional and Individual Strategy-Proofness Properties. Soc. Choice Welfare 33, 287–309 (2009)
Manjunath, V.: Efficient and strategy-proof social choice when preferences are single-dipped. Forthcoming in Int. J. Game Theory (2013)
Moulin, H.: On Strategy-proofness and Single Peakedness. Public Choice 35, 437–455 (1980)
Moulin, H.: Axioms of cooperative decision making. Econometric Society Monograph (1988)
Muller, E., Satterthwaite, M.A.: On the Equivalence of Strong Positive Association and Strategy-Proofness. J. Econ. Theory 14, 412–418 (1977)
Rothstein, P.: Order Restricted Preferences and Majority Rule. Soc. Choice Welfare 7, 331–342 (1990)
Saporiti, A.: Strategy-proofness and single-crossing. Theoretical Economics 4, 127–163 (2009)
Satterthwaite, M.: Strategy-Proofness and Arrow's Conditions: Existence and Correspondence Theorems for Voting Procedures and Social Welfare Functions. J. Econ. Theory 10, 187–217 (1975)
Serizawa, S.: Power of Voters and Domain of Preferences Where Voting by Committees Is Strategy-Proof. J. Econ. Theory 67, 599–608 (1995)
Sprumont, Y.: Strategyproof Collective Choice in Economic and Political Environments. Can. J. Econ. 28(1), 68–107 (1995)

Weighted Quasi-Arithmetic Means: Utility Functions and Weighting Functions

Yuji Yoshida

Faculty of Economics and Business Administration, University of Kitakyushu
4-2-1 Kitagata, Kokuraminami, Kitakyushu 802-8577, Japan
yoshida@kitakyu-u.ac.jp

Abstract. This paper discusses weighted quasi-arithmetic means from viewpoint of a combined index of utility functions and weighting functions, which represent stochastic risk in economics. The combined index characterizes decision maker's attitude and background risks in stochastic environments by conditional expectation representations of weighted quasi-arithmetic means. The first-order stochastic dominance and risk premium are demonstrated using weighted quasi-arithmetic means and aggregated mean ratios, and they are characterized by the combined index. Finally, examples of weighted quasi-arithmetic mean and aggregated mean ratio for various typical utility functions are given.

1 Introduction

This paper deals with weighted quasi-arithmetic means of an interval. Weighted quasi-arithmetic means are important tools in subjective estimation of data in decision making such as management, artificial intelligence and so on ([3–5]), and it is also strongly related to utility and stochastic risk in economics ([6]). Kolmogorov [9] and Nagumo [10] studied the aggregation operators and Aczél [1] developed the theory regarding weighted aggregation. Yoshida [12–15] has studied weighted quasi-arithmetic means of an interval by utility functions and weighting functions from viewpoint of subjective decision making. In relation to decision making, a weighted quasi-arithmetic mean is defined as follows. For a continuous strictly increasing function $f : [a, b] \mapsto (-\infty, \infty)$ as decision maker's utility function and for a continuous function $w : [a, b] \mapsto (0, \infty)$ as weighting function, a *weighted quasi-arithmetic mean* on a closed interval $[a, b]$ is given by

$$f^{-1}\left(\int_a^b f(x)w(x)\,\mathrm{d}x \Big/ \int_a^b w(x)\,\mathrm{d}x\right).$$

Hence, it represents a *mean value* given by real number $c(\in [a, b])$ satisfying

$$f(c)\int_a^b w(x)\,\mathrm{d}x = \int_a^b f(x)w(x)\,\mathrm{d}x$$

in the *first mean value theorem for integration*. We investigate the weighted quasi-arithmetic means by a combined index regarding utility functions and

V. Torra et al. (Eds.): MDAI 2013, LNAI 8234, pp. 25–36, 2013.

weighting functions extending the results in Yoshida [13–15]. Weighting functions are corresponding to stochastic risk in economics. Using conditional expectation representations of weighted quasi-arithmetic means, the combined index characterizes decision maker's attitude and background risk in stochastic environments. The first-order stochastic dominance and risk premium are also demonstrated using weighted quasi-arithmetic means and aggregated mean ratios.

In Section 2, we give the definitions of *weighted quasi-arithmetic means* and *aggregated mean ratios* of weighted quasi-arithmetic means by interior ratios, and we show the relation among weighted quasi-arithmetic mean, aggregated mean ratio and decision maker's preference/attitude based on his utility and weighting. In economics, decision maker's attitudes, for example risk neutral, risk averse and risk loving, are characterized by Arrow-Pratt index of utility functions([2, 11, 7, 8]), and risks in stochastic environments are given as an index of weighing functions. In Section 3, this paper characterizes weighted quasi-arithmetic means and mean ratios by not only utility functions but also weighing functions as a combined index. Next we investigate properties of weighted quasi-arithmetic means and aggregated mean ratios regarding combinations of utility functions and weighting functions. Representing weighted quasi-arithmetic means by conditional expectations, we investigate relation between the index for stochastic risks and risk premium in economics. We also discuss the first-order stochastic dominance using weighted quasi-arithmetic means. Finally, in Section 4, we show a lot of examples of the weighted quasi-arithmetic means and the aggregated mean ratios with various typical utility functions, and we demonstrate their relations with the classical quasi-arithmetic means.

2 Weighted Quasi-Arithmetic Means and Their Properties

In this section, we introduce weighted quasi-arithmetic means and aggregated mean ratios with utility functions and weighting functions, and we discuss sufficient conditions on utility functions and weighting functions to characterize decision maker's attitude based on quasi-arithmetic mean and aggregated mean ratio. Let D be a fixed interval which is not a singleton and we call it a domain. Let $\mathcal{C}(D)$ be the set of all nonempty bounded closed subintervals of D and let $\mathcal{C}(D)_< := \{[a,b] \in \mathcal{C}(D) | a < b\}$. Let $f : D \mapsto (-\infty, \infty)$ be a continuous strictly increasing function for utility, and let $w : D \mapsto (0, \infty)$ be a continuous function for weighting. For a closed interval $[a,b] \in \mathcal{C}(D)_<$, a mapping $M_w^f : \mathcal{C}(D) \mapsto D$ given by

$$M_w^f([a,b]) := f^{-1}\left(\int_a^b f(x)w(x)\,\mathrm{d}x \Big/ \int_a^b w(x)\,\mathrm{d}x\right) \tag{1}$$

is called *weighted quasi-arithmetic mean* with specified weighting w. For a closed interval $[a,b] \in \mathcal{C}(D)_<$ we define an interior ratio $\theta_w^f(a,b)$ from a position of weighted quasi-arithmetic mean $M_w^f([a,b])$ on the interval $[a,b]$ by

$$\theta_w^f(a,b) := \frac{M_w^f([a,b]) - a}{b-a}. \tag{2}$$

Dujmović [3–5] studied a *conjunction/disjunction degree*, which is a similar type of ratio in the power case, for computer science. This paper discusses their characterizations from viewpoint of economics. Hence we have the following results.

Lemma 1 ([13]). *Let f and g be two C^2-class utility functions on D. Let $[a,b] \in \mathcal{C}(D)_<$. Then the following (a) – (c) are equivalent.*

(a) $f''/f' \leq g''/g'$ *on* (a,b).
(b) $M_w^f([c,d]) \leq M_w^g([c,d])$ *for all* $[c,d]$ *satisfying* $[c,d] \subset [a,b]$ *and* $c < d$.
(c) $\theta_w^f(c,d) \leq \theta_w^g(c,d)$ *for all* $[c,d]$ *satisfying* $[c,d] \subset [a,b]$ *and* $c < d$.

When we may choose two utility functions f and g as decision makers' utilities, Lemma 1 says that utility f yields more risk averse results than g if $f''/f' \leq g''/g'$ on (a,b). Similarly inequality $\theta_w^f(a,b) \leq \theta_w^g(a,b)$ implies that aggregated mean ratio $\theta_w^f(a,b)$ is more risk averse than $\theta_w^g(a,b)$. Hence $-f''/f'$ is called *Arrow-Pratt index* and it implies the degree of decision maker's absolute risk aversion in micro-economics ([2, 11]). The following lemma implies the properties of weighted quasi-arithmetic mean M_w^f and ratio θ_w^f concerning weighting w.

Lemma 2 ([14, 15]). *Let $w : D \mapsto (0,\infty)$ and $v : D \mapsto (0,\infty)$ be two C^1-class weighting functions. Let $[a,b] \in \mathcal{C}(D)_<$. Then the following (a) – (c) are equivalent.*

(a) $w'/w \leq v'/v$ *on* (a,b).
(b) $M_w^f([c,d]) \leq M_v^f([c,d])$ *for all* $[c,d]$ *satisfying* $[c,d] \subset [a,b]$ *and* $c < d$.
(c) $\theta_w^f(c,d) \leq \theta_v^f(c,d)$ *for all* $[c,d]$ *satisfying* $[c,d] \subset [a,b]$ *and* $c < d$.

Arrow-Pratt index $-f''/f'$ indicates the degree of absolute risk aversion, and the index $-w'/w$ is related to *background risks* of stochastic environments in economics ([8, 15]). In next section, using representation of conditional expectations, we characterize a combination of Arrow-Pratt index $-f''/f'$ and *stochastic risk index* $-w'/w$.

3 Decision Making under Risk

In this paper, we focus on weighting functions w as risk factors of stochastic environments in weighted quasi-arithmetic mean (1) and we characterize it in relation to conditional expectation. Let D be a fixed domain and let $f : D \mapsto (-\infty,\infty)$ be a fixed continuous strictly increasing function for utility. Let (Ω, P) be a probability space, where P is a non-atomic probability measure on Ω.

Definition 1. For random variables X and Y on Ω, it is said that random variable X is *dominated by* random variable Y in the sense of *the first-order stochastic dominance* if

$$P(X < x) \geq P(Y < x) \text{ for any real number } x. \tag{3}$$

Hence the following result is well-known for the first-order stochastic dominance in economics (Arrow [2], Gollier [7], Eeckhoudt et al. [8]).

Lemma 3. *Let X and Y be random variables on Ω. Then, random variable X is dominated by random variable Y in the sense of the first-order stochastic dominance if and only if it holds that*

$$E(f(X)) \leq E(f(Y)) \tag{4}$$

for any increasing utility function $f : (-\infty, \infty) \mapsto (-\infty, \infty)$ satisfying tail condition $\lim_{x\to\pm\infty} f(x)(P(X < x) - P(Y < x)) = 0$.

The first-order stochastic dominance (3) means that stochastic environment X is risky than stochastic environment Y, and it shows in (4) that all decision makers estimate stochastic environment X smaller than stochastic environment Y with respect to their expected utilities. Then decision makers prefer stochastic environment Y to stochastic environment X with their any increasing utility functions f. Let X be a real random variable on Ω with a C^1-class density function w on $(-\infty, \infty)$. Since conditional expectation of utility $f(X)$ is

$$E(f(X) \mid a < X < b) = \frac{E(f(X)1_{\{a<X<b\}})}{P(a < X < b)} = \frac{\int_a^b f(x)w(x)\,\mathrm{d}x}{\int_a^b w(x)\,\mathrm{d}x}, \tag{5}$$

it holds that

$$M_w^f([a, b]) = f^{-1}\left(\frac{\int_a^b f(x)w(x)\,\mathrm{d}x}{\int_a^b w(x)\,\mathrm{d}x}\right) = f^{-1}(E(f(X) \mid a < X < b)) \tag{6}$$

for real numbers $a, b\,(a < b)$, where $1_{\{\cdot\}}$ implies the characteristic function of a set. From Lemma 2 and (6), we have the following result together with Lemma 3.

Lemma 4 ([13]). *Let X and Y be random variables on Ω which have C^1-class density functions w and v on $(-\infty, \infty)$ respectively. If*

$$\frac{w'}{w} \leq \frac{v'}{v} \quad \text{on } (-\infty, \infty), \tag{7}$$

then random variable X is dominated by random variable Y in the sense of the first-order stochastic dominance.

Eq. (7) is a sufficient condition for the first-order stochastic dominance (3) where stochastic environment X is risky than stochastic environment Y. Hence we find that (7) is useful to estimate risk-levels of stochastic environments and it is easy to check in actual problems. In this paper, we call $-w'/w$ *stochastic risk index*. We note that the first-order stochastic dominance (3) is a risk criterion in global area $D = (-\infty, \infty)$ for stochastic environments and it is represented by integrals in (4), however stochastic risk index $-w'/w$ can measure risks even in local areas because it is represented by differentials.

Next we discuss risk premiums regarding risk averse in financial management ([7, 8]). For simplicity, in this paper we take *initial wealth* is zero. Let $[a,b] \in \mathcal{C}(D)_<$. Let X be a random variable on Ω, which implies a *stochastic environment with some risk*. Decision making with utility f is called *risk averse on* (a,b) if

$$E(f(X) \mid a < X < b) \leq f(E(X \mid a < X < b)). \tag{8}$$

A sufficient condition for risk averse is that utility function f is concave. Let w be a density function on D for random variable X. Hence, in the following (9), real number $\pi_w^f(a,b)$ is called *risk premium on* (a,b) ([7, 8]) if it satisfies

$$E(f(X) \mid a < X < b) = f(-\pi_w^f(a,b)). \tag{9}$$

Eq.(9) means that decision maker accepts the risk arising from random variable X by paying risk premium $\pi_w^f(a,b)$.

Lemma 5 ([15]). *Let f be a continuous strictly increasing utility function on D. Let X be a random variable on Ω which has C^1-class density function w on D. The risk premium in (9) is given by*

$$\pi_w^f(a,b) = -M_w^f([a,b]). \tag{10}$$

Then we obtain the following two theorems. Theorem 1 gives an equivalence relation between combined index and weighted quasi-arithmetic means, and Theorem 2 gives an equivalence relation between combined index and risk premiums.

Theorem 1. *Let $[a,b] \in \mathcal{C}(D)_<$. Let f and g be C^2-class strictly increasing utility functions on D. Let X and Y be random variables on Ω which have C^1-class density functions w and v respectively. Then the following (a) – (c) are equivalent.*

(a) $f''/f' + 2\,w'/w \leq g''/g' + 2\,v'/v$ *on* (a,b).
(b) $M_w^f([c,d]) \leq M_v^g([c,d])$ *for all* $[c,d]$ *satisfying* $[c,d] \subset [a,b]$ *and* $c < d$.
(c) $\theta_w^f(c,d) \leq \theta_v^g(c,d)$ *for all* $[c,d]$ *satisfying* $[c,d] \subset [a,b]$ *and* $c < d$.

Theorem 2. *Let $[a,b] \in \mathcal{C}(D)_<$. Let f and g be C^2-class strictly increasing utility functions on D. Let X and Y be random variables on Ω which have C^1-class density functions w and v respectively. Then the following (a) and (b) are equivalent.*

(a) $f''/f' + 2\,w'/w \leq g''/g' + 2\,v'/v$ *on* (a,b).
(b) $\pi_w^f(c,d) \geq \pi_v^g(c,d)$ *for all* $[c,d]$ *satisfying* $[c,d] \subset [a,b]$ *and* $c < d$.

From Theorems 1 and 2, we find that combined index

$$\frac{f''}{f'} + 2\,\frac{w'}{w} \tag{11}$$

must be essential risk index of stochastic market where decision makers participates in.

4 Examples

In this section, we give examples for weighted quasi-arithmetic means which are presented in the previous sections. When we give a fixed domain D, a continuous strictly increasing function $f : D \mapsto (-\infty, \infty)$ and a fixed continuous function $w : D \mapsto (0, \infty)$, we can define weighted quasi-arithmetic mean $M_w^f([a, b])$ of an interval $[a, b] \in \mathcal{C}(D)$ by (1). We check movement of aggregated mean ratio $\theta_w^f(a, b)$, which is given by (2), with respect to parameters a and b in local regions and global regions in each example. First we discuss several examples of utility functions f.

Example 1.

(i) *(Linear case)* Let $D = (-\infty, \infty)$ and take utility function $f(x) = x$ for $x \in D$. Then $f''(x)/f'(x) = 0$. For a closed interval $[a, b] \in \mathcal{C}(D)_<$, we define *risk neutral weighted mean* $N_w([a, b])$ and its aggregated mean ratio $\nu_w(a, b)$ by

$$N_w([a,b]) := \int_a^b x\ w(x)\,\mathrm{d}x \Big/ \int_a^b w(x)\,\mathrm{d}x \tag{12}$$

and

$$\nu_w(a,b) := \frac{N_w([a,b]) - a}{b-a} = \int_a^b (x-a)w(x)\,\mathrm{d}x \Big/ \int_a^b (b-a)w(x)\,\mathrm{d}x. \tag{13}$$

Take weighting function $w(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_n x^n$ on $D = (0, \infty)$ with positive constants $c_0, c_1.c_2, \cdots, c_n$. Then

$$\frac{f''(x)}{f'(x)} + 2\,\frac{w'(x)}{w(x)} = 2\,\frac{\sum_{k=0}^{n-1}(k+1)c_{k+1}x^k}{\sum_{k=0}^{n} c_k x^k}. \tag{14}$$

For $[a, b] \subset D$ such that $a < b$, we have

$$N_w([a,b]) = \frac{\sum_{k=0}^{n} \frac{1}{k+2} c_k (b^{k+2} - a^{k+2})}{\sum_{k=0}^{n} \frac{1}{k+1} c_k (b^{k+1} - a^{k+1})}.$$

From Yoshida [13, Theorem 5.10], it holds that $\lim_{b \downarrow a} \nu_w(a, b) = \lim_{a \uparrow b} \nu_w(a, b) = 1/2$,

$$\lim_{a \downarrow 0} \nu_w(a,b) = \frac{\sum_{k=0}^{n} \frac{1}{k+2} c_k b^{k+2}}{\sum_{k=0}^{n} \frac{1}{k+1} c_k b^{k+1}} \quad \text{and} \quad \lim_{b \to \infty} \nu_w(a,b) = \frac{n+1}{n+2}.$$

(ii) *(Power case)* Take utility function $f(x) = x^r$ and weighting function $w(x) = x^\alpha$ on $D = (0, \infty)$ with constants r, α satisfying $r \neq 0$. Then

$$\frac{f''(x)}{f'(x)} + 2\,\frac{w'(x)}{w(x)} = \frac{r-1}{x} + \frac{2\alpha}{x}. \tag{15}$$

Hence we can deal with not only $r > 0$ for increasing function $f(x) = x^r$ but also $r < 0$ for decreasing function $f(x) = x^r$. For $[a, b] \subset D$ such that $a < b$, weighted quasi-arithmetic mean is given by the following $M^{(r)}_{(\alpha)}([a, b]) := M^f_w([a, b])$:

$$M^{(r)}_{(\alpha)}([a,b]) = \left(\frac{(1+\alpha)(b^{1+\alpha+r} - a^{1+\alpha+r})}{(1+\alpha+r)(b^{1+\alpha} - a^{1+\alpha})}\right)^{1/r}$$

if $r \neq 0, \alpha \neq -1, \alpha + r \neq -1$. The limiting values regrading r and α are

$$\lim_{\alpha \to -r-1} M^{(r)}_{(\alpha)}([a,b]) = ab\left(\frac{r(\log b - \log a)}{b^r - a^r}\right)^{1/r} \quad \text{if } r \neq 0,$$

$$\lim_{\alpha \to -1} M^{(r)}_{(\alpha)}([a,b]) = \left(\frac{r(\log b - \log a)}{b^r - a^r}\right)^{-1/r} \quad \text{if } r \neq 0,$$

$$\lim_{r \to 0} M^{(r)}_{(\alpha)}([a,b]) = \exp\left(\frac{b^{\alpha+1}\log b - a^{\alpha+1}\log a}{b^{\alpha+1} - a^{\alpha+1}} - \frac{1}{\alpha+1}\right) \quad \text{if } \alpha \neq -1,$$

$$\lim_{\alpha \to -1}\lim_{r \to 0} M^{(r)}_{(\alpha)}([a,b]) = \sqrt{ab},$$

$$\lim_{r \to -\infty} M^{(r)}_{(\alpha)}([a,b]) = a,$$

$$\lim_{r \to \infty} M^{(r)}_{(\alpha)}([a,b]) = b.$$

From Yoshida [13, Corollary 5.4] we also have

$$\theta^f_w(a,b) \lesseqgtr \nu_w(a,b) \quad \text{if } r \lesseqgtr 1.$$

From Yoshida [13, Theorems 5.9 and 5.10], it holds that $\lim_{b \downarrow a} \theta^f_w(a,b) = \lim_{a \uparrow b} \theta^f_w(a,b) = 1/2$ and

$$\lim_{a \downarrow 0} \theta^f_w(a,b) = \lim_{b \to \infty} \theta^f_w(a,b) = \left(\frac{1+\alpha}{1+\alpha+r}\right)^{1/r}.$$

(iii) *(Logarithmic case)* Take concave utility function $f(x) = r \log x$ and weighting function $w(x) = x^\alpha$ on $D = (0, \infty)$ with constants r, α satisfying $r > 0$. Then

$$\frac{f''(x)}{f'(x)} + 2\frac{w'(x)}{w(x)} = -\frac{1}{x} + \frac{2\alpha}{x}. \tag{16}$$

For $[a, b] \subset D$ such that $a < b$, we can check

$$M^f_w([a,b]) = \exp\left(\frac{b^{\alpha+1}\log b - a^{\alpha+1}\log a}{b^{\alpha+1} - a^{\alpha+1}} - \frac{1}{\alpha+1}\right),$$

and Yoshida [13, Corollary 5.4] implies $\theta^f_w(a,b) < \nu_w(a,b)$. Yoshida [13, Theorems 5.9 and 5.10] also imply $\lim_{b \downarrow a} \theta^f_w(a,b) = \lim_{a \uparrow b} \theta^f_w(a,b) = 1/2$ and

$$\lim_{a \downarrow 0} \theta^f_w(a,b) = \lim_{b \to \infty} \theta^f_w(a,b) = \exp\left(-\frac{1}{\alpha+1}\right).$$

(iv) *(Exponential case)* Take convex utility function $f(x) = e^{sx}$ and weighting function $w(x) = x^{\alpha}$ on $D = (0, \infty)$ with constants r, α satisfying $s > 0$. Then

$$\frac{f''(x)}{f'(x)} + 2\,\frac{w'(x)}{w(x)} = s + \frac{2\alpha}{x}. \tag{17}$$

For $[a, b] \subset D$ such that $a < b$, we can check

$$M_w^f([a,b]) = \frac{1}{s}\log\left(\frac{(1+\alpha)(\Gamma(1+\alpha,-sb) - \Gamma(1+\alpha,-sa))}{s^{1+\alpha}(b^{1+\alpha} - a^{1+\alpha})}\right)$$

and Yoshida [13, Corollary 5.4] implies $\nu_w(a,b) < \theta_w^f(a,b)$, where we put $\Gamma(\alpha+1, z) = \int_z^{\infty} x^{\alpha}e^{-x}dx$ for $z \geq 0$. From Yoshida [13, Theorem 5.9], we obtain $\lim_{b\downarrow a}\theta_w^f(a,b) = \lim_{a\uparrow b}\theta_w^f(a,b) = 1/2$.

(v) Take utility function $f(x) = x^r$ and weighting function $w(x) = x^{\alpha}e^{-\beta x}$ on $D = (-\infty, \infty)$, where $r \neq 0$. Then

$$\frac{f''(x)}{f'(x)} + 2\,\frac{w'(x)}{w(x)} = \frac{r-1}{x} + 2\left(\frac{\alpha}{x} - \beta\right) \tag{18}$$

for $x \in D$. Then for $[a, b] \in \mathcal{C}(D)_<$ we have

$$M_w^f([a,b]) = \left(\frac{\Gamma(1+\alpha+r,\beta b) - \Gamma(1+\alpha+r,\beta a)}{\beta^r(\Gamma(1+\alpha,\beta b) - \Gamma(1+\alpha,\beta a))}\right)^{1/r},$$

where $\Gamma(\cdot,\cdot)$ is defined by $\Gamma(c,z) := \int_z^{\infty} x^{c-1}e^{-x}\,\mathrm{d}x$ for $c > 0$.

(vi) Take utility function $f(x) = e^{sx}$ and weighting function $w(x) = x^{\alpha}e^{-\beta x}$ on $D = (-\infty, \infty)$, where $s \neq 0$. Then

$$\frac{f''(x)}{f'(x)} + 2\,\frac{w'(x)}{w(x)} = s + 2\left(\frac{\alpha}{x} - \beta\right) \tag{19}$$

for $x \in D$. Let $[a, b] \subset D = (-\infty, \infty)$ such that $a < b$. Then, for $[a, b] \in \mathcal{C}(D)_<$ we have

$$M_w^f([a,b]) = \frac{1}{s}\log\left(\frac{\beta^{1+\alpha}(\Gamma(1+\alpha,(\beta-s)b) - \Gamma(1+\alpha,(\beta-s)a))}{(\beta-s)^{1+\alpha}(\Gamma(1+\alpha,\beta b) - \Gamma(1+\alpha,\beta a))}\right).$$

(vii) Take utility function $f(x) = x^r e^{sx}$ and weighting function $w(x) = x^{\alpha}e^{-\beta x}$ on $D = (-\infty, \infty)$, where $r \neq 0$. Then

$$\frac{f''(x)}{f'(x)} + 2\,\frac{w'(x)}{w(x)} = \frac{-r + (r+sx)^2}{(r+sx)x} + 2\left(\frac{\alpha}{x} - \beta\right) \tag{20}$$

for $x \in D$. Let $[a, b] \subset D = (-\infty, \infty)$ such that $a < b$. Then for $[a, b] \in \mathcal{C}(D)_<$ we have $M_w^f([a, b]) =$

$$\frac{r}{s}L\left(\frac{s}{r}\left(\frac{\beta^{1+\alpha}(\Gamma(1+\alpha+r,(\beta-s)b) - \Gamma(1+\alpha+r,(\beta-s)a))}{(\beta-s)^{1+\alpha+r}(\Gamma(1+\alpha,\beta b) - \Gamma(1+\alpha,\beta a))}\right)^{1/r}\right),$$

where L is the inverse function of function $x \mapsto xe^x$.

(viii) Take utility function $f(x) = e^{sx}$ and weighting function $w(x) = e^{-\beta x - \gamma x^2}$ on $D = (-\infty, \infty)$, where $s \neq 0$. We have

$$\frac{f''(x)}{f'(x)} + 2\,\frac{w'(x)}{w(x)} = s - 2(\beta + 2\gamma x)$$

for $x \in D$. Let $[a, b] \subset D = (-\infty, \infty)$ such that $a < b$. Then for $[a, b] \in \mathcal{C}(D)_<$ it holds that

$$M_w^f([a,b]) = \frac{1}{s}\log\left(\frac{\exp\left(\frac{(\beta-r)^2-\beta^2}{4\gamma}\right)\left(\operatorname{erf}\left(\frac{\beta-r+2\gamma b}{2\sqrt{\gamma}}\right) - \operatorname{erf}\left(\frac{\beta-r+2\gamma a}{2\sqrt{\gamma}}\right)\right)}{\operatorname{erf}\left(\frac{\beta+2\gamma b}{2\sqrt{\gamma}}\right) - \operatorname{erf}\left(\frac{\beta+2\gamma a}{2\sqrt{\gamma}}\right)}\right),$$

where $\operatorname{erf}(z) := \frac{2}{\sqrt{\pi}}\int_0^z e^{-x^2}\,\mathrm{d}x$.

Next we give three examples as applications of Theorems 1 and 2. The following examples show the cases that two decision making with utilities f and g are compared.

Example 2.

(i) *(Square root case and logarithmic case)* Let domain $D = (0, \infty)$. Take concave utility functions $f(x) = \sqrt{x}$ and $g(x) = \log x$ on D and take weighting functions $w(x) = x^\alpha e^{-\beta x - x^2/4}$ and $v(x) = \lambda e^{-\lambda x}$. Then we have

$$\frac{f''(x)}{f'(x)} + 2\,\frac{w'(x)}{w(x)} = -\frac{1}{2x} + \frac{2\alpha}{x} - 2\beta - x, \tag{21}$$

$$\frac{g''(x)}{g'(x)} + 2\,\frac{v'(x)}{v(x)} = -\frac{1}{x} - 2\lambda. \tag{22}$$

Therefore it follows

$$\frac{f''(x)}{f'(x)} + 2\,\frac{w'(x)}{w(x)} \leq \frac{g''(x)}{g'(x)} + 2\,\frac{v'(x)}{v(x)} \iff x^2 - 2(\lambda - \beta)x - 2\alpha - \frac{1}{2} \geq 0$$

for $x \in D$. If $(\lambda - \beta)^2 + 2\lambda + 1/2 \leq 0$,

$$\frac{f''(x)}{f'(x)} + 2\,\frac{w'(x)}{w(x)} \leq \frac{g''(x)}{g'(x)} + 2\,\frac{v'(x)}{v(x)} \qquad \text{for all } x \in D.$$

If $(\lambda - \beta)^2 + 2\lambda + 1/2 > 0$,

$$\frac{f''(x)}{f'(x)} + 2\,\frac{w'(x)}{w(x)} \leq \frac{g''(x)}{g'(x)} + 2\,\frac{v'(x)}{v(x)} \iff x \in (-\infty, x_-] \cup [x_+, \infty),$$

where $x_\pm := \lambda - \beta \pm \sqrt{(\lambda - \beta)^2 + 2\lambda + 1/2}$. From Theorem 1, we obtain $\theta_w^f(a,b) < \theta_v^g(a,b)$ for $[a,b] \in \mathcal{C}(D)$ such that $a < b$, where $\theta_w^f(a,b)$ is aggregated mean ratio given by $f(x)$ and $\theta_v^g(a,b)$ is aggregated mean ratio given by $g(x)$. This shows that $f(x)$ is more risk averse than $g(x)$ as decision making.

(ii) *(Exponential case and logarithmic case)* Let domain $D = (0, \infty)$. Take concave utility functions $f(x) = 1 - e^{-2\lambda x}$ and $g(x) = \log x$ on D and take weighting functions $w(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\,\sigma^2}\right)$ and $v(x) = \sqrt{x}\, e^{-\beta x - x^2/2}$. Then we have

$$\frac{f''(x)}{f'(x)} + 2\,\frac{w'(x)}{w(x)} = -\frac{2(x - \mu + \lambda\sigma^2)}{\sigma^2}, \tag{23}$$

$$\frac{g''(x)}{g'(x)} + 2\,\frac{v'(x)}{v(x)} = -2(x + \beta). \tag{24}$$

Therefore, if $\sigma^2 \neq 1$,

$$\frac{f''(x)}{f'(x)} + 2\,\frac{w'(x)}{w(x)} \leq \frac{g''(x)}{g'(x)} + 2\,\frac{v'(x)}{v(x)} \iff x \begin{cases} \leq x_3 & \text{if } \sigma^2 > 1 \\ \geq x_3 & \text{if } \sigma^2 < 1, \end{cases}$$

where $x_3 = \frac{(\lambda-\beta)\sigma^2 - \mu}{\sigma^2 - 1}$. If $\sigma^2 = 1$,

$$\frac{f''(x)}{f'(x)} + 2\,\frac{w'(x)}{w(x)} \leq \frac{g''(x)}{g'(x)} + 2\,\frac{v'(x)}{v(x)} \text{ for all } x \in D \iff \mu + \beta - \lambda \leq 0.$$

From Theorem 1, we obtain $\theta^f_w(a,b) < \theta^g_v(a,b)$ for $[a,b] \subset (0,1]$ such that $a < b$ and we also obtain $\theta^f_w(a,b) > \theta^g_v(a,b)$ for $[a,b] \subset [1,\infty)$ such that $a < b$, where $\theta^f_w(a,b)$ is aggregated mean ratio given by $f(x)$ and $\theta^g_v(a,b)$ is aggregated mean ratio given by $g(x)$. This shows that $f(x)$ is more risk averse than $g(x)$ in the region $(0,1)$ and that $f(x)$ is more risk loving than $g(x)$ in the region $(1,\infty)$. This example shows that decision makers' attitudes are comparable in each local area using the index $f''/f' + 2\,w'/w$.

(iii) *(Weighted quasi-arithmetic means and conditional expectations)* Finally we show the relation between weighted quasi-arithmetic means and conditional expectations and their application to economics. We give an example for Theorems 1 and 2 by normal distributions on stochastic environments. Let domain $D = (0, \infty)$. Take concave utility functions $f(x) = x^r$ and $g(x) = x^s$ on D. Let random variables X and Y have normal distributions on Ω with density functions w and v respectively as follows: Let μ_X and μ_X be the means and let σ_X and σ_Y be the standard deviations for w and v respectively,

$$w(x) = \frac{1}{\sqrt{2\pi\sigma_X^2}} \exp\left(-\frac{(x-\mu_X)^2}{2\,\sigma_X^2}\right)$$

and

$$v(x) = \frac{1}{\sqrt{2\pi\sigma_Y^2}} \exp\left(-\frac{(x-\mu_Y)^2}{2\,\sigma_Y^2}\right)$$

for real numbers x. Then we have

$$\begin{aligned}
&\frac{f''(x)}{f'(x)} + 2\,\frac{w'(x)}{w(x)} \le \frac{g''(x)}{g'(x)} + 2\,\frac{v'(x)}{v(x)} \\
&\iff \frac{r-1}{x} - \frac{2(x-\mu_X)}{\sigma_X^2} \le \frac{s-1}{x} - \frac{2(x-\mu_Y)}{\sigma_Y^2} \\
&\iff (r-s)\sigma_X^2\sigma_Y^2 - 2(\sigma_X^2\mu_Y - \sigma_Y^2\mu_X)x + 2(\sigma_X^2 - \sigma_Y^2)x^2 \le 0 \\
&\iff x \in D_{\le} := \begin{cases}
(-\infty,\infty) & \text{if } \sigma_X < \sigma_Y \text{ and } \eta \le 0 \\
(-\infty, x_-] \cup [x_+,\infty) & \text{if } \sigma_X < \sigma_Y \text{ and } \eta > 0 \\
[x_-, x_+] & \text{if } \sigma_X > \sigma_Y \text{ and } \eta < 0 \\
\emptyset & \text{if } \sigma_X > \sigma_Y \text{ and } \eta \ge 0 \\
[x_4,\infty) & \text{if } \sigma_X = \sigma_Y \text{ and } \mu_X < \mu_Y \\
(-\infty, x_4] & \text{if } \sigma_X = \sigma_Y \text{ and } \mu_X > \mu_Y \\
(-\infty,\infty) & \text{if } \sigma_X = \sigma_Y \text{ and } \mu_X = \mu_Y,
\end{cases}
\end{aligned}$$

where $x_4 = \frac{(r-s)\sigma_X^2}{2(\mu_Y-\mu_X)}$, $\eta := (\sigma_X^2\mu_Y - \sigma_Y^2\mu_X)^2 - 2(r-s)(\sigma_X^2 - \sigma_Y^2)\sigma_X^2\sigma_Y^2$, $x_\pm := \frac{\sigma_X^2\mu_Y - \sigma_Y^2\mu_X \pm \sqrt{\eta}}{(r-s)\sigma_X^2\sigma_Y^2}$. By Theorems 1 and 2 we get $M_w^f([a,b]) \le M_v^f([a,b])$ and $\pi_w^f(a,b) = -M_w^f([a,b]) \ge -M_v^f([a,b]) = \pi_v^f(a,b)$ for subintervals $[a,b] \subset D_{\le}$. Further if $\sigma_X < \sigma_Y$ and $\eta \le 0$, all agents prefers stochastic environment Y to stochastic environment X for his any increasing utility f, i.e. it holds that $E(f(X)) \le E(f(Y))$ for any increasing utility function f, which is equivalent that X is dominated by Y in the sense of the first-order stochastic dominance.

5 Conclusions

We have analyzed weighted quasi-arithmetic means with utility functions and weighting for random factors in stochastic environments. We have investigated a lot of examples of weighted quasi-arithmetic means and aggregated mean ratio for various typical utility functions. Stochastic dominance is a risk criterion in a global area for stochastic environments. Using combined index $f''/f' + 2w'/w$, we can analyze risks even in local areas. Combined index $f''/f' + 2w'/w$ will be useful and easy to calculate in actual problems.

References

1. Aczél, J.: On weighted mean values. Bulletin of the American Math. Society 54, 392–400 (1948)
2. Arrow, K.J.: Essays in the Theory of Risk-Bearing, Markham, Chicago (1971)
3. Dujmović, J.J.: Weighted Conjunctive and disjunctive means and their application in system evaluation. Univ. Beograd. Publ. Elektotech. Fak. Ser. Mat. Fiz. 483, 147–158 (1974)
4. Dujmović, J.J., Larsen, H.L.: Generalized Conjunction/disjunction. International Journal of Approximate Reasoning 46, 423–446 (2007)

5. Dujmović, J.J., Nagashima, H.: LSP method and its use for evaluation of Java IDEs. International Journal of Approximate Reasoning 41, 3–22 (2006)
6. Fishburn, P.C.: Utility Theory for Decision Making. John Wiley and Sons, New York (1970)
7. Gollier, G.: The Economics of Risk and Time. MIT Publishers (2001)
8. Eeckhoudt, L., Gollier, G., Schkesinger, H.: Economic and Financial Decisions under Risk. Princeton University Press (2005)
9. Kolmogoroff, A.N.: Sur la notion de la moyenne. Acad. Naz. Lincei Mem. Cl. Sci. Fis. Mat. Natur. Sez. 12, 388–391 (1930)
10. Nagumo, K.: Über eine Klasse der Mittelwerte. Japanese Journal of Mathematics 6, 71–79 (1930)
11. Pratt, J.W.: Risk Aversion in the Small and the Large. Econometrica 32, 122–136 (1964)
12. Yoshida, Y.: Aggregated mean ratios of an interval induced from aggregation operations. In: Torra, V., Narukawa, Y. (eds.) MDAI 2008. LNCS (LNAI), vol. 5285, pp. 26–37. Springer, Heidelberg (2008)
13. Yoshida, Y.: Quasi-arithmetic means and ratios of an interval induced from weighted aggregation operations. Soft Computing 14, 473–485 (2010)
14. Yoshida, Y.: Weighted quasi-arithmetic means and conditional expectations. In: Torra, V., Narukawa, Y., Daumas, M. (eds.) MDAI 2010. LNCS, vol. 6408, pp. 31–42. Springer, Heidelberg (2010)
15. Yoshida, Y.: Weighted quasi-arithmetic means and and a risk index for stochastic environments. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems (IJUFKS) 16(suppl.), 1–16 (2011)

Toward a General Framework for Information Fusion

Didier Dubois[1,2], Weiru Liu[2], Jianbing Ma[3], and Henri Prade[1]

[1] IRIT, CNRS & Université de Toulouse, France
[2] School of Electronics, Electrical Engineering and Computer Science, Queens University, Belfast, UK
[3] School of Design, Engineering and Computing, Bournemouth University, Bournemouth, UK

Abstract. Depending on the representation setting, different combination rules have been proposed for fusing information from distinct sources. Moreover in each setting, different sets of axioms that combination rules should satisfy have been advocated, thus justifying the existence of alternative rules (usually motivated by situations where the behavior of other rules was found unsatisfactory). These sets of axioms are usually purely considered in their own settings, without in-depth analysis of common properties essential for all the settings. This paper introduces core properties that, once properly instantiated, are meaningful in different representation settings ranging from logic to imprecise probabilities. The following representation settings are especially considered: classical set representation, possibility theory, and evidence theory, the latter encompassing the two other ones as special cases. This unified discussion of combination rules across different settings is expected to provide a fresh look on some old but basic issues in information fusion.

1 Introduction

In information fusion, each piece of information is assumed to come from a different source (measurement device or expert opinion) and the fusion is a process aiming at grasping what is known about a situation being observed. This contrasts with preference aggregation where preferences merely reflect what some agent would like the result to be, and the aggregation process is more about building compromises than finding what the true state of a situation is. The pieces of information to be fused may be inconsistent, and are often pervaded with uncertainty, which must be reflected on the result.

The information fusion problem is met in different representation settings, ranging from the merging of logical knowledge/belief bases supposed to encode the states of mind of agents about the perception of a situation ([11] in classical logic, [1] in possibilistic logic for the merging of stratified or prioritized bases), to numerical-based frameworks, such as, probability theory [20], evidence theory [17], possibility theory [10], or imprecise probability theory [22]. It is worth-noticing that all the abovementioned settings can handle epistemic uncertainty and incomplete knowledge with the exception of probability theory that often accounts with variability and randomness, while the Bayesian approach to subjective probability yields a questionable representation of incomplete information [5]. In that respect, it is important to keep in mind the fact that, formally speaking, evidence theory encompasses both probability theory and

V. Torra et al. (Eds.): MDAI 2013, LNAI 8234, pp. 37–48, 2013.

possibility theory as particular cases; in turn, evidence theory can be seen as a particular imprecise probability system [23]; and binary-valued possibility theory is nothing but a Boolean representation for imprecise pieces of information at work in propositional or epistemic logic.

It is striking to observe that the information fusion problem until now has been discussed independently in each setting. Sometimes, specific postulates that govern fusion operations are provided [21,11,14]. Moreover in each setting, various combination rules have been advocated as behaving properly (on the basis of good properties) as opposed to the unsatisfactory behavior of other rules. In practice, we are faced with many combination rules (their number is still increasing!), and several postulate systems. It is worthwhile to provide a more unified view of the problem.

In this paper, we aim to propose common properties of fusion operators valid in any setting. They do account for various existing axiomatic systems proposed in specific settings. These properties are stated at the semantic level, rather than at the syntactic one (unlike [11]), since probabilistic settings do not have a well-established logical counterpart. Moreover, the semantical level is especially appropriate for laying bare the practical meaning of the combination rules. This provides a common ground for a rational exploration of fusion methods, despite the heterogeneity of existing frameworks. Particular instantiations of these common properties in the different settings are then considered.

The rest of the paper is organized as follows. The next section introduces eight core properties, before considering their instantiations, in Section 3 in the classical set representation and in the possibility theory setting, and in Section 4 in the context of evidence theory, in which many different combination rules have been proposed. These properties provide a basis for comparing these alternative rules.

2 Core Properties

In order to define a set of required properties that make sense in different settings ranging from logic to imprecise probability, we consider an abstract notion of information item, denoted by T, supplied by sources. Let $\Omega = \{\omega_1, ..., \omega_{|\Omega|}\}$ be a finite, non-empty set of possible worlds (e.g. the range of some unknown quantity), one of which is the true one. There are n experts/sources and the i_{th} expert/source is denoted by i. Let T_i be the information provided by i, e.g., T_i may be a basic belief assignment, a possibility distribution, or a knowledge base. $T = f(T_1, \ldots, T_n)$ denotes the fusion result using aggregation operator f over a set of information items T_i. To any information item, we associate the following features:

- The subset $\mathcal{S}(T) \subseteq \Omega$, called the *support* of T, contains the set of values considered possible by information T. It means that $\omega_i \notin \mathcal{S}(T) \iff \omega_i$ is impossible.
- Its *core* $\mathcal{C}(T) \subseteq \Omega$ contains the set of values considered fully plausible according to information T. The idea is that, by default, if information T is taken for granted, a first guess for the value of x should be an element of $\mathcal{C}(T)$. Clearly, $\mathcal{C}(T) \subseteq \mathcal{S}(T)$.
- *Internal Consistency* An information item T is said to be weakly (resp. strongly) consistent if $\mathcal{S}(T) \neq \emptyset$ (resp. $\mathcal{C}(T) \neq \emptyset$) otherwise information T is totally (resp. weakly) inconsistent. In the following, we assume $\mathcal{C}(T) \neq \emptyset$ for each source.

Strong consistency is assumed for inputs of a merging process, and weak consistency at worst for the output.

- *Mutual consistency* T and T' are said to be weakly mutually consistent when $\mathcal{S}(T) \cap \mathcal{S}(T)' \neq \emptyset$ and strongly so when $\mathcal{C}(T) \cap \mathcal{C}(T)' \neq \emptyset$.
- *Information ordering*: $T \sqsubseteq T'$ expresses that T provides at least as much information as T'. In particular, $T \sqsubseteq T'$ should imply $\mathcal{S}(T) \subseteq \mathcal{S}(T')$.
- *Plausibility ordering*: If consistent, information T induces a partial preorder $\succeq_T$ expressing relative plausibility: $\omega \succeq_T \omega'$ means that ω is at least as plausible as (or *dominates*) ω' according to T. We write $\omega \sim_T \omega'$ if $\omega \succeq_T \omega'$ and $\omega' \succeq_T \omega$. Of course, if $\omega \in \mathcal{S}(T), \omega' \notin \mathcal{S}(T)$, then $\omega \succ_T \omega'$ (ω is strictly more plausible than ω').

The *vacuous information*, expressing total ignorance is denoted by $T^\top$. Then the plausibility ordering is flat: $\mathcal{S}(T^\top) = \mathcal{C}(T^\top) = \Omega$ and $\omega \sim_{T^\top} \omega' \; \forall \omega, \omega' \in \Omega$.

The process of merging information items, supplied by sources whose reliability levels are not known, is guided by a few first principles (already in [21]):

- It is a basically symmetric process as the sources play the same role and supply information of the same kind;
- We try to use as many information items as possible in the fusion process, so as to get a result that is as precise and useful as possible. However, the result should not be arbitrarily precise, but faithful to the level of informativeness of the inputs.
- Information fusion should try to solve conflicts between sources, while neither dismissing nor favoring any of them without a reason.

These principles are implemented in the postulates listed below, called *core properties*, which are meant to be natural minimal requirements, independent of the actual representation framework.

Property 1: Unanimity.

When all sources agree on some results, then the latter should be preserved. Minimal conditions are

(a) Possibility preservation. If for all sources ω is possible, then so should the fusion result assert: if $\forall i, \omega \in \mathcal{S}(T_i)$ then $\omega \in \mathcal{S}(f(T_1, ..., T_n))$.

(b) Impossibility preservation. If all sources believe that a possible world ω is impossible, then this ω cannot become (even slightly) possible after fusion. This can be expressed as $\mathcal{S}(f(T_1, ...T_n)) \subseteq \mathcal{S}(T_1) \cup ... \cup \mathcal{S}(T_n)$.

Property 2: Informational Monotonicity.

If a set of agents provides less information than another set of *non-disagreeing* agents, then fusing the former inputs should not produce a more informative result than fusing the latter. The weakest such requirement is:

Weak Informational Monotonicity. if $\forall i, T_i \sqsubseteq T'_i$, then $f(T_1, ...T_n) \sqsubseteq f(T'_1, ...T'_n)$, provided that all the inputs are globally strongly mutually consistent.

Property 3: Consistency Enforcement.

This property requires that fusing individually consistent inputs should give a consistent result. At best: $\mathcal{C}(f(T_1, ...T_n)) \neq \emptyset$. At least: $\mathcal{S}(f(T_1, ...T_n)) \neq \emptyset$.

Property 4: Optimism.
In the absence of specific information about source reliability, one should assume as many sources as possible are reliable, in agreement with their observed mutual consistency. In particular: If $\cap_{i=1}^{n}\mathcal{C}(T_i) \neq \emptyset$ then $f(T_1, ..., T_n) \sqsubseteq T_i, \forall i = 1,...,n$. In general, it should be assumed that at least one source is reliable.

Property 5: Fairness. The fusion result reconciles all sources. Hence, the result of the fusion process should keep something from each input, i.e., $\forall i = 1,...,n, \mathcal{S}(f(T_1,...,T_n)) \cap \mathcal{S}(T_i) \neq \emptyset$.

Property 6: Insensitivity to Vacuous Information.
Sources that provide vacuous information should not affect the fusion result: $f_n(T_1, \ldots, T_{i-1}, T^{\top}, T_{i+1}, \ldots, T_n) = f_{n-1}(T_1, \ldots, T_{i-1}, T_{i+1}, \ldots, T_n)$

Property 7: Commutativity.
Inputs from multiple sources are treated on a par, and the combination should be symmetric (up to their relative reliability).

Property 8. Minimal Commitment.
The result of the fusion should be as little informative as possible (in the sense of $\sqsubseteq$) among possible results that satisfy the other core properties.

Some comments are in order. The general core properties proposed here have counterparts in properties considered in different particular settings ; see especially [21] and also [11]. Let us further discuss each of them.

Possibility and impossibility preservation can be found in possibility theory [14] and imprecise probability [21]. It makes sense to request more than possibility preservation: plausibility preservation, replacing supports by cores [21]. The strongest form of Unanimity (Prop. 1) is *idempotence*: if $\forall i, T_i = T$, $f(T_1, ..., T_n) = T$. However, while it makes sense if sources are redundant, adopting it in all situations forbids reinforcement effects to take place when sources are independent [9]. Our Unanimity properties minimally respect the agreement between sources. A slightly more demanding requirement which leaves room for reinforcement effects can be: *Local Ordinal Unanimity*: $\forall\omega$ and ω', if ω is at least as plausible as ω', then so should it be after fusion. e.g., ω dominates ω'. Formally: if $\forall i, \omega \succeq_{T_i} \omega'$, then $\omega \succeq_{f(T_1,...,T_n)} \omega'$.

Informational Monotonicity (Prop.2), adopted as a general property in [14] should be restricted to when information items supplied by sources do not contradict each other. Indeed, if conflicting, it is always possible to make these information items less informative in such a way that they become consistent. In that case the result of the fusion may become very precise by virtue of Optimism Prop. 4, and in particular, more informative than the union of the supports of original precise conflicting items of information.

Consistency enforcement (Prop. 3) is instrumental if the result of the merging is to be useful in practice: one must extract something non-trivial, even if tentative, from available information. It is a typical requirement from the logical area [11] and a property taken for granted by numerical approaches (viz. Dempster rule of combination, but also for imprecise probabilities [21]). Still, when the representation setting is refined enough, there are gradations in consistency requirements, and Prop. 3 can be interpreted in a flexible way. For example, the re-normalisation of belief functions or possibility distributions obtained by merging is not always compulsory, even if sub-normalisation expresses a form of inconsistency.

Optimism (Prop. 4) underlies the idea of making the best of the available information: If items of information are globally consistent with each other, there is no reason to question the reliability of the sources. It is again a typical assumption in logical settings [11], but Walley [21] tries to formulate a similar property. In case of strong inconsistency, this assumption is not sustainable. Note that in the latter case (in particular if $\cap_{i=1}^{n} \mathcal{S}(T_i) = \emptyset$), and under the Impossibility Preservation property (1b), the support of the result should be at worst the union of the supports of inputs, i.e., $\mathcal{S}(f(T_1, ...T_n)) \subseteq \mathcal{S}(T_1) \cup ... \cup \mathcal{S}(T_n)$, now assuming that at least one source is reliable (still a form of optimism in the presence of inconsistency). The latter requirement sounds natural for two sources only, but may be found overcautious for many sources. In particular, Optimism will lead to replace any group K of strongly consistent sources, by a single source that is more informative than and in agreement with each of them.

Fairness (Prop. 5) ensures that all input items participate to the result. At the same time, it favors no source by forbidding any input to be derived from the output result in the case of inconsistency. Note that different versions of the Fairness property can be found in the literature. In particular, a form of this property was already suggested by Walley [21] for imprecise probabilities. In the logical setting [11], the counterpart of the condition $\mathcal{S}(f(T_1,...,T_n)) \cap \mathcal{S}(T_i) \neq \emptyset$ is required to hold either for each i, or for none. The possibility that it holds for none sounds highly debatable using supports, from a knowledge fusion point of view, while it may be acceptable when fusing preferences, which is more a matter of trade-off, or when supports are changed into cores.

Insensitivity to Vacuous Information (Prop. 6) looks obvious, not to say redundant, but dispensing with it may lead to uninformative results. It appears again in the Walley postulates [21] for merging sets of probabilities. Prop. 6 implicitly admits that a non informative source is assimilated to one that does not express any opinion, and is typical of information fusion. It excludes probabilistic fusion rules like averaging, since it is sensitive to vacuous information (represented, e.g., by a uniform distribution).

Commutativity (Prop. 7) is characteristic of fusion processes as opposed to revision where prior knowledge may be altered by input information. In contrast, information fusion deals with inputs received in parallel. So, commutativity makes sense, if no information is available on the reliability of sources.

Minimal Commitment is a very important postulate that applies in many circumstances. It is central in all uncertainty theories handling incomplete information under different terminologies, including in logic-based approaches (where it is implicit). It considers as possible any state of affairs not explicitly discarded. It is called *principle of minimal specificity* in possibility theory [10], *principle of Minimal Commitment* in evidence theory [19], and it underlies the so-called *natural extension* in imprecise probability theory [22]. This is a cautious principle that is nicely counterbalanced by the Optimism postulate, and this equilibrium is sometimes useful to characterise the unicity of fusion rules: Optimism provides an upper limit to the set of possible worlds and Minimal Commitment a lower limit.

Some other properties may be required in aggregation processes, such as associativity, which makes computation more efficient, but lacking associativity is not a fatal flaw in itself (e.g., the MCS rule below), if the rule can be defined for n sources.

3 Merging Set-Valued Information: Hard Constraints

The most elementary setting one may first consider is the one of sets, whereby any information item is a subset of possible worlds, which restricts the unknown location of the true state, the simplest account of an epistemic state. Let us assume that the information items T_i are classical subsets. Then $\mathcal{S}(T_i) = T_i$, the relation $\sqsubseteq$ is set inclusion, and $\omega \succ_T \omega'$ if $\omega \in T$ and $\omega' \notin T$, while $\omega \sim_T \omega'$ if $\omega, \omega' \in T$ or $\omega, \omega' \notin T$.

If the inputs are globally consistent, i.e., if $\cap_{i=1,n} T_i \neq \emptyset$, one should have the inclusion $f(T_1,...,T_n) \subseteq \cap_{i=1}^n T_i$ by Prop. 4 (Optimism). By Possibility preservation (1a), $\cap_{i=1}^n T_i \subseteq f(T_1, ..., T_n)$. Thus, $f(T_1, ..., T_n) = \cap_{i=1}^n T_i$ in case of global consistency. Let us now consider the case of two inconsistent pieces of information T_1 and T_2 such that $T_1 \cap T_2 = \emptyset$. By Prop. 5 (Fairness), one should have $f(T_1, T_2) \cap T_1 \neq \emptyset$ and $f(T_1, T_2) \cap T_2 \neq \emptyset$. Moreover by Impossibility preservation (1b), one should have $f(T_1, T_2) \subseteq T_1 \cup T_2$. This leads to $f(T_1, T_2) = A_1 \cup A_2$ with $\emptyset \neq A_x \subseteq T_x$ for $x = 1, 2$. Minimal Commitment leads us to take $A_x = T_x$ for $x = 1, 2$.

This reasoning clearly extends to the case of more than two pairwise inconsistent information pieces: by Fairness, $f(T_1, ..., T_n)$ should be of the form $A_1 \cup ... \cup A_n$, $\emptyset \neq A_i \subseteq T_i$ for $i = 1, \ldots, n$. Let $I \subset \{1,..,n\}$ be a maximal consistent subset (MCS) of sources, i.e., $T^I = \cap_{i \in I} T_i \neq \emptyset$ and $T^I \cap T_j = \emptyset$ if $j \notin I$. Then the partial result should be $A_j = \cap_{i \in I} T_i, \forall j \in I$ by Minimal Commitment and Optimism. Given two MCSs I and I', $T^I \cap T^{I'} = \emptyset$. Hence at most one subset I of sources is correct. Optimism dictates that at least one subset I of sources is so. We thus get the general combination rule

$$f(T_1, ..., T_n) = \bigcup_{I \in MCS(\{1,...,n\})} \bigcap_{i \in I} T_i \tag{1}$$

where MCS($\{1, \ldots, n\}$) is the set of maximal consistent subsets of sources. It was first proposed by [15]. It satisfies all core properties.

This rule exhibits an apparent discontinuity when moving from a consistent situation to an inconsistent one, since shrinking two subsets that overlaps may lead from situations with more and more precise fusion results to a situation with an imprecise result. However, nothing forbids independent sources to provide information pieces having a narrow intersection. But such a precise result may sometimes become all the more debatable as its precision increases. Some approaches cope with inconsistency in fusion problems by a similarity-based enlargement of the supports and cores of information pieces [16].

4 Possibility Theory

The possibility theory framework is a graded extension of the previous setting. Subsets are replaced by possibility distributions π, which are mappings from Ω to $[0, 1]$ that rank-order interpretations ($\omega \succeq_T \omega'$ if $\pi(\omega) \geq \pi(\omega')$). The support is $\mathcal{S}(\pi) = \{\omega | \pi(\omega) > 0\}$ and the core is $\mathcal{C}(\pi) = \{\omega | \pi(\omega) = 1\}$. A strongly consistent possibility distribution is such that $\mathcal{C}(\pi) \neq \emptyset$. The consistency degree $Cns(\pi_i, \pi_j) = \max_\omega \min(\pi_i(\omega), \pi_j(\omega))$ between two distributions ranges from 1 when there is a

common ω that is fully possible, to 0 when the supports do not overlap. The information ordering is relative specificity ($\pi_i \sqsubseteq \pi_j \iff \pi_i \leq \pi_j$).

The most basic combination rules extend conjunction and disjunction, especially the Minimum rule $\min(\pi_1, ..., \pi_n)$ and the Maximum rule $\max(\pi_1, ..., \pi_n)$; other conjunctions can be t-norms t such as product instead of $\min$, which creates a reinforcement effect. The conjunctive rules do not obey the strong form of consistency enforcement.

The latter property justifies the renormalized conjunctive fusion rule (RCF) [8]

$$\hat{\bigwedge}(\pi_1, ..., \pi_n) = \frac{\bigwedge(\pi_1, ..., \pi_n)}{Cns(\pi_1, ..., \pi_n).} \tag{2}$$

It is undefined as soon as $Cns(\pi_1, ..., \pi_n) = 0$ (strong conflict). When $\bigwedge$ is product, this rule is well-known and is associative, but associativity is generally not preserved with other t-norms. This kind of fusion rule is used in logic-based merging using distances [11] instead of possibility distributions (see [1] for the connection between the two approaches). However this kind of rule cannot cope with strongly mutually inconsistent sources. We can extend the MCS rule in at least two ways:

$$MCS1(\pi_1, ..., \pi_n) = \max_{I \in MCS(\{\mathcal{C}(\pi_1), ..., \mathcal{C}(\pi_n)\})} \bigwedge_{i \in I} \pi_i \tag{3}$$

$$MCS0(\pi_1, ..., \pi_n) = \max_{I \in MCS(\{\mathcal{S}(\pi_1), ..., \mathcal{S}(\pi_n)\})} \hat{\bigwedge}_{i \in I} \pi_i \tag{4}$$

In fact, each of MCS1, MCS0 selects maximal consistent subsets in a specific way. Once this principle is chosen, the same reasoning holds as in the crisp case, and we obtain for the above three rules for merging possibility distributions:

Proposition 1. *The RCF rule (2) does not satisfy Consistency Enforcement nor Fairness (when undefined). The extended-MCS rules (3,4) satisfy all core properties.*

MCS1 is much demanding on mutual consistency of sources and yields plain disjunction if cores of π_i are disjoint. MCS0 is less demanding and more optimistic: it yields $\hat{\bigwedge}(\pi_1, ..., \pi_n)$ if all supports overlap.

Another fusion rule for possibility distribution that applies the classical MCS rule to all cuts of the input possibility distributions has been recently proposed [4]. It satisfies all basic postulates but it yields a belief function, as resulting cuts are no longer nested.

5 Evidence Theory

In evidence theory, a piece of information is modeled by a *basic belief assignment (bba)* m which is a mapping from 2^Ω to $[0, 1]$ such that $\sum_{A \subseteq \Omega} m(A) = 1$. A bba is consistent if $m(\emptyset) \neq 0$. A is called a focal element of m if $m(A) > 0$. Let $\mathcal{F}_m$ be the set of focal elements of m. Let $\mathcal{S}(m)$ denote the union of the focal elements: if $\mathcal{F}_m = \{A_1, .., A_n\}$, then $\mathcal{S}(m) = \bigcup_{i=1}^n A_i$ is the support of m. The *vacuous* bba m^Ω is such that $m(\Omega) = 1$.

From a bba m, two dual functions, bel and pl called *belief and plausibility functions* respectively, are defined as $bel(A) = \sum_{B \subseteq A} m(B)$, and $pl(A) = \sum_{B \cap A \neq \emptyset} m(B)$, while the commonality function q is defined by $q(A) = \sum_{A \subseteq B} m(B)$.

Evidence theory is rich enough to include as particular cases i) sets (when there is one focal element), ii) probabilities (when focal elements are singletons), and iii) possibility theory (when focal elements are nested). The *contour* function C_m of the bba m, which is the plausibility function of the singletons, $C_m(\omega) = \sum_{A \subseteq \Omega, \omega \in A} m(A)$, reduces to a possibility distribution $\pi_m = C_m$ when focal elements are nested, and then $pl(A) = \max_{\omega \in A} \pi_m(\omega)$ is a possibility measure. The contour function reduces to a probability distribution if the focal elements are singletons.

We now examine issues related to plausibility and information ordering, and inconsistency between bbas.

Plausibility Ordering. In evidence theory, from a representation point of view the contour function is a natural option for comparing possible worlds ($\omega_1 \succeq_m^{con} \omega_2$ iff $C_m(\omega_1) \geq C_m(\omega_2)$). In addition to this standard ordering, we define a more basic partial ordering relation on possible worlds induced by the bba.

Definition 1. *Let $\omega_1, \omega_2 \in \Omega$. Then ω_1 dominates ω_2 w.r.t. m, denoted by $w_1 \succeq_m^{dom} \omega_2$ iff for any $A \subseteq \Omega \setminus \{\omega_1, \omega_2\}$, $m(A \cup \{\omega_1\}) \geq m(A \cup \{\omega_2\})$.*

Proposition 2. *$\succeq_m^{dom}$ is a reflexive and transitive relation. Moreover $\omega_1 \succeq_m^{dom} \omega_1$ implies $C_m(\omega_1) \geq C_m(\omega_2)$.*

Inconsistency. The degree of inconsistency (or conflict) of two bbas m_1 and m_2 is measured by the mass received by the empty set as the result of the conjunction of m_1 and m_2 viewed as random sets: $m_{1\wedge 2}(\emptyset) = \sum_{A \cap B = \emptyset} m_1(A) m_2(B)$. It is the counterpart of $1 - Cns(\pi_1, \pi_2)$ using product instead of min. However, it has been pointed out in [12] that $m_{1\wedge 2}(\emptyset)$ is not a convincing measure of conflict, since two identical bba's usually have a non zero degree of conflict. To get a more satisfactory measure of conflict one may avoid using products $m_1(A)m_2(B)$ that presuppose source independence, and replace them by a joint mass $x(A, B)$ whose marginals are m_1 and m_2[6]. Then we can define a better inconsistency index, such that $Inc(m, m) = 0$:

$$Inc(m_1, m_2) = \inf_x \sum_{B \in \mathcal{F}_1, C \in \mathcal{F}_2 : B \cap C = \emptyset} x(B, C)$$

Note that $Inc(m_1, m_2) = 0$ whenever there exists a joint mass $x(A, B)$ whose marginals are m_1 and m_2 that assigns zero mass to all disjoint focal sets, which corresponds to saying that the two credal sets (families of probabilites) $\{P : P(A) \geq Bel_1(A), \forall A\}$ and $\{P : P(A) \geq Bel_2(A), \forall A\}$ have a non-empty intersection [2]. So we can call this index one of probabilistic consistency. Its calculation requires the use of linear programming. It is easy to see that $Inc(m_1, m_2) \leq m_{1\wedge 2}(\emptyset)$.

Alternatively we can adopt definitions that do not rely on numerical values of bba's: two mass functions m and m' with focal sets $\mathcal{F}$ and $\mathcal{F}'$ are said to be

- *Weakly mutually consistent* if $\exists E \in \mathcal{F}, E' \in \mathcal{F}' : E \cap E' \neq \emptyset$ (note that it implies that $m_{1\wedge 2}(\emptyset) < 1$, hence $Inc(m_1, m_2) < 1$ as well)
- *Strongly (or logically [3]) mutually consistent* if $\forall E \in \mathcal{F}, \forall E' \in \mathcal{F}' : E \cap E' \neq \emptyset$ (note that it does imply that $Inc(m_1, m_2) = m_{1\wedge 2}(\emptyset) = 0$).

Information Ordering. In the literature, different information orderings in evidence theory have been proposed for comparing the information contents of bba's (see e.g. [7]). We here only consider the one that can be expressed in terms of mass functions, and echoes the above inconsistency index. It is the strongest information ordering among those previously introduced in the literature.

Definition 2 (Specialization). *Let m_1 and m_2 be two bbas over Ω, m_1 is a specialization of m_2 (denoted by $m_1 \sqsubseteq_s m_2$) if and only if there exists a joint mass $x(A, B)$ whose marginals are m_1 and m_2 , such that $x(A, B) = 0$ whenever $A \not\subseteq B$, $A \in \mathcal{F}_1, B \in \mathcal{F}_2$.*

We are in a position to propose one possible instantiation of the basic fusion postulates, for two sources here, denoting by m_{12} the result:

1. **Unanimity** Possibility and impossibility preservation.
2. **Weak Information Monotonicity**: If m_1 and m_2 are strongly consistent, and moreover $m_1 \sqsubseteq_s m_1'$, $m_2 \sqsubseteq_s m_2'$ then $m_{12} \sqsubseteq_s m_{12}'$
3. **Consistency enforcement**:
   - $\sum_{E \subseteq S} m_{12}(E) = 1$ (strong version)
   - $\sum_{E \subseteq S} m_{12}(E) > 0$ (weak version)
4. **Optimism**
   - If m_1 and m_2 are strongly mutually consistent, then $m_{12} \sqsubseteq_s m_i, i = 1, 2$.
   - There exists a joint bba $x(\cdot, \cdot)$ whose marginals are m_1 and m_2, such that $m_{12} \sqsubseteq_s m_1 \oplus m_2$, with $m_1 \oplus m_2(E) = \sum_{F,G:E=F \cup G} x(F, G)$.
5. **Fairness**: Each m_i should be weakly consistent with m_{12}.
6. **Insensitivity to Vacuous Information**: If $m_1(\Omega) = 1$ then $m_{12} = m_2$
7. **Symmetry**: $m_{12} = m_{21}$
8. **Minimal Commitment**: m_{12} should be minimally specific for specialization.

5.1 Checking Some Existing Combination Rules

Several rules have been proposed in evidence theory for merging information, apart from the well-known Dempster's rule of combination. We first focus on the main rules.

$$m_{Dem}(C) = \frac{\sum_{A,B:A \cap B=C} m_1(A)m_2(B)}{1 - \sum_{A,B:A \cap B=\emptyset} m_1(A)m_2(B)} \text{ (Dempster's rule)} \tag{5}$$

$$m_{Sm}(C) = \sum_{A,B \subseteq \Omega, A \cap B=C} m_1(A)m_2(B) \text{ (Smets' rule)} \tag{6}$$

$$m_{Ya}(C) = \begin{cases} \sum_{A,B:A \cap B=C} m_1(A)m_2(B) \text{ if } C \neq \Omega \text{ (Yager's rule)} \\ m_1(\Omega)m_2(\Omega) + \sum_{A \cap B=\emptyset} m_1(A)m_2(B) \text{ if } C = \Omega \end{cases} \tag{7}$$

$$m_{DP}(C) = \sum_{A,B:A \cap B=C} m_1(A)m_2(B) + \sum_{A,B:A \cup B=C, A \cap B=\emptyset} m_1(A)m_2(B). \tag{8}$$

All four fusion rules presuppose independence between sources, as an additional assumption, which enforces the choice of $x(\cdot, \cdot) = m_1(\cdot) \cdot m_2(\cdot)$. It reduces the scope of

the Minimal Commitment axiom to the choice of a set-theoretic combination for focal sets. The main difference between Dempster's rule and the three other rules respectively proposed in [19] (see also [18]) [24], [8] concern the way the mass $(m_1 \otimes m_2)(\emptyset) = \sum_{A,B:A\cap B=\emptyset} m_1(A)m_2(B)$ is re-allocated. In Dempster's rule, the renormalization by division enforces strong consistency of the result, when the two bba's are weakly mutually consistent (otherwise the operation is not defined). Smets's rule simply keeps this mass on $\emptyset$, whilst Yager's rule assigns it to Ω.

All four fusion rules coincide with each other if the two bba's are strongly consistent. Then all postulates are satisfied. When $\sum_{A\cap B=\emptyset} m_1(A)m_2(B) = 1$, m_{Dem} is not defined due to a total conflict between the sources, which violates the Consistency Enforcement postulate, like for the normalized conjunctive rule of possibility theory. When the two bba's are weakly mutually consistent, the result is consistent since $m_{Dem}(\emptyset) = 0$. Dempster's rule of combination is over-optimistic in case of weak consistency; it may fail to satisfy the second Optimism condition, due to renormalization (it would satisfy it if we replace it by the weaker condition $\mathcal{S}(m_{12}) \subseteq \mathcal{S}(m_1) \cup \mathcal{S}(m_2)$).

In Smets rule, the mass assigned to the empty set $m_S(\emptyset)$ may be different from 0. Smets rule does not respect the consistency enforcement principle, even if it is always defined, since it may deliver the plain empty set in case m_1 and m_2 are strongly inconsistent. Like Dempster rule of combination, Smets' rule is purely conjunctive, hence does not behave in agreement with the postulates in case of partial mutual inconsistency. The Fairness axiom formally fails with this fusion rule like for Dempster's, because it is not compatible with the failure of the consistency enforcement postulate.

Yager's rule is similar to Smets' rule except that $(m_1 \otimes m_2)(\emptyset)$ is added to $m_{Ya}(\Omega)$ instead of leaving it in $m_{Ya}(\emptyset)$, just changing conflict into ignorance (a form of renormalization). It does not respect Unanimity, nor Optimism and in particular impossibility preservation is clearly violated. In fact, this rule is far too cautious in the presence of conflicts.

Three of the four above rules are conjunctive, while the last one, proposed in [8] extends the basic fusion rule (1) for sets to belief functions (hence it is a special case of the MSC rule). It is a hybrid rule, like Yager's, that contains both conjunctive and disjunctive elements. It is more informative than Yager's. This fusion rule satisfies all fusion postulates like the MCS fusion rule for two sets, which it generalizes.

Dempster rule and Smets rule are associative, while the others are not. However, Dubois and Prade combination rule can be readily extended to $n > 2$ sources using the MCS rule on all n-tuples of focal sets.

We may complement Unanimity with Local Ordinal Unanimity with respect to dominance ordering: for two possible worlds ω and ω', $\omega \succeq_1^{dom} \omega'$ and $\omega \succeq_2^{dom} \omega'$ then $\omega \succeq_{12}^{dom} \omega'$. Indeed, we can prove the following:

Proposition 3. *Smets, Yager and Dempster combination rules obey Local Ordinal Unanimity with respect to the dominance ordering.*

It is still unclear whether this result holds for the 4th fusion rule. The above results are summarized by the following Table 5.1 (all above rules are symmetric).

rule/Prop	Una	Mono	Cons	Opti	Fair	Vacuous	Min-Com
Dempster	Yes[1]	Yes	Strong[1]	No[3]	Yes[1]	Yes	No[3]
Smets	Yes[2]	Yes	No	Yes	No	Yes	Yes[1]
Yager	No	Yes	Strong	No	Yes	Yes	Yes
DP	Yes	Yes	Strong	Yes	Yes	Yes	Yes

- 1. Only when there is no strong global inconsistency
- 2. Trivially in case of strong global inconsistency
- 3. Overoptimistic in case of weak inconsistency

All the fusion rules considered above assume source independence but can be extended by replacing the product of bba's $m_1(F)m_2(G)$ by a suitably chosen joint mass function $x(E,F)$ whose marginals are m_1 and m_2 [3]. The main difference is that we can replace strong consistency by probabilistic consistency, that is all four fusion rules would coincide with $m_{12}(E) = \sum_{E=F\cap G} x(E,F)$ if m_1 and m_2 are mutually consistent in the sense that $Inc(m_1,m_2) = 0$. However there may be several minimally specific fusion rules, some of which are idempotent [3], if we leave the choice of $x(E,F)$ open.

6 Concluding Remarks

In this paper, we have provided a general framework for analyzing fusion operators proposed in different settings, in a unified way. Due to space limitation, we have concentrated the presentation on three types of representation using classical sets, possibility theory, and evidence theory respectively, considering only a representative sampling of operators. It is clear that the analysis may be applied more systematically, as well as to other settings, whether numerical (such as imprecise probabilities [22,21]), ordinal[13] or yet logical [11]. The latter case comes down to viewing the set of models of a knowledge base K as the core $\mathcal{C}(T_K)$ of the corresponding information item T_K. We did not discuss the case of single probability distributions as they only support weighted arithmetic means [20], which violates Insensitivity to Vacuous Information (assuming the latter is expressed by uniform probability distributions). When distinct, they always conflict, but taking their convex hull satisfies all postulates [21]. Beyond our core properties, that are usually completely intuitive, and should be satisfied by any reasonable fusion rule, other less universal properties, may make sense in specific contexts. For instance a discontinuous fusion rule in a continuous setting is questionable (e.g. Dempster's rule is oversensitive to small changes of input values). Some properties are useful in some situations but not possessed by many rules (e.g., idempotency when sources are redundant). Moreover, when the representation setting becomes richer, more options are available for expressing the properties with various strengths. Adapting the basic postulates to prioritized merging is another line for further work.

References

1. Benferhat, S., Dubois, D., Kaci, S., Prade, H.: Possibilistic merging and distance-based fusion of propositional information. Annals of Mathematics and Artificial Intelligence 34(1-3), 217–252 (2002)

2. Chateauneuf, A.: Combination of compatible belief functions and relation of specificity. In: Advances in the Dempster-Shafer Theory of Evidence, pp. 97–114. John Wiley & Sons, Inc., New York (1994)
3. Destercke, S., Dubois, D.: Idempotent conjunctive combination of belief functions: Extending the minimum rule of possibility theory. Information Sciences 181(18), 3925–3945 (2011)
4. Destercke, S., Dubois, D., Chojnacki, E.: Possibilistic information fusion using maximal coherent subsets. IEEE Tr. Fuzzy Systems 17, 79–92 (2009)
5. Dubois, D.: The role of epistemic uncertainty in risk analysis. In: Deshpande, A., Hunter, A. (eds.) SUM 2010. LNCS, vol. 6379, pp. 11–15. Springer, Heidelberg (2010)
6. Dubois, D., Prade, H.: On the unicity of Dempster rule of combination. Int. J. of Intelligent Systems 1, 133–142 (1986)
7. Dubois, D., Prade, H.: A set-theoretic view of belief functions: logical operations and approximations by fuzzy sets. Int. J. General Systems 12 (1986)
8. Dubois, D., Prade, H.: Representation and combination of uncertainty with belief functions and possibility measures. Computational Intelligence 4, 244–264 (1988)
9. Dubois, D., Prade, H.: Possibility theory and data fusion in poorly informed environments. Control Engineering Practice 2, 811–823 (1994)
10. Dubois, D., Prade, H.: Possibility theory: qualitative and quantitative aspects. In: Handbook of Defeasible Reasoning and Uncertainty Management Systems, vol. 1, pp. 169–226. Kluwer (1998)
11. Konieczny, S., Pino Pérez, R.: Logic based merging. J. Philosophical Logic 40(2), 239–270 (2011)
12. Liu, W.: Analyzing the degree of conflict among belief functions. Artificial Intelligence 170, 909–924 (2006)
13. Maynard-Reid, P., Lehmann, D.: Representing and aggregating conflicting beliefs. In: Proc. of Int. Conf on Principles of Knowledge Representation and Reasoning (KR 2000), pp. 153–164 (2000)
14. Oussalah, M.: Study of some algebraic properties of adaptative combination rules. Fuzzy Sets and Systems 114, 391–409 (2000)
15. Rescher, N., Manor, R.: On inference from inconsistent premises. Theory and Decision 1, 179–219 (1970)
16. Schockaert, S., Prade, H.: An inconsistency-tolerant approach to information merging based on proposition relaxation. In: Proc. 24th AAAI Conf. on Artificial Intelligence, pp. 363–368 (2010)
17. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press (1976)
18. Smets, P.: Data fusion in the transferable belief model. In: Proc. of Intern. Conf. on Information Fusion, Paris (2000)
19. Smets, P., Kennes, R.: The transferable belief model. Artificial Intelligence 66(2), 191–234 (1994)
20. Wagner, C., Lehrer, K.: Rational Consensus in Science and Society. D. Reidel (1981)
21. Walley, P.: The elicitation and aggregation of beliefs. Technical Report 23, University of Warwick, UK (1982)
22. Walley, P.: Statistical Reasoning with Imprecise Probabilities. Chapman and Hall (1991)
23. Walley, P.: Measures of uncertainty in expert systems. Artificial Intelligence 83, 1–58 (1996)
24. Yager, R.: On the Dempster-Shafer framework and new combination rules. Information Sciences 41, 93–138 (1987)

Facility Location and Social Choice via Microaggregation

Josep Domingo-Ferrer

Universitat Rovira i Virgili
Dept. of Computer Engineering and Mathematics
UNESCO Chair in Data Privacy
Av. Països Catalans 26
E-43007 Tarragona, Catalonia
josep.domingo@urv.cat

Abstract. Microaggregation is a cardinality-constrained clustering problem that arose in the context of data privacy. In microaggregation, the number of clusters is not fixed beforehand, but each cluster must have at least k elements. We illustrate in this paper that microaggregation can be applied for decision making in areas other than privacy. Specifically, we focus on the service facility location problem and on game theory (coalition formation and social choice).

Keywords: Microaggregation, Decision making, Service facility location problem, Cooperative games, Coalitions, Social choice.

1 Introduction

Microaggregation [2,4] is a clustering problem that originally arose in data anonymization for privacy protection. Rather than fixing beforehand the number of desired clusters, in microaggregation one fixes the minimum size of clusters. Clusters are formed using a criterion of maximum within-cluster similarity and each cluster should contain at least k elements. In the anonymization application, elements are records corresponding to individual respondents and the records in a cluster are replaced by the centroid cluster before publication; this ensures that an individual's published record is indistinguishable from the records of at least another $k-1$ individuals (k-anonymity).

The optimal solution to the microaggregation problem is defined to be the one that maximizes the sum of within-cluster similarities, while respecting the constraint that all clusters must contain at least k elements. Finding an optimal solution for the microaggregation problem can be done in polynomial time only if the data elements are one-dimensional [7]. In the general multi-dimensional case, the problem has been shown to be NP-hard [14].

Several heuristics have been published in the literature that provide good solutions to the microaggregation problem. Most of them deal with numerical elements (*e.g.* see survey in [8]), but extensions for ordinal and nominal data

V. Torra et al. (Eds.): MDAI 2013, LNAI 8234, pp. 49–57, 2013.

have also been proposed [4,5,6]. Some of the proposed heuristics are approximations [3,11] in the sense that they can be proven to yield solutions within a specific bound of the optimal solution.

The motivation of this paper is to explore applications of microaggregation other than anonymization; in particular, we describe several applications to decision making.

1.1 Contribution and Plan of This Paper

As pointed out above, the microaggregation problem is a well-studied NP-hard problem and several efficient heuristics aimed at solving it have been proposed.

Although microaggregation arose in the field of data anonymization, it is a general problem that can also arise in many other application areas. In this paper, we explore the use of microaggregation in facility location and game theory. Specifically, Section 2 describes how the location of service facilities can be viewed as a microaggregation problem. Section 3 illustrates two uses of microaggregation in game theory: detecting natural coalitions in cooperative games and reducing the number of strategies to facilitate rational social choice. Conclusions and avenues for future research are outlined in Section 4.

2 Microaggregation and Service Facility Location

A well-known problem in operations research is the *simple plant location problem* (SPLP), also known under a variety of alternative names (warehouse location problem [9], uncapacitated facility location problem [10], etc.). The most popular statement of this problem is as follows:

- Let $I = \{1, \cdots, m\}$ be a set of candidate locations for industrial plants producing some product. A plant can be opened in any location $i \in I$ at a cost f_i. Each opened plant can provide an unlimited amount of product.
- Let $J = \{1, \cdots, n\}$ be a set of customers such that customer j needs an amount b_j the product.
- Let c_{ij} be the unit transportation cost from plant i to customer j.
- The problem is to decide at which locations should plants be opened and the quantity x_{ij} of product to be supplied by plant i to customer j.

Mathematically, the SPLP can be formulated as the following constrained minimization problem:

$$\min \left(\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{i=1}^{m} f_i y_i \right)$$

subject to

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad \forall j \in J$$

$$0 \leq x_{ij}/b_j \leq y_i \text{ and } y_i \in \{0,1\}, \quad \forall i \in I \text{ and } \forall j \in J$$

where y_i indicates whether a plant is opened at location i ($y_i = 1$) or not ($y_i = 0$).

Imagine now that, instead of industrial plants that produce some products, we want to find locations for service facilities that must give service to users. Examples of such service facilities could be hospitals, schools, sports facilities, etc. Let us call this problem the *service facility location problem* (SFLP). Let us point out some fundamental differences between the SPLP and the SFLP, which make the heuristics designed for SPLP unsuitable for SFLP:

- Whereas the SPLP assumes that what is transported is the product, what is transported in SFLP are the users who must reach their service facility to use it. Hence, even if transportation costs were extremely low in terms of money, users do not wish to travel long distances to reach their service facility.
- For the above reason, each user wishes their service facility to be as close as possible to them. However, for cost reasons, opening a service facility at each single user's location is not affordable. Rather, for the investment to be justified, each service facility must be shared by at least k users.

From the above observations, it follows that *the SFLP is in fact a microaggregation problem*, because:

1. Clusters of at least k user locations must be formed in order to locate a service facility at the centroid of each cluster.
2. The Euclidean distance from each user location to the location of the corresponding service facility must be as small as possible. This amounts to maximizing the sum of within-cluster similarities as pointed above, for the special case of within-cluster similarity being the inverse of the sum of the Euclidean distances from the user locations in a cluster to the cluster centroid where the facility is located.

Being equivalent to a multivariate microaggregation problem (locations are bivariate), the SFLP is an NP-hard problem [14]. Any microaggregation heuristic for multivariate numerical data (*e.g.* [2,4,3,11]) can be used to find a reasonably good solution to the SFLP. In fact, the approximation heuristic in [3] offers a solution that can be proven to be within a factor of $(2k-1)\max(2k-1, 3k-5)$ of the optimal one, where optimality means minimum sum of within-cluster Euclidean distances from locations to centroids. Even better, the more recent approximation heuristic in [11] offers a solution within a factor of $8(k-1)$ of the optimal one. The actual solutions returned by all the above-mentioned heuristics are usually very close to the optimal solution; in particular they are much closer than guaranteed by the theoretical approximation bounds (which are worst-case).

Since most microaggregation heuristics run in time quadratic in the number n of elements (user locations in our case), it may be necessary to use blocking for large n. In the SFLP blocking means that, rather than solving the problem

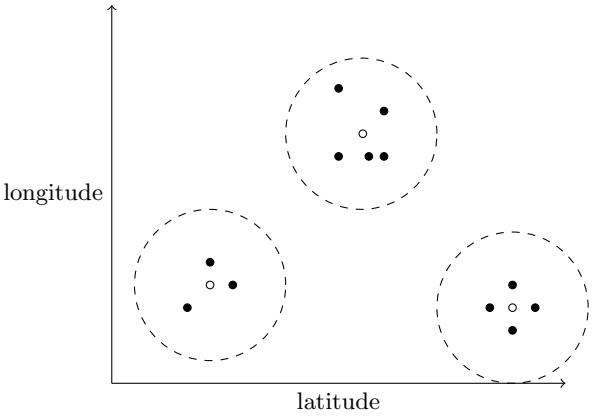


Fig. 1. Variable-size microaggregation with $k = 3$ to obtain a solution of the service facility location problem with $n = 12$ users. Black dots indicate user locations and white dots indicate proposed service facility locations.

for all users/citizens in a large geographic area (*e.g.* a country), one would independently solve instances of the problem in manageable subdivisions of the large area (*e.g.* in each state, province, county, etc.).

Figure 1 depicts an example with $n = 12$ users, a minimum of $k = 3$ users needed to justify a new service facility and service facility locations obtained with variable-size microaggregation.

3 Microaggregation and Game Theory

In game theory, a *cooperative game* is a game in which groups of players, called *coalitions*, may enforce cooperative behavior. Hence, the game can be viewed as a competition between coalitions of players, rather than between individual players.

We first give some background on game theory (Section 3.1). Then we describe how microaggregation can be used to detect natural coalitions in cooperative games (Section 3.2), and to reduce the number of strategies to facilitate rational social choice (Section 3.3).

3.1 Background on Game Theory

A game is a protocol between a set of n *players*, $\{P_1, \cdots, P_n\}$. Each player P_i has her own *set of possible strategies*, say S_i. To play the game, each player i selects a strategy $s_i \in S_i$. We will use $\mathbf{s} = (s_1, \cdots, s_n)$ to denote the vector of strategies selected by the players and $\mathbf{S} = \Pi_{i=1}^{n} S_i$ to denote the set of all possible ways in which players can pick strategies.

The vector of strategies $\mathbf{s} \in \mathbf{S}$ selected by the players determines the outcome for each player, which can be a payoff or a cost. In general, the outcome will

be different for different players. To specify the game, we need to state for each player a preference ordering on these outcomes by giving a complete, transitive, reflexive binary relation on the set of all strategy vectors $\mathbf{S}$. The simplest way to assign preferences is by assigning, for each player, a value for each outcome representing the payoff of the outcome (a negative payoff can be used to represent a cost). A function whereby player P_i assigns a payoff to each outcome is called a utility function and is denoted by $u_i : \mathbf{S} \longrightarrow \mathbb{R}$.

For a strategy vector $\mathbf{s} \in \mathbf{S}$, we use s_i to denote the strategy played by P_i and s_{-i} to denote the $(n-1)$-dimensional vector of the strategies played by all other players. With this notation, the utility $u_i(\mathbf{s})$ can also be expressed as $u_i(s_i, s_{-i})$.

A strategy vector $\mathbf{s} \in \mathbf{S}$ is a *dominant strategy solution* if, for each player P_i and each alternate strategy vector $\mathbf{s}' \in \mathbf{S}$, it holds that

$$u_i(s_i, s'_{-i}) \geq u_i(s'_i, s'_{-i}) \tag{1}$$

In plain words, a dominant strategy $\mathbf{s}$ is the best strategy for each P_i, independently of the strategies played by all other players.

A strategy vector $\mathbf{s} \in \mathbf{S}$ is said to be a *Nash equilibrium* if, for any player P_i and each alternate strategy $s'_i \in S_i$, it holds that

$$u_i(s_i, s_{-i}) \geq u_i(s'_i, s_{-i})$$

In plain words, no player P_i can change her chosen strategy from s_i to s'_i and thereby improve her payoff, assuming that all other players stick to the strategies they have chosen in $\mathbf{s}$. A Nash equilibrium is self-enforcing in the sense that once the players are playing such a solution, it is in every player's best interest to stick to her strategy. Clearly, a dominant strategy solution is a Nash equilibrium. Moreover, if the solution is strictly dominant (*i.e.* when the inequality in Expression (1) is strict), it is also the unique Nash equilibrium. See [13] for further background on game theory.

3.2 Detecting Natural Coalitions in Cooperative Games

We assume in this section that the set of strategy vectors is finite, *i.e.*

$$\mathbf{S} = \{\mathbf{s}^1, \cdots, \mathbf{s}^m\}$$

We can represent each player P_i as an m-dimensional vector $u_i(\mathbf{S})$ whose components specify the normalized payoffs the player obtains under each strategy, according to his/her utility function $u_i(\cdot)$:

$$u_i(\mathbf{S}) = \left(\frac{u_i(\mathbf{s}^1)}{\max_{l=1}^{m} u_i(\mathbf{s}^l)}, \cdots, \frac{u_i(\mathbf{s}^m)}{\max_{l=1}^{m} u_i(\mathbf{s}^l)} \right)$$

We now can cluster vectors $u_i(\mathbf{S})$ to obtain clusters of players with "similar" interests, in the sense that they derive similar payoffs from the various strategies. Two important remarks are in order here:

- Clusters indicate "natural" coalitions, in the sense that players with similar interests may tend to rally: they wish to enforce the same strategy vectors and avoid the same strategy vectors[1].
- The "centroid player" of each cluster could be taken as the prototypical player representing the coalition of players in the cluster. In this way, a cooperative game involving the natural coalitions can be approximately transformed into a non-cooperative game between prototypical players. Finding solutions in non-cooperative games (that is, dominant strategy vectors or Nash equilibria mentioned above) is normally easier.

If a single cluster containing all players is created, the homogeneity of that cluster is likely to be low and the prototypical player of that cluster is unlikely to accurately represent the interests of all players. On the other hand, if some clusters are much smaller than others, the coalitions corresponding to the smaller clusters will have much less power to enforce dominant strategies or equilibria than the coalitions corresponding to the larger clusters. Hence, there is a tradeoff between the coalition power and the representativeness of the prototypical player. Resorting to microaggregation to form clusters ensures bounds on coalition sizes:

- If a fixed-size microaggregation heuristic is chosen (*e.g.* MDAV, [4]) then the sizes of all coalitions are k except for one coalition having size at most $2k-1$. Choosing this kind of heuristics establishes equal power (size) for all coalitions as a primary goal and prototype representativeness as a secondary goal.
- If a variable-size microaggregation heuristic is used (*e.g.* [3,11]), then the sizes of all coalitions lie between k and $2k-1$, where the precise sizes are automatically selected by the heuristic in order to maximize within-cluster homogeneities. Choosing this kind of heuristic establishes prototype representativeness as a primary goal and equal power as a secondary goal.

Figure 2 depicts an example with two strategies, 12 players and three natural coalitions that can be formed when using variable-size microaggregation with $k = 3$.

3.3 Facilitating Social Choice

Social choice is a theoretical framework that studies how to combine individual preferences, interests or welfares to reach a collective decision or social welfare in some sense [1]. The Nakamura number [12] measures the degree of rationality of collective decision rules, such as voting rules. If the number of alternatives (candidates, options, etc.) to choose from is less than the Nakamura number,

[1] We discard here coalitions including players with similar utilities for the highest-paying strategy vectors *only*. The reason is that these are weaker coalitions, because they will break up if players not in the coalition manage to enforce a strategy vector that is not among the highest-paying ones for the coalition.

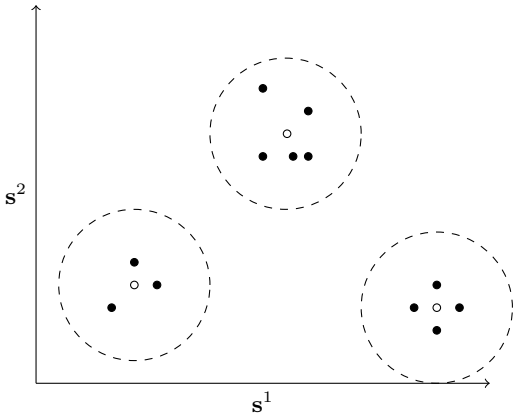


Fig. 2. Toy example with 12 players being clustered into 3 natural coalitions, for $m = 2$ strategy vectors $\mathbf{s}^1$ and $\mathbf{s}^2$, and variable-size microaggregation with $k = 3$. Black dots indicate actual players; white dots indicate the prototype player for each coalition; all coordinates are assumed to be normalized.

then the voting rule will identify "best" alternatives without any problem. In contrast, if the number of alternatives is greater than or equal to the Nakamura number, the voting rule will fail to identify "best" alternatives for some pattern of voting (*i.e.* for some tuple of voters' preferences), because a voting paradox will arise: a cycle of preferences will appear, like alternative a being socially preferred to alternative b, b to c and c to a.

The above discussion motivates the *relevance of being able to reduce the number of alternatives* in such a way that the new alternatives are as representative as possible of the old alternatives. We propose to use microaggregation to implement such a reduction.

Let us assimilate voters to players and alternatives to strategy vectors. We can represent each strategy vector $\mathbf{s}^j$ as an n-dimensional vector $\mathbf{u}(\mathbf{s}^j)$ whose components specify the normalized payoffs $\mathbf{s}^j$ brings to each player, according to the players' utility functions $u_1(\cdot)$ to $u_n(\cdot)$:

$$\mathbf{u}(\mathbf{s}^j) = \left(\frac{u_1(\mathbf{s}^j)}{\max_{l=1}^{m} u_i(\mathbf{s}^l)}, \cdots, \frac{u_n(\mathbf{s}^j)}{\max_{l=1}^{m} u_i(\mathbf{s}^l)} \right)$$

We now can cluster vectors $\mathbf{u}(\mathbf{s}^j)$ to obtain clusters of "similar" strategy vectors, in the sense that they provide similar payoffs to players/voters. The "centroid strategy" of each cluster can be taken as the prototypical strategy that will be used to replace the strategies in the cluster, thereby reducing the total number of strategies/alternatives.

If a single cluster containing all strategies is created, the homogeneity of that cluster is likely to be low and the prototypical strategy of that cluster is unlikely to accurately represent the interests of all players/voters (for example, think of

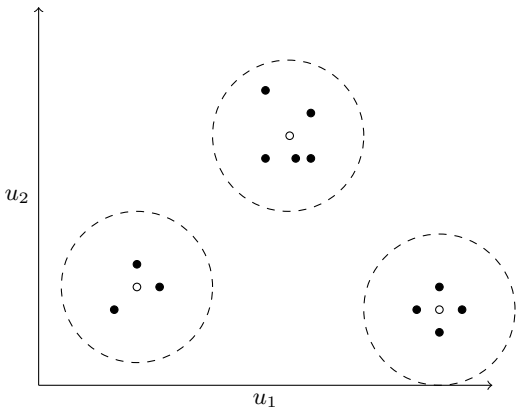


Fig. 3. Toy example with $n = 2$ voters/players with utility functions u_1 and u_2 in which 12 original alternatives/strategy vectors are reduced to 3 prototype alternatives/strategy vectors, using variable-size microaggregation with $k = 3$. Black dots indicate original strategies; white dots indicate prototype strategies; all coordinates are assumed to be normalized.

a country with a single party). On the other hand, if one goes for a less dramatical reduction of alternatives, it would seem fair to reduce the granularity of all original alternatives to a similar level, perhaps with more similar alternatives being included in larger clusters. This is exactly what variable-size microaggregation heuristics (*e.g.* [3,11]) offer: the sizes of all clusters lie between k and $2k - 1$, where the precise sizes are automatically selected by the heuristic in order to maximize within-cluster homogeneities. Choosing this kind of heuristic seeks to obtain prototype strategies with similar representativeness of the original strategies. The smallest possible value of k ought to be taken that brings the final number of alternatives/strategy vectors below the desired threshold (for example, the Nakamura number of the game).

Figure 3 shows a toy example with $n = 2$ voters/players where 12 original alternatives/strategy vectors are reduced to 3 prototype alternatives/strategies, using variable-size microaggregation with $k = 3$.

4 Conclusions

Although microaggregation was a problem that arose and was studied in the context of data anonymization, we claim that it is relevant in other application domains. In this paper, we have sketched its application to decision making. Specifically, microaggregation heuristics have been shown to offer solutions to the service facility location problem. Also, microaggregation can be helpful in game theory. Indeed, in cooperative games it helps detecting natural coalitions (players with similar interests). In social choice it can be used to reduce the

number of alternatives with minimum loss of information, in order to facilitate rational voting.

Acknowledgments and Disclaimer. This work was partly supported by the Government of Catalonia under grant 2009 SGR 1135, by the Spanish Government through projects TIN2011-27076-C03-01 "CO-PRIVACY" and CONSOLIDER INGENIO 2010 CSD2007-00004 "ARES", and by the European Comission under FP7 projects "DwB" and "Inter-Trust". The author is partially supported as an ICREA Acadèmia researcher by the Government of Catalonia. The author is with the UNESCO Chair in Data Privacy, but he is solely responsible for the views expressed in this paper, which do not necessarily reflect the position of UNESCO nor commit that organization.

References

1. Arrow, K.J.: Social Choice and Individual Values. Yale University Press, New Haven CT (1951)
2. Domingo-Ferrer, J., Mateo-Sanz, J.M.: Practical data-oriented microaggregation for statistical disclosure control. IEEE Transactions on Knowledge and Data Engineering 14(1), 189–201 (2002)
3. Domingo-Ferrer, J., Sebé, F., Solanas, A.: A polynomial-time approximation to optimal multivariate microaggregation. Computers & Mathematics with Applications 55(4), 714–732 (2008)
4. Domingo-Ferrer, J., Torra, V.: Ordinal, continuous and heterogeneous k-anonymity through microaggregation. Data Mining and Knowledge Discovery 11(2), 195–212 (2005)
5. Domingo-Ferrer, J.: Marginality: A numerical mapping for enhanced exploitation of taxonomic attributes. In: Torra, V., Narukawa, Y., López, B., Villaret, M. (eds.) MDAI 2012. LNCS, vol. 7647, pp. 367–381. Springer, Heidelberg (2012)
6. Domingo-Ferrer, J., Sánchez, D., Rufian-Torrell, G.: Anonymization of nominal data based on semantic marginality. Information Sciences 242, 35–48 (2013)
7. Hansen, S.L., Mukherjee, S.: A polynomial algorithm for optimal univariate microaggregation. IEEE Transactions on Knowledge and Data Engineering 15(2), 1043–1044 (2003)
8. Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Schulte Nordholt, E., Spicer, K., De Wolf, P.P.: Statistical Disclosure Control. Wiley, New York (2012)
9. Khumawala, B.M.: An efficient branch and bound algorithm for the warehouse location problem. Management Science 18(12), B-718–B-731 (1972)
10. Krarup, J., Pruzan, P.M.: The simple plant location problem: survey and synthesis. European Journal of Operational Research 12(1), 36–81 (1983)
11. Laszlo, M., Mukherjee, S.: Approximation bounds for minimum information loss microaggregation. IEEE Transactions on Knowledge and Data Engineering 21(11), 1643–1647 (2009)
12. Nakamura, K.: The vetoers in a simple game with ordinal preferences. International Journal of Game Theory 8(1), 55–61 (1979)
13. Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.V. (eds.): Algorithmic Game Theory. Cambridge University Press, New York (2007)
14. Oganian, A., Domingo-Ferrer, J.: On the complexity of optimal microaggregation for statistical disclosure control. Statistical Journal of the United Nations Economic Commission for Europe 18(4), 345–354 (2001)

Ordering Pareto Sets with Fuzzy Inference Systems

Sandra Sandri, José Carlos Becceneri, and Roberto Luiz Galski

Instituto Nacional de Pesquisas Espaciais
12201-970, São José dos Campos, SP
sandra.sandri@inpe.br, jcbecce@hotmail.com, galski@ccs.inpe.br

Abstract. Even though elements in the Pareto set of a given multi-objective problem represent optimal solutions, additional information may be available about a preference on these solutions. We propose to use the Pareto frontier obtained for a problem to discard dominated solutions and a fuzzy system to order the non-dominated ones.

1 Introduction

In single-objective optimization problems, we are interested in optimizing a single function by either maximization or minimization. In multi-objective optimization, we have a set of different objectives that need to be optimized simultaneously. When some of them are conflicting, usually there exists no single optimal solution, but rather a family of trade-offs (non-dominated solutions), which need not be the global optimum solution of any of the objectives when taken separately [7]. In this case, the quality of a solution to the problem is no longer assessed by a single value but by a set of values, each of which corresponding to a certain goal.

The complete set of non-dominated solutions is called a Pareto set; the set of values of the multi-objective function associated to those solutions is called the Pareto frontier. In real-world problems, it is very often impracticable to determine the complete Pareto set, either because the search space is infinite, or because obtaining a solution is a costly process. Moreover, more often than not, what is obtained is only an estimate of the Pareto set, formed by the collection of non-dominated solutions obtained in the process.

After the Pareto set (or its estimate) is found for a problem, usually a single solution has to be chosen by the decision-maker. To select this solution, a subset can be extracted to be examined in light of other criteria than the objective functions themselves. Sometimes it is possible to create this subset by making a visual inspection of the Pareto frontier (or its estimate) and select the most interesting ones. However, in problems with a large number of objectives, the visualization of the Pareto frontier is often hard or even impossible.

Automatically choosing a single element from the Pareto set obtained from a given problem has been addressed by several papers in the literature. A straightforward method consists in the choosing the solution whose evaluation is closer to the baricenter of the points in the frontier. Another consists in taking the solution whose evaluation is the closest to the origin of the Cartesian coordinates system for the problem, considering that the objective functions are all positive. A more complex method consists in adopting as the final solution for a problem, the one that results on the smallest loss

V. Torra et al. (Eds.): MDAI 2013, LNAI 8234, pp. 58–69, 2013.

for each of the objectives individually [9]. As stated by [4], these approaches produce a ranking on the basis of an arbitrary criterion of merit, obtained by combining the multiple decision criteria into one scalar index.

Another way to order solutions in a Pareto set is to consider user preferences on the search space. Fuzzy Sets Theory and Possibility Theory [5] are the basis of systems that use imperfect information furnished by an expert, in the form of rules and preferences, to solve problems in all kinds of fields. The combination between multi-objective optimization and fuzzy systems has been addressed by several authors, but few in what regards ranking the Pareto set (see [3] a survey). Many of these works use the fuzzy sets framework to help a method, such as genetic algorithms, ant colonies optimization, etc, to obtain the Pareto set itself (see [2] for a survey).

Here we are interested in using fuzzy sets to order a set of non-dominated solutions, according to the following general strategy, inspired from one proposed in [8] for robot control (see also [1]):

1. Find feasible solutions, according to the problem criteria and available data.
2. Find a sufficiently large set of non-dominated solutions.
3. Order these solutions incorporating subjective knowledge.
4. Find the most preferred solution, using other criteria.

We propose to use a fuzzy system to assign user satisfaction degrees to the compound objective values. Then we take the optimal solutions found for the problem and order them according to the satisfaction degree that were obtained by the objective function values calculated for each of these solutions. In problems for which the selection of the final solution is taken by a group of decision-makers, a fuzzy system can be created to model the preferences of each of these decision-makers individually, and then combine the results to obtain a single final ordering.

This paper is organized as follows. In Section 2 we formally address multi-objective problems, defining solution dominance, Pareto sets and Pareto frontiers, and in Section 3 we give a brief introduction to fuzzy sets theory and fuzzy systems. Then in Section 4, we propose a general framework to optimization based on the one proposed in [8] and show how fuzzy systems can be used to implement this general approach. Section 5 brings a brief discussion about how the proposed approach could be useful in a real-world application and Section 6 finally brings the conclusion. The concepts are illustrated using a simple example presented in Section 2.

2 Multi-objective Problems

In single-objective optimization problems, we are interested in optimizing a function $f : S \rightarrow \Omega$, by either maximization or minimization. We call f a *cost function* or an *objective function* and S and Ω are respectively called the *search* and the *objective* spaces. If f is the objective function of a minimization problem, $s_0 \in S$ is an *optimal solution* for that problem when

$$\forall s \in S, f(s) \geq f(s_0)$$

It is worthy noting that for some problems, there exist several optimal solutions in set S, all of them attaining the same optimal value for f.

If a problem has I goals to achieve, the objective function is not modelled by a single mathematical function but rather by a set of objective functions $\{f_1, f_2, ..., f_I\}$, where $\forall i, f_i : S \rightarrow \Omega_i$. This (compound) objective function is then a mapping $F : S \rightarrow \Omega$, where $\Omega = \Omega_1 \times ... \times \Omega_I$.

A minimization problem can be generally stated as:

$$\begin{aligned} &\text{Minimize } f_i(s);\ i = 1 \text{ to } I, \\ &\text{Subject to:} \\ &\quad g_k(s) \leq 0;\ k = 1 \text{ to } K \text{ (Inequality constraints)} \\ &\quad h_l(s) = 0;\ l = 1 \text{ to } L \text{ (Equality constraints)} \\ &\quad s_{min} \leq s \leq s_{max} \text{ (Boundary conditions)} \end{aligned}$$

Below, we first present an example that will be used in the remaining of the text. Then we define some important concepts, such as solution dominance, Pareto set and Pareto frontier.

2.1 Running Example

In the following, we present a very simple example to better illustrate what would be conflicting objectives in a multi-objective problem. Let us suppose we want to fly from one city to the other, possibly with stops, at a given date. We only consider trips that cost at most R$ 1000 and last at most 20 hours. Formally, we have the following functions, variables and sets:

- $s \in S$
- $f_{cost} : S \rightarrow [1, 1000]$
- $f_{duration} : S \rightarrow [1, 20]$

Variable s identifies a trajectory (a path), consisting of sequence of legs, each of which identified with data such as the departure and arrival cities, the airport names, the time schedule, the air company used in that leg, etc. All possible trajectories between two cities at a given time are collected in set S. Functions f_{cost} and $f_{duration}$ respectively describe the total cost and duration of the trajectory. The following table brings examples of trajectories.

Let us further suppose that we want to minimize the cost of the ticket and the total duration of the journey. Our compound objective function is thus $F : S \rightarrow [1, 1000] \times [1, 20]$. Formally, our optimization problem is stated as:

$$\begin{aligned} &\text{Minimize } f_{cost}(s) \text{ and } f_{duration}(s) \\ &\qquad\qquad s \in S \end{aligned}$$

2.2 Pareto Frontier and Pareto Sets

In the following we define the dominance relation between any two solutions, the Pareto frontier and the Pareto set, in a multi-objective problem.

Table 1. Fragment of trajectories data base; optimal solutions are indicated by "×" and non-optimal ones with "•"

s	$cost$	$duration$
t_1 ×	401	8
t_2 ×	400	15
t_3 ×	200	19
t_4 •	610	8
t_5 •	270	19
t_6 •	210	20

By definition, a solution $s \in S$ *dominates* a solution $s' \in S$ in a minimization framework when:

1. s is not worse than the s' in any of the objectives of the problem, i.e.

$$\forall i \in [1, I], f_i(s) \leq f_i(s')$$

2. s is strictly better than s' in relation to at least one goal, i.e.

$$\exists i \in [1, I], f_i(s) < f_i(s')$$

In our example, we want to minimize the cost and duration of the trip. The ideal solution would be a single trajectory s_0 in S that minimizes $F(s) = (f_{cost}(s), f_{duration}(s))$. In practice, it may be impossible to reach this global objective, as the fastest route is rarely the most economical.

Considering only the trajectories in Table 1, we see that the set of non-dominated solutions is given by $\{t_1, t_2, t_3\}$. The remaining do not belong to that set because trajectory t_4 is dominated by t_1, whereas t_5 and t_6 are dominated by t_3.

The *Pareto set* (PS) of the problem consists of all the non-dominated solutions in S. Formally, we have:

$$PS = \{s \in S \mid \nexists\, s' \in s, s \text{ is dominated by } s'\}$$

Let $PS = \{s_1, s_2, ..., s_p\}$ be a Pareto set with p elements. The *Pareto frontier* (PF) is defined as:

$$PF = \{F(s_1), F(s_2), ..., F(s_p)\}$$

Note that the Pareto set and Pareto frontier only exist as such when all the solutions are known, otherwise we only have estimates, because a new solution may dominate solutions that were not dominated previously.

3 Fuzzy Systems

Fuzzy Sets Theory was proposed by Professor Lofti Zadeh at the University of California in 1965 [13] to address the vagueness aspect of information (see also [5]). It later

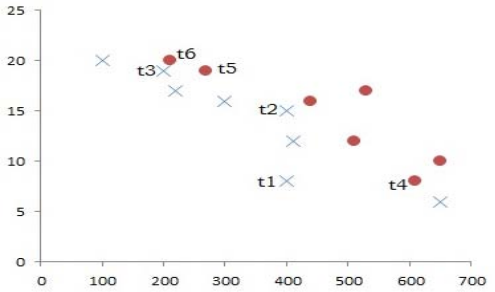


Fig. 1. Cost and duration of flights in the trajectories problem: the points correspond to non-dominated solutions and the crosses to the dominated ones

gave rise to a large number of concepts, operations and measures that are applicable to all kinds of disciplines in science.

Fuzzy sets theory is the starting point in the development of several types of fuzzy systems. Fuzzy Inference systems (FIS) are described as universal approximators that can be used to model the nonlinear relationships between inputs and outputs. Most of these systems are a particular case of the class of Rule-Based Systems, that use rules of thumb of the type " If <*premise 1*> and ... and <*premise n* > then <*conclusion*>. Usually, four main tasks are carried out in the execution of such type of systems: encoding fuzzification (or encoding), inference, composition and defuzzification (or decoding) [12]. A FIS can be created by encoding the knowledge of an expert in a particular field, or through the use of a learning algorithm, such as Neural Networks.

In the following, we briefly present some basic definitions from fuzzy sets theory and a basic FIS framework. We then extend the running example in a fuzzy context.

3.1 Basic Definitions and FIS Framework

A fuzzy set A defined on a universe of discourse X is characterized by a membership function $A : X \rightarrow [0,1]$[1]. The value $A(x)$ indicates the degree of compatibility of element x in X to the concept expressed by fuzzy set A: $A(x) = 0$ indicates that x is not compatible with A, $A(x) = 1$ indicates that x is fully compatible with A, and $0 < A(x) < 1$ indicates x is only partially compatible with A.

Many of fuzzy rule-based systems use a rule base $R = \{R_1, ..., R_m\}$ of the type R_j : If $x_1 = A_{1,j}$ and ... and $x_n = A_{n,j}$ then $y_j = B_j$, where each x_i (respec. y) takes value in its respective domain X_i (respec. Y). Each x_i is called a linguistic variable and have a set of k_i fuzzy sets (called fuzzy terms) $T_i = \{T_{i,1}, ...T_{i,k_i}\}$ associated to it. For every rule R_j in R, we have $\forall i, j, A_{i,j} \in T_i$.

Let $x^* = \{x_1^*, ..., x_n^*\}$ denote the input vector, i.e., each x_i^* denotes the realization of variable $x_i \in X_i$. Let $\top : [0,1]^2 \rightarrow [0,1]$ (respect. $\perp : [0,1]^2 \rightarrow [0,1]$) be a T-norm (respec. T-conorm), an operator that is commutative, associative and monotonic, with 1 (respec. 0) as neutral element.

[1] Here we use the same word to name a fuzzy set and its membership function.

Inference in such systems can be seen as a function $FIS\_solution : X \rightarrow Y$, usually calculated as described below.

- ▷ **Step 1**: The *compatibility* between the i-th premise of rule R_j with its corresponding input x_i^* is calculated as

$$\alpha_{i,j} = A_{i,j}(x_i^*), \quad 1 \leq i \leq n, \quad 1 \leq j \leq m \tag{1}$$

- ▷ **Step 2**: The *global compatibility* of rule R_j, with x^* is calculated as:

$$\alpha_j = \top(\alpha_{1,j}, \ \ldots, \alpha_{n,j}), \quad 1 \leq j \leq m \tag{2}$$

- ▷ **Step 3**: A fuzzy set C_j is calculated as the value for y according to rule R_j, given x^*, using an implication function I (see below) as:

$$C_j(y) = Imp(\alpha_j, B_j(y)), \quad \forall y \in Y \tag{3}$$

- ▷ **Step 4**: Using an operator ∇, the various C_js are aggregated as single fuzzy set C, representing the global solution of the problem given x_0^*, as:

$$C(y) = \nabla(C_1(y), \ \ldots C_m(y)), \ \ \forall y \in Y \tag{4}$$

- ▷ **Step 5**: A single value y^* from Y is calculated as the solution of the problem vector x^*, a process known as *deffuzification*, as

$$FIS\_solution : X \rightarrow Y(x^*) = y^* = def(C), \tag{5}$$

where $def : \mathcal{F}(y) \rightarrow Y$ and $\mathcal{F}(y)$ is the set of fuzzy sets that can be constructed on Y.

Function Imp is usually either a T-norm or a residuated implication operator [5]. When Imp is a T-norm, ∇ is usually a T-conorm. When Imp is a residuated implication operator, ∇ is usually a T-norm.

3.2 Extension of the Running Example

Let us now suppose that the level of satisfaction of a given user with respect to a given trajectory, can be modelled using a fuzzy rule based system, as follows.

- Input variables $cost \in [1, 1000]$ and $duration \in [1, 20]$
- Output variable $satisfaction \in [0, 10]$
- $T_{cost} = \{Cheap, Medium, Expensive\}$
- $T_{duration} = \{Short, Medium, Long\}$
- $T_{satisfaction} = \{VeryBad, Bad, Medium, Good, VeryGood\}$

Table 2 and Figure 2 respectively bring the rule base and the fuzzy terms for our problem, for a traveller for whom flight cost is more important than flight duration. Figure 3 depicts the output surface $FIS\_solution$ obtained using the so-called Mamdani fuzzy system [5], with $\top = \min$ and $Imp = \max$. We can clearly see that the preferred solutions are those that with low cost and short flight duration.

Table 2. Rule base for our problem; rows and columns refer to $cost$ and $duration$, respectively

	Short	*Medium*	*Long*
Expensive	*Bad*	*VeryBad*	*VeryBad*
Medium	*Good*	*Medium*	*Bad*
Cheap	*VeryGood*	*VeryGood*	*Medium*

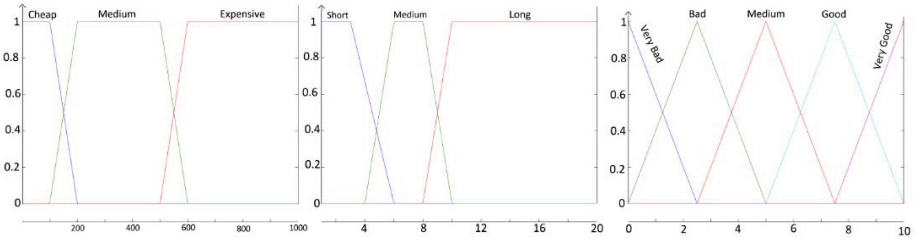


Fig. 2. Fuzzy terms for the trajectories problem: a) T_{cost}, b) $T_{duration}$ and c) $T_{satisfaction}$

4 A Proposal to Order Pareto Sets

In the following we propose a general approach to order the solutions in the Pareto set associated with a given multi-objective problem. Then we study how fuzzy systems can be used as a means to implement the general approach.

4.1 A General Algorithm to Order Pareto Sets

In complex problems, an optimization algorithm O does not generate the complete search space S of a given problem, but a subset $S_O \subseteq S$. Consequently, the Pareto set PS, containing the optimal solutions from S is usually also not completely generated. Moreover, more often than not, instead of a subset of PS, at the end of the execution of O, we obtain a set of solutions $PS^e \subseteq S_O$ that represents only an estimate of PS. It is interesting to note that set PS^e may not grow monotonically with S_O. In other words, if more solutions are visited, i.e. S_O is enlarged to a set ${S_O}'$, the previous set of non-dominated solutions PS^e may not be a subset of $PS^{e'}$, because solutions previously considered optimal may become dominated by those in ${S_O}' - S_O$.

Here we are interested in ordering set PS^e found by a generic optimization algorithm, according to the preferences of a user, modeled by a function called SAT, abbreviated from "satisfaction". We propose the following algorithm to order solutions in a Pareto sets, inspired on [8] (see also [1]).

ALGORITHM
Let S and Ω be the search and objective spaces, respectively. LetPS be the Pareto set for S, according to the problem characteristics and available data. Let $F = \{f_1, f_2, ..., f_I\}$, where $\forall i, f_i : S \rightarrow \Omega_i$, be the set of objective functions that have to be optimized. Let $Q = \{q_1, q_2, ..., q_J\}$, where $\forall j, q_j : S \rightarrow \Gamma_j$, be a set of functions that are not the

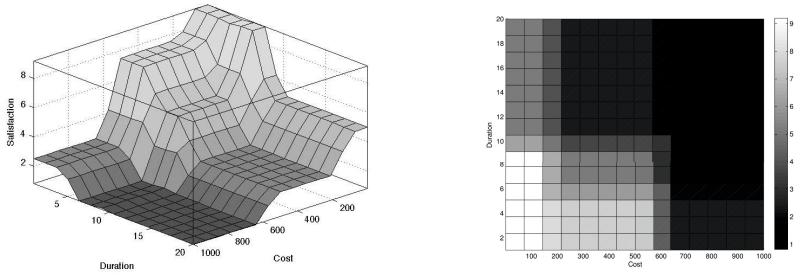


Fig. 3. Two views of output surface $FIS\_solution$ for the trajectories problem

subject of optimization. Let $SAT : \Omega_1 \times ... \times \Omega_m \times \Gamma_1 \times ... \times \Gamma_k \rightarrow [0, 1]$ be a function that models the user satisfaction with the options in S. Let O be an algorithm that aims at optimizing the functions in F.

1. Find a set of solutions $S_O \subseteq S$, by applying an algorithm O on the available data.
2. Gather the non-dominated solutions $PS^e \subseteq S_O$, according to the set of objective functions $\{f_1(s), ..., f_I(s)\}$.
3. Apply function $SAT : \Omega_1 \times ... \times \Omega_I \times \Gamma_1 \times ... \times \Gamma_J \rightarrow [0, 1]$, to the options in PS^e.
4. Order the pairs (solution, evaluation) of the set $\{(s, eval(s)) \mid s \in PS^e, eval(s) = SAT(f_1(s), ..., f_I(s), q_1(s), ..., q_J(s))\}$.
5. Choose the final solution from the ordered set.

Note that the user preference may take into account not only the functions to be optimized but also others, that although not subject to optimization, are relevant to the user. In our problem, those functions could for instance model that the user, although only wanting to optimize cost and duration, prefers trajectory with only a small number of stops, or companies in which the user is a frequent traveller, etc.

For many problems, the final solution is simply the first one in the ordered set. But for some complex problems, obtaining the ordered set is just a step towards selecting the final solution, that will depend on other criteria than minimality of the objective functions, as will be described in Section 4.

Function SAT may be the result of another system, such as learning systems, e.g. neural networks, knowledge-based systems, e.g. fuzzy systems, or data mining, such as case-based reasoning systems. In the following we propose a fuzzy approach to model the preferences of a user, in which function SAT is the output of a fuzzy inference system.

4.2 Ordering Pareto Sets with Fuzzy Systems

A fuzzy rule-based system can be used to model a satisfaction function SAT. For that, we just have to make

$$\forall s \in S, SAT(s) = FIS\_solution(f_1(s), ..., f_I(s), q_1(s), ..., q_J(s)),$$

where SAT is defined in the algorithm formulated in 4.1 and $FIS\_solution$ is calculated by Equation (6) in 3.1, associating each function in $F \cup Q$ with an input variable $x \in X$ in the fuzzy system.

In our example, variables $cost$ and $duration$ in X are associated to functions f_{cost} and $f_{duration}$ in F, respectively, and $Q = \emptyset$. The satisfaction calculated for the trajectories in Table 1 are found in Table 3. Trajectory t_1 clearly stands out among the optimal solutions. We see that many optimal trajectories are considered as desirable as dominated ones, which shows that optimization and preference do not necessarily go together. Function SAT implemented by the fuzzy system was capable of distinguishing between t_1 and t_4 that have the same duration but a large difference in cost. In the same way, t_1 is considered clearly better than t_2, that has practically the same cost but differs significantly on cost duration.

Table 3. Ordered fragment of trajectories data base with *SAT* function evaluation; optimal solutions are indicated by "×" and non-optimal ones with "•"

s	f_{cost}	$f_{duration}$	SAT
t_1 ×	401	8	5
t_2 ×	400	15	2.5
t_3 ×	200	19	2.5
t_5 •	270	19	2.5
t_6 •	210	20	2.5
t_4 •	610	8	.8

Similar approaches could be used here, instead of the adopted Mamdani model, such as the fuzzy residuated implication approach proposed in [10]. Moreover, the output surface could also be obtained using an approach that combines preference modeling, using the Sugeno integral, with a fuzzy inference system as described in Section 3, but employing a fuzzy residuated implication approach [6]. The advantage in this case is that the system would be able to incorporate existing preferences by a user in relation to the objectives.

Here we have used only the objective functions as the input of the fuzzy system, but it is possible to also incorporate other variables of interest. In our trajectories example, we could for instance take into account the air companies, the distance to the airports, etc.

5 A Potential Application

This work has been conceived to help engineers from the Brazilian National Institute for Space Research to select equipment layouts to be adopted in a given satellite. Figure 4 brings the evaluations of a set of solutions for the allocation of eight objects (batteries, transponders, etc)on one of the satellite panels, obtained by the evolutionary-based multi-objective optimizer M-GEO [7]. The problem is described using six project variables (mass, dissipated heat and three geometric dimensions). The solutions are evaluated considering three conflicting objectives: obtaining the center of mass of the system

(the equipments) close to a target one (f_1), obtaining a homogeneous distribution of heat inside the satellite (f_2), and minimizing the sum of the distances between equipments inside a subsystem (f_3), in order to reduce the length of cables connecting them (see details in [11]).

Having a set of solutions in hand, the engineers select a small set for visual inspection, and consider other criteria than the ones used in the basic problem descriptor. Then these solutions can suffer small (in most cases) modifications using past experience on other similar problems. Figure 5 brings a set of six solutions, whose evaluations are depicted in Figure 4b. The set consists of the three best single objective solutions (best solution for each objective in the Pareto frontier), two based on the frontier baricenter and one calculated using the Minimal Loss Criterion [9].

It is important to select a reasonably small set of solutions to be further examined, because the the examination process of each solution is very time-consuming. Choosing which solutions to check is made difficult by several factors. One of them is the fact that the influence of each object in the layout cannot be appreciated by the objective functions, since they are an aggregation of other functions on those objects. Moreover, in problems with three objective functions visualizing the evaluation of the solutions is already bothersome; with more dimensions it becomes impossible. The problem described here is a

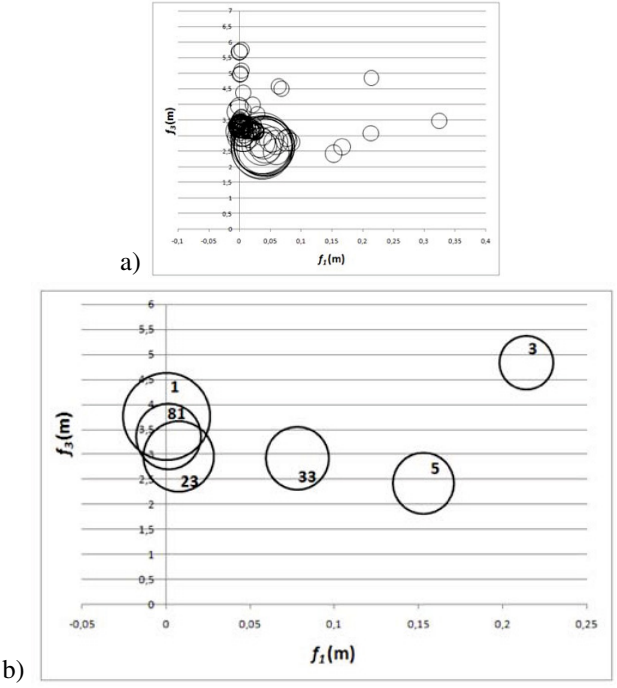


Fig. 4. Evaluation of a spacecraft equipment layout problem solutions (the larger the bubble, the higher the value of objective function f_3) for: a) all solutions b) selected solutions (Source: [11])

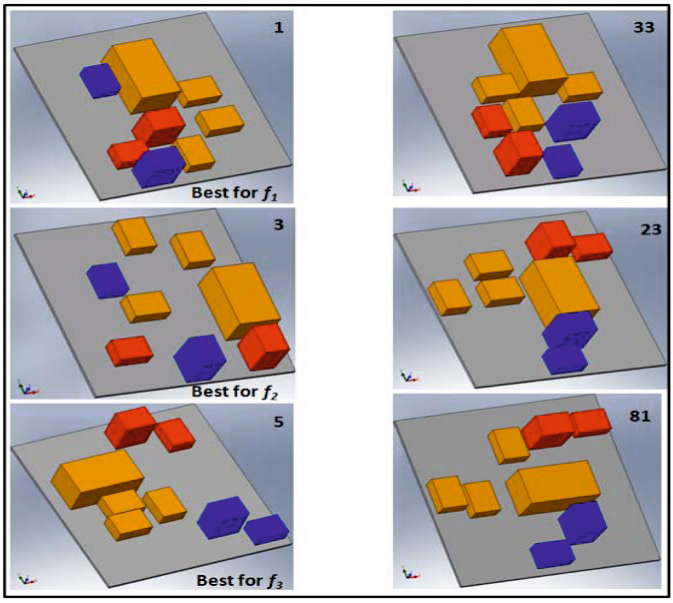


Fig. 5. Solutions for the problem of allocating objects inside a satellite, considering 3 objective functions, corresponding to the evaluations depicted in Figure 4b (Source: [11])

simplification of a more complex one, that involves the allocation of objects on panels disposed in three dimensions.

The proposed approach in this kind of application is interesting for various reasons. First of all, the solutions can be ranked according to knowledge that is available but not otherwise considered in the phase of solution selection. Moreover, such a system can be used in an interactive manner, by changing the rules or terms to privilege one or other objective. Finally, more than one expert can create a fuzzy system and either the satisfaction surfaces or the rankings can be aggregated to produce a better overall ordering of optimal solutions.

6 Concluding Remarks

In many applications involving multi-objective problems, it is important to guarantee optimality of at least one of a set of conflicting objectives. Once the set of non dominated-solutions (the Pareto set or an approximation of it) is found, it is often necessary to extract a subset of these solutions to be examined more closely, to finally choose a single solution to be implemented. On the other hand, it is often the case that there exists available knowledge about the preferred solutions to the problem at hand that can be modelled by means of fuzzy rules and terms. In a nutshell, in this work we propose to use a multi-objective optimizer to discard dominated solutions and a fuzzy system to order the non-dominated ones.

One of the main advantages of the proposed approach is that available knowledge about the problem can be easily incorporated in the task of ordering non-dominated solutions. Moreover, fuzzy systems allow for interactively which gives flexibility to the end user. As future work, we intend to adapt the proposed approach in the creation of spacecraft equipments layouts, incoportaing available knowledge that is not taken into account by the optimization procedure adopted so far [11].

Acknowledgments. The authors would like to thank Gilberto Pedro da Silva Júnior for implementing the fuzzy system for the running example and F. Souza, E.M. Rocco and W.A. Santos for discussions on the use of fuzzy systems for the spacecraft equipment layout problem.

References

1. Coello, C.A.C., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems. Springer (2007)
2. Coello, C.A.C.: Handling Preferences in Evolutionary Multiobjective Optimization: A Survey. In: 2000 Congress on Evolutionary Computation (2000)
3. Xu, J., Zhou, X.: Fuzzy-Like Multiple Objective Decision Making. Springer (2011)
4. Das, I.: A preference ordering among various Pareto optimal alternatives. Structural Optimization 18, 30–35 (1999)
5. Dubois, D., Prade, H.: Possibility Theory: An Approach to Computerized Processing of Uncertainty. Plenum Press, New York (1988)
6. Dubois, D., Fargier, H., Sandri, S.: Fuzzy MCDM and the sugeno integral. In: Huynh, V.-N., Nakamori, Y., Lawry, J., Inuiguchi, M. (eds.) Integrated Uncertainty Management and Applications. AISC, vol. 68, pp. 257–268. Springer, Heidelberg (2010)
7. Galski, R.L.: Desenvolvimento de versões aprimoradas híbridas, paralela e multiobjetivo do método da otimização extrema generalizada e sua aplicação no projeto de sistemas espaciais (INPE-14795-TDI/1238). PhD Thesis in Appleid Computation, Instituto Nacional de Pesquisas Espaciais, SJCampos (2006) (in Portuguese)
8. Pirjanian, P., Matari, M.J.: Multi-Robot Target Acquisition Using Multiple Objective Behavior Coordination. In: Proc. International Conference on Robotics and Automation (ICRA 2000), San Francisco, CA, pp. 2696–2702 (2000)
9. Rocco, E.M., Souza, M.L.O., Prado, A.F.B.A.: Station Keeping of Costellations Using Multiobjective Strategies. Mathematical Problems in Engineering, vol. 2013. Hindawi Publishing Corporation (2013) dx.doi.org/10.11 (2013) 55/2013/476451
10. Sandri, S., Sibertin-Blanc, C.: A multicriteria System Using Gradual Fuzzy Rules and Fuzzy Arithmetic. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 16(suppl.1), 17–34 (2008)
11. Souza, F.L., Galski, R.L., Rocco, E.M., Becceneri, J.C., Dos Santos, W.A., Sandri, S.A.: A tool for Multidisciplinary Design Conception of Spacecraft Equipment Layout (submitted)
12. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its application to modeling and control. IEEE Trans. Systems, Man, and Cybernetics 15, 116–132 (1985)
13. Zadeh, L.A.: Fuzzy sets Information and control 8(3), 338–353 (1965)

A Comparison of Two Approaches for Situation Detection in an Air-to-Air Combat Scenario

Anders Dahlbom

Informatics Research Centre, University of Skövde,
P.O. Box 408, SE-541 28 Skövde, Sweden
anders.dahlbom@his.se

Abstract. Combat survivability is an important objective in military air operations, which involves not being shot down by e.g. enemy aircraft. This involves analyzing data and information, detecting and estimating threats, and implementing actions to counteract threats. Beyond visual range missiles can today be fired from one hundred kilometers away. At such distances, missiles are difficult to detect and track. The use of techniques for recognizing hostile aircraft behaviors can possibly be used to infer the presence and for providing early warnings of such threats. In this paper we compare the use of dynamic Bayesian networks and fuzzy logic for detecting hostile aircraft behaviors.

Keywords: Threat assessment, situation recognition, behavior recognition, behavior detection, Bayesian networks, fuzzy logic.

1 Introduction

Combat survivability is in military operations concerned with survival of the own aircraft and entails analyzing data and information, detecting and estimating threats, and implementing actions to counteract detected threats. Threats can be defined as elements designed to inflict damaging effects, force undesirable maneuvers or degrade system effectiveness [1]. Two main types of threats that can affect the combat survivability of an aircraft can be discerned: enemy ground and sea based firing units and enemy fighter aircraft [2]. Although both types of threats are important to consider, this paper focuses on threats in the air, i.e. threats posed to the own aircraft by enemy aircraft.

The threat of an enemy aircraft can be determined as a combination of two parameters: intent and capability [3, 4], where capability refers to an opposing agent's ability to inflict injury or damage, and intent refers to the will or determination of an enemy to do so [4]. Determining the capability of an enemy entails platform identification and from this inferring the capabilities of the platform in relation to the own aircraft [3]. Intent is more difficult to determine since it cannot often be directly observed, and instead involves reasoning around the future behavior of the enemy based on its behavior [3].

Beyond visual range (BVR) missiles can today be fired from more than 100kms, and at such distances it can be very hard to detect and track them

V. Torra et al. (Eds.): MDAI 2013, LNAI 8234, pp. 70–81, 2013.

directly with the on-board sensor systems. In order to get early warnings of such imminent threats, the behavior of enemy aircraft can be analyzed to possibly infer the action of firing long-range air-to-air missiles. This can be cast as a situation recognition problem [5, 6] which in its essence concerns defining a number of relations that in sequence or in parallel define situation types that are of interest and then try to identify instances of these in data.

1.1 Related Work

Both deterministic and probabilistic methods have been used for addressing the problem of detecting complex patterns. Deterministic approaches include the use of logic and temporal constraint propagation for situation and chronicle recognition in e.g. environment surveillance and networks surveillance applications [7, 8]. The use of chronicle recognition and temporal constraint propagation has also been investigated in air scenarios [9]. Also related is work on complex event recognition using rule based approaches with extensions for modeling temporal constraints [10]. A rule-based approach has also been proposed for recognition of behaviors in maritime surveillance applications [11]. Deterministic recognition of complex behaviors has also been addressed using Petri nets, for e.g. complex event recognition in video surveillance [12, 13], for modeling plan and activity prototypes in automated scene recognition [14], and for multi-agent activity recognition in basketball games [15]. Petri nets have also been investigated for situation recognition in surveillance scenarios [16, 6].

Probabilistic approaches for detecting complex behaviors include the use of hidden Markov models (HMMs) and dynamic Bayesian networks (DBNs) for recognizing traffic situations [17]. The use of Bayesian networks (BNs) is also popular in the surveillance domain, e.g. for detecting insider threats in information systems [18], for signature based detection of maritime situations [19]. Highly related is also work on using and constructing DBNs for recognition [20]. Besides the use of graphical models the problem of recognizing interesting situations has also been addressed using fuzzy logic [21].

Also related is work on threat evaluation in ground based air defense situations. In threat evaluation, the objective is to estimate the level of threat that individual (enemy) objects pose to one's own defended assets [4]. The threat evaluation problem has been addressed using e.g. rule-based systems [22, 23], BNs [24–26], evidential networks [27], DBNs [28, 29] and fuzzy logic [30–32]. A main difference compared with this work, is that the focus here is on explicitly representing specific situations that evolve over time.

1.2 Problem

This paper focuses on intent inference based on recognized aircraft behaviors that play out over time. A number of requirements can be identified: (1) there can be multiple types of interesting situations thereof the interest to make use of methods for describing situations at an abstract level, (2) situations may play out over time requiring representations that allow for temporal relations, (3)

sensor data does not provide a perfect nor complete view of the world, which puts requirements on methods to be able to cope with uncertainty, and (4) it can be important to continuously get estimates that represents the likelihood, or similar, that a specific situation is possibly taking place.

In previous work [33] we have investigated the use of DBNs for detecting one type of hostile aircraft behavior. This paper compares the use of DBNs and fuzzy logic on this task. Situation types defined using fuzzy logic has the advantage of being defined in human interpretable terms, thus possibly making it easier to define interesting situations for human experts. Furthermore, in the DBN approach we relied upon crisp variable discretization. Fuzzy logic allows for fuzzy discretization, which can be beneficial. However, DBNs allow for temporal relations to explicitly be used, which classical fuzzy logic does not.

2 The Interesting Situation

One type of interesting BVR situation is coupled to the behavior of firing long range air-to-air missiles. This situation has previously been discussed in [33] and is illustrated in figure 1.

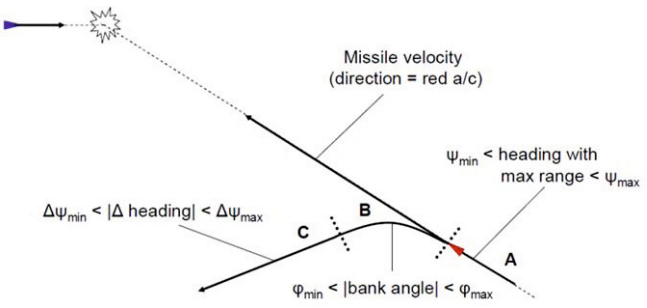


Fig. 1. Illustration of the interesting situation. A, B, and C denote three different phases of the situation (adapted from [33]).

The purpose in the first phase (A) is to move the launching aircraft (a/c) so that the target is within its weapon range. The range of the weapon is highly dependent on the velocity of the carrier and to maximize weapon range the velocity of the launching a/c needs be aligned, at least horizontally, with line-of-sight to hit point. The hit point is where the target will be located when the missile arrives sometime in the future. The next phase (B) begins shortly after launching the missile. In this phase the launching a/c turns to decrease the opponent's weapon range while maintaining radar to missile data link communication and radar coverage of the target. It is important to uphold communication with the missile to continuously transmit target data since the missile is not capable to track the target by itself at this distance. The last phase (C) starts when the

heading has changed to such a degree that radar field of view limit is reached. Turning more would yield loss of radar to missile communication and a loss of radar coverage of the target.

2.1 Variables for Detection

The information that is available to the recognition process is track data, including position and velocity of target, provided by the onboard sensor systems. Additional information may also be received over data links. Recognition however takes place at a more abstract level, and the process of defining a situation recognition system involves (1) defining which measures to use and how these can be extracted or calculated from data, and (2) defining symbols and situation types using the defined measures. The first of these steps is based on previous work [33] and is here recapitulated. The definition of symbols and patterns is carried out in sections 3 and 4 for DBNs and fuzzy logic, respectively.

In previous work [33] three measures were used when defining a DBN for recognizing the interesting situation: distance between the target and the own a/c (D), distance at closest point of approach of the two platforms ($DCPA$), and relative angle between the two platforms (RA). Distance and DCPA are coupled to phase A. The first criterion in this phase can be calculated using the Euclidean distance (a maximum weapon range of 100km is assumed), D. The second criterion in the first phase is that a weapon fired from the present position of the target should have a future interception point with the own a/c. This can be estimated using $DCPA$ which denotes the distance between two objects at their closest point of approach (CPA), given their present positions and velocities. In the second and third phases of the situation, the enemy platform should turn, but not too much. The relative angle (RA) was used for capturing both of these phases.

$DCPA$ can be calculated using equation 1, where $TCPA$ (time to CPA) can be calculated using equation 2, p_1 is our position, p_2 the position of the enemy, v_1 our velocity, v_2 the velocity of the enemy, and v_r the relative velocity between the two platforms. Note that since the interest here is to calculate a future interception distance between us and a weapon fired from the enemy platform, the velocity vector of the enemy needs to be normalized and then multiplied with the assumed weapon speed. RA can be calculated using equation 3, where p_1 is our position, p_2 is the position of the enemy and where v is the velocity vector of the enemy.

$$DCPA = D(p_1 + v_1 \cdot TCPA, p_2 + v_2 \cdot TCPA) \tag{1}$$

$$TCPA = \begin{cases} -\frac{\overrightarrow{p_2 - p_1} \cdot v_r}{|v_r|^2} & \text{if } v_r \cdot v_r > 0 \wedge |v_r| > 0 \\ \infty & \text{otherwise} \end{cases} \tag{2}$$

$$RA = cos^{-1}(\overrightarrow{p_2 - p_1} \cdot v) \tag{3}$$

3 Detection Using Dynamic Bayesian Networks

A BN [34] enables for a compact representation of a full joint probability distribution and is constructed as a directed acyclic graph, where nodes represent random variables and where edges denote conditional dependence between variables. For each node, a joint probability distribution is formed together with its direct ancestors, referred to as conditional probability tables. BNs can be seen as representing cause and effect relations amongst nodes, and given evidence for some variables that have been observed (information variables), the posterior probability distributions for other variables can be determined (hypothesis variables). Although classical BNs allow for causal relations to be modeled, they do not allow for modeling temporal dynamics.

A DBN [35] on the other hand allows for modeling dynamic systems. DBNs are a generalization of HMMs and BNs which allows for causal time dependencies to be modeled. In a DBN a set of time slices is depicted. In addition to conditional dependencies to other nodes in the same time slice, nodes are in DBNs also able to have dependencies to nodes in previous time slices (not necessarily restricted to only the previous time slice).

3.1 Random Variables, Discretization and Network Structure

The experiments presented in this paper have used the DBN presented in [33], and is only briefly recapitulated here. The three variables presented in section 2.1, distance (D), distance at closest point of approach (DCPA), and relative angle (RA) have been used for defining information variables in the DBN. Related to the first phase are the two random variables WithinWeaponRange (WWR) and WeaponInterceptionDistance (WID), defined in equations 4 and 5 where S denotes the own a/c, and where T denotes the enemy platform. The second and third phases of the situation are captured by a single random variable InRadarCoverage (IRC), which is defined in equation 6.

$$WWR = \begin{cases} True & \text{if } D(S,T) < 100000 \\ False & \text{otherwise,} \end{cases} \tag{4}$$

$$WID = \begin{cases} Short & \text{if } DCPA < 500 \\ Medium & \text{if } DCPA < 5000 \\ Large & \text{otherwise,} \end{cases} \tag{5}$$

$$IRC = \begin{cases} Inside & \text{if } RA < 45 \\ OnEdge & \text{if } 45 \leq RA \leq 65 \\ Outside & \text{otherwise.} \end{cases} \tag{6}$$

In addition to these, two hypothesis variables have also been identified: MissileLaunched and MissileInAir. MissileInAir is the main hypothesis variable that we want to calculate the likelihood of since it indicates the likelihood of the

situation taking place. The MissileLaunched variable is used for separation of the first phase from the combined second and third phases. The DBN is shown in figure 2. Additional aspects that could be included are e.g. target platform type, identity, origin, radar and missile boundaries and general threat level.

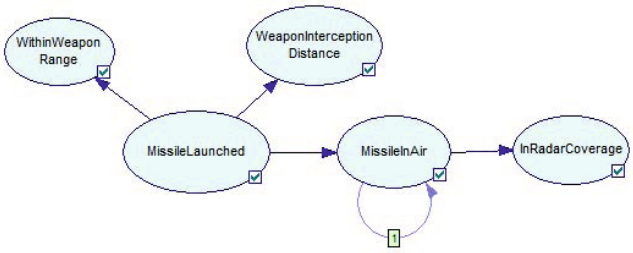


Fig. 2. A DBN describing the interesting situation. The DBN has five nodes, three information nodes and two hypothesis nodes (adapted from [33]).

4 Detection Using Fuzzy Logic

Fuzzy logic is based on the concept of fuzzy sets which are sets that in contrast to crisp sets do not have clearly defined boundaries. In classical set theory, an element is either a member of a set or it is not a member of a set. In fuzzy set theory, elements can have a degree of membership in a set. Formally, given a universe of discourse X, a fuzzy set A in X is defined as a set of ordered pairs $A = \{x, \mu_A(x) | x \in X\}$, where x is an element in X and where $\mu_A(x) \rightarrow [0, 1]$ is a membership function denoting the degree of membership that the element x has in the fuzzy set A.

The process of using fuzzy inference involves three steps: fuzzification, inference, and defuzzification. In the first step, numerical input variables are mapped to fuzzy sets using membership functions (often specified using graphs) and fuzzy linguistic terms (e.g. high, medium). In the second step, a set of fuzzy rules and operators are used to make inferences. The result of the inference step is one or more fuzzy sets which in the third step are converted back to numerical output.

4.1 Fuzzy Variables, Membership Functions and Rules

The implemented fuzzy inference system uses the same set of input variables as the DBN; distance (D), DCPA, and relative angle (RA). Two primary output variables have been defined, Launch and Guide. These refer to missile launch opportunity and missile guidance, respectively. Additionally, two more input variables have been defined to capture the temporal aspects of the situation, WL (Weapon Launched) and WG (Weapon Guidance). The first of these, WL takes on the maximum value of the Launch parameter from previous time steps.

The reasoning is that once a launch opportunity has been detected, then it will still have been valid in the past in the next time step. The second of these, WG, is assigned the output of the guide parameter in the previous time step. The reasoning for this is that something is being guided over time and this variable tries to capture the temporal aspects of this. Finally, a third output parameter has been defined, WIA (Weapon In Air), and this is actually the output fuzzy set that we are interested in using for inferring the possibility of a missile having been launched. The rules that have been used in the fuzzy inference system are shown below, and the membership functions for variables are shown in figure 3.

```
1. D == Close && DCPA == Short => Launch = True (1)
2. D == Far => Launch = False (1)
3. D == Close && DCPA == Medium => Launch = True (0.5)
4. DCPA == Long => Launch = False (1)
5. D == Close && RA == Edge => Guide = True (1)
6. RA == Outside => Guide = False (1)
7. D == Far => Guide = False (1)
8. D == Close && WG == True => Guide = True (0.5)
9. D == Close && WL == True && WG == True => WIA = True (1)
10. WL == False => WIA = False (1)
11. WG == False => WIA = False (1)
12. WG == False => Guide = False (0.25)
```

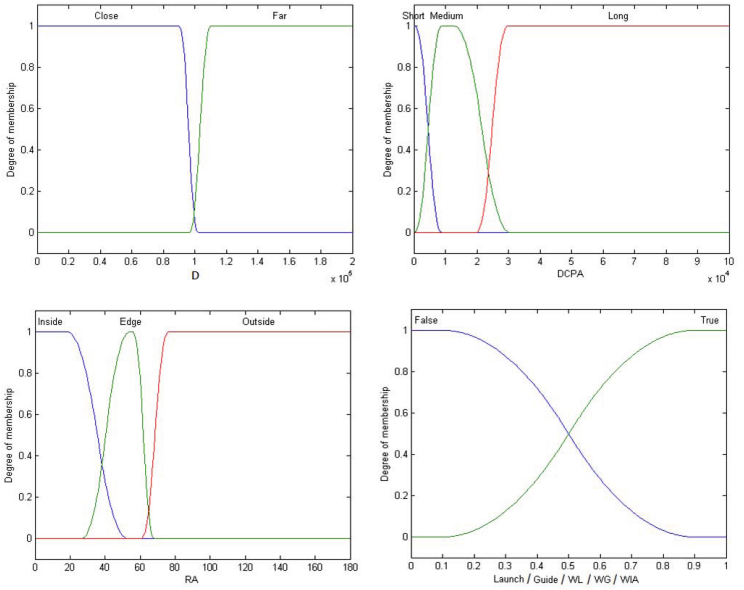


Fig. 3. Membership functions used in the fuzzy inference system. The same membership function is used for the variables Launch, Guide, WL, WG, and WIA.

5 Experimental Results

5.1 Experimental Setup

Two situations have been simulated using Matlab (figure 4), one in which the interesting behavior occurs and another where only parts of the interesting situation occurs. The two situations have been used to carry out two experiments, one in which the two techniques are used directly on the output of the simulated tracks, and one in which a measurement model has been used to add noise to the data. The DBN has been defined evaluated using Genie [36] and the fuzzy inference system has been implemented in the fuzzy logic toolbox in Matlab.

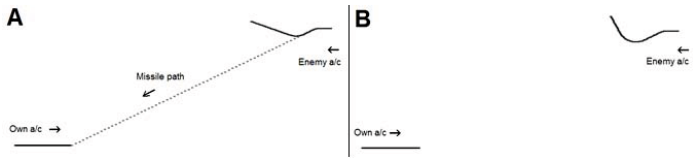


Fig. 4. Illustration of one situation representing the gimbal turn (A) and one situation not representing the gimbal turn (B). In the figures, Own a/c represents the position of the own aircraft, Enemy a/c represents the position of the enemy, and in (A) Missile path depicts the path of the missile, from Enemy a/c to Own a/c (adapted from [33]).

5.2 Detection Using Ground Truth Data

Figure 5 illustrates the results for the two approaches when using ground truth data (both situations a and b). As can be seen, the shapes of the curves are similar although represented using different measures. The output from the fuzzy inference system however rises more quickly for both situations, compared to the output from the DBN. It is also interesting to note that the output from the fuzzy inference system is more plateau like. This can however depend on the DBN smoothing the output since the curve has been calculated as a whole in Geenie. In case the output had been iteratively calculated for each time step, then the output might have been slightly different.

5.3 Detection Using Noisy Data

Figure 6 illustrates the results for the two approaches when using noisy data. Again, for the first situation (a), the shapes of the curves are similar for the first of the two situations. Even the fluctuations from time 50 and onwards have some resemblance. For the second situation (b) there is however some difference between the output from the two approaches. The fuzzy logic approach behaves very similar to the case of ground truth data, rise and plateau until the enemy a/c overturns. The output from the DBN however stops rising rather quickly. Again, a reason for this may be coupled to smoothing.

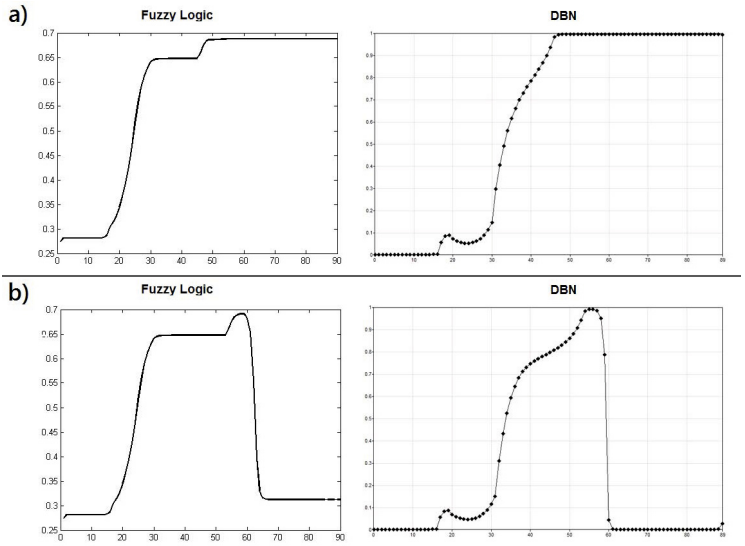


Fig. 5. Results of using the fuzzy inference system (to the left) and the DBN (to the right) on the data set that represents the interesting situation (a) and on the data set that does not represent the situation (b)

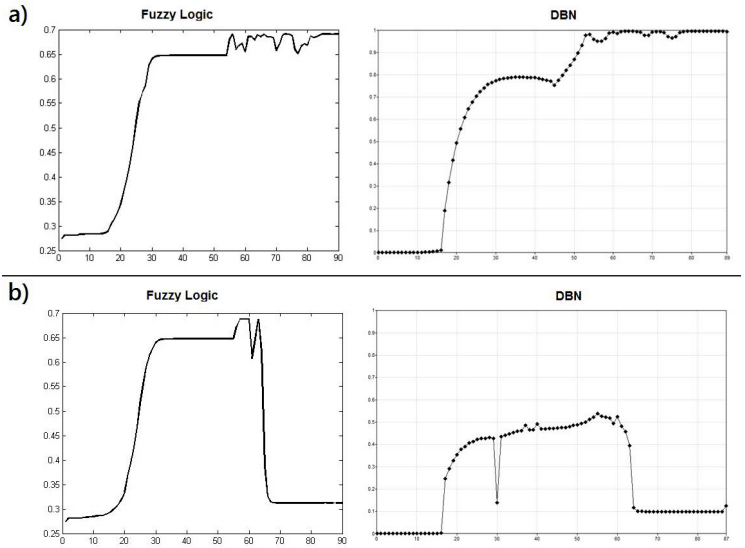


Fig. 6. Results when using the fuzzy inference system and the DBN on the noisy data representing the interesting situation (a) and when using noisy data that does not represent the situation (b)

6 Conclusion

Combat survivability is an important objective in military air operations, which involves not being shot down by e.g. enemy aircraft. This involves analyzing data and information, detecting and estimating threats, and implementing actions to counteract detected threats. BVR missiles can today be fired from more than 100kms, and at such distances it can be very hard to detect and track them directly with the onboard sensor systems. In order to get early warnings of such imminent threats, techniques for detecting enemy behaviors can be used.

In this paper we have carried out an initial experiment to compare the use of fuzzy logic and DBNs on the task of recognizing one type of interesting air-to-air missile launch behavior. In the experiment, the two techniques behave similarly. The output from using the fuzzy logic approach is more similar with and without noise compared to the output of the DBN, suggesting that the fuzzy logic approach is more stable when using noisy data. As an effect however, the DBN seems to better separate the two situations when using noisy data. This may however be an effect of smoothing. It is not certain that the same differentiation would have been achieved if the DBN were to be iteratively queried as more data becomes available. More experiments are needed to look into this.

Future work will be carried out on in four directions: (1) investigate more carefully how the techniques behave when varying the amount and type of noise, (2) investigate if and how the techniques can be used for recognizing multi-actor situations, (3) carry out investigations using other types of scenarios as well as real-world data, and (4) investigate other techniques, e.g. temporal fuzzy logic.

Acknowledgments. This research has been supported by the Infofusion Research Program (University of Skövde, Sweden) in partnership with Saab AB and the Swedish Knowledge Foundation under grant 2010/0230 (UMIF). I would like to acknowledge Per-Johan Nordlund at Saab AB for providing simulated data, discussions and descriptions.

References

1. Ball, R.E.: The Fundamentals of Aircraft Combat Survivability Analysis and Design. AIAA, New York (1985)
2. Helldin, T., Erlandsson, T., Niklasson, L., Falkman, G.: Situational adapting system supporting team situation awareness. In: Proceedings of SPIE 7833, Unmanned/Unattended Sensors and Sensor Networks VII (2010)
3. Nguyen, X.T.: Threat assessment in tactical airborne environments. In: Proceedings of the 5th International Conference on Information Fusion, Annapolis, Maryland, USA, July 7-11 (2002)
4. Paradis, S., Benaskeur, A., Oxenham, M., Cutler, P.: Threat evaluation and weapons allocation in network-centric warfare. In: Proceedings of the 8th International Conference on Information Fusion, Philadelphia, PA, July 25-29 (2005)
5. Dahlbom, A., Niklasson, L., Falkman, G., Loutfi, A.: Towards template-based situation recognition. In: Mott, S., Buford, J.F., Jakobson, G., Mendenhall, M.J. (eds.) Intelligent Sensing, Situation Management, Impact Assessment, and Cyber-Sensing, Orlando, FL, USA, April 15-17, vol. 7352, p. 735205. SPIE (2009)

6. Dahlbom, A.: Petri nets for Situation Recognition. PhD thesis, Örebro University, Sweden (2011)
7. Dousson, C., Gaborit, P., Ghallab, M.: Situation recognition: Representation and algorithms. In: Proceedings of the 13th International Joint Conference on Artifical Intelligence (IJCAI), Chambery, France, August 28-September 3, vol. 1, pp. 166–172. Morgan Kaufmann (1993)
8. Dousson, C., Le Maigat, P.: Chronicle recognition improvement using temporal focusing and hierarchization. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI), Hyderabad, India, January 6-12 (2007)
9. Coradeschi, S., Vidal, T.: Accounting for temporal evolutions in highly reactive decision-making. In: Proceedings of the Fifth International Workshop on Temporal Representation and Reasoning, Sannibel Island, FL, USA, May 16-17 (1998)
10. Walzer, K., Groch, M., Breddin, T.: Time to the rescue - supporting temporal reasoning in the rete algorithm for complex event processing. In: Bhowmick, S.S., Küng, J., Wagner, R. (eds.) DEXA 2008. LNCS, vol. 5181, pp. 635–642. Springer, Heidelberg (2008)
11. Edlund, J., Grönkvist, M., Lingvall, A., Sviestins, E.: Rule based situation assessment for sea-surveillance. In: Dasarathy, B.V. (ed.) Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications, Kissimmee, Florida, USA, April 19-20, vol. 6242. SPIE (2006)
12. Ghanem, N., DeMenthon, D., Doermann, D., Davis, L.: Representation and recognition of events in surveillance video using petri nets. In: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop, Washington, DC, June 27-July 2, vol. 7, pp. 112–120. IEEE Computer Society (2004)
13. Lavee, G., Borzin, A., Rivlin, E., Rudzsky, M.: Building petri nets from video event ontologies. In: Bebis, G., et al. (eds.) ISVC 2007, Part I. LNCS, vol. 4841, pp. 442–451. Springer, Heidelberg (2007)
14. Castel, C., Chaudron, L., Tessier, C.: What is going on? a high level interpretation of sequences of images. In: Buxton, B.F., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1065, pp. 13–27. Springer, Heidelberg (1996)
15. Perše, M., Kristain, M., Perš, J., Kovačič: Recognition of multi-agent activities with petri nets. In: Proceedings of the 17th International Electrotechnical and Computer Science Conference, Portorož, Slovenia, pp. 217–220 (September 2008)
16. Dahlbom, A., Niklasson, L., Falkman, G.: Situation Recognition and Hypothesis Management Using Petri Nets. In: Torra, V., Narukawa, Y., Inuiguchi, M. (eds.) MDAI 2009. LNCS, vol. 5861, pp. 303–314. Springer, Heidelberg (2009)
17. Meyer-Delius, D., Plagemann, C., Burgard, W.: Probabilistic situation recognition for vehicular traffic scenarios. In: IEEE International Conference on Robotics and Automation, Kobe, Japan, May 12-17 (2009)
18. Laskey, K., Alghamdi, G., Wang, X., Barbar'a, D., Shackelford, T., Wright, E., Fitzgerald, J.: Detecting threatening behavior using bayesian networks. In: Proceedings of the 13th Conference on Behavioral Representation in Modeling and Simulation (BRIMS), Arlington, Virginia, USA, May 17-20 (2004)
19. Fooladvandi, F., Brax, C., Gustavsson, P., Fredin, M.: Signature-based activity detection based on bayesian networks acquired from expert knowledge. In: Proceedings of the 12th Internation Conference on Information Fusion (Fusion 2009), Seattle, WA, USA, July 6-9, pp. 436–443 (2009)
20. Fischer, Y., Beyerer, J.: Defining dynamic bayesian networks for probabilistic situation assessment. In: Proceedings of the 15th International Conference on Information Fusion, Singapore, July 9-12 (2012)

21. Foo, P.H., Ng, G.W., Ng, K.H., Yang, R.: Application of intent inference for air defense and conformance monitoring. Journal of Advances in Information Fusion 4(1), 3–26 (2009)
22. Liebhaber, M.J., Feher, B.A.: Surface warface threat assessment: Requirements definition. Technical report, SSC San Diego, San Diego, CA (2002)
23. Liebhaber, M.J., Feher, B.: Air threat assessment: Research, model, and dispaly guidelines. In: Proceedings of the 2002 Command and Control Research and Technology Symposium (2002)
24. Okello, N., Thoms, G.: Threat assessment using bayesian networks. In: Proceedings of the 6th International Conference on Information Fusion, Cairns, Queensland, Australia (2003)
25. Johansson, F., Falkman, G.: A bayesian network approach to threat evaluation with application to an air defense scenario. In: Proceedings of the 11th International Conference on Information Fusion (Fusion 2008), Cologne, Germany, June 30-July 3, pp. 1352–1358 (2008)
26. Digioia, G., Panzieri, S.: Infusion: A system for situation and threat assessment in current and foreseen scenarios. In: 2012 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situaton Awareness and Decision Support, New Orleans, LA (2012)
27. Benavoli, A., Ristic, B., Farina, A., Oxenham, M., Chisci, L.: An application of evidential networks to threat assessment. IEEE Transactions on Aerospace and Electronic Systems 45(2), 620–639 (2009)
28. Hou, Y., Guo, W., Zhu, Z.: Threat assessment based on variable parameter dynamic bayesian network. In: Proceedings of the 29th Chinese Control Conference, Beijing, China (2010)
29. Wang, Y., Sun, Y., Li, J.Y., Xia, S.T.: Air defense threat assessment based on dynamic bayesian network. In: The 2012 International Conference on Systems and Informatics, Yantai, China (2012)
30. Liang, Y.: An approximate reasoning model for situation and threat assessment. In: Proceedings of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery, Haikou, Hainan, China (2007)
31. Johansson, F., Falkman, G.: A Comparison between Two Approaches to Threat Evaluation in an Air Defense Scenario. In: Torra, V., Narukawa, Y. (eds.) MDAI 2008. LNCS (LNAI), vol. 5285, pp. 110–121. Springer, Heidelberg (2008)
32. Kumar, S., Dixit, A.M.: Threat evaluation modelling for dynamic targets using fuzzy logic approach. In: Proceedings of the International Conference on Computer Science and Engineering, Dubai, UAE (2012)
33. Dahlbom, A., Nordlund, P.J.: Detection of hostile aircraft behaviors using dynamic bayesian networks. In: Proceedings of the 16th International Conference on Information Fusion (2013)
34. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Mateo (1988)
35. Murphy, K.P.: Dynamic Bayesian Networks: Representation, Inference and Learning. PhD thesis, University of California, Berkeley (2002)
36. DSL: Genie and smile. Decision Systems Laboratory, University of Pittsburgh, Pittsburgh, PA, USA

Web 2.0 Tools to Support Decision Making in Enterprise Contexts

Raquel Ureña and Enrique Herrera-Viedma

Department of Computer Science and Artificial Intelligence
C/ Periodista Daniel Saucedo Aranda s/n, 18071, Granada
{raquel,viedma}@decsai.ugr.es

Abstract. Nowadays Web 2.0 provides users a unique framework not only to find information but also to express their opinions and collaborate and interact in real time. Web 2.0 includes applications such as blogs, wikis, RSS, pod-casting, mashups, and social networks. These applications aggregate the collective intelligence of millions of users and therefore new tools to develop decision making processes adapted to the new virtual environments need to be developed. In this contribution we analyse how Web 2.0 tools are used to improve cooperation and social decision making in the enterprise context and what are the challenges that need to be accomplished to take fully advantage of them.

Keywords: Decision making, Web 2.0, Enterprise 2.0

1 Introduction

We live in a world where technology has changed the way people communicate, interact, get information and do business. Web 2.0 is the common term for advanced Internet technologies and applications including social networks, blogs, wikis, RSS, podcasting and mashups. All these tools and applications are often known as Social Media Technologies, SMT [12]. One of the most significant differences between traditional Web and the Web 2.0 is that in the latter the content is user generated, and there is greater collaboration among Internet users.

Web 2.0 communities provide a framework to collaborate, negotiate, communicate, and interact allowing their users to take advantage of values such as democratic participation, collaboration, collective intelligence and knowledge sharing on a massive scale beyond geographical barriers. All these values are extremely useful in social decision making processes which consist on the extraction and aggregation of individuals' information to generate a global solution. Therefore Web 2.0 communities are considered as very powerful tools for decision support systems. This enormous on-line collective provides two potential benefits: Firstly, such a large, dispersed population captures statistical collective intelligence which leads to the knowledge generation through the weighted averaging of independent, individual judgements; and secondly, some systems benefit from the ability to amplify expertise. That is , if each individual in a collective

V. Torra et al. (Eds.): MDAI 2013, LNAI 8234, pp. 82–93, 2013.

is more likely than not to be correct, then as the size of the group scales, the probability of the collective decision being correct moves toward certainty [5]. From the point of view of the industry many believe that understanding these new applications and technologies and using their benefits early will stand organizations in good stead to greatly improve internal business and decision making processes.

In this paper we analyse how the Web 2.0 communities are used to improve collaboration and decision making in the enterprise contexts. So, the paper is set out as follows. Section 2 describes how the most outstanding Web 2.0 technologies have become very powerful tools to support decision making systems. Section 3 shows the utilization of the SMT in enterprise contexts, discussing its advantages and drawbacks. In Section 4 we present current trends and challenges of the utilization of SMT technologies in collaborative environments. And finally, Section 5 points out our conclusions.

2 Web 2.0 Communities

New Web 2.0 technologies have provided a new framework in which virtual communities can be created in order to collaborate, communicate, share information, resources and so on. This very recent kind of communities aggregates the collective intelligence of their users existing on the Web to extract information such as behaviours, opinions, popularity, trends, knowledge and customs [16]. Particularly, some of the most common on-line Web 2.0 communities are:

1. **Folksonomy** is a tool for information retrieval which connects users to resources via tags. A tag can be seen as an interpretation that a user makes about a particular resource. Folksonomies are generated indicating the popularity of a particular term to describe a particular resource. Folksonomies such as del.icio.us provide the user with a personalized view of the emergent structure of the Web and the user' self interest improves its ability to do the same for others. Another example of folksonmy is CiteULike, a social bookmarking site for academic context which organizes users' favourite papers into a personal library that any other user can consult. Thus, every user's library serves as that user's bookmarks as well as an impersonal recommendation list for other users who have liked one or more resources in that library.
2. **Recommender systems** manage information overload by acting as a search function to provide a personalized subset of the total collection. They are personalized because they track each user behavior, pages viewed, purchases, and ratings to come up with recommendations. Most recommender systems rely on an item-item algorithm, which calculates the distance between each pair of items according to how closely users who have rated them agree. Distances between pairs of items are usually based on the ratings of thousands or millions of users, so they tend to be relatively stable over time. Some popular recommender systems are: Amazon which offers personalized suggestions to

their on line shoppers, Netflix which suggests videos to watch, Facebook's friend suggestions, Last.fm which ic a popular music website based in the United Kingdom, and Pandora which builds personalized music streams.

3. **Discussion forums** represent Web online discussion communities where users share information or discuss about selected topics. In many of these communities some simple group decision making schemes, as referendum or voting systems are usually used. For example, services like PollDaddy allow to create online surveys and polls where users can vote about the best alternative to choose for a given decision problem. Moreover Smartocracy [13] is a social software system for collective decision making. The system is composed of a social network that links individuals using trust degrees and allows to make good decisions and a decision network that links individuals to their voted-on solutions.
4. **Wiki** is a server software that allows users to freely create and edit Web page content using any Web browser. It is a highly distributed way to gather, create, and share knowledge. Its main purpose is to capture the collective knowledge held by participants such that the resulting documents transcend the abilities of individual contributors. The result is a network of collaboratively generated documents that contains the authorial wisdom of all its contributors. Wikis provide a new way of solving problems based on a transformation on the way the knowledge is generated, shared and stored. There are no any automatic mechanism to leverage Wikis in decision making environments. However their capability of organizing and storing information from multiple users in a dynamic way converts them in a excellent complementary tool to support decision making processes. The most outstanding example of a Siki based system is the online encyclopedia Wikipedia. In [1] a consensus model designed for Web 2.0 communities and its application in Wikipedia were presented. This model is aimed to minimize the main problems that this type of communities present such as low and intermittent participation rates, difficulty of establishing trust relations and so on. This model includes some delegation and feedback mechanisms to improve the speed of the process and its convergence towards a solution of consensus.
5. **Social networks** are the major achievements of the Web 2.0 technologies. They are Web sites where people create their own virtual spaces (or home page), on which they post pictures, write blogs, share ideas, and link to other Web locations which they find interesting. As a result, they form online communities comprised of people who share similar interests. There is a wide range of social networks with different targets, from Web sites to share personal information with friends, to places where you can expose your professional capabilities, or even places to share your opinions about your trips.

3 Enterprise 2.0

It is widely known that incorporating social business is becoming imperative to improve customer communication and engagement, build loyal partner networks

and improve internal collaboration. Enterprise 2.0 refers to the deployment of Web-based social software tools and services, such as Wikis, blogs, forums, RSS feeds, opinion polls, community chats and social networking, to facilitate enterprise collaboration. It includes social and networked modifications to corporate Intranets and other classic software platforms used by large companies to organize their communication.

3.1 Using Popular Web 2.0 Tools in the Business Context

Most popular social networks can facilitate knowledge management and transfer in complex, dynamic enterprise environments by developing new relationships between colleagues of the firm or from other firms, advertising new products as well as attracting prospective clients [12]. Some examples are:

1. **Facebook:** Using Facebook in the organizational environment leads to establish relationships with colleagues inside across the firm and outside the firm providing a way of expertise sourcing an sharing. It offers the possibility of advertise products, spreads the employees' network and attracts prospective customers or clients.
2. **Twitter:** In an enterprise context, Twitter is useful for employees to share expertise, post progress updates and rapidly disseminate information. It is interesting to note the rise of Twitter in all types of enterprise social networking.
3. **LinkedIn:** It is a professional oriented social networking site that allows users to share expertise and gain new insights from discussions with like-minded professionals in private groups. In many companies LinkedIn is used both to recruit talents and identify sales leads. For example, IBM provides knowledge sharing via LinkedIn answers and its own social network.
4. **Blogs:** A blog is as a web based journal authored by one of multiple writers, which serves as a platform to articulate thoughts , feelings, ideas observations and issues of relevance. Readers can contribute responding to posts as comments. Blogs sparks conversation and debate and enables to share knowledge and information.
5. **RSS feeds:** They provide a channel for subscribing to content sharing common social tags. That enables visibility of content and allows information providers to syndicate their content.
6. **Google aps (Google Docs and Google Groups):** Google Docs allows users to create word-processing, spreadsheet and presentation applications that are Web-hosted and can be remotely accessed by any authorized user. These documents can be edited simultaneously by multiple users. On the other hand, Google groups allow an extension of Google Docs into collaboration space where users can create, share, and work on documents as well as start discussions, upload multi-media files and manage content.
7. **Wiki:** Wiki technology is increasingly being used in corporate environments to facilitate a variety of organisational tasks that include the codification of organisational knowledge and the formulation of corporate communities

of practice, as well as more specific processes such as the development of collaborative information systems, the interactions of the enterprise with third parties, management activities and organisational response in crisis situations [11].

3.2 Enterprise SMT Based Tools

While social networking's success among consumers is well-known, enterprise social media tools are still struggling to gain a place in organizations. However companies are starting to recognize the potential value that enterprise social media technology can deliver, particularly around departmental and cross-department collaboration [8]. Enterprise social media technology adapts and combines features such as employee profiles, activity streams, microblogging, discussion forums, Wikis, groups of friends, tagging, rating and reviewing of content for workplace use with the primary goals of better connecting members of an organization and promoting knowledge-sharing between different employees and departments.

Although Facebook and LinkedIn have avoided tailoring their products for corporate use, there is a wide range of tools supporting enterprise collaboration which goes from point solutions like Yammer and Socialcast to SaaS-based solutions like Salesforce.com's or Chatter, and solutions from well known companies like Microsoft, IBM, and Cisco. In the Gartner, Inc.'s 2012 Magic Quadrant for Social Software in the Workplace we can find 21 ESN Vendors classified as niche players, challengers, visionaries and leaders. Among them we can highlight:

1. **Cisco's WebEx Social**: This tool provides a secure, business-focused Facebook-like experience compatible with other Cisco's communications platforms, such as WebEx conferencing, Jabber and Cisco Unified Communications Manager. This tool is mainly focused on networking, employees can follow one another, and finding an expert in an area becomes as simple as a Google search.
2. **Microsoft's Yammer and SharePoint:** On the one hand, SharePoint is a repository of business documents and institutional knowledge. Files can be uploaded, shared, archived and edited, while Wikis and discussion threads can help capture conversations for posterity. On the other hand, Yammer covers real-time interactions with a series of mobile and Web applications that combine the simplicity of Twitter with more extensive features, such as organizational chart mapping, polls and groups. Microsoft is working on the integration of both platforms.
3. **IBM connections:** It is a secure social software platform that helps employees engage with networks of expertise, and integrates business processes. Users can quickly set up their own profiles, create and manage groups and share files, status updates and wiki pages. Users can access this platform everywhere from desktop or mobile devices. Over time, connections become an expertise repository. That is, it allows users seeking out and finding the answer to their questions or else quickly discovering who might have the

answer based on profiles or past discussions. The key capabilities of these platform include Activity stream with the most relevant events on the user's network and social analytics for connection components which provides new trends in content, social activity and expertise for better decision-making. It also provides a team oriented platform to keep in touch all the members in a project including also the main tasks and milestones for the project.

4. **Socialtext:** It is a social software which provides the employees with facilities to create, share and manage content, and effectively collaborate within their enterprise. It offers capabilities to generate and edit content, such as blogs, wikis, activities, etc., and automates the ability to create pages and track their progress along the way. Moreover it helps employees to find the most relevant people in their network to connect and collaborate with. Socialtext makes it easy to integrate with other enterprise tools such as CRM (Customer relationship management), ERP (Enterprise resource management), HR and content management systems.
5. **Jive:** It is a social business platform which enables people to connect, collaborate and communicate from anywhere. From the point of view of the employees inside companies Jive provides collaborative employees' networks. Externally, it supports customer communities to improve service, support and customer satisfaction. This platform also encourages the engagement and participation by using built-in game mechanics and rewards. Moreover it provides on-line support to quickly capture and share new ideas by brainstorming including voting and rating mechanisms and includes task planning tools.
6. **Salesforce.com:** It is a cloud based software which provides a Customer Relation Management platform as well as engage clients, employees and sales representatives on a social network. It includes a social networking plug-in that enables the user to join the conversation about their company on social networking Web sites, provides analytical tools and other services including email, chat, and accesses to customers' entitlement and contracts. This solution is comprised of several tools: Sales, Service Cloud, Data Cloud (including Jigsaw), Collaboration Cloud (including Chatter) and Custom Cloud. The sales cloud is a real time collaborative tool which enables users to control all the relevant information related to the company's sales process. It is designed to manage marketing campaign spending and performance across a variety of channels from a single application, tracks opportunity-related data including milestones, decision makers and customer communications. Chatter is a real-time collaboration platform which provides the users with updates via a real-time news stream. Users can also form groups and post messages on each other's profiles to collaborate on projects.

Table 1 summarizes the main characteristics of the tools explained above.

3.3 Advantages of Enterprise 2.0

Using SMT in the companies context exhibites substantial benefits which range from the way in which the information is spread and shared to the new way

Table 1. Enterprise 2.0 Tools

Tool	Main characteristics
Cisco's WebEx social	Facebook like experience
	Focus on networking
Microsoft's Yammer	Real time communications
	Mobile version
	Microbloging
	Support for groups collaboration
	Polling system
Microsoft SharePoint	On line repository of business document
	wiki
	discussion threads
IBM connections	Social software platform
	users' profile
	document sharing
	Activity stream
	team support
	social analytics
	wiki
Social text	facilities to create, share and manage content
	user's profile
	expert finding
Jive	provides collaborative employees' networks
	supports customer communities
	Encourages the engagement using built-in game mechanics and rewards.
	On line support to brainstorming
	Polling system
	tasks planning tools
Salesforce.com	Cloud based solution
	Social network for employees,clients,and sales representatives
	Analytical tools

in which the problems are solved taking advantage of the collective intelligence and fostering the mass collaboration [15]. In the following subsections these advantages are analysed posing some real examples of companies which leverage Web based collaboration tools.

1. Improving Communication, Collaboration, and Advertising: Among the various possible outcomes of incorporating SMT is the possibility to easily share information within different departments of an organisation allowing a constant stream of user defined data and developing an ambient awareness of other's behaviours as well as increasing the potential discovery of knowledge from previously unconnected sources. SMT also makes possible the exchange of information outside the organizational boundaries, with organizations and institutions that have a previous relationship with the company, offering a mechanism whereby contractors can develop and maintain relationships and share knowledge and information beyond the exact terms of the service agreement regardless of the affiliation or geographical dispersion. In such a way, companies can obtain customers' feedback.

These ways of communication are also exploited to advertise products and attract new clients, and therefore, reducing the advertising cost while targeting a bigger audience. In such a way, Social networking sites give businesses a fantastic opportunity to widen their circle of contacts allowing organizations to reach out and select groups or individuals and target them and their network of contacts personally boosting the companies reputation. Some examples of the corporate use of social networks for marketing activities can be found in [17,9].

Companies can use Wikis to supplement regular collaboration tools within its global teams and cross department collaboration. Blogs for communication and sharing within the members of the company but also with the clients and RSS feeds for news and business information dissemination. Using forums and discussion groups firms can also obtain feedback from their customers about their products and identify what should they improve or even get ideas about new products. In a recent survey about using SMT in the working environment majority of the respondents agree on that social communities could improve collaboration project work helping employees to get quicker answer to their questions and to easily find experts on relevant topics. Moreover they felt that the use of SMT could help with information overload by lowering the amount of email traffic, diverting instead to more open approaches of communication.

2. Support to Decision Making: SMT based tools can provide support for managerial decision making through analysis of the data collected in social networks. Typical examples include identifying key performers, locating experts, soliciting ideas, developing possible solutions to complex problems (e.g., using the answer functions on LinkedIn), and analyzing managerial connection networks to facilitate succession planning. Furthermore, some data from SMT are analysed by firms using data mining and machine learning procedures to track behaviours and explore new trends to recognize current problems and develop

new products. Some examples of companies which use SMT for managerial issues are: (i) Deloitte Touche Tohmatsu's social network (D Street), which was established to assist the company's human resource management team in downsizing and regrouping, building networks of experts, and retaining talents [3]; (ii) Hoover Inc. has established a social network that makes use of Visible Path's relationship management technology to identify target business users to build relationships and discover ways to reach specific users; (iii) Ypodimatopoulos et al. presented in [18] a problem-solving application for discovering expertise by leveraging the professional social network of its employees.

3. Training and Learning: Some companies employ virtual worlds, for training purposes since they allow training via virtual simulation. For example, Cisco makes use of Second Life on its virtual campus for product training and executive briefings, and IBM offers training exercises to its field service teams through the simulation of project management and customer interaction in virtual worlds. Not only enterprises but also other well-known institutions such universities leverage social technologies to develop virtual campus in which students and professors can collaborate and share information.

4. Knowledge Management: These applications involve employee-driven activities such as knowledge discovery, idea creation, maintenance, sharing, transfer, and dissemination. Areas of application include the discovery of experts and the mapping of communities of expertise. These large-scale activities are known as *crowdsourcing*, *collective intelligence, mass collaboration,* and the *power of the crowd* [10]. A good example is innocentive.com, a social network that attracts the participation of a huge community of scientists to solve science-related problems, usually for a cash reward.

Furthermore, many companies have created retiree corporate social networks to take advantage of their knowledge and expertise. These former employees possess huge amounts of knowledge that can be used for productivity increases and problem solving.

3.4 Risk of Using Social Media Technology in the Enterprise 2.0

The threats and the exposure to security and business risks, arising from careless employees engaging in online communities is now an issue of great relevance for enterprises: Employees can disclose not only their personal information but also confidential business data. In this context, enterprises face not only productivity loss due to employee spending time using SMT; a greater concern is the possible threat of information leakage caused by incautious posts or explicit references to private business information [14]. The audience of SNs is so broad that besides customers, business competitors, and partners, also hackers may access such information, potentially gaining competitive advantages and causing the targeted enterprise financial losses, both in the short and long term. The risk of attackers

exploiting SMT data warehouses is on the rise, due also to the tools available to them (e.g. data aggregator and data mining tools). In table 2 we summarize the main pros and cons of using SMT in the work place.

Table 2. Pros and concerns of using Enterprise 2.0 tools

Pros	Cons
Expanding market research	Security risks
Low cost marketing	Viruses and malware
Expertise source	Low productivity
More eficient communications inside and outside the firm	High investment in social software
On line training and learning	Reputation and legal liability
Sharing expertise	Information leackage
Better organization of knowledge	Employees reluctance to use SMT
Faster way to attract new clients	
Easily select groups or individuals and target them personally	
Easily way to keep up to date	
Improving collaboration	

4 Trends and Future Work

We are at the very beginning of the utilization of SMT technologies in collaborative environments such as the enterprise. Therefore, there are still many potential research issues. Most of them are especially aimed to bridge the gap between big data and big strategy, that is to take advantage of the massive volume and detail of information captured by enterprises, the rise of multimedia, social media, and the Internet of Things.

4.1 Development of New Tools to Support Computer Assisted Decision Making Adapted to the Web 2.0 Communities

We identify as a key challenge the development of new tools and algorithms to support computer assisted decision making taking advantage of web 2.0 technologies, and integrate these tools in the already existing platforms. These tools should be able to provide better "participation architectures" that allow sharing data, trusting user as co-developers, harnessing collective intelligence, etc... as well as overcome some of the inherent problems of the Web 2.0 communities such us large user base, heterogeneity in the users' background, low and intermittent participation rates, the dynamism of the Web 2.0 frameworks and difficulty of establishing trust relations. Moreover some validation tools must be developed to assess the quality of the decision process and the validity of the obtained results.

4.2 Expertise Seeking Using Web 2.0 Technologies

A a key aspect in many business and engineering contexts is the appropriate selection of the experts and the way their opinions are aggregated depending on

their background, expertise and the quality of the opinions provided. In many big companies they have a huge staff world wide and sometimes, they struggle to find the most suitable experts to be integrated in a work team or take part in a decision making process. Therefore, new tools to find experts using companies' staff data-bases or web-based scientific- academical networks such as Google Schoolar, the Web of Knowledge or DBLP need to be developed. In such a way, the selection of experts whose background really fits the problem to be solved would be ensured.

4.3 Using Big Data to Improve the Decision Making Process

Many decision making problems require gathering and analysing information to define the problem and identify possible solutions. To do so, it is required to obtain the opinion of people such as clients, prospective users or even knowing the general opinions and trends. Traditionally, to get clients information companies use polls. However it is necessary to develop more sophisticated on line polling systems in which the specific profile of each survey respondent would be taken into account to aggregate the results.

Another effective way to gather information is to extract the knowledge from corporate databases or from the Web 2.0 communities, such as Internet forums, groups of bloggers, social network services, etc, which provides a platform in which users can collectively contribute and also generate massive content. Such an enormous amount of data it is widely known as 'Big Data'. One of the most straightforward manners of taking advantage of Big Data, is by creating transparency, that is, organizing raw data in such a way that making them more accessible and easy to understand in a timely manner. For example, by clustering similar objects, by showing the evolution of certain features along the time, or even by identifying possible trends. Properly understanding entire datasets could also lead to substantially improve the decision making process [19]. To do that, new ways of organizing and extract knowledge from Big Data need to be developed. These tools would provide the experts with a high level insight about a huge entire dataset helping them to make the most appropriate decision minimizing the risks.

5 Conclusions

We have analysed how the Web 2.0 tools can be used to improve cooperation and social decision making in the enterprise context. We have presented the most popular Web 2.0 technologies and shown their application in enterprise domains. We have also analysed the concept of Enterprise 2.0, discussing its characteristics. We have explained which are the main social tools specially designed to support the Enterprise 2.0. Finally, some current trends, open questions and prospects in the topic have been pointed out.

Acknowledgments. This work has been developed with the financing of the Andalusian Excellence Projects TIC-05299 and TIC-5991, the FEDER funds in Project TIN2010-17876.

References

1. Alonso, S., Pérez, I.J., Cabrerizo, F.J., Herrera-Viedma, E.: A linguistic consensus model for web 2.0 communities. Applied Soft Computing 13, 149–157 (2013)
2. Bjelland, O.M., Wood, R.C.: An Inside View of IBMs 'Innovation Jam. MIT Sloan Management Review (2008)
3. Brandel, M.: The new employee connection: Social networking behind the firewall (2008)
4. Brynjolfsson, E., McAfee, A.: Beyond enterprise 2.0. MIT Sloan Management Review (2007)
5. Condorcet, M.: Essai sur l'Application de l'Analyse a la Probabilité des Décisions Rendues á la Pluralité des Voix. Imprimerie Royale, Paris (1785)
6. Gibson, S.: Web 2.0 tools gain enterprise acceptance (2009)
7. Lai, L., Turban, E.: Groups formation and operations in the web 2.0 environment and social networks. Group Decision and Negotiation 17, 387–402 (2008)
8. Li, C.: Making the business case for enterprise social networks, ALTIMETER REPORT (February 2012)
9. Li, C., Bernoff, J.: Groundswell: Winning in a World Transformed by Social Technologies. Harvard Business School Press, Boston (2008)
10. Libert, B., Spector, J.: We Are Smarter Than Me: How to Unleash the Power of Crowds in Your Business. Wharton School Publishing (2007)
11. Lykourentzou, I., Dagka, F., Papadaki, K., Lepouras, G., Vassilakis, C.: Wikis in enterprise settings: a survey. Enterprise Information Systems 6, 1–53 (2012)
12. Murphy, G.D.: Using web 2.0 tools to facilitate knowledge transfer in complex organisational environments: A primer. In: ICOMS Asset Management Conference (ICOMS 2010), June 21-25. University of Adelaide, South Australia (2010)
13. Rodriguez, M.A., Steinbock, D.J., Watkins, J.H., Gershenson, C., Bollen, J., Grey, V., de Graf, B.: Smartocracy: Social networks for collective decision making. In: 40th Annual Hawaii International Conference on Systems Science (HICSS 2007), Waikoloa (2007)
14. Squicciarini, A., Rajasekaran, S.D., Mont, M.C.: Using modeling and simulation to evaluate enterprises' risk exposure to social networks. Computer 44, 66–73 (2011)
15. Turban, E., Bolloju, N., Liang, T.P.: Enterprise social networking: Opportunities, adoption, and risk mitigation. J. Org. Computing and E. Commerce 21, 202–220 (2011)
16. Watkins, J., Rodriguez, M.: A survey of web-based collective decision making systems. In: Nayak, R., Ichalkaranje, N., Jain, L. (eds.) Evolution of the Web in Artificial Intelligence Environments, vol. 130, pp. 243–277. Springer, Heidelberg (2008)
17. Weber, L.: Marketing to the Social Web: How Digital Customer Communities Build Your Business. Wiley, Hoboken (2009)
18. Ypodimatopoulos, P., Vukovic, M., Laredo, J., Rajagopal, S.: Server hunt: Using enterprise social networks for knowledge discovery in it inventory management. In: 6th World Congress on Services (2010)
19. Frank, C.: Improving decision making in the world of big data (2012)

Using the Logarithmic Generator Function in the Spoken Term Detection Task*

Gábor Gosztolya

MTA-SZTE Research Group on Artificial Intelligence
of the Hungarian Academy of Sciences and University of Szeged
H-6720 Szeged, Tisza Lajos krt. 103., Hungary
ggabor@inf.u-szeged.hu

Abstract. Spoken term detection is a task in artificial intelligence where user-entered keywords are to be looked for in a huge audio database. In one common approach the recordings are first converted into phoneme-sequences, and the actual search is performed in this space. During search, instead of performing the default multiplication of basic phoneme operation probabilities, applying a triangular norm can significantly improve system accuracy. We used an application-oriented method for triangular norm representation and tuning, namely the logarithmic generator function. In practice this proved to be quite successful and led to a relative error reduction score of 16%.

Keywords: triangular norms, additive generator function, artificial intelligence, speech processing, spoken term detection, keyword spotting.

1 Introduction

Among the range of fuzzy functions, *triangular norms* (or *t-norms*) [7] have a significant number of successful applications in the literature, especially in artificial intelligence (AI) problems such as image enhancement [4], image blending [13], classifier combination [3], speech recognition [9], and multimodal biometrics [14]. What is common among these AI tasks, and what makes them a good area for applying t-norms, is that they usually rely on aggregating lower-level probabilities (outputs of single classifiers, phoneme probabilities of short excerpts of speech, confidence scores of different biometrical identifier systems etc.). The standard approach for this aggregation is to simply calculate the product of these individual probability values (*naive Bayes* approach), relying on the assumption that these components are independent. While this assumption leads to a nice and elegant mathematical formulation and also behaves well in practice, in most cases it is clearly false, which calls for the use of other operators. However, these

* This publication is supported by the European Union and co-funded by the European Social Fund. Project title: Telemedicine-focused research activities in the fields of mathematics, informatics and medical sciences. Project number: TÁMOP-4.2.2.A-11/1/KONV-2012-0073.

V. Torra et al. (Eds.): MDAI 2013, LNAI 8234, pp. 94–104, 2013.

operators still have to express an AND-like relation of the arguments. Triangular norms are just the conjunctive operators of fuzzy logics, and they are an ideal choice for such tasks.

In this paper we will focus on an AI task, that of spoken term detection (STD, sometimes also referred to as *keyword spotting* or KWS), which is a quite recent topic within speech technology. It seeks to provide a way to search for user-entered keywords in a huge archive of audio recordings. Recent approaches [27,28] view this task in a dictionary-independent way, where search is performed only by relying on the acoustic model and using only general language information (e.g. probability values of consecutive phoneme pairs or triplets). This rules out the approach of simply performing automatic speech recognition (ASR [25]) on the recordings, storing the resulting *word sequence*, and performing a text search in this textual representation, since this approach prevent users from finding words (usually proper nouns) which were not present in the dictionary used in the speech recognition step.

One common approach in STD is to represent the recordings as mere phoneme-sequences, to which the phonemes of the search term are matched one by one. The overall probability of such a phoneme-sequence pairing is usually computed as the product of the individual probability values. In this paper we experimented with triangular norms when performing this aggregation; among the wide range of possible t-norms we chose an application-oriented representation.

In this paper we will describe the STD problem, focusing on the approach using phoneme-sequences. Then we will describe the t-norm representation chosen (the logarithmic generator function), present the test results, and finally analyze them.

2 The Spoken Term Detection Task

In the spoken term detection task we seek to find the user-entered natural language expressions (*terms* or *keywords*) in an audio database (the set of *recordings*). An STD method returns a list of *hits*, each of which contains the point of occurrence, the term found, and a probability value that can be used to rank the hits. In contrast to other information retrieval tasks, in STD the order of the hits does not matter; the probability value of the returned hits is only used to filter the hit list further by using a decision threshold, keeping just the more probable elements.

In STD, a user expects a quick response for his input, thus we have to scan hours of recordings in a few seconds (or less). To achieve this, the task is usually separated into two distinct parts. In the first one, steps requiring intensive computation are performed without knowing the actual search term, resulting in some intermediate representation. Then, when the user enters the keyword(s), a (quick) search is performed in this representation. We will focus on the approach where the intermediate representation is the most probable phoneme sequence, since it permits a very quick search while still retaining good accuracy [20].

The most probable phoneme sequence for each recording is usually generated by some standard speech recognition technique. Then, for a term w with the phonemes $w_1, w_2, \ldots, w_n$, we look for all the non-overlapping phoneme sequences (L) for which

$$P(w|L) \geq P_{\min}, \tag{1}$$

where $P_{\min}$ is a threshold set previously. Making the standard assumption that the successive phonemes are independent, we get

$$P(w|L) = \prod_{i=1}^{n} P(w_i|l_i) \geq P_{\min}, \tag{2}$$

where $l_1, \ldots, l_n$ are the phonemes of the phoneme sequence L. To compensate for errors in the phoneme sequence representations, phoneme insertions, deletions and substitutions are allowed. This means that w_i or l_i can be empty (λ), so

$$P(w|L) = \prod_{i=1}^{m} P(w_i|l_i) \geq P_{\min}, \tag{3}$$

where by omitting the $w_i = \lambda$ values from the sequence $w_1, \ldots, w_m$ we get the term w, and without the $l_i = \lambda$ values $l_1, \ldots, l_m$ forms L. $P(w_i|\lambda)$ represents the probability of deleting phoneme w_i (if $w_i \neq \lambda$), $P(\lambda|l_i)$ means the probability of inserting phoneme l_i (if $l_i \neq \lambda$), while $P(w_i|l_i)$ is the probability of substituting w_i for l_i in the case where neither w_i nor l_i is λ (but it may be that $w_i = l_i$). The optimal pairs can be found by calculating the edit (or Levenshtein) distance [22]. The probability values of the phoneme operations can be computed from the errors of the phoneme recognizer: after performing phoneme classification on recordings with known real phonetic transcriptions, the probability values of phoneme insertions, deletions and substitutions can be readily calculated by comparing the resulting phoneme sequences to ground truth ones (i.e. from the confusion matrix [21,10]).

Note that in equations (2) and (3) we made the assumption that the consecutive phonemes are independent, which allowed us to decompose $P(w|L)$ into a product of lower-level probability values. This assumption is clearly false owing to the continuous motion of the vocal chords, the tongue and the mouth [31], so we can replace product with other operators as long as they behave well in practice. As triangular norms also represent AND-like relations of values in the range $[0, 1]$, which is just what we need for combining probability values of phoneme operations (insertions, deletions and substitutions), we may expect them to work well in this task. Furthermore, several norms (e.g. [5,26,1,6]) have one or more parameters, allowing us to fine-tune them to the actual problem. For these reasons, we will apply t-norms in the STD task.

3 The Logarithmic Generator Function

One advantage of using triangular norms is their tunability: they can be adapted to the requirements of the given problem. With respect to this, however, there

could be great differences among various t-norm families depending on how the range of triangular norms they contain matches the ideal performance needed for our actual application [11]. On the basis of our earlier findings [9,12] it is usually better to concentrate on the additive generator function f [26,17], since we have plenty of room to adjust it to suit our actual needs. This can be viewed as triangular norm construction [18,8], with respect to the criterion that the applied triangular norm representation must be easy to handle.

Recall that a strict, continuous and Archimedean triangular norm T can be written in the form

$$T(x, y) = f^{-1}\big(f(x) + f(y)\big), \tag{4}$$

where f is the *additive generator* of T, and it is a continuous, strictly decreasing function on the interval $[0, 1]$; $f(0) = \infty$ and $f(1) = 0$. Moreover, for a given T, f is unique up to a scalar factor, so the triangular norm applied can also be represented by its generator function. If we could find a suitable way to model this function f, we could fine-tune its behaviour to suit our needs. To achieve an optimal performance we have to find a flexible yet simple representation, preferably one which is application-oriented.

The additive generator is widely examined in the literature (e.g. [19,23,18]). However, for an actual application we need an application-oriented approach instead of a theory-oriented solution, as we have to pay attention also to computer arithmetics (like the ability of avoiding underflowing, being able to easily handle values in a different order of magnitude, etc.). Due to these reasons we chose the logarithmic generator function for triangular norm representation, which we will describe next.

3.1 The Logarithmic Generator Function

To understand the logic of the logarithmic generator function [12], we should first consider its application context. In a typical case we have a number of probability estimates as input $(p_1, p_2, \ldots, p_k)$, and a t-norm T; and we need to calculate

$$T^k(p_1, p_2, \ldots, p_k) = T(\ldots T(T(p_1, p_2), p_3), \ldots, p_k). \tag{5}$$

Now using the transcript $T(x, y) = f^{-1}(f(x) + f(y))$ we have that

$$T^k(p_1, p_2, \ldots, p_k) = f^{-1}\Big(\sum_{i=1}^{k} f(p_i)\Big). \tag{6}$$

In our environment, and in most artificial intelligence tasks, to avoid numerical underflowing, instead of a probability value p we use the cost value $c = -\log p$. This step also implies that we use cost addition instead of probability multiplication, and perform (aggregated) cost minimization instead of probability maximization. The triangular norms, however, work only on probability values. To overcome this difficulty, first we incorporate this conversion into Eq. (6), i.e. we will use

$$-\log\Big(f^{-1}\Big(\sum_{i=1}^{k} f(e^{-c_i})\Big)\Big). \tag{7}$$

It is straightforward to include the calculation of the negative exponential into f; hence, the logarithmic generator function is defined as

$$\phi(x) = f(e^{-x}). \tag{8}$$

Now we can write

$$\phi^{-1}\Big(\sum_{i=1}^{k} \phi(c_i)\Big) = -\log\Big(f^{-1}\Big(\sum_{i=1}^{k} f(e^{-c_i})\Big)\Big) \tag{9}$$

$$= -\log T(e^{-c_1}, e^{-c_2}, \ldots, e^{-c_k}), \tag{10}$$

so using the logarithmic generator function $\phi(x)$ in exactly the same way as we used the additive generator function $f(x)$ will lead to a calculation of the same triangular norm T, only with the corresponding cost values instead of the probabilities both as arguments and as the result. As $f(x) : [0, 1] \to [0, \infty)$ was a strictly decreasing function with $f(1) = 0$, the logarithmic generator function $\phi(x) : [0, \infty] \to [0, \infty]$ is strictly increasing, and $\phi(0) = 0$. The additive generator function is unique up to a multiplicative constant for any given T t-norm, so the same is true for the logarithmic generator function.

3.2 Representing the Logarithmic Generator Function

Now we will turn to modeling this logarithmic generator function. Almost any representation could be used for this task; we chose to model it with a piecewise linear one for two basic reasons. First, it is quite simple to handle: both ϕ and ϕ^{-1} can be implemented very easily. Second, it is a very flexible representation: the family of all strict t-norms with a piecewise linear logarithmic generator $\phi : [0, \infty] \to [0, \infty]$ with finitely many breakpoints, such that $\lim_{x\to\infty} \phi'(x) = 1$, is dense in the family of all strict t-norms with respect to the topology of uniform convergence. A proof of this just involves a standard compactness argument.

Henceforth let $\phi = \phi_{a_1,\ldots,a_N}^{m_1,\ldots,m_N} : [0, \infty] \to [0, \infty]$ be the piecewise linear, strictly increasing function with break points on the domain as $0 = a_0 < a_1 < \ldots < a_N < a_{N+1} = \infty$ and with positive steepness values $m_1 < \ldots < m_N$, respectively, and $m_{N+1} = \lim_{x\to\infty} \phi'(x) = 1$. That is,

$$\phi(x) = (x - a_j)m_{j+1} + \sum_{i=1}^{j} (a_j - a_{j-1})m_j, \qquad a_j \le x < a_{j+1}. \tag{11}$$

If the a_i control points are fixed, ϕ can be described by a vector of N steepness values, making it easy to optimize. Furthermore, the function ϕ is unique up to a positive multiplicative constant; by setting m_{N+1} to 1, we fix exactly one of these equivalent representations. The actual function f (and hence, the triangular norm T) can be easily calculated from ϕ, being a piecewise exponential function with $N + 1$ negative exponents. It will be continuous, but not smooth (except when $m_1 = \ldots = m_N = 1$, which is just the product case).

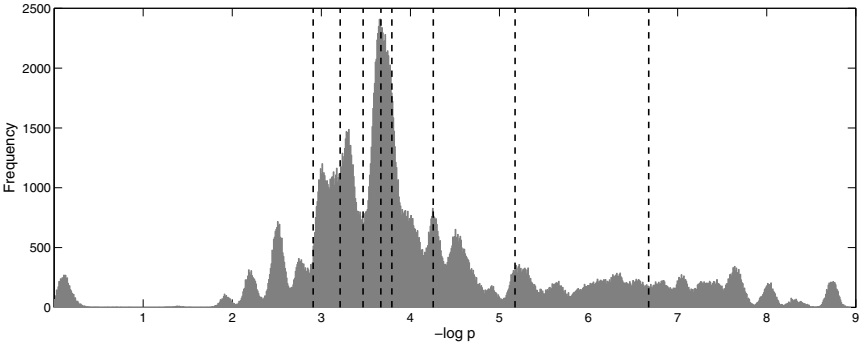


Fig. 1. A (smoothed) histogram of the $-\log p$ values encountered during performing STD, and the suggested positioning of the $N = 8$ control points

This way, by keeping all the a_j values fixed, this problem can be simplified to that of a maximization task in an N-dimensional space: we seek to maximize the accuracy of the spoken term detection system as a function of $\mathbf{m}$. As for the choice of the control points, we have the possibility to set them at values where they represent our problem as accurately as possible. Since it is also nontrivial, next we will present a method for control point assignment.

3.3 The Choice of Control Points

Optimizing the logarithmic generator function means performing a search in an N-dimensional vector space. To aid this search process we should avoid the presence of irrelevant or redundant dimensions, so we should try to give each one the same importance. The main idea behind the general method introduced for this purpose in [12] is to create statistics of the values occurring during use, i.e. note which x and y values are passed to the $T(x, y)$ operator (and thus to the generator function f). Owing to the commutative property we do not need to distinguish between the two arguments x and y. Next, we calculate a histogram of the $-\log$ of recorded values: for each value we note how many times it appears. Afterwards, we divide this histogram into $N + 1$ parts with equal-sized areas: the control points will be the borders between these regions (see Fig. 1). This way about the same number of evaluations will fall into each region between two adjacent control points, making each steepness value (roughly) equally important. An advantage of this method is that it is quite general regarding the actual task, since it requires only a statistic of appearing cost values, and it also has only parameter (N).

Now we have presented the logarithmic generator function, which allows us to represent and fine-tune a triangular norm in an application-oriented manner. We have also described a general methodology to fit it into a given problem by positioning the a_i control points. Next, we will focus on the actual application.

4 Experiments and Results

Having defined the problem and the logarithmic generator function, we turn to the testing part: we introduce the evaluation methodology, the testing environment and the way of testing, then present and analyze the test results.

4.1 The Evaluation Metrics

A Spoken Term Detection system returns a list of hits for a query. Given the correct list of hits, we should rate the performance of the system to compare different configurations. In STD, instead of standard information retrieval metrics such as precision (the ratio of correct hits found to the hits returned) and recall (the ratio of correct hits found to all the correct hits), usually some other, albeit similar measures are used. Here, we will mainly use the Actual Term-Weighted Value (ATWV) [24], which is defined as

$$ATWV = 1 - \frac{1}{T}\sum_{t=1}^{T}\big(P_{Miss}(t) + \beta P_{FA}(t)\big), \tag{12}$$

where T is the number of terms, $P_{Miss}(t)$ is the probability of missing the term t (in fact, the opposite of recall for the term t) and $P_{FA}(t)$ is the probability of getting a false alarm. These values are defined as

$$P_{Miss}(t) = 1 - \frac{N_C(t)}{N_T(t)} \quad \text{and} \quad P_{FA}(t) = 1 - \frac{N_{FA}(t)}{T_{speech} - N_T(t)}, \tag{13}$$

where $N_C(t)$ is the number of correct hits returned, $N_{FA}(t)$ is the number of false alarms, $N_T(t)$ is the total number of real occurrences of term t, and T_{speech} is the duration of recordings in seconds. Usually the penalty factor for false alarms (β) is set to 1000. A system achieving perfect detection (having precision and recall scores of 100%) has an ATWV score of 100%; a system returning no hits has a score of 0%; while a system which finds all occurrences, but produces 3.6 false alarms for each term and speech hour also has a score of 0% [24]. An older and more permissive metric is the Figure-of-Merit (FOM), which is the mean of recall scores when we allow only $1, 2, \ldots 10$ false alarms per hour per keyword.

Note that although ATWV uses all the hits returned, a threshold value was still used, namely $P_{\min}$ from Eq. (3). A carelessly chosen threshold constant leads to a worse ATWV score than optimal; due to this, usually it is worth calculating *max-ATWV* (or MTWV), which is a (theoretical) upper bound of ATWV, where we take the maximal ATWV score of all N-best lists of the hit list returned. It summarizes the performance of the system if the probability threshold $P_{\min}$ has been optimally chosen. Fortunately, the metric FOM does not rely on this threshold value.

Table 1. The accuracy values obtained when using different kinds of triangular norms

T-norm used	N	Development Set		Test Set		
		MTWV	FOM	ATWV	MTWV	FOM
Log. gen., optimized for MTWV	8	71.32%	89.81%	65.38%	67.43%	86.67%
	16	68.44%	88.15%	64.21%	67.43%	87.31%
Log. gen., opt. MTWV + FOM	8	70.71%	91.13%	69.22%	69.75%	88.55%
	16	65.25%	92.79%	65.86%	66.03%	90.11%
Product (baseline)		57.31%	91.13%	63.29%	63.78%	89.96%

4.2 The Testing Environment

We used audio recordings of Hungarian news broadcasts taken from 8 different TV channels for testing. The 70 broadcasts were divided into three groups: the first, largest one (about 5 hours long) was used for training purposes. The second part (about 1 hour long) was the *development set*: these recordings were used to fine-tune the t-norm and get the corresponding threshold. The third part was the *test set* (about 2 hours long), used for the final evaluation of system performance. We chose 25 words and expressions as search terms, coming up in the news recordings quite frequently; they varied between 6-16 phonemes (2-6 syllables) in length. The phoneme sequence intermediate representations were produced by Artificial Neural Networks [2] used in two consecutive steps [29], applying the standard MFCC $+\Delta + \Delta\Delta$ feature set [15] with phoneme bigrams as a dictionary-independent language model, using the HTK tool [30].

4.3 The Testing Process

To set the control point a_i values, we used the histogram-based method described in Section 3.3. It requires a statistic of the actual probability values, which was obtained in a simple way. Assuming that the distribution of the phonemes of search terms mirror those of the recordings, we calculated the ratio of the occurrence of each phoneme in the training data set. Next, we chose two phonemes according to this distribution, and noted the probability values of deleting the first phoneme, inserting the second one, and replacing the first phoneme by the second one. This process was repeated 100 000 times, some white Gaussian noise was added to the generated values to smooth the resulting discrete values, and we chose the control points based on this histogram.

We performed the optimization of the steepness values by using the freely available Snobfit package [16]. We maximized for just the MTWV metric, and for the MTWV and FOM metric combined, and experimented with $N = 8$ and $N = 16$ control points, which meant a total of 4 tests. We optimized it by performing STD on the development set; the steepness values associated with the optimal score were then evaluated on the test set, using the corresponding threshold value. To ensure stability, we took all the vectors that produced an optimal score on the development set, and calculated their mean for each steepness m_i. In the

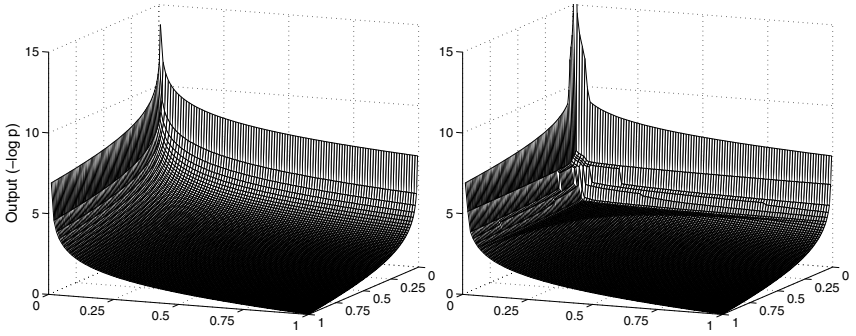


Fig. 2. The product t-norm (left) and the optimized t-norm using the logarithmic generator function (right). The x and y axes show the two argument probability values, while the z axis show the resulting cost value (i.e. -log p).

end, we got five scores for each case: MTWV and FOM for the development set, and ATWV (using the threshold value we got on the development set), MTWV (using the optimal threshold for the test set) and FOM for the test set.

4.4 Results

Table 1 lists the accuracy scores obtained using the logarithmic generator function. Examining the scores attained on the development set, we can see that all the optimized metric values significantly increased compared to the baseline scores. The settings $N = 16$ produced somewhat worse scores than $N = 8$, which is probably due to the *curse of dimensionality*: the number of tests required increases exponentially with the number of dimensions. Turning to the test set results, we see that in some cases the ATWV score is much lower than MTWV, reflecting threshold instability (i.e. $P_{\min}$ obtained on the development set was not optimal for the test set). In general, error reduction in the test set was not as successful, which could be partly due to *overfitting*: the optimization resulted in a development set-specific t-norm. To avoid this side effect, incorporating other metrics (in our case FOM) into the objective function of optimization seems to be a good idea, as in these cases there were only minor differences in the corresponding MTWV and ATWV scores. This is probably because different evaluation metrics measure the performance of a configuration in a somewhat different manner; in our case ATWV focuses on the top of the hit list, whereas FOM takes less probable hits into account as well. Trying to satisfy both metrics at the same time might result in a more balanced hit list, being better *in general*.

We should also stress that the resulting MTWV and ATWV scores exceeded those of the product norm in every case. Focusing on the case $N = 8$ when we optimized both for MTWV and FOM, we achieved an ATWV score of 69.22%, which, compared to the baseline score of 63.29%, means a relative error reduction score over 16%, this being quite a significant improvement in STD accuracy.

The product t-norm and the best-performing logarithmic generator function can be seen in Figure 2 (where, to emphasize the differences between the two norms, the z axis has a log scale). It can be seen that the two norms are quite different, reflecting the fact that the product operator is suboptimal for this task, and, unlike the logarithmic generator function, it could not be tuned either.

5 Conclusions

In a common approach of spoken term detection, user-entered queries are processed by matching their phonemes to the phonemes of recordings one at a time. In this task usually the phoneme operations are assumed to be independent, hence the product of their probabilities is taken; but using a triangular norm instead of multiplication can improve the system accuracy. In this work we applied an application-oriented representation of t-norms, and achieved a significant improvement in system accuracy and resulted in a relative error reduction of 16% this way.

References

1. Aczél, J., Alsina, C.: Characterizations of some classes of quasilinear functions with applications to triangular norms and to synthesizing judgements. Methods Oper. Res. 48, 3–22 (1984)
2. Bishop, C.: Neural Networks for Pattern Recognition. Clarendon Press, Oxford (1995)
3. Bonissone, P., Goebel, K., Yan, W.: Classifier fusion using triangular norms. In: Roli, F., Kittler, J., Windeatt, T. (eds.) MCS 2004. LNCS, vol. 3077, pp. 154–163. Springer, Heidelberg (2004)
4. Deng, G.: A parametric generalized linear system based on the notion of the t-norm. IEEE Transactions on Image Processing 22(7), 2903–2910 (2012)
5. Dombi, J.: A general class of fuzzy operators, the De Morgan class of fuzzy operators and fuzziness measures induced by fuzzy operators. Fuzzy Sets and Systems 8, 149–163 (1982)
6. Dombi, J.: Towards a general class of operators for fuzzy systems. IEEE Transaction on Fuzzy Systems 16(2), 477–484 (2008)
7. Dubois, D., Prade, H.: Fundamentals of Fuzzy Sets. Kluwer (2000)
8. Fodor, J.C.: A remark on constructing t-norms. Fuzzy Sets and Systems 41(2), 195–199 (1991)
9. Gosztolya, G., Dombi, J., Kocsor, A.: Applying the Generalized Dombi Operator family to the speech recognition task. Journal of Computing and Information Technology 17(3), 285–293 (2009)
10. Gosztolya, G., Kocsor, A.: A hierarchical evaluation methodology in speech recognition. Acta Cybernetica 17(2), 213–224 (2005)
11. Gosztolya, G., Kocsor, A.: Using triangular norms in a segment-based automatic speech recognition system. International Journal of Information Technology and Intelligent Computing (IT & IC) (IEEE) 1(3), 487–498 (2006)
12. Gosztolya, G., Stachó, L.L.: Aiming for best fit t-norms in speech recognition. In: Proceedings of SISY (IEEE), Subotica, Serbia, pp. 1–5 (September 2008)

13. Grundland, M., Vohra, R., Williams, G.P., Dodgson, N.A.: Cross dissolve without cross fade: Preserving contrast, color and salience in image compositing. In: Proceedings of Computer Graphics Forum, vol. 25, pp. 577–586 (2006)
14. Hanmandlu, M., Grover, J., Gureja, A., Gupta, H.: Score level fusion of multimodal biometrics using triangular norms. Pattern Recognition Letters 32(14), 1843–1850 (2011)
15. Huang, X., Acero, A., Hon, H.W.: Spoken Language Processing. Prentice Hall (2001)
16. Huyer, W., Neumaier, A.: Snobfit – stable noisy optimization by branch and fit. ACM Transactions on Mathematical Software 35(2), 1–25 (2008)
17. Jenei, S.: On Archimedean triangular norms. Fuzzy Sets and Systems 99(2), 179–186 (1998)
18. Jenei, S.: A general method for constructing left-continuous t-norms. Fuzzy Sets and Systems 136(3), 263–282 (2003)
19. Jenei, S., Pap, E.: Smoothly generated Archimedean approximation of continuous triangular norms. Fuzzy Sets and Systems (Special Issue "Triangular norms") 104, 19–25 (1999)
20. Katsurada, K., Sawada, S., Teshima, S., Iribe, Y., Nitta, T.: Evaluation of fast spoken term detection using a suffix array. In: Proceedings of Interspeech, pp. 909–912 (2011)
21. Kohavi, R., Provost, F.: Glossary of terms. Editorial for the Special Issue on Applications of Machine Learning and the Knowledge Discovery Process 30(2/3) (February/March 1998)
22. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady 10(8), 707–710 (1966)
23. Ling, C.H.: Representation of associative functions. Publ. Math. Debrecen 12, 189–212 (1965)
24. Pinto, J., Hermansky, H., Szöke, I., Prasanna, S.: Fast approximate spoken term detection from sequence of phonemes. In: Proceedings of SIGIR, Singapore (2008)
25. Rabiner, L., Juang, B.H.: Fundamentals of Speech Recognition. Prentice Hall (1993)
26. Schweizer, B., Sklar, A.: Associative functions and statistical triangle inequalities. Publ. Math. Debrecen 8, 169–186 (1961)
27. Szöke, I., Schwarz, P., Matejka, P., Burget, L., Karafiát, M., Fapso, M., Cernocky, J.: Comparison of keyword spotting approaches for informal continuous speech. In: Proceedings of Interspeech, pp. 633–636 (2005)
28. Tejedor, J., Wang, D., King, S., Frankel, J., Colas, J.: A posterior probability-based system hybridisation and combination for spoken term detection. In: Proceedings of Interspeech, Brighton, UK, pp. 2131–2134 (September 2009)
29. Tóth, L.: A hierarchical, context-dependent Neural Network architecture for improved phone recognition. In: Proceedings of ICASSP, pp. 5040–5043 (2011)
30. Young, S.: The HMM Toolkit (HTK) (software and manual) (1995), `http://htk.eng.cam.ac.uk/`
31. Young, S.: Statistical modelling in continuous speech recognition. In: Proceedings of UAI, Seattle, pp. 562–571 (2001)

Emotion Detection Using Hybrid Structural and Appearance Descriptors

David Sanchez-Mendoza, David Masip, Xavier Baró, and Àgata Lapedriza

Scene Understanding and Artificial Intelligence Lab (SUNAI),
Internet Interdisciplinary Institute (IN3),
Open University of Catalonia (UOC), Barcelona
{dsanchezmen,dmasipr,xbaro,alapedriza}@uoc.edu
http://in3.uoc.edu/

Abstract. In computer vision the facial expression recognition descriptors extracted from raw images are categorized as *structural* or *appearance* descriptors. A lot of effort has been done in the literature for improving both type of descriptors for making them more robust; in most cases, both types of descriptors have been used separately. In this work we propose a hybrid model that uses both descriptors for emotion inferring. Our model is based in detecting Action Units and uses a probabilistic approach for emotion prediction based on an ensemble of *Support Vector Machine* classifiers. Fully detailed inner workings of the method are provided for experiment replication as well as detailed results to assess emotion inferring performance.

Keywords: Computer Vision, Machine Learning, Emotion Analysis.

1 Introduction

Emotions are essential in our everyday lives. Whether intentionally or unintentionally, humans express different kinds of emotion during their daily routine in very different scenarios and situations.

Emotion expression has a lot to do with non-verbal body language. However, the expression of the face has a crucial role in emotions, being probably the most important part of the body when dealing with them. Actually, some basic emotions could be inferred just by analyzing the movements of the face; for this reason, trying to detect emotions based on the expression of the face is a common practice.

Ekman and Friesen (both psychologists) developed, back in 1978, the Facial Action Coding System (FACS) [1]: a complete system to define in a quite formal approach all the movements that a person can do with the muscles of its face. The most important feature of FACS is that, since it is based just on human anatomy, there is no room for subjectivity or different interpretations due to different cultural backgrounds.

The FACS basic units are called Action Units (AU). each one of them (there are up to 46) defines a single movement that human beings are able to do with their faces. Figure 1) shows some examples of Action Units.

V. Torra et al. (Eds.): MDAI 2013, LNAI 8234, pp. 105–116, 2013.

Fig. 1. Some examples of Action Units, they correspond, respectively, to AU#1 (inner brow raiser), AU#2 (outer brow raiser), AU#15 (lip corner depressor), AU#20 (lip stretcher) and AU#27 (mouth stretch). (Extracted from *http://www.cs.cmu.edu/∼face/facs.htm*).

It is known that a particular emotion can be expressed as a combination of some Action Units, while an emotion could be inferred given the set of all the Action Units involved in it. Our system is based in these ideas: emotions based on the detection of Action Units.

There are some Action Units that are relatively easy to detect by analyzing the movement of certain key points, like the ones located in the eyebrows or in the lips; however, some of them are easier to be detected using the appearance (the intensity value of a point in the image) rather than its movement or its location.

For instance, figure 2 shows a person performing, among others, Action Units #1 (inner brow raiser), #2 (outer brow raiser) and #27 (mouth stretch); all of them are suitable to be detected by analyzing the movement (how its location varies along a video) of certain key points such as the ones located on the eyebrows and on the lips. On the other hand, figure 2 also shows a person doing (among others) Action Unit #9 (nose wrinkle): this Action Unit is extremely difficult to detect just by tracking the movement of the key points involved in it. Rather than that, it is much easier to be detected analyzing the evolution of the appearance of the nose region.

Fig. 2. Left column person doing (among others) Action Units #1 (inner brow raiser), #2 (outer brow raiser) and #27 (mouth stretch); all of them should be detected analyzing the movement of some particular points. Right column person doing (among others) Action Unit #9 (nose wrinkle); it is very difficult to detect this Action Unit analyzing just the movement of key points, instead of it the evolution of the appearance of the nose region should be used. (Original images from Cohn-Kanade database).

Our method uses a two-layer approach hybrid model that takes advantage of both approaches: the fist step learns predictive models for each one of the Action

Units using separately both structural and appearances descriptors; the second step involves a probabilistic approach so that the Action Units detection results are combined by means of a probabilistic weighted ensemble of binary classifiers. Additionally, we will introduce a probabilities table relating emotions and Action Units classifiers, our results show consistency with the accepted psychology model that establishes the relationship between Action Units activation and the emotion being detected.

We briefly discuss previous work on face expression recognition (including both face detection and used descriptors) in section 2. The inner workings of the method are explained in depth in section 3 while the experimental settings and the results are detailed within section 4. Finally, results are briefly discussed in section 5.

2 State of the Art

Most of the facial expression recognition systems available in the literature, for instance [2], [3], [4], [5] and [6], are composed (among others) by two main phases.

The first one is face detection. The vast majority of existing images of faces have a cluttered background potentially including other objects apart from the face itself. The task of the face detector is to get rid of the background objects and pick just the region of the image in which the face is located. Currently face detection can be considered a solved problem for the frontal view case, although locating faces in unconstrained pose and in presence of occlusions is still an active research topic.

The second is the extraction of visual characteristics (descriptors) from the detected face. It is desirable that any descriptor could be invariant to illumination changes and rotation. On the other hand, regarding with dimensionality reduction, the size of the descriptor should be significantly smaller than the image size.

Among the most commonly used descriptors in computer vision are Gabor filters [7], SIFT [8], histograms of oriented gradients [9] (HOG), local binary patterns [10], and so on.

2.1 Face Detection and Facial Emotion Detection

One of the most widely used algorithms for face detection is the one developed by Viola and Jones [11]. The algorithm proposed in [11] detects faces almost in real time.

Viola-Jones algorithm uses the concept of *integral image*: instead of working with the pixel intensities they compute a set of features that recall Haar Basis functions. This computation involves just a few operation per pixel. The algorithm takes advantage of a machine learning approach in which many predictive models are built by selecting a small set of relevant features using AdaBoost [12]; then, a set of more complex models are build in a cascade fashion in order to speed up the detection process by focusing on the relevant regions of the image (see figure 3).

A more recent face detection algorithm was introduced by Ramanan and Zhu [3]. In [3] not only the face is detected, but also the pose is estimated and

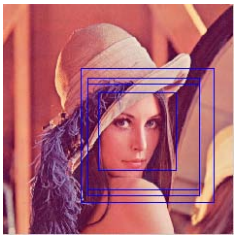


Fig. 3. Results of Viola-Jones algorithm ran on famous Lenna image. (Extracted from *http://www.mathworks.es*).

the landmark or key points of the face are located. Therefore, Ramanan-Zhu algorithm proposes a unified approach to solve the three tasks simultaneously by encoding elastic deformation of faces using mixtures of trees (see figure 4).

Fig. 4. Results of Ramanan-Zhu algorithm on a "wild" image. The algorithm is able to detect faces in a wide variety of positions and orientations. (Extracted from [3]).

Regarding to the descriptors usage for achieving emotion detection: examples of facial emotion detection using appearance descriptors can be found in [10], [13] and [14]. On the other hand, [15], [5] and [4] make use of facial landmarks and structural descriptors. Finally, a hybrid approximation by means of using simultaneously both descriptors is found in [2].

3 Emotion Detection Method

3.1 Structural Descriptors

Structural descriptors have to do with the location (rather than its color intensity) of certain key points (we call them *landmark* points) on the image. In particular, the structural descriptors we used are based in the work done by Rojas et al. [6].

Not all the points on the face are equally relevant for Action Units detection purposes. Ideally, we are interested in points that suffer enough variation from one Action Unit to another; for instance, the center of the left eyebrow has much more variation than the tip of the nose, therefore we will be more interested in this first landmark rather than in the second one.

Concretely, we have defined the structural descriptors only over the landmarks depicted in figure 5.

Fig. 5. Most significant landmarks that have been used for building the structural descriptors. (Original image from Cohn-Kanade database).

Let us describe the structural descriptor extracted from one particular frame f_t that belongs to a particular video F having T frames:

Let $P_t = \{p_1^t, ..., p_k^t\} : \forall i = 1, ..., k \;\; p_i^t \in \mathbb{N}_0^2$ be the k locations of the landmarks points within a particular frame f_t. For this particular frame the structural descriptor is defined as the concatenation of four blocks:

The first block is the difference between the position of a landmark with the same landmark on the first frame. It is defined as:

$$p_i^t - p_i^0 \quad \forall i = 1, ..., k$$

The second block are the differences among all the landmarks within the same frame. It is defined as:

$$I_{fd}^t = p_i^t - p_j^t : \quad \forall i, j = 1, ..., k : i \neq j$$

The third block contains the differences defined in second block with respect to the first frame. It is defined as:

$$F_{ad} = I_{fd}^t - I_{fd}^0$$

Finally, the fourth block comprises again the differences defined in second block but now with respect to the previous frame. It is defined as:

$$F_{pd} = I_{fd}^t - I_{fd}^{t-1}$$

3.2 Appearance Descriptors

Unlike structural descriptors, appearance descriptors are computed with the intensity values of a particular region of the image rather than the location of the landmark points.

We decided to use a bank of 12 Gabor filters to extract the appearance descriptor of a particular frame. The different Gabor filters belonging to the bank are parametrized according to scale $\lambda = \{11 \times 11, 24 \times 24, 37 \times 37\}$ and angle $\theta = \{0, \frac{\pi}{2}, -\frac{\pi}{2}, pi\}$. Therefore, our Gabor filters are defined as:

$$G(\theta, \lambda) = \exp\left(-\frac{x^2 + \gamma^2 y^2}{2\sigma^2}\right) \cos\left(\frac{2\pi x}{\lambda}\right)$$

where $x = i\cos\theta + j\sin\theta$, $y = -i\sin\theta + j\cos\theta \quad \forall i, j = 1, ..., N$ being N the size of the particular filter according to its scale.

Let f_t be a particular frame that belongs to a video F having T frames.

Let $G(\theta, \lambda)$ be a Gabor filter with λ scale and θ angle.

The appearance descriptor for image f_t, call it $\hat{f}_t$, is obtained by filtering f_t with $G(\theta, \lambda)$ $\forall\lambda = \{11 \times 11, 24 \times 24, 37 \times 37\}, \forall\theta = \{0, \frac{\pi}{2}, -\frac{\pi}{2}, pi\}$.

As we did with the structural descriptor, the final appearance descriptor will be defined by the concatenation of two blocks using the differences of the filtered image with the previous ones, as follows:

The first block is about the difference between the current frame and the initial one, it is defined as:

$$\hat{f}_t - \hat{f}_0$$

The second block contains the difference between the current frame and the previous one, it is defined as:

$$\hat{f}_t - \hat{f}_{t-1}$$

Finally, since not all the regions of the face are equally important, and also due to dimensionality reduction purposes, we do not use the whole face; we use instead the regions corresponding to eyes, nose and mouth (see figure 6). Given the high redundancy of the Gabor filters response, and still with the aim of reducing (even more) the dimensionality, Principal Component Analysis (PCA) is applied by picking dimensions (components) so that 95% of the data variance is kept.

3.3 Action Units Detection

Support Vector Machines (SVM) were introduced by Vapnik [16] and have been successfully proven in many pattern recognition and machine learning tasks. SVMs try to find the maximum-margin hyper-plane that separates positive and negative examples for a specified class. In our case, when training a SVM for a particular Action Unit, positive examples of the SVM would be the ones containing the corresponding Action Unit, while the negatives ones are those that do not contain it.

In order to perform Action Units detection a classifier is trained for each Action Unit with a one-vs-all strategy using a SVM with linear kernel. Furthermore, for each one of the Action Units two SVMs are learned, one using the appearance

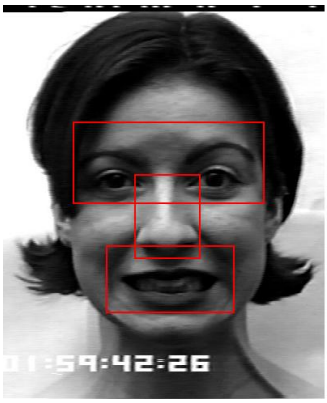


Fig. 6. Regions of the face being used for the appearance descriptor based on Gabor filters. (Original image from Cohn-Kanade database).

descriptor and another using the structural descriptor of the images. All classifiers are trained using the well-known Cohn-Kanade database in its extended version [2].

In all the classifiers the cost parameter is optimized by means of a cross-validation process on the training set. This parameter controls how strict is the classifier when accepting errors (misclassified data). Optimizing the *cost* guarantees that the classifier will have no over-fitting and, therefore, it will generalize some knowledge or patterns (and not just memorizing it) from the data it has been trained with.

Let $AUS = \{AU_1, ..., AU_k\}$ be the set of k action units that have to be learned. The Action Unit detection process ends up with $2k$ classifiers separated in two blocks, half of them using the structural descriptors and the other half using the appearance ones, as follows:

- $CS = \{c_{s1}, ..., c_{sk}\}$: set of classifiers using structural descriptors so that $c_{si} \in CS$ is a classifier that uses structural descriptors for learning $AU_i \in AUS$.
- $CA = \{c_{a1}, ..., c_{ak}\}$: set of classifiers using appearance descriptors so that $c_{ai} \in CA$ is a classifier that uses appearance descriptors for learning $AU_i \in AUS$.

3.4 Table of Probabilities for Emotion Inferring

The emotions that we want to infer are the seven basic emotion categories. Concretely this set is $E = \{Anger, Contempt, Disgust, Fear, Happy, Sadness, Surprise\}$.

Let $C = \{CS \cup CA\} : |C| = 2k$ being $c(x)$ the output of classifier $c \in C$ given input x.

Let $TE_i = \{te_{i1}, ..., te_{ip}\}$ be the test subset containing examples in which emotion $E_i \in E$ is present.

Let TOP be the $|E| \times |C|$ table of probabilities so that columns are ordered as:

$$c_{s1}, c_{a1}, c_{s2}, c_{a2}, ..., c_{sk}, c_{ak}$$

being

$$TOP_{ij} = P(c_j(x) \text{ predicted AU is present} \mid E_i \text{ present in } x) =$$

$$\frac{\#\{c_j(te_{im}); c_j(te_{im}) = 1\} \ \ \forall m = 1, ..., p}{|TE_i|}$$

$$\forall i = 1, ..., |E| \ \ \forall j = 1, ..., |C|$$

Inferring the Emotion of a New Example

Let im be a new image example in which a single emotion $e \in E$ is present.

Let $r \in \{0, 1\}^{|C|}$ so that:

$$r_i = \begin{cases} 0 & \text{if } c_i \text{ predicts that its corresponding action unit is present in } im \\ 1 & \text{if } c_i \text{ predicts that its corresponding action unit is } \textbf{not} \text{ present in } im \end{cases}$$

$$\forall i = 1, ..., 2k$$

Once r is set the emotion that is finally inferred according to:

$$\arg\max_{e} \quad r \cdot TOP_{e,\cdot}$$

Emotion Recognition Process Overview

In figure 7 a summary of the whole emotion recognition process is shown. It depicts all the steps that are followed from the original images to the Action Units classifiers and, afterwards, the table for emotion inferring.

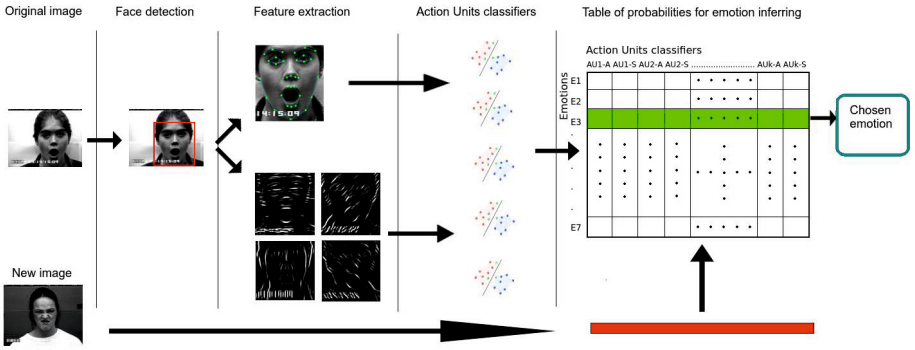


Fig. 7. Summary of the whole emotion recognition process

4 Experimentation and Results

4.1 Experimental Setting

Let us introduce within this section the inner workings of the tools we have been using for both Action Units classifiers training and emotion inferring.

Action Units classifiers have been trained using the Cohn-Kanade database (CK) [2]: regarding with Action Units detection, it contains 593 posed sequences in which full FACS coding of peak expression frames is provided. CK database include 118 different participants from 18 to 50 years, being the 69% of them female, 81% Euro-American, 13% Afro-American and 6% other groups. All image sequences were digitalized into 640×490 pixel matrices with 8-bit gray-scale or 24-bit color values.

As previously stated, the classifier used is the linear kernel Support Vector Machine, particularly the LIBSVM implementation [17]. The *cost* parameter is optimized using cross-validation on the training set and then the classifier is retrained with all the training set using the best *cost*.

The calculation of each one of the cells TOP_{ij} of the table of probabilities is done by means of a leave-one-out person-independent cross-validation, as follows:

Let $IND = \{ind_1, ..., ind_{118}\}$ be the set of different individuals present in CK database. Each one of the $ind_n \in IND$ has its own image sequences.

The classifier $c_j \in C$ is trained $|IND|$ times. The n-th iteration uses the sequences $s_{TR} \in IND \setminus \{ind_n\}$ as the training set and the sequences $s_{TE} \in ind_n$ as the test set. As stated before, each built classifier for each iteration makes, at the same time, another cross-validation process (this time over $IND \setminus \{ind_n\}$ sequences) so that the *cost* parameter is optimized.

The calculation of the whole table of probabilities TOP using the CK database involves no more than 20 seconds in a standard desktop computer.

The TE_i set described within section 3.4 is formed by all the sequences s_{TE} (for all the $|IND|$ iterations) containing the emotion $E_i \in E$. Predicting a test example e involves very few computations, in particular: running all the $c \in C$ AU classifiers using e as input and computing one scalar product for each $E_i \in E$.

4.2 Emotion Detection Results

The heatmap of figure 8 shows a representation of the table of probabilities for emotion inferring: the rows represent the emotions while the columns represent the different classifier for Action Units detection using whether structural or appearance descriptors.

Using the Table of Probabilities depicted on figure 8 the emotion of all the sequences of the CK database has been inferred. The matrix in figure 9 shows a comparison of the prediction results with the ground truth gives us the confusion matrix shown on figure 9.

5 Discussion

Analyzing the Table of Probabilities, depicted in heat-map style on figure 8, we observe that these probabilities are completely coherent with the accepted emotion description in terms of facial action units.

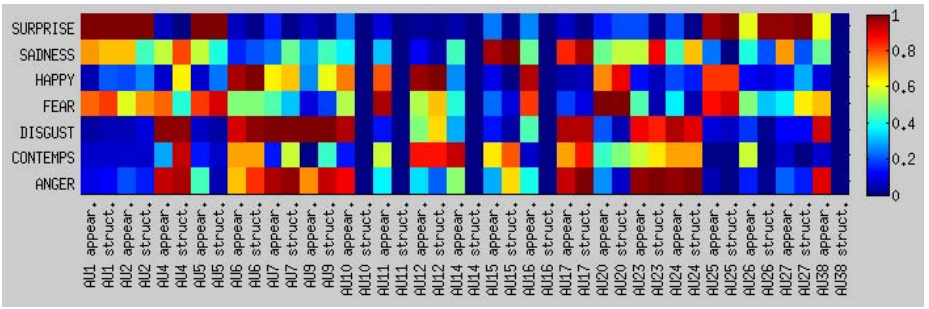


Fig. 8. Heatmap representation of the Table of Probabilities for emotion inferring

	anger	*contempt*	*disgust*	*fear*	*happiness*	*sadness*	*surprise*
anger	**0.73**	0.04	0.00	0.00	0.00	0.23	0.00
contempt	0.07	**0.64**	0.00	0.00	0.07	0.14	0.07
disgust	0.25	0.01	**0.68**	0.00	0.05	0.01	0.00
fear	0.00	0.00	0.00	**0.75**	0.25	0.00	0.00
happiness	0.00	0.03	0.00	0.09	**0.88**	0.00	0.00
sadness	0.00	0.09	0.00	0.00	0.00	**0.91**	0.00
surprise	0.00	0.00	0.00	0.07	0.01	0.06	**0.85**

Fig. 9. Confusion matrix resulting from the emotions predictions of the CK database using the Table of Probabilities

For instance, happiness is expected to contain Action Unit #12: as expected, both classifiers for this action have a very high probability to be activated (see probabilities on figure 8).

On the other hand, an interesting phenomenon happens with disgust emotion: it is expected to have Action Units #9 or #10: looking at the Table of Probabilities we can see that both classifiers for Action Unit #9 have high probability, however, for the Action Unit #10 just the appearance classifier has high probability; this means that is easier to detect this Action Unit using appearance descriptors rather than using the structural ones. It must be said that this is very suitable since in Action Unit #10 (upper lip raiser, see figure 10) there is not a clear trajectory of any landmark point that helps to detect it, however there is a noticeable change on the region around the lower part of the nose.

Regarding with the confusion matrix shown in figure 9 we want to remark that an accuracy of 0.77 is a very promising result since we are dealing with a 7-class problem, in this sense we improve largely the chance accuracy.

There are some emotions that have better results than the others. Happiness, sadness and surprise are the ones that have more accuracy (> 0.85) while others like anger, contempt or disgust have a lower one (< 0.65). An effort in both finding better descriptors and improving the classifiers has to be done to improve the detection of these emotions.

An example of this difficulty can be found in disgust emotion: while the 64% of examples containing this emotion are well classified, the 25% of them are classified as anger. This is happening because both emotion are, in some way,

Fig. 10. Detail of person performing Action Unit #10: upper lip raiser. (Extracted from *http://www.cs.cmu.edu/~face/facs.htm*).

similar and then, for some examples, the classifiers are not able to detect the differences among them.

6 Conclusion

We introduced a probabilistic weighted ensemble to infer emotions from video based on Action Unit Detection. Yet simple, the model intrinsically correlates with the intuitive idea of where appearance and structural descriptors should be used given its nature. We have also explained both structural and appearance descriptors and in which situations one should be more convenient instead of the other.

Finally, the results of the whole method using the Cohn-Kanade database are introduced: both the Table of Probabilities and the confusion matrix to asses the quality of the predictions.

References

1. Ekman, P., Friesen, W.V.: Facial action coding system: A technique for the measurement of facial movement. palo alto. Consulting Psychologists Press, CA (1988); Ellsworth, P.C., Smith, C.A.: From appraisal to emotion: Differences among unpleasant feelings. Motivation and Emotion 12, 271–302 (1978)
2. Lucey, P., Cohn, J.F., Kanade, T., Saragih, J., Ambadar, Z., Matthews, I.: The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 94–101. IEEE (2010)
3. Zhu, X., Ramanan, D.: Face detection, pose estimation, and landmark localization in the wild. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2879–2886. IEEE (2012)
4. Valstar, M., Pantic, M.: Fully automatic facial action unit detection and temporal analysis. In: Conference on Computer Vision and Pattern Recognition Workshop, CVPRW 2006, p. 149. IEEE (2006)
5. Valstar, M.F., Patras, I., Pantic, M.: Facial action unit detection using probabilistic actively learned support vector machines on tracked facial point data. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, CVPR Workshops, p. 76. IEEE (2005)
6. Rojas, M.: On the use of geometric and appearance information for facial analysis. PhD thesis, University of Barcelona (2012)
7. Bartlett, M.S., Littlewort, G., Frank, M., Lainscsek, C., Fasel, I., Movellan, J.: Recognizing facial expression: machine learning and application to spontaneous behavior. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 2, pp. 568–573. IEEE (2005)

8. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
9. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 1, pp. 886–893. IEEE (2005)
10. Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: Application to face recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(12), 2037–2041 (2006)
11. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, vol. 1, pp. I–511. IEEE (2001)
12. Freund, Y., Schapire, R.E.: A desicion-theoretic generalization of on-line learning and an application to boosting. In: Vitányi, P.M.B. (ed.) EuroCOLT 1995. LNCS, vol. 904, pp. 23–37. Springer, Heidelberg (1995)
13. Calonder, M., Lepetit, V., Strecha, C., Fua, P.: BRIEF: Binary robust independent elementary features. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 778–792. Springer, Heidelberg (2010)
14. Kumano, S., Otsuka, K., Yamato, J., Maeda, E., Sato, Y.: Pose-invariant facial expression recognition using variable-intensity templates. International Journal of Computer Vision 83(2), 178–194 (2009)
15. Gizatdinova, Y., Surakka, V.: Feature-based detection of facial landmarks from neutral and expressive facial images. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(1), 135–139 (2006)
16. Vapnik, V.: The nature of statistical learning theory. Springer (2000)
17. Chang, C.-C., Lin, C.-J.: Libsvm: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST) 2(3), 27 (2011)

A Lazy Learning Approach for Self-training

Eva Armengol

Artificial Intelligence Research Institute (IIIA - CSIC),
Campus de la UAB, 08193 Bellaterra, Catalonia, Spain
eva@iiia.csic.es

Abstract. Self-Training methods are a family of methods that use some supervised method to assign class labels to the unlabeled examples. The resulting model is useful to predict the classification of unseen new domain objects. Most common supervised methods used inside self-training are the inductive ones. In this paper we propose to use the lazy learning method LID to assign classes to the unlabeled examples. A lazy approach such as the one of LID allows to reason by similarity around the labeled examples. Thus, when an unlabeled example is classified as belonging to a class we are sure that it shares relevant features with some labeled examples.

Keywords: Machine learning, Semi-supervised learning, Self-Training, Lazy Learning methods, Classification.

1 Introduction

Classification problems are those that handle domain objects that can be grouped in classes [5]. The goal of classification is to characterize univocally each one of the classes by means of one or more descriptions. There are many learning methods that can be used to solve classification problems and most of them take into account a set of known examples to build a model of the domain. These known examples are *labelled* since the class to which they belong is known. One of the most common families of learning methods used to build domain models are the *inductive learning methods*. In these methods, given the set of known examples belonging to a class C_i, the goal is to build a generalized description for C_i (commonly a disjunction of several descriptions) that is satisfied by all the examples of C_i and none of the examples belonging to the other classes. The more amount of examples are known the more accurate the induced model is.

However with the growth availability of data a new problem appears: some of the known examples are not labelled. In fact, what happens is that the *majority* of the known examples are not labelled. This means that inductive learning methods are not longer applied since only a few of the examples can be used to build the domain model and therefore that model is not accurate enough. A different approach in these cases could be the use of unsupervised learning methods such as *clustering methods* that are able to handle both labeled and unlabelled examples. The main concern in this situation is that they cannot

V. Torra et al. (Eds.): MDAI 2013, LNAI 8234, pp. 117–125, 2013.

take great benefit of the labeled examples. This is the main motivation of using *Semi-supervised learning methods*.

Semi-supervised learning methods are mainly concerned to deal with a huge amount of data the majority of which are not labeled. In [12] there is an excellent survey about semi-supervised learning methods. In that paper authors classify semi-supervised learning methods as *transductive* when the goal is to construct a domain model, and *inductive* when the goal is to classify unseen objects. Because we are interested on predict classifications, in this paper we focus on inductive methods. These methods, according to [12] can be classified in the following three families:

- *Generative methods*. They are probabilistics and assume that a domain model is an identifiable mixture distribution [10] in a way that assume that examples have a distribution from which labels to the unlabeled examples can be assigned. If the assumed distribution is not the one fitting the examples, the final model could be totally incorrect; otherwise, these methods have a good performance. The most common algorithm on this family is the Expectation-Maximization (EM) algorithm [7].
- *Self-training*. The basic idea is to assess the classification of unlabeled examples from the classes of labeled examples. Thus a supervised learning method is used to classify the unlabeled examples which are then added to the set of labeled examples. Algorithms used in this family are, for instance inductive learning methods such as ID3 [11].
- *Co-training* [4]. The idea is to split the attributes involved in the descriptions of the known examples in two disjoint subsets. Each one of these subsets is given to a classifier in order to propose a class for the object. Thus, each classifier proposes a class label according to the information that it has about the object. When the two classifiers agree on the class the unlabeled object is then labeled as belonging to that class.

Behind these methods there is the idea of selection of unlabeled examples. It has been proved [8] that an intelligent selection of either the unlabeled methods to be labeled or the split of attributes in the case of co-training, can significantly improve the accuracy of the final model.

Authors working on semi-supervised learning assume that the process of labeling is a time-consuming task; therefore, the goal is how to automatically take benefit of the unlabeled data. Clearly, what we need is to label them, therefore it is necessary to explore how to assign labels to all these unlabeled examples or, at least, to a majority of them.

In this paper we focus on self-training. We assume we have a huge amount of data and only a few of them are labeled, however we would use all them to predict the classification of unseen objects. Self-training uses a supervised learning method on the labelled examples in order to assign class labels to the unlabeled examples. The labeling process is commonly performed using either an inductive learning method or the k-NN method in the context of support vector machines [9]. In our approach we propose to use a lazy learning method called

```
Function LID (p, S_{D_i}, D_i, C)
    if stopping-condition(S_{D_i})
        then return class(S_{D_i})
        else f_d := Select-attribute (p, S_{D_i}, C)
             D_{i+1} := Add-attribute(f_d, D_i)
             S_{D_{i+1}} := Discriminatory-set (D_{i+1}, S_{D_i})
             LID (p, S_{D_{i+1}}, D_{i+1}, C)
    end-if
end-function
```

Fig. 1. The LID algorithm: p is the problem to be solved, D_i is the similitude term, S_{D_i} is the discriminatory set associated with D_i, C is the set of solution classes, $class(S_{D_i})$ is the class $C_i \in C$ to which all elements in S_{D_i} belong.

LID [2] to label the examples. This method, in addition to classify an object is capable to give an explanation of the proposed classification. Such explanation is, in fact, a description similar to the ones produced by inductive methods. As we already suggested in [1] we can store these descriptions and finally we obtain a partial model of the domain. In the current paper we do not use these descriptions but only the labelled examples, however in the future we plan to experiment with these descriptions and with partial models of the domain.

The paper is organized as follows. First in Section 2 we introduce LID, the lazy learning method that we use in our experiments. In Section 3 we explain how to perform self-training with LID. In Section 4 we describe the experiments we carried out and the obtained results. The paper ends with the conclusions and the future work.

2 The Lazy Induction of Descriptions Method

Lazy Induction of Descriptions (LID) is a lazy learning method for classification tasks. LID determines which are the most relevant attributes of a new problem and searches in a case base for cases sharing these relevant attributes. The problem is classified when LID finds a set of relevant attributes whose values are shared by a subset of cases all of them belonging to a same class. The description formed by these relevant features is called *similitude term* and the set of cases satisfying the similitude term is called *discriminatory set*.

Given a problem for solving p, the LID algorithm (Fig. 1) initializes D_0 as a description with no attributes, the discriminatory set S_{D_0} as the set of cases satisfying D_0, i.e., all the available cases, and C as the set of solution classes into which the known cases are classified. Let D_i be the current similitude term and S_{D_i} be the set of all the cases satisfying D_i. When the stopping condition of LID is not satisfied, the next step is to select an attribute for specializing D_i.

The specialization of D_i is achieved by adding attributes to it. Given a set F of attributes candidate to specialize D_i, LID selects the most discriminatory

attribute in F using a distance measure. Such distance is used to compare each partition $\mathcal{P}_f$ induced on S_{D_i} by an attribute f with the correct partition $\mathcal{P}_c$. The *correct partition* has as many sets as solution classes. Each attribute $f \in F$ induces in S_{D_i} a partition $\mathcal{P}_f$ with as many sets as the number of different values that f takes in the cases contained in S_{D_i}. Given a distance measure Δ and two attributes f and g inducing respectively partitions $\mathcal{P}_f$ and $\mathcal{P}_g$, we say that f is *more discriminatory* than g iff $\Delta(\mathcal{P}_f, \mathcal{P}_c) < \Delta(\mathcal{P}_g, \mathcal{P}_c)$. This means that the partition $\mathcal{P}_f$ is closer to the correct partition than the partition $\mathcal{P}_g$.

Let f_d be the most discriminatory attribute in F. The specialization of D_i defines a new similitude term D_{i+1} by adding to D_i the attribute f_d. The new similitude term $D_{i+1} = D_i \cup \{f_d\}$ is satisfied by a subset of cases in S_{D_i}, namely $S_{D_{i+1}}$. Next, LID is recursively called with $S_{D_{i+1}}$ and D_{i+1}. The recursive call of LID has $S_{D_{i+1}}$ instead of S_{D_i} because the cases that are not satisfied by D_{i+1} will not satisfy any further specialization. Notice that the specialization reduces the discriminatory set at each step, i.e., we get a sequence $S_{D_n} \subseteq S_{D_{n-1}} \subseteq \ldots \subseteq S_{D_0}$.

The selection of the most discriminatory attribute is heuristically done using the LM distance [6] over the candidate attributes. Let us recall its definition: Let X be a finite set of objects; $\mathcal{P} = \{P_1, \ldots, P_n\}$ be a partition of X in n sets; and $\mathcal{Q} = \{Q_1, \ldots, Q_m\}$ be a partition of X in m sets. The LM distance between them is computed as follows:

$$LM(\mathcal{P}, \mathcal{Q}) = 2 - \frac{I(\mathcal{P}) + I(\mathcal{Q})}{I(\mathcal{P} \cap \mathcal{Q})}$$

where

$$I(\mathcal{P}) = -\sum_{i=1}^{n} p_i log_2 p_i; \quad p_i = \frac{|P_i|}{|X|}$$

$$I(\mathcal{Q}) = -\sum_{j=1}^{m} p_j log_2 p_j; \quad p_j = \frac{|Q_j|}{|X|}$$

$$I(\mathcal{P} \cap \mathcal{Q}) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p_{ij} log_2 p_{ij}; \quad p_{ij} = \frac{|P_i \cap Q_j|}{|X|}$$

Given a partition $\mathcal{P}$ on a set X, $I(\mathcal{P})$ is the average information of $\mathcal{P}$ and it measures the randomness of the distribution of elements of X over the n classes of the partition. The quantity represented by $I(\mathcal{P} \cap \mathcal{Q})$ is the mutual average information of the intersection of the partitions $\mathcal{P}$ and $\mathcal{Q}$.

LID has two stopping situations: 1) all the cases in the discriminatory set S_{D_j} belong to the same solution class C_i, or 2) there is no attribute allowing the specialization of the similitude term. When the stopping condition (1) is satisfied p is classified as belonging to C_i. When the stopping condition (2) is satisfied, S_{D_j} contains cases from several classes and p cannot be uniquely classified. The outcome of LID in the stopping condition (1) is both a class label and a similitude term, which justifies the classification of p. The last similitude term can also be interpreted as a partial description of the class C_i. It is partial because all the

examples satisfying it belong to the class C_i, however there are examples in C_i that do not satisfy this similitude term; therefore, it does not characterizes all the class.

3 Self-training with LID

In this section we explain how to perform self-learning with LID. Given a set of labeled examples $\mathcal{L}$ and a set of unlabeled examples $\mathcal{U}$ the goal is to use the examples in $\mathcal{L}$ to label those in $\mathcal{U}$. At the end of labelling process, the set of the labeled examples, $\mathcal{L}$ and those in $\mathcal{U}$ newly labeled, can be used to predict the classification of unseen examples. A self-training method uses the labeled examples with a supervised learning method to assign labels to (some) examples of $\mathcal{U}$. In our approach, the supervised method we use is LID. The common self-training algorithm is the following:

1. Select (randomly) an example $u_i \in \mathcal{U}$.
2. Run LID to classify u_i. Let s be the solution proposed by LID.
3. If s is a multiple solution (more than one class), then u_i cannot be classified. Actualize $\mathcal{U} = \mathcal{U} - \{u_i\}$ and go to step 1.
4. If s is one class then assign s as the label for u_i. Let D_s be the description given by LID to explain the classification of u_i.
5. Retrieve the subset $\mathcal{U}_s$ of the $u_k \in \mathcal{U}$ satisfying D_s and assign to all of them the label s.
6. Actualize $\mathcal{U} = \mathcal{U} - \mathcal{U}_s$ and go to step 1

The general idea of a self-training method is to assign a class label to each one of the examples in $\mathcal{U}$. Commonly, self-training methods assign labels to all the examples in $\mathcal{U}$. The use of LID has two main differences with respect to the use of any other method. The first one is that at the end of the labelling process some examples in $\mathcal{U}$ can remain unlabeled. This happens when LID finishes with the stopping condition (2), i.e., when it cannot find a unique solution class and gives a multiple classification. The second difference is that LID does not need to be run for all the examples in $\mathcal{U}$. Let us suppose that LID has classified an example $u_i \in \mathcal{U}$ as belonging to the class C_i because of the similitude term D_i. All other examples in $\mathcal{U}$ that satisfy D_i will also be classified as belonging to C_i. So, in fact, it is not necessary to run LID for all the examples in $\mathcal{U}$.

Behind the step 5 of the previous algorithm there is the idea of local approximation. The object u_i has been classified as belonging to the class s because it shares important features with some labeled examples belonging to s. The description D_s is a pattern describing the part of the problem space around u_i and this subset of labeled examples; therefore, we assume that all the other examples satisfying this pattern will also belong to s. Notice that, by construction, any of the labeled examples satisfying D_s belong to a class different than s. We also want to remark that at the end of the process some of the unlabeled examples can remain unlabeled.

Table 1. Number of domain objects, attributes and classes of the UCI datasets used in the experiments

Dataset	♯Objects	♯Attributes	♯Classes
Monks1	432	6	2
Monks2	432	6	2
Monks3	432	6	2
Solar Flare	1066	10	2
Vehicle	1728	6	4

Many semi-supervised learning methods can only deal with binary domains, i.e., those having only two solution classes. Our approach is not aware of the number of classes of the domain since it explores areas of the problem space around the known problems.

As we pointed out in [1], the set of all D_j gives us a partial model of the domain which can be used for prediction in the same way that inductive models. We do not experiment now with that model but only use the examples in $\mathcal{L}$ and those of $\mathcal{U}$ newly labeled to predict the class of unseen objects, i.e., we take a lazy approach for classify new objects. In the next section we explain the experimentation.

4 Experiments

The goal of the experiments is to prove that using a lazy learning approach such as LID to assign labels to unlabeled examples is a good approach. We used 5 data sets from the UCI repository [3]: Monks1, Monks2, Monks3, Solar Flare and Vehicle (Table 1 shows the characteristics of these domains). We chosen these domains mainly because they have a "reasonable" number of domain objects although most of them have only two solution classes. Other data sets having more classes, as for instance Soybean that has 17 classes, have not enough number of objects to experiment with a semi-supervised approach. However, the data set Vehicle is a good example of data set having more than two classes and help us to prove the feasibility of our approach.

First of all we separated a 10% of the domain objects to form the test set TS to be used for prediction. Then, we need to determine the sets $\mathcal{L}$ and $\mathcal{U}$. Thus, from all the remaining examples we randomly selected a proportion p to act as the set $\mathcal{L}$ and for the rest we deleted their class label and considered them as unlabeled. Next step has been to apply the self-training procedure described in previous section to increase the number of labelled examples. Finally, with all the labeled examples (those already labeled at the beginning and the new ones) as case base, we used LID to predict the class of the objects in TS.

With this basis we carried out several trials. First we experimented with different values of p in order to check some changes depending of the proportion of labeled and unlabeled examples. Thus we taken p to be 5%, 10% and 15% of the available objects (i.e., excluding those in TS). For each one of these values

Table 2. Comparison of the predictivity accuracy average of the experiments between LID in mode supervised and self-training with LID taking the 5%, 10%, and 15% of labeled examples. Between parenthesis is also show the percentage of multiple classifications.

Dataset	LID	5%	10%	15%
Monks1	96.582 (17.177)	65.845 (21.744)	66.413 (12.397)	67.042 (13.248)
Monks2	48.620 (24.233)	56.814 (13.027)	57.842 (16.702)	56.961 (19.083)
Monks3	100 (0)	85.347 (15.179)	92.550 (5.391)	95.229 (5.650)
Solar Flare	81.966 (42.453)	80.384 (9.339)	77.399 (10.596)	89.731 (7.989)
Vehicle	95.984 (5.652)	78.874 (13.272)	83.105 (12.927)	84.829 (12.971)

we performed 10 trials. In each trial we chosen randomly the elements of $\mathcal{L}$. This allows us to obtain results that are independent of the labeled examples in $\mathcal{L}$. Finally, we repeated all the experiments picking up different examples to form the test set.

Table 2 shows the accuracy average of all the experiments and also the accuracy of LID when all the objects are labeled. The accuracy has been computed without taking into account the percentage of multiple answers, i.e., we considered multiple answers as a no classification. That is to say, results have to be read as follows: when the systems gives a unique solution, the $X\%$ of times is the correct one. Between parenthesis there is also the percentage of multiple classifications.

Concerning accuracies, we seen that the semi-supervised approach is comparable to the lazy learning approach. Best results for semi-supervised approach are obtained with 15% of labeled data improving the results of LID in Monks2 and Solar Flare and being very similar in Monk3. This result was expected since as many labeled examples we have better should be the final model. We also seen than semi-supervised approach produces low percentage of multiple classifications. This is specially important in Monks2, where LID produces very bad results.

In a general view we can see that using the semi-supervised approach the accuracy decreases. This is an expected fact because the classification is made with a low percentage of examples having labels that can be assured as correct (since the other ones have labels automatically assigned). Focusing on the concrete data sets, we seen that the accuracy on the Solar Flare highly increases taking the semi-supervised approach. This increment is explained by the fact that there are less multiple classifications probably due to the regularities of the domain, allowing the construction good local approximations around labelled examples. Concerning Monks2 and Vehicle, we seen that supervised and the semi-supervised approaches give similar results. This could be due to the inverse reason that for the Solar Flare: these two domains are less regular and the frontiers of the classes are not clear. The data sets Monks1 and Monks3 have the expected behavior since the semi-supervised approach is less accurate than LID and the accuracy increases as much as labelled examples are available.

It is also clear that the examples contained in the labeled set $\mathcal{L}$ highly influences the process of labeling. Since we chosen randomly the examples in $\mathcal{L}$, if one class C_i is not well represented by these examples, then the unseen examples belonging to C_i will not be well classified.

Table 3 shows the average of correct classifications of the examples in $\mathcal{U}$. For Solar Flare and Vehicle the percentage of correct classifications is high, meaning that in these case we can obtain a good model domain with self-training. This good model is reflected on the accuracy results and, as we already mentioned, in the Solar Flare the accuracy is even better than using LID on the data set completely labeled. For all three Monks data sets, we see that there are many errors in labeling examples. By a detailed analysis of the trials we have seen that, depending on the examples of the set $\mathcal{L}$, sometimes all the examples of one of the classes have been erroneously labeled. Therefore the case base on which LID is working to classify unseen objects is biased toward one of the classes. In other words, LID is capable to classify correctly examples of one class but the ones in the other class sometimes produce errors. Monks data set is balanced, that is to say, there is the same number of objects in both classes. Therefore we also see that despite the incorrect labeling of one of the classes, LID is robust enough to correctly classify unseen objects of that class, otherwise the high accuracy achieved in Monks3 could not be explained.

5 Conclusions and Future Work

In this paper we introduced an approach for self-training that uses a lazy learning method called LID for labeling known examples without class label. The novelty is the use of a symbolic method as LID since other approaches using lazy learning methods such as the k-NN, work on the context of support vector machines or probabilistic models. We experimented on several well known data sets and the results prove the feasibility of the approach. The expected behavior is that the accuracy of the semi-supervised approach decreases with respect the one exhibited by a supervised approach (i.e., with all the known examples labeled). Using the description that LID builds to explain the classification, we obtain good patterns allowing to classify some unlabeled examples. In some domains, as for instance in Monks2 and in Solar Flare they allow to increase the accuracy.

We take a lazy approach for predict the class of unseen domain objects. However it should also possible to use the descriptions build by LID to form a partial

Table 3. Percentage of correct classifications of the examples in $\mathcal{U}$ using LID taking as case base the set $\mathcal{L}$

Dataset	5%	10%	15%
Monks1	42.090	61.957	59.706
Monks2	38.111	47.800	48.879
Monks3	54.404	42.600	52.533
Solar Flare	73.156	78.786	66.049
Vehicle	74.347	76.717	70.574

model of the domain. The idea is to store these descriptions during the self-training process. Each one of the descriptions is associated to a class, i.e., is a partial description of a class. Therefore, when an unseen example matches one of the descriptions it can be classified as belonging to that class.

Acknowledgments. The author thanks Àngel García-Cerdaña his helpful comments and suggestions. This research is partially funded by the Spanish MICINN projects COGNITIO (TIN-2012-38450-C03-03) and EdeTRI (TIN2012-39348-C02-01), and the grant 2009-SGR-1434 from the Generalitat de Catalunya.

References

1. Armengol, E.: Building partial domain theories from explanations. Knowledge Intelligence 2/08, 19–24 (2008)
2. Armengol, E., Plaza, E.: Lazy induction of descriptions for relational case-based learning. In: De Raedt, L., Flach, P. (eds.) ECML 2001. LNCS (LNAI), vol. 2167, pp. 13–24. Springer, Heidelberg (2001)
3. Bache, K., Lichman, M.: UCI machine learning repository (2013)
4. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, COLT 1998, pp. 92–100. ACM, New York (1998)
5. Clancey, W.J.: Heuristic classification. Artificial Intelligence 27(3), 289–350 (1985)
6. López de Mántaras, R.: A distance-based attribute selection measure for decision tree induction. Machine Learning 6, 81–92 (1991)
7. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B 39(1), 1–38 (1977)
8. Huang, J., Sayyad-Shirabad, J., Matwin, S., Su, J.: Improving co-training with agreement-based sampling. In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (eds.) RSCTC 2010. LNCS, vol. 6086, pp. 197–206. Springer, Heidelberg (2010)
9. Iggane, M., Ennaji, A., Mammass, D., El Yassa, M.: Self-training using a k-nearest neighbor as a base classifier reinforced by support vector machines. International Journal of Computer Applications 56(6), 43–46 (2012)
10. Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T.M.: Text classification from labeled and unlabeled documents using EM. Machine Learning 39(2/3), 103–134 (2000)
11. Quinlan, J.R.: Induction of decision trees. Machine Learning 1(1), 81–106 (1986)
12. Zhu, X.: Semi-supervised learning literature survey. Technical report, Computer Science, University Wisconsin-Madison, Madison, WI, Tech. Rep. 1530 (2005)

Combining Recommender and Reputation Systems to Produce Better Online Advice

Audun Jøsang[1], Guibing Guo[2], Maria Silvia Pini[3], Francesco Santini[4], and Yue Xu[5]

[1] University of Oslo, Norway
josang@ifi.uio.no
[2] NTU, Singapore
gguo1@e.ntu.edu.sg
[3] University of Padova, Italy
pini@dei.unipd.it
[4] INRIA - Rocquencourt, France
francesco.santini@inria.fr
[5] QUT, Australia
yue.xu@qut.edu.au

Abstract. Although recommender systems and reputation systems have quite different theoretical and technical bases, both types of systems have the purpose of providing advice for decision making in e-commerce and online service environments. The similarity in purpose makes it natural to integrate both types of systems in order to produce better online advice, but their difference in theory and implementation makes the integration challenging. In this paper, we propose to use mappings to subjective opinions from values produced by recommender systems as well as from scores produced by reputation systems, and to combine the resulting opinions within the framework of subjective logic.

1 Introduction

Recommender systems [1] and reputation systems [8,14] are similar in the sense that both collect data of members in a community in order to provide advice to those members. However, there are also fundamental differences. Recommender systems assume that different people inherently have different tastes, and hence value things subjectively. In contrast, reputation systems assumption that all members in a community value things under the same criteria, i.e. objectively. Said differently, when a recommender system indicates that a user probably does not like a given resource, it does not mean that there is anything wrong with it. However, when a reputation system produces a low value for a resource, one can assume that its quality is poor. We use the term *"resource"* to denote the thing (or item) being rated. The purpose of recommender systems is mainly to generate suggestions about resources that a user *a priori* is not aware of but would probably be interest in. The purpose of reputation systems is to provide advice about resources that the user already is aware of and interested in.

V. Torra et al. (Eds.): MDAI 2013, LNAI 8234, pp. 126–138, 2013.

On this background there is a strong potential for combining the two types of systems.

However, it is quite challenging to make an effective integration of the output results produced by recommender and reputation systems, given the following three-fold. First, in general the advices generated from different systems are distinct and heterogeneous. This is because different systems may use different forms of feedback and evaluate the performance based on different criteria. Second, the result from reputation systems reflects the collective opinions of a whole community whereas the result from recommender systems only represents the collective opinions of a local community, i.e. the users with similar preference. Third, the uncertainty of the generated advice should be taken into consideration. The uncertainty is typically due to the small number of received ratings, and will hinder the usefulness of ratings in decision making. To address these issues, we propose to use mappings to subjective opinions from the respective output results of recommender and reputation systems, so that the outputs are homogeneous and hence can be easily integrated and fused. Subjective logic [11] is a probabilistic framework capable of coping with the uncertainty in evidences.

We denote *recommendation values* and *reputation scores* as the outputs derived from recommender systems and reputation systems, respectively. Reputation systems produce reputation scores, e.g. in the range 0 – 5 stars. We assume a Bayesian/Dirichlet reputation system where the collected feedback ratings can be converted to subjective opinions, see Section 4.1 for details. Besides, a recommender system derives predictive recommendation values in the range $[0, 1]$[1], which will be converted to subjective opinions, see Section 4.2 for details. To integrate both reputation scores and recommendation values, we introduce the CasMin operator in Section 5. Finally, in Section 5.3 we show via an example that the advice produced by our approach is better than that produced by either a recommender system or a reputation system alone. To the authors' best knowledge, our work is the first effort in the literature to fuse outputs from recommender systems and reputation systems in order to produce better advice.

2 Related Work

Both recommender systems and reputation systems have been extensively and separately studied for decades. Recommender systems, as an essential component of e-commerce and online service applications, provide users with personalized high-quality recommendations to mitigate the well-known *information overload* problem. Collaborative filtering (CF) is a widely adopted technique to generate recommendations using the ratings of like-minded users [1]. The basic principle is that users with similar tastes in the past will also favour the same resources in the future. CF techniques can be classified into memory-based and model-based approaches. However, CF inherently suffers from two severe issues: *data sparsity* and *cold start* [1], due to the fact that users – especially new users – typically

[1] The ratings given by users are normalized in the range [0,1] if necessary.

have rated only a few resources. The uncertainty of predictions arises from such conditions where none or only few ratings are available for recommendations.

Many approaches have been proposed to reduce the uncertainty and improve the accuracy of recommendations. One direction of work is to develop new similarity measures in order to identify more reliable similar users [2]. However, the uncertainty due to few ratings of similar users cannot be handled. Model-based approaches [12,15] generally handle these issues better than memory-based approaches in terms of efficiency and accuracy. This is because global rating data is used to train a prediction model whereas memory-based approaches work on local rating data. The main drawback is that the trained static model is difficult to adapt to real-time increasing ratings. Another direction is to incorporate social relationships, such as trust-aware recommender systems [13]. The underlying principle assumption is that trust and taste are strongly and positively correlated [8]. Our work follows this general direction, i.e. to integrate taste and trust. The difference is that our approach takes the global perspective of resources (reputation scores) rather than the local perspective of users (social ties). In addition, the integration that we study is based on directly fusing taste and trust, rather than on moderating taste recommendations with trust.

Attacks against recommender systems are usually summarized as *shilling attacks* [3,4] where bogus rating profiles are injected to promote or degrade some resources. Although effective methods have been designed for memory-based CF, the research on robust model-based CF are not well studied [4]. Reputation systems are often built upon the assumption that user feedback may be fake and unreliable, and that various kinds of attacks could be conducted to influence the formation of reputation scores [10].

Reputation systems also suffer from the cold start problem. Remember that reputation systems generate scores based on feedback (or ratings) from members in a community [14,8]. When only little feedback is available, it is like a cold start situation where the derived reputation scores will be less reliable. Uncertainty can also increase when feedback greatly conflicts [17]. Users also tend to give mostly positive feedback which results in the derived reputation scores being less distinguishable and hence less useful.

In a nutshell, combining scored from both recommender systems and reputation systems can provide users with more accurate and robust online advice than either of the scores can in isolation. However, to date the integration of the two types of systems has not been studied in the literature.

3 Subjective Opinions

In this section, we will first introduce the notation and formation of subjective opinions used for fusing taste and trust. We also depict the mappings to binomial opinions from the multinomial ratings which is the common form of feedback in reputation systems and recommender systems.

3.1 Opinions Formation and Representation

A subjective opinion expresses belief about states of a state space called a *"frame of discernment"* or *"frame"* for short. In practice, a state in a frame can be regarded as a statement or proposition, so that a frame contains a set of statements. Let $X = \{x_1, x_2, \ldots, x_k\}$ be a frame of cardinality k, where x_i $(1 \leq i \leq k)$ represents a specific state. An opinion distributes belief mass over the reduced powerset of the frame denoted as $\mathcal{R}(X)$ defined as:

$$\mathcal{R}(X) = \mathcal{P}(X) \setminus \{X, \emptyset\} \ , \tag{1}$$

where $\mathcal{P}(X)$ denotes the powerset of X and $|\mathcal{P}(X)| = 2^k$. All proper subsets of X are states of $\mathcal{R}(X)$, but the frame X and the empty set $\emptyset$ are not states of $\mathcal{R}(X)$, in line with the hyper-Dirichlet model [5]. $\mathcal{R}(X)$ has cardinality $\kappa = 2^k - 2$.

An opinion is a composite function that consists of a belief vector $\boldsymbol{b}$, an uncertainty parameter u and base rate vector $\boldsymbol{a}$ that take values in the interval $[0, 1]$ and that satisfy the following additivity requirements.

$$\text{Belief additivity: } u_X + \sum_{x_i \in \mathcal{R}(X)} \boldsymbol{b}_X(x_i) = 1. \tag{2}$$

$$\text{Base rate additivity: } \sum_{i=1}^{k} \boldsymbol{a}_X(x_i) = 1, \ \text{ where } \ x_i \in X. \tag{3}$$

The opinion of user A over the frame X is denoted as $\omega_X^A = (\boldsymbol{b}_X, u_X, \boldsymbol{a}_X)$, where $\boldsymbol{b}_X$ is a belief vector over the states of $\mathcal{R}(X)$, u_X is the complementary uncertainty mass, and $\boldsymbol{a}_X$ is a base rate vector over X, all seen from the viewpoint of belief owner A.

The belief vector $\boldsymbol{b}_X$ has $(2^k - 2)$ parameters, whereas the base rate vector $\boldsymbol{a}_X$ only has k parameters. The uncertainty parameter u_X is a simple scalar. Thus, a general opinion contains $(2^k + k - 1)$ parameters. However, given that Eq.(2) and Eq.(3) remove one degree of freedom each, opinions over a frame of cardinality k only have $(2^k + k - 3)$ degrees of freedom. The probability projection of hyper opinions is the vector $\mathbf{E}_X$ expressed as:

$$\mathbf{E}_X(x_i) = \sum_{x_j \in \mathcal{R}(X)} \boldsymbol{a}_X(x_i/x_j)\, \boldsymbol{b}_X(x_j) + \boldsymbol{a}_X(x_i)\, u_X, \quad \forall x_i \in \mathcal{R}(X) \tag{4}$$

where $\boldsymbol{a}_X(x_i/x_j)$ denotes relative base rate, i.e. the base rate of subset x_i relative to the base rate of (partially) overlapping subset x_j, defined as follows:

$$\boldsymbol{a}_X(x_i/x_j) = \frac{\boldsymbol{a}_X(x_i \cap x_j)}{\boldsymbol{a}_X(x_j)}, \ \forall\, x_i, x_j \subset X. \tag{5}$$

Equivalent probabilistic representations of opinions, e.g. as Beta pdf (probability density function) or a Dirichlet pdf, offer an alternative interpretation of subjective opinions in terms of traditional statistics [11].

The term *hyper opinion* is used for a general opinion [11]. A *multinomial opinion* is when the belief vector $\boldsymbol{b}_X$ only applies to elements $x_i \in X$, not in $\mathcal{R}(X)$. *Binomial opinions* apply to binary frames and have a special notation as described below. Let $X = \{x, \overline{x}\}$ be a binary frame, then a binomial opinion about the truth of state x is the ordered quadruple $\omega_x = (b, d, u, a)$ where:

b, *belief*: belief mass in support of x being true;
d, *disbelief*: belief mass in support of $\overline{x}$ (NOT x);
u, *uncertainty*: uncertainty about probability of x;
a, *base rate*: non-informative prior probability of x.

The special case of Eq.(2) in case of binomial opinions is expressed by Eq.(6).

$$b + d + u = 1. \tag{6}$$

Similarly, the special case of the probability expectation value of Eq.(4) in case of binomial opinions is expressed by Eq.(7).

$$\mathrm{E}_x = b + au. \tag{7}$$

Binomial and multinomial opinions can be visualised as a point inside a simplex. Binomial opinions can thus be visualised as a point inside an equal sided triangle, and a trinomial opinion as a point inside a tetrahedron.

3.2 Mapping to Binomial Opinions

Multinomial opinions represent a generalisation of binomial opinions, and hyper opinions represent a generalisation of multinomial opinions. It can be necessary to project hyper opinions onto multinomial opinions, or to project multinomial opinions onto binomial opinions. For example, a reputation system where ratings are given in the form of 1-5 stars can represent reputation scores as multinomial opinions over a frame of five states, each of which represents a specific number of stars. In this case, a reputation score represented as a multinomial opinion can be projected to a binomial opinion over a binary frame, as explained below.

Let $X = \{x_1, \ldots, x_k\}$ be a frame where the k states represent linearly increasing rating levels, e.g. so that x_i represents an i-star rating. Let $Y = \{y, \overline{y}\}$ be a binary frame where y and $\overline{y}$ indicate *high quality* and *low quality* of a resource, respectively. Assume that a reputation score or recommendation value is represented as the multinomial opinion $\omega_X = (\boldsymbol{b}_X, u_X, \boldsymbol{a}_X)$ over the frame X, and that a binomial opinion $\omega_y = (b_y, d_y, u_y, a_y)$ over Y is required. The linear projection from the multinomial opinion ω_X on X onto the binomial opinion ω_y on Y is defined by:

$$\begin{cases} u_y = u_X \\ b_y = \sum_{i=1}^{k} b_{x_i} \frac{(i-1)}{(k-1)} \\ d_y = 1 - b_y - u_y \\ a_y = \sum_{i=1}^{k} a_{x_i} \frac{(i-1)}{(k-1)} \end{cases} \tag{8}$$

where $\frac{(i-1)}{(k-1)}$ indicates the relative weight, and hence the belief in a higher level x_i will have more weight in the formation of binary belief and base rate. As the default base rates on X is defined by $a_{x_i} = 1/k$, the default base rate of y is computed as follows:

$$a_y = \sum_{i=1}^{k} \frac{1}{k} \frac{(i-1)}{(k-1)} = \frac{1}{k(k-1)} \sum_{i=1}^{k} (i-1) = \frac{1}{k(k-1)} \frac{k(k-1)}{2} = 1/2 \; . \qquad (9)$$

The advantage of the projection of Eq.(8) is to provide the flexibility of analysing reputation scores and recommendation values independently of the frame cardinality.

4 Determining Opinions

This section details the procedures to derive subjective opinions, i.e. reputation scores and recommendation values from reputation systems and recommender systems, respectively.

4.1 Opinions Derived from Reputation Systems

A reputation system generally applies to services or goods that can be rated on one or multiple aspects, such as the set (*expected quality*, *seller communication*, *shipment timeliness*, *shipment charges*) in case of `eBay.com`. In case only a single aspect can be rated, it is typically the overall quality of a specific service or target. Each aspect can be rated with a specific level out of l levels such as 1 to 5 stars. It is also common that an aspect is rated with only two possible levels such as *Thumbs Up* and *Thumbs Down*.

Opinions for each aspect can be derived from such ratings. The frame for each aspect is the set of discrete rating levels, so that in case ratings can be given as 1 to 5 stars the frame has five states. Let X denote the frame of cardinality k, $r(x_i)$ be the number of ratings of type x_i, and $\omega_X = (\boldsymbol{b}_X, u, \boldsymbol{a}_X)$ be a multinomial opinion on X. The more ratings collected, the smaller the uncertainty becomes. The opinion ω_X can be determined from the ratings $r(x_i)$ according to Eq.(10):

$$\forall x_i \in X \quad \begin{cases} \boldsymbol{b}(x_i) = \frac{\boldsymbol{r}(x_i)}{W + \sum_{i=1}^{\kappa} \boldsymbol{r}(x_i)} \\ u \quad\quad = \frac{W}{W + \sum_{i=1}^{\kappa} \boldsymbol{r}(x_i)} \end{cases} \qquad (10)$$

where $W = 2$ is the non-informative prior weight with default value dictated by the requirement of having a uniform pdf (probability density functions) over binary frames when no evidence other than the domain base rate is available. The value would e.g. be $W = 3$ in case it were required to have a uniform pdf over a ternary frame. However, higher values for W make the probability distribution less sensitive to new evidence, so the value $W = 2$ is adopted [16].

An opinion derived according to Eq.(10) can thus represent a reputation score which can be mapped to a probability value, or to a simple user friendly representation e.g. in the form of 1 to 5 stars. A reputation score can also be adjusted as a function of time, reliability of the rater, etc. [6].

A rating is expressed as a specific level corresponding to a singleton state in the frame. In case there are more than two rating levels, the derived opinions are multinomial. In case only two types of ratings can be given, e.g. as *Thumbs Up* and *Thumbs Down*, the frame is binary so the opinions are binomial.

Reputation scores represented as multinomial opinions can be mapped to a binomial opinion according to Eq.(8) by assuming that each rating level corresponds to a value in the range $[0, 1]$, for more details see [9].

4.2 Opinions Derived from Recommender Systems

As an example, we describe a user-based CF method to generate recommendations [1]. The task of CF methods is to predict the preference (or rating) of a given resource (or item) for an active user, based on the rating histories of the active user as well as other participants in the community.

We keep the symbols s, v for users and i, j for items. Let $r_{v,i}$ denote a rating given by user v on item i, and let I_v denote the set of items that user v previously has rated. The mean rating of user v is computed by:

$$\overline{r}_v = \frac{1}{|I_v|} \sum_{i \in I_v} r_{v,i} \ . \tag{11}$$

Let $N_{s,j}$ denote the neighbourhood of an active user s constrained by having rated item j, i.e. the set of users who have rated (some of) the same items as user s and who have also rated the specific target item j. In general, only the top-K most similar users will be selected as the neighbourhood. The prediction $p_{s,j}$ for user s on target item j is computed by:

$$p_{s,j} = \overline{r}_s + \kappa \sum_{v \in N_{s,j}} w(s, v)(r_{v,j} - \overline{r}_v), \tag{12}$$

where κ is a normalisation factor and $w(s, v)$ represents the similarity between users s and v. There are several ways to compute user similarity, where the most commonly used method is the Pearson correlation coefficient [1]:

$$w(s, v) = \frac{\sum_{i \in I_{s,v}} (r_{s,i} - \overline{r}_s)(r_{v,i} - \overline{r}_v)}{\sqrt{\sum_{i \in I_{s,v}} (r_{s,i} - \overline{r}_s)^2 \sum_{i \in I_{s,v}} (r_{v,i} - \overline{r}_v)^2}}, \tag{13}$$

where $I_{s,v}$ represents the set of items that both users s and v has rated, and $w(s, v)$ is located in the range of $[-1, 1]$. A problem for similarity computation is that in case of none or only few commonly rated items, i.e. the size of $I_{u,v}$ is small, the computed similarity is not reliable which results the predicted value uncertain. This problem is called *cold start*. However, when representing predictions in terms of subjective opinions, the degree of uncertainty can be explicitly

expressed. Below is described a method by means of which subjective opinions can be derived from raw CF predictions.

The derivation is based on three intuitive assumptions. First, the uncertainty of the derived prediction opinion as expressed by Eq.(10) is a decreasing function of the number of ratings by similar users in $N_{s,j}$. Second, the probability expectation value of the derived prediction opinion as expressed by Eq.(7) is equal to the prediction of Eq.(12). Third, Eq.(6) also holds. Thus the set of equations below emerges.

$$\begin{cases} u_j^s = \frac{W}{W + \sum\limits_{v \in N_{s,j}} |I_{s,v}|} \\ p_{s,j} = b_j^s + a u_j^s \\ 1 = b_j^s + d_j^s + u_j^s \end{cases} \Rightarrow \begin{cases} u_j^s = \frac{W}{W + \sum\limits_{v \in N_{s,j}} |I_{s,v}|} \\ b_j^s = p_{s,j} - a u_j^s \\ d_j^s = 1 - b_j^s - u_j^s \end{cases} \tag{14}$$

where $W = 2$ is the non-informative prior weight. As before $N_{s,j}$ is the neighbourhood of user s constrained by having rated item j, and $I_{s,v}$ is the set of items that both users s and v have rated.

Although Eq.(14) is obtained from a user-based CF method, it can be easily adapted to item-based methods by:

$$\begin{cases} u_j^s = \frac{W}{W + \sum\limits_{i \in N_{s,j}} |U_{i,j}|} \\ p_{s,j} = b_j^s + a u_j^s \\ 1 = b_j^s + d_j^s + u_j^s \end{cases} \Rightarrow \begin{cases} u_j^s = \frac{W}{W + \sum\limits_{i \in N_{s,j}} |U_{i,j}|} \\ b_j^s = p_{s,j} - a u_j^s \\ d_j^s = 1 - b_j^s - u_j^s \end{cases} \tag{15}$$

where $N_{s,j}$ is the neighbourhood of item j, i.e. the set of items that have been rated by the users who also rated target item j as well as (some of) items rated by user s, and $U_{i,j}$ is the set of users who rated both items i and j. As before, generally only the top-K most similar items will be selected as the neighbourhood for rating prediction.

5 Combining Recommender and Reputation Values

After obtaining the subjective opinions from reputation systems and recommender systems respectively, the question is how they can be combined. We present the cascading minimum common belief fusion (CasMin) as a relatively conservative operator for fusing rating levels expressed as opinions. The detailed algorithm of CasMin fusion is also given below, and the usage is exemplified at the end of this section.

5.1 Cascading Minimum Common Belief Fusion

Various belief fusion models can be used to model specific situations. It is often challenging to determine the correct or the most appropriate fusion operator for a specific situation, see e.g. [7] for a discussion. We now present a new fusion model called *Cascading Minimum Common Belief Fusion* (CasMin) which is applicable when the states in the frame represent ordered levels.

When fusing belief masses on the highest order state in the frame, the greatest belief mass in one argument is reduced to match the belief mass in the other argument to produce the mutual minimum belief mass on that state. The amount of belief mass removed from the greatest belief mass is cascaded to the belief mass of the next inferior state in the frame and so forth until the lowest order state in the frame is reached. Belief mass from the least arguments can also be matched by uncertainty mass from the other argument, so that uncertainty typically is reduced, and belief mass in the lowest order states typically is increased.

An example situation is company investment where weighted ratings are given by analysts expressed as (1) *strong sell*, (2) *sell*, (3) *hold*, (4) *buy*, (5) *strong buy*. An investor might want to determine conservative company ratings based on the CasMin fusion model, so that in case a single analyst gives a low rating to a company on a specific level then the CasMin rating on that level is low even if all the other analysts give a high rating to the same company on that level. The conservative property of this fusion operator is useful in situations of possible bias in the arguments such as market analysis, where analysts tend to avoid negative opinions as they typically receive flack from the management teams and pressure that they may lose access to the companies that they cover.

The case that we are interested in is about giving advice that is confirmed by both recommendation values and reputation scores for resources. CasMin fusion provides a conservative fusion model for this situation because it takes the smallest of reputation score and recommendation value on each level, starting from the highest level, and on each level cascading the overshooting values down to the level below. A high CasMin fusion result, i.e. with large scores/values for high levels, can only be obtained when both reputation scores and recommendation values are high. In this way, the advice produced by CasMin fusion will be more conservative than that provided by reputation systems or recommender systems alone. We will describe the details of CasMin fusion in next sub section.

5.2 CasMin Fusion Operator

Let $X = \{x_1, \ldots x_k\}$ be an ordered frame where x_k is considered to be the highest order state predefined by a recommender or reputation system. The reduced powerset of X is denoted $\mathcal{R}(X)$ with cardinality κ. Assume that there are two opinions ω_X^A and ω_X^B over the frame X where the superscripts A and B represent the belief owners. The two opinions can be mathematically merged using the CasMin operator which in expressions is denoted as: $\omega_X^{(A \underline{\downarrow} B)} = \text{CasMin}(\omega_X^A, \omega_X^B)$.

The CasMin operator requires binomial or multinomial opinions, so in case of hyper opinion arguments, first project to binomial or multinomial opinions as described by Eq.(16), where the beliefs of the hyper opinion ω'_X are denoted as $\boldsymbol{b}'_X$, and the the beliefs of the multinomial opinion ω_X are denoted as $\boldsymbol{b}_X$.

$$\boldsymbol{b}_X(x_i) = \sum_{x_j \in \mathcal{R}(X)} \boldsymbol{a}_X(x_i/x_j)\, \boldsymbol{b}'_X(x_j)\ , \forall x_i \in X, \tag{16}$$

With multinomial opinions arguments the CasMin fusion operation proceeds according to the algorithm of Fig.1. Specifically, it first acts on the belief on the

highest level state x_k and finally on the belief on lowest level state x_1. Line 2 ensures that the belief on the A-argument is always greater than that of the B-argument, by executing a swap operation if necessary. For each level x_i, there are two possible cases, i.e. whether the A-argument's belief is less than or equal to the sum of the B-argument's belief and uncertainty (lines 3-7) or not (lines 8-13). In either case, (a part of) the B-argument's uncertainty can compensate for it's belief value being less than that of the A-argument (lines 4, 9-10). The remaining minimum belief value will be assigned to both A and B's arguments (lines 5, 12), and then the differences between the new and previous belief values (lines 6, 11) will be cascaded to the next inferior state x_{i-1} (line 14). This procedure will be repeated until the frame is finished. Finally, user A's new opinion represents the fused result and will be returned (line 16).

1. FOR i = k to 2 DO {
2. IF $\boldsymbol{b}_X^A(x_i) \leq \boldsymbol{b}_X^B(x_i)$ THEN {Swap(ω_X^A, ω_X^B);}
3. IF $u_X^B > (\boldsymbol{b}_X^A(x_i) - \boldsymbol{b}_X^B(x_i))$ THEN {
4. $u_X^B = u_X^B - (\boldsymbol{b}_X^A(x_i) - \boldsymbol{b}_X^B(x_i))$;
5. $\boldsymbol{b}_X^B(x_i) = \boldsymbol{b}_X^A(x_i)$;
6. $b_{\text{cascade}} = 0$;
7. }
8. ELSE {
9. $\boldsymbol{b}_X^B(x_i) = \boldsymbol{b}_X^B(x_i) + u_X^B$;
10. $u_X^B = 0$;
11. $b_{\text{cascade}} = \boldsymbol{b}_X^A(x_i) - \boldsymbol{b}_X^B(x_i)$;
12. $\boldsymbol{b}_X^A(x_i) = \boldsymbol{b}_X^B(x_i)$;
13. }
14. $\boldsymbol{b}_X^A(x_{i-1}) = \boldsymbol{b}_X^A(x_{i-1}) + b_{\text{cascade}}$;
15. }
16. $\omega_X^{(A \underline{\downarrow} B)} = \omega_X^A$;

Fig. 1. Algorithm for the CasMin belief fusion operator

The CasMin operator is commutative, associative and idempotent, and a totally uncertain opinion acts as the neutral element for the CasMin operator.

5.3 Example

We consider the case of providing advice about hotels through a web site such as e.g. `tripadvisor.com`. It is assumed that a recommender system tracks user preferences, and that a reputation system allows users to rate hotels.

With the method described in Eq.(10) the reputation system can produce scores expressed as multinomial opinions. With the method described in Eq.(8) the multinomial opinions can be transformed into binomial opinions.

The recommender system can also use a multi-aspect and multi-level representation of ratings. A user can rate general satisfaction high even if another aspect such as cleanliness is rated low, e.g. in case cleanliness is not an important preference for the user. The recommender system is thus able to identify hotels that match the users personal preference. The recommendation values for each hotel and each user are expressed as binomial opinions using Eq.(14) or Eq.(15).

The recommender system identifies a list of hotels based on the ratings given by the user and other travelers. The recommender system can predicted that the user will like the hotels because other users with similar tastes have rated the hotels with satisfaction. In contrast, the reputation system offers community-wide scores for each hotel. The CasMin operator produces conservative results in the sense that hotels must simultaneously have high recommendation values and high reputation scores. The numerical example of Table 1 illustrates the result of fusing two such opinions according to the CasMin algorithm of Fig.1.

Table 1. Fusion of reputation scores and recommendation values

Hotel	Rep. Ratings	Multinomial Rep. Score	Binomial Rep. Score	Rec. Value	CasMin Advice
Hotel I	$r(x_5) = 50$	$b_{x_5} = 0.65$	$b = 0.81$	$b = 0.1$	$b = 0.30$
	$r(x_4) = 10$	$b_{x_4} = 0.13$	$d = 0.16$	$d = 0.7$	$d = 0.70$
	$r(x_3) = 10$	$b_{x_3} = 0.13$	$u = 0.03$	$u = 0.2$	$u = 0.00$
	$r(x_2) = 0$	$b_{x_2} = 0.00$			
	$r(x_1) = 5$	$b_{x_1} = 0.06$			
		$u_X = 0.03$			
Hotel II	$r(x_5) = 5$	$b_{x_5} = 0.06$	$b = 0.16$	$b = 0.7$	$b = 0.19$
	$r(x_4) = 0$	$b_{x_4} = 0.00$	$d = 0.81$	$d = 0.1$	$d = 0.61$
	$r(x_3) = 10$	$b_{x_3} = 0.13$	$u = 0.03$	$u = 0.2$	$u = 0.20$
	$r(x_2) = 10$	$b_{x_2} = 0.13$			
	$r(x_1) = 50$	$b_{x_1} = 0.65$			
		$u_X = 0.03$			
Hotel III	$r(x_5) = 50$	$b_{x_5} = 0.65$	$b = 0.81$	$b = 0.7$	$b = 0.81$
	$r(x_4) = 10$	$b_{x_4} = 0.13$	$d = 0.16$	$d = 0.1$	$d = 0.19$
	$r(x_3) = 10$	$b_{x_3} = 0.13$	$u = 0.03$	$u = 0.2$	$u = 0.00$
	$r(x_2) = 0$	$b_{x_2} = 0.00$			
	$r(x_1) = 5$	$b_{x_1} = 0.06$			
		$u_X = 0.03$			

Table 1 shows the results of analysing three separate hotels called Hotel I, II and III, respectively. In case of Hotel I and Hotel II where the recommendation values and reputation scores are in conflict, the fused belief value is small. The only strong result is for Hotel III where both the recommendation value and reputation score are positive. In addition, as shown in cases of Hotel I and Hotel III, it is often the case for reputation systems that the scores have a strong positive bias, reducing the utility and discriminating power of the reputation system.

The advantage of combining recommender systems and reputation systems is to amplify the discriminating power.

6 Conclusions

Since both recommender systems and reputation systems support decision making we believe that combining both types of systems may produce better advice than any individual systems can do alone. However, the significant differences in the underlying theory and implementation make such integration challenging. In this paper, we proposed a method to represent reputation scores and recommendation values within the framework of subjective logic. We also proposed the new CasMin fusion operator in order to fuse the results from recommender and reputation systems in a conservative fashion, i.e. so that high results can only be obtained when both reputation scores and recommendation values are high for a given resource. The proposed method was illustrated with a hypothetical example. In future research we intent to apply the method to real data in order to judge its usefulness.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering 17(6), 734–749 (2005)
2. Ahn, H.J.: A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. Information Sciences 178(1), 37–51 (2008)
3. Burke, R., O'Mahony, M.P., Hurley, N.J.: Robust collaborative recommendation. In: Recommender Systems Handbook, pp. 805–835 (2011)
4. Gunes, I., et al.: Shilling attacks against recommender systems: a comprehensive survey. Artificial Intelligence Review, 1–33 (2012)
5. Hankin, R.K.S.: A Generalization of the Dirichlet Distribution. Journal of Statistical Software 33(11), 1–18 (2010)
6. Jøsang, A., Bhuiyan, T., Xu, Y., Cox, C.: Combining Trust and Reputation Management for Web-Based Services. In: Proceedings of the 5th International Conference on Trust, Privacy & Security in Digital Business (TrustBus 2008), Turin (September 2008)
7. Jøsang, A., Costa, P.C.G., Blash, E.: Determining Model Correctness for Situations of Belief Fusion. In: Proceedings of the 16th International Conference on Information Fusion (FUSION 2013), Istanbul (July 2013)
8. Jøsang, A., et al.: A survey of trust and reputation systems for online service provision. Decision Support Systems 43(2), 618–644 (2007)
9. Jøsang, A., Luo, X., Chen, X.: Continuous Ratings in Discrete Bayesian Reputation Systems. In: Karabulut, Y., Mitchell, J., Herrmann, P., Jensen, C.D. (eds.) Trust Management II. IFIP, vol. 263, pp. 151–166. Springer, Boston (2008)
10. Jøsang, A., Golbeck, J.: Challenges for Robust of Trust and Reputation Systems. In: Proceedings of the 5th International Workshop on Security and Trust Management (STM 2009), Saint Malo (September 2009)

11. Jøsang, A., Hankin, R.K.S.: Interpretation and Fusion of Hyper Opinions in Subjective Logic. In: Proceedings of the 15th International Conference on Information Fusion (FUSION 2012), Singapore (July 2012)
12. Koren, Y.: Collaborative filtering with temporal dynamics. Communications of the ACM 53(4), 89–97 (2010)
13. Massa, P., Avesani, P.: Trust-aware recommender systems. In: Proceedings of the 2007 ACM Conference on Recommender Systems, RecSys 2007 (2007)
14. Mui, L., Mohtashemi, M., Ang, C.: A probabilistic rating framework for pervasive computing environments. In: Proceedings of the MIT Student Oxygen Workshop, SOW 2001 (2001)
15. Shi, Y., Karatzoglou, A., Baltrunas, L., Larson, M., Hanjalic, A., Oliver, N.: Tfmap: Optimizing map for top-n context-aware recommendation. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2012), pp. 155–164 (2012)
16. Walley, P.: Inferences from Multinomial Data: Learning about a Bag of Marbles. Journal of the Royal Statistical Society 58(1), 3–57 (1996)
17. Wang, Y., Singh, M.P.: Evidence-based trust: A mathematical model geared for multiagent systems. ACM Transactions on Autonomous and Adaptive Systems (TAAS) 5(4), 14 (2010)

Pushing Constraints into a Pattern-Tree

Andreia Silva and Cláudia Antunes

Department of Computer Science and Engineering
Instituto Superior Técnico – Technical University of Lisbon
Lisbon, Portugal
{andreia.silva,claudia.antunes}@ist.utl.pt

Abstract. Frequent Itemset Mining, or just pattern mining, plays an important role in data mining, aiming for the discovery of frequent co-occurrences in data. However, existing techniques still suffer from two bottlenecks that difficult the analysis and actual application of their results: they usually return a large number of patterns, and these patterns usually do not reflect user expectations. The most accepted and common approach to minimize these drawbacks is to define the user needs through constraints, and use them to filter and return less but more interesting patterns. Several types of constraints have been proposed in the literature, along with some algorithms that are able to incorporate them. However, there is no unified algorithm able to push any type of constraint. In this work we propose to push constraints into pattern mining through the use of a pattern-tree structure to efficiently store, check and prune the patterns. We define in detail a set of strategies to push each type of constraint, and a generic algorithm that is able to combine these strategies and incorporate any constraint into a pattern-tree.

Keywords: Pattern Mining, Pattern-Tree, Constraints, Monotonicity.

1 Introduction

An important line of research that has gained attention in recent years consists in the development of data mining techniques that are able to incorporate the existing domain knowledge into the search process. Indeed, one of the common criticisms pointed out to data mining, in particular to pattern mining, is the fact that it generates a huge number of patterns, independent of user expertise, making it very hard to understand and use the results.

The use of constraints to filter the results is the most common and used approach to focus the algorithms only on what is really interesting. They are an efficient way to reduce the number of returned patterns and increase the efficacy of pattern mining, by returning less but more interesting results, in the user and application points of view.

Several types of constraints have been proposed in the literature, along with a set of different algorithms that are able to push each individual type of constraint. How to enforce these constraints into pattern mining is non trivial, and depends heavily on the constraint in question. Fortunately, studies show that a large

V. Torra et al. (Eds.): MDAI 2013, LNAI 8234, pp. 139–151, 2013.

number of constraints in pattern mining have some particular properties that allow the exploration of efficient strategies to prune the search space.

In this paper we propose set of strategies to push constraints that follow certain properties into pattern mining algorithms, through the use of a tree to store the patterns. We also propose a generic algorithm, named *CoPT* (Constraint Pushing into a Pattern-Tree), that combines and implements these strategies and is able to incorporate any constraint, taking advantage of its properties. By filtering the results according to the user constraints, *CoPT* returns less and more interesting results.

We formalize the problem in section 2, and describe the existing constraint properties in section 3. Section 4 presents the proposed strategies and algorithm, and experiments are shown in section 5. Finally, section 6 concludes the work.

2 Problem Statement

The oldest and most studied constraint in pattern mining (PM) is the minimum support threshold [1], which states that, to be interesting, a pattern must have a support higher than the given threshold. In fact, what we call traditional PM corresponds to the discovery of *frequent* itemsets from data. Hence, constrained PM is perceived as the use of constraints beyond the minimum support, i.e. the discovery of *frequent* itemsets that *satisfy some constraints*.

Formally, let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of distinct literals, called items. A subset of items is denoted as an itemset. A superset of an itemset X is also an itemset, that contains all items in X and more. The support of an itemset X is the number of occurrences in the dataset, and X is frequent if its support is no less than a predefined minimum support threshold: $sup(X) \geq \sigma \in [0, 1]$.

Definition 1. *A constraint C is a predicate on the powerset of I [9], i.e. $C : 2^I \mapsto \{true, false\}$. An itemset X satisfies a constraint C, if $C(X) = true$.*

A pattern corresponds to a frequent itemset that satisfies the constraint C, i.e. if $sup(X) \geq \sigma \wedge C(X) = true$. And the problem of *constrained frequent pattern mining* is to find all patterns in a dataset.

3 Constraints

Constraints are a common way to represent user expectations [2]. Essentially, constraints are filters on the data or on the results, that capture application semantics and allow the users to somehow control the search process and focus de algorithms on what is really interesting.

The use of constraints in data mining is mostly associated with the PM task. In this context, constraints are an efficient way to reduce the number of patterns and increase the efficacy of the mining process, by returning less but more interesting results, in the user and application points of view.

There are many types of constraints. According to their semantics and form, they can be divided in several categories, from *content* constraints, filtering the

content of the discovered patterns, to *length* constraints, limiting the number of items in each pattern, and to more complex types, such as *temporal* constraints, taking into account the temporal dimension.

How to enforce these constraints into pattern mining is non trivial, and depends heavily on the constraints in question. Performing an extensive search is not a viable solution mostly due to the size of the search space.

Fortunately, studies show that constraints have some properties that provide efficient strategies to prune the search space and improve the selection of patterns that satisfy them. These *"nice"* properties [8] are described next.

In its basis, a constraint can be *anti-monotonic*, *monotonic*, or none.

Anti-monotonicity (AM). *A constraint is said anti-monotonic if and only if, whenever an itemset X violates it, so does any superset of X.*

The minimum support threshold is the best known and simple example of an AM constraint [1], according to which an itemset is frequent if its support is greater or equal to a user defined threshold. It is AM in the sense that if an itemset is infrequent, so does any of its supersets.

Anti-monotonicity, if used actively, can drastically reduce the search space [10,7,8,3]. It allows the algorithms to prune earlier, with less effort, minimizing the computational cost, and at the same time maximizing the efficacy of the results. However, it is not possible to ensure the efficiency of pushing this type of constraints, since it depends on their selectivity [3]: the less selective, the less can be discarded, and the less efficient it usually is.

Monotonicity (M). *A constraint is said monotonic if and only if, whenever an itemset X satisfies it, so does any superset of X.*

An example is an item constraint of the type $C(X) = (\{i, j\} \subseteq X)$. If an itemset satisfies the constraint (i.e. contains all the known items), all supersets also satisfy it, because they contain the same items and more. However, if an itemset violates it, a superset can satisfy it, by introducing the missing items.

Monotonic constraints can also be used to improve the efficiency of pattern mining, by avoiding multiple unnecessary tests [10,7,8,3].

In addition, constraints can also be, at the same time, *succinct*.

Succinctness (S). In its essence, *a constraint is succinct if it is possible to enumerate all possible patterns, based only on items from the alphabet I* [7].

A simple example is the value constraint $C(X) = (X.price \leq €100)$. It is succinct because we can select from the alphabet all items X_1 with $price \leq €100$, and the itemsets that satisfy the constraint are exactly only those in the strict powerset of X_1. This is a *succinct anti-monotonic* constraint (SAM), since supersets of itemsets with some item with $price > €100$ will never satisfy it.

Taking into account the succinctness of a constraint allows the algorithms to prune more and earlier, in the case of SAM constraints, and to avoid more constraint checks, in the case of a SM constraint [7,5,3].

There are, of course, constraints that are not overall anti-monotonic neither monotonic, and therefore it is not easy to push them in an efficient way. However,

with some assumptions, many of them can be converted and treated as that. In this sense, constraints can also be *prefix-monotone* or *mixed-monotone*.

Prefix-Monotonicity. *A constraint is prefix-monotone*[1] *if there is an order of items that allows the algorithms to treat it as anti-monotonic or monotonic [9].* By fixing an order on items, each transaction can be seen as a sequence, and therefore we can use the notion of prefixes and suffixes, as the first or last items in the ordered transaction, respectively.

A constraint is prefix-monotone, if it is *prefix anti-monotonic* (PAM) or *prefix monotonic* (PM). Formally, a constraint C is prefix anti-monotonic (resp. prefix monotonic) if there is an order R over the set of items, and assuming each itemset $X = i_1 i_2 ... i_n$ is ordered accordingly to order R, such that, whenever an itemset X violates (resp. satisfies) C, so does any itemset with X as prefix ($X' = X \cup \{i_{n+1}\} = i_1 i_2 ... i_n i_{n+1}$).

For example, an aggregate constraint like $C(X) = (avg(X) \geq 20)$, is not monotonic neither anti-monotonic. But, if we order the items in a value-descending order, an itemset X has higher average than its supersets X'. This means that, if X violates C, also will all its supersets X'. Thus, C is prefix anti-monotonic. With a similar reasoning, the same C is prefix monotonic if items are ordered in a value-ascending order. In this case, if X satisfies C, also all its supersets X'.

Mixed-Monotonicity. *A constraint is mixed-monotone if it can be considered both anti-monotonic and monotonic, at the same time, for different groups of possible values (positive and negative)*[6].

Formally, let the set of items I be divided into two disjoint groups based on their monotonicity relating to a constraint C: let I^{AM} be the set of anti-monotonic items, and I^M, the set of monotonic items. Then, a constraint is mixed monotone if, for any itemset X: (a) whenever X satisfies C, all supersets of X formed by adding items from the I^M group, also satisfies C; and (b) whenever X violates C, all supersets of X formed by adding items from the I^{AM} group, also violates C.

This property was proposed in particular for *sum* constraints of the form $sum(X)\theta v$, where itemset X may contain positive or negative numerical values (or zero), v is also a positive or negative constant (or zero), and $\theta \in \{>, \geq, <, \leq\}$. The aggregate constraint $C(X) = (sum(X) \geq v)$, for example, is monotonic for positive values (including zero), and anti-monotonic for negative values.

Most of existing constraints fall into one of these properties. This makes it possible to generalize and create strategies for pushing constraints that follow them. There are several algorithms for pushing constraints of a specific type or that have a specific property. The problem is that those algorithms are specific, and there is no general algorithm capable of incorporating any constraint, and still taking advantage of constraint properties at the same time.

[1] Prefix-monotone constraints were first proposed with the name of *convertible constraint* [9,8].

Srikant et al. [10,11] were the first to introduce item constraints, the first different from minimum support. They proposed three apriori-based algorithms, *MultipleJoins*, *Reorder* and *Direct*, that are able to deal with boolean combinations of these constraints, i.e. of the form $i \in S$ or $i \notin S$. Succinct constraints were first proposed by Ng et al. [7], as well as an apriori-based algorithm, called *CAP* (Constrained APriori). Later on, [5] proposed *FPS* (FP-tree based mining of Succinct constraints) that uses the same approach but in a pattern-growth algorithm. These algorithms are only able to push succinct constraints. Pei et al. [9] proposed prefix-monotone constraints as well as a pattern growth algorithm, *FIC* (Frequent Itemset mining with Convertible constraints), that is able to push them into the discovery process, by growing only valid prefixes. Finally, mixed monotone constraints were recently proposed by Leung et al. [6], in particular for *sum* constraints, along with a pattern-growth algorithm *FPM* (Frequent Pattern mining for Mixed monotone constraints).

4 Push Constraints in a Pattern Tree

In this paper we propose a set of strategies to push constraints that have *nice* properties into pattern mining, through the use of a pattern-tree structure. These are post-processing strategies that, combined with the properties of the pattern-tree, make it possible to efficiently filter the results accordingly to any constraint.

We also propose an algorithm, called *CoPT* (Constraint Pushing into a Pattern-Tree), that implements these strategies and is able to incorporate any of those constraints and therefore return less and more interesting results. As a post-processing algorithm, any traditional pattern mining algorithm can be used before to search for frequent itemsets, and its results, kept in a pattern-tree, can be processed directly by *CoPT*.

A pattern-tree is a compact prefix tree structure that holds information about patterns. Each node contains an item and a support, and edges link items that occur together, forming the itemsets. Therefore, each node in a pattern-tree corresponds to an itemset, composed of the items from the root to this node, and the support attached to this node. As a prefix tree, itemsets that share the same prefix also share the same nodes corresponding to that prefix.

Since there are often a lot of sharing of frequent items among patterns, the size of the tree is usually much smaller than having them in a list or a table, and the search for an itemset is usually much faster.

Note that if an itemset $(a, b, c) : 5$ is a frequent itemset, then both a, b, c, (a, b), (a, c) and (b, c) are also frequent, with support higher or equal to 5, and therefore they are also in the pattern-tree. This means that, for each itemset in the tree, all elements of its strict powerset are also in the tree. This may seam undesirable or redundant at a first look, but it is a important property that facilitates the pruning of the tree while searching.

4.1 Constraint Pushing Strategies

In order to push constraints into a pattern-tree, we define a set of strategies that can be used, based on constraint properties. A naive approach is to perform a simple depth-first search (DFS) to traverse the tree and test all nodes for all types of constraints (note that, when we test a node for a constraint, we mean that we test the itemset corresponding to that node). However, not all nodes need to be tested. For example, if an itemset of a node violates an AM constraint, no superset will satisfy it, and therefore there is no need to test the children of that node, neither to keep them in the tree. Hence, we can take advantage of constraint properties and perform a constrained DFS, stopping the search at some points and avoiding unnecessary tests.

Another possible approach is to push the constraint right before inserting each itemset in the pattern-tree. However, while this may be better in terms of memory, because the pattern-tree would be smaller, this means that we have to test every itemset. By scanning the tree, we may skip the constraint checking of a lot of itemsets.

Furthermore, constraints can be used, not only to filter the results, but also to prune the pattern-tree and remove invalid itemsets for future accesses.

We describe next the strategies for pushing constraints with each property.

Anti-Monotonicity: Pushing an AM constraint (C_{AM}) is pretty straightforward. While performing a DFS, if the node:

(a) Satisfies C_{AM}: keep it in the tree and return it as a pattern;
(b) Violates C_{AM}: there is no need to search its subtree because all supersets also violate the constraint. Therefore we can prune the tree and remove this node, as well as all of its children.

Monotonicity: To incorporate a monotonic constraint (C_M), we cannot remove nodes that violate it, because the supersets of this node (its children) can satisfy it. So, while traversing the tree, if the node:

(a) Satisfies C_M: keep it in the tree and return it as a pattern. Do the same for each node in its subtree, without testing for the constraint; (Note that if we are just pruning the tree, not yet returning the patterns, we do not even need to scan the subtree, because all supersets satisfy the constraint, and there is nothing to remove.)
(b) Violates C_M: If it is a leaf node (has no supersets), we can remove it, as well as all parents that become a leaf because of this elimination. If it is not a leaf, continue the search to its children, since they can satisfy the constraint.

Succinctness: In the presence of a succinct constraint, we can apply the strategies for C_{AM} or C_M, whether it is succinct anti-monotonic (C_{SAM}) or succinct monotonic (C_{SM}), respectively. However, the succinctness of a constraint allow us to know, from the outset, which items satisfy or not satisfy the constraint. Therefore, we can use that to take advantage of this property, and obtain a more efficient search.

With this in mind, we can first divide the items into two groups: items that satisfy or are necessary to the satisfaction of the constraint, I^s; and items that violate, or are not necessary to the satisfaction of the constraint, I^v. And before inserting itemsets into the pattern-tree, we can order the itemsets according to those groups.

C_{SAM}: With a SAM constraint, single items that violate it can be discarded. If we order items in itemsets so that I^v appears before I^s (I^v closer to the root and I^s to the leafs), when applying the C_{AM} strategy, we only need to check the first level of the pattern-tree. If the node violates the constraint, remove it and its sub-tree; if the node satisfies, all of its children will also satisfy, because they belong to I^s, so we can return all of them as patterns, without testing for the constraint.

C_{SM}: In the case of a SM constraint, I^s contains the mandatory items and I^v the optional items. If an itemset with items from I^s satisfy the constraint, all of its supersets formed by adding items from I^s or I^v also satisfy it. Itemsets with items only from I^v violate the constraint. In this sense, if we order itemsets so that items from I^s appear first than items from I^v, when applying the C_M strategy, we only need to do it until the first node from I^v, because if we arrive to a node like this and still need to test the constraint, it means it has not been satisfied by items from I^s, and next items also cannot satisfy it because they are optional, therefore we do not need to test anything more.

Prefix-Monotonicity: Since prefix-monotone constraints can only be treated as AM (C_{PAM}) or M (C_{PM}) constraints if items are ordered by a particular order, we just need to sort the itemsets according to that order before inserting them in the pattern-tree, and apply the C_{AM} or C_M strategy, respectively. Otherwise, we have to traverse the whole tree and check all nodes for the constraint.

Mixed-Monotonicity: Mixed-monotone constraints (C_{Mix}) are both AM and M, for different groups of values. In this case, we just have to divide the items into those groups: I^{AM} and I^M, and put I^M before I^{AM} in the tree, i.e. sort itemsets so that items from the I^M group appear above items from I^{AM}. The idea is to start with the C_M strategy, until a node that satisfies it, or a node from I^{AM} appears. From that node, we can apply the C_{AM} strategy and prune invalid nodes from its sub-tree. So, for each node, start with the monotone strategy:

1. Monotone strategy: If the itemset:
   (a) Satisfies C_{Mix}: Keep it in the tree and return it as a pattern. We can now change to the anti-monotone strategy and proceed;
   (b) Violates C_{Mix}: If it is a leaf, remove it, as well as all parents that become a leaf. If it is a node from I^{AM}, remove it, and all its sub-tree. Otherwise, continue to its children.
2. Anti-monotone strategy: If the itemset satisfies the constraint, keep it in the tree and return it as a pattern. If it violates the constraint, prune the tree from this node removing it and all its children.

4.2 Algorithm

Since there are a lot of similarities between the strategies presented above, they can be combined into one single generic strategy or algorithm. We propose therefore the algorithm *CoPT* (Constraint Pushing into a Pattern-Tree), that is able to efficiently and effectively push any constraint into a pattern-tree.

Algorithm 1. *CoPT* Pseudocode

```
Input: Support σ, Dataset D, Constraint C
Output: All frequent itemsets that satisfy C
  if C has order then
    order ← best order for C
  p-tree ← empty tree with order order
  run a pattern mining algorithm with σ and D, and insert results into the p-tree
  L ← pushConstraint(p-tree, C)
  return L

  Patterns ← pushConstraint(Pattern-Tree p-tree, Constraint C)
  L ← ∅
  for all Node N, children of the root of p-tree do
    remove? ← push(N, C, {}, L)
    if remove? is true then
      remove N from root
  return L

  boolean ← push(Node N, Constraint C, Itemset itset, Patterns L)
  isPattern? ← true, current ← itset ∪ N.item : N.support
  if Constraint is not null then
    if C is Succinct and N.item ∈ C.I^v then
      return true // remove this node
    if current satisfies C then
      if C is Monotonic or C is Succinct then
        if C is Mixed then
          Change C to AM for next children
        else
          C ← null // no need to test any children
    else
      if C is Anti-monotonic then
        return true
      isPattern? ← false
  if isPattern? is true then
    L ← L ∪ current
  for all Node T, children of N do
    remove? ← push(T, C, current, L)
    if remove? is true then
      remove T from N
  if isPattern? is false and N is leaf then
    return true
  return false
```

The pseudo-code of the algorithm is presented in Algorithm 1.

Essentially, to push a constraint, *CoPT* first checks what is the order of items for that constraint, and creates an empty pattern-tree with it (if there is no order, items are put in the pattern-tree in a support-descending order, which is known to improve the compactness of the tree [4]). Then a traditional pattern mining algorithm can run over the dataset to get frequent itemsets. While running it, results are inserted in the pattern-tree (note that the algorithm does not need any change. Only the pattern-tree knows how to sort and insert the itemsets). After that, we can push the constraint into the pattern-tree.

So, following function **push**, for each node, *current* corresponds to the itemset composed of items from root to this node, and until proved otherwise, it is a pattern. If there is no constraint to check (e.g. a C_M already satisfied), add it as a pattern and do the same for all children. Otherwise, (1) if the constraint C is succinct (SAM or SM) and the node violates it, it can be removed; (2) if current satisfies C: (a) C is mixed and we can change the strategy to AM; (b) C is monotonic and no child needs testing; or (c) C is succinct AM, and only the first level of the tree needs testing. (3) if current violates C, it is not a pattern, and if C is AM we can prune the tree from here. After checking the constraints, if the node was not pruned, we can test its children. Finally, after pushing C into the children, if the node is not a pattern and is a leave, we can remove it.

5 Performance Evaluation

The goal of these experiments is to analyze the behavior of our algorithm in the presence of all types of constraints, and prove that *CoPT* is able to effectively and efficiently push them into a pattern-tree, taking advantage of its properties.

In these experiments we use a transaction database automatically generated by the program developed at IBM Almaden Research Center [1]. The dataset has $10k$ transactions, with an average of 25 items per transaction and a domain of 1000 items (with values from zero to 1000). In addition, in order to test the mixed-monotone constraint, we consider an equivalent dataset but with negative values, by making values vary from -500 to 500.

We analyze the time needed to push the constraints on these datasets, as well as the size of the pruned pattern-tree and the number of constraint checks the algorithm needed to make. Since the behavior of the algorithm can depend on the selectivity of the constraints, we use it in our experiments. Selectivity is defined as the ratio of frequent itemsets that violate the constraint, over the total number of frequent itemsets, i.e. how much we can filter. We also tested several minimum supports, and since results are consistent, we present the results for a support of 0.5%, and results presented correspond to the average of several runs with different constraints with equivalent selectivity. Also, to have a term of comparison, we test our algorithm against a version that checks all nodes for the constraints (i.e. not taking into account constraint properties), named $CoPT^+$.

The traditional pattern mining algorithm used was FP-Growth [4], since it is an efficient algorithm that does not suffer from the candidate generation problem.

The computer used to run the experiments was an Intel Core i7 CPU at 2GHz (Quad Core), with 8GB of RAM and using Mac OS X Server 10.7.5 and the algorithm was implemented using the Java (JVM version 1.6.0_37).

As the basis of our algorithm, the pattern-tree plays an important role in these experiments. Independently of the constraint, the size of the pattern-tree after pushing the constraint is most of the times smaller than the original one, because it does not contain leaves that violate it. As the selectivity increases, the more itemsets violate the constraint. In the case of an AM constraint (either AM, SAM or PAM), the number of nodes in the final pattern-tree corresponds to the number of frequent itemsets that satisfy the constraint (the number of patterns). In the case of M constraints (M, SM, PM and Mix), this might not be true, since nodes that violate the constraint have to be kept if there is some superset that satisfy the constraint.

In fact, the time needed by the traditional unconstrained pattern mining algorithm corresponds to the bulk of time needed: about 5 hours for these settings. After having the patterns in a pattern tree, and due to its compact nature, it is fast (compared to pattern mining) to look for patterns that satisfy some constraint, even constraints with no nice properties ($CoPT^+$) and with less selectivity. Fig. 1, 3 and 5 show the time needed for pushing AM, M and Mix constraints into a pattern-tree, respectively. We can see there that pushing constraints taking into account their properties ($CoPT$) takes less time than testing all nodes ($CoPT^+$), for every constraint property. For all AM and $Succinct$ constraints, as the selectivity increases, the time needed to prune the tree decreases, since they can eliminate earlier more itemsets that violate it. On the contrary, M and SM constraints tend to increase the time needed, because they take more time until finding itemsets that satisfy it (so that they can stop checking the constraint). The time is therefore related to the number of constraint checks.

These constraint checks are also an important part of the algorithm, since theoretically, taking advantage of constraint properties results in less tests. Fig. 2, 4 and 6 show interesting results about that. For AM constraints (AM and PAM), the number of tests decreases with the increase of selectivity, because the number of itemsets that violate and can be discarded increases. For M constraints (both M and PM) the trend is reversed. This happens because the M strategy only stops checking when itemsets satisfy the constraints. If there are more itemsets that violate (more selectivity), more itemsets need to be tested. Using the succinctness of constraints brings the most improvements, both in time needed and in constraint checks avoided. The number of tests for succinct constraints does not depend on the selectivity, because only and all nodes of the first level of the tree need to be tested (in this case, about 800 nodes). Note that the tree has more than 300 thousand nodes, and only 800 need to be checked. Finally, Mix constraints have a "mix" of the behavior of M and AM constraints. As the selectivity increases, more itemsets belonging to both groups of values violate the constraint, and the more violating itemsets from I^{AM}, the more can be pruned, but the more violating itemsets from I^M, the more constraint checks are required. Hence, there is a tradeoff between both strategies.

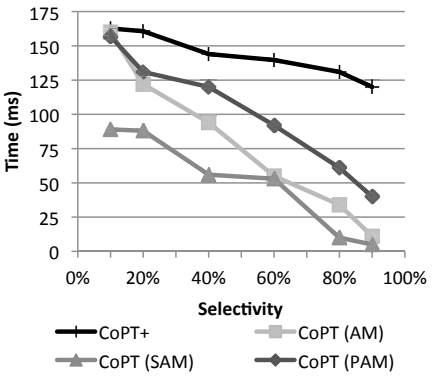


Fig. 1. Time with AM

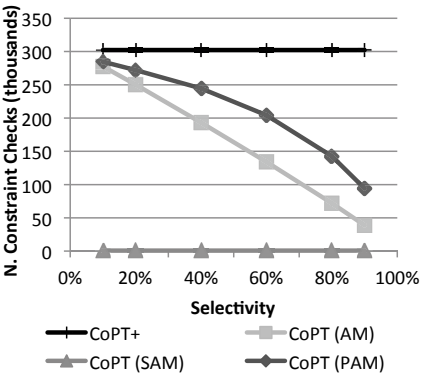


Fig. 2. Checks with AM

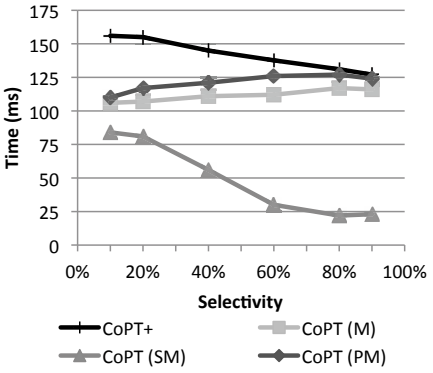


Fig. 3. Time with M

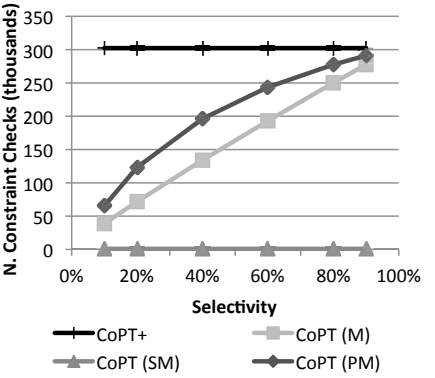


Fig. 4. Checks with M

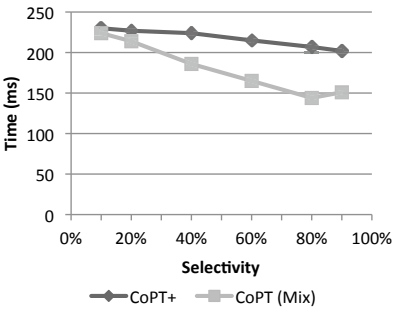


Fig. 5. Time with $Mixed$

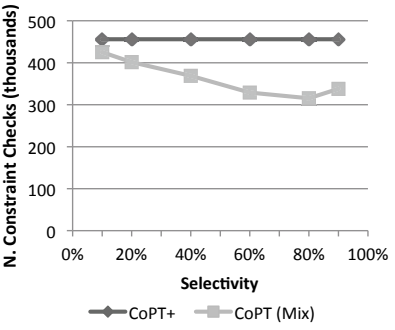


Fig. 6. Checks with $Mixed$

6 Conclusions

In this paper, we propose a new set of post-processing strategies for pushing constraints into pattern mining, through the use of the efficient pattern-tree structure. These strategies take advantage of constraint properties, so that we can filter earlier the frequent itemsets that satisfy each constraint, and avoid unnecessary tests. We also propose a general algorithm, named *CoPT*, that combines the defined strategies and is able to push any constraint into a pattern-tree, and still taking advantage of their properties.

Experimental results show that the algorithm is effective and efficient. It needs a small amount of time to push and prune the pattern-tree, even for constraints with small selectivity, and checks much less nodes and needs less time than an approach that does not take into account constraint properties.

Despite the benefits of *CoPT*, it is a post-processing approach. This means that some traditional pattern mining algorithm must run first to discover all frequent itemsets. This usually takes much time, and results in a large quantity of frequent itemsets that need to be again evaluated. As future work, we intend to create a more balanced approach and use the strategies proposed here to filter itemsets during the actual discovery process.

This work is partially supported by FCT – Fundação para a Ciência e a Tecnologia, under research project D2PM (PTDC/EIA-EIA/110074/2009) and PhD grant SFRH/BD/64108/2009.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: VLDB 1994: Proc. of the 20th Intern. Conf. on Very Large Data Bases, pp. 487–499. Morgan Kaufmann, San Francisco (1994)
2. Bayardo, R.J.: The hows, whys, and whens of constraints in itemset and rule discovery. In: Boulicaut, J.-F., De Raedt, L., Mannila, H. (eds.) Constraint-Based Mining. LNCS (LNAI), vol. 3848, pp. 1–13. Springer, Heidelberg (2006)
3. Boulicaut, J.-F., Jeudy, B.: Constraint-based data mining. In: The Data Mining and Knowledge Discovery Handbook, pp. 399–416. Springer (2005)
4. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: Proc. of the 2000 ACM SIGMOD, pp. 1–12. ACM, New York (2000)
5. Leung, C.K.-S., Lakshmanan, L.V.S., Ng, R.T.: Exploiting succinct constraints using fp-trees. SIGKDD Explor. Newsl. 4(1), 40–49 (2002)
6. Leung, C.K.-S., Sun, L.: A new class of constraints for constrained frequent pattern mining. In: Proc. of the 27th Annual ACM Symposium on Applied Computing (SAC 2012), pp. 199–204. ACM (2012)
7. Ng, R., Lakshmanan, L., Han, J., Pang, A.: Exploratory mining and pruning optimizations of constrained associations rules. In: Proc. of the 1998 ACM SIGMOD Int. Conf. on Management of Data, pp. 13–24. ACM (1998)
8. Pei, J., Han, J.: Constrained frequent pattern mining: a pattern-growth view. SIGKDD Explor. Newsl. 4(1), 31–39 (2002)

9. Pei, J., Han, J., Lakshmanan, L.V.S.: Mining frequent itemsets with convertible constraints. In: Proc. of the 17th Int. Conf. on Data Engineering (ICDE 2001), pp. 433–442. IEEE, Washington (2001)
10. Srikant, R., Agrawal, R.: Mining generalized association rules. In: Proc. of the 21th Int. Conf. on Very Large Data Bases (VLDB 1995), pp. 407–419. Morgan Kaufmann Publishers Inc., San Francisco (1995)
11. Srikant, R., Vu, Q., Agrawal, R.: Mining association rules with item constraints. In: Proc. of the 3rd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD 1997), pp. 67–73 (1997)

Generalization of Quadratic Regularized and Standard Fuzzy *c*-Means Clustering with Respect to Regularization of Hard *c*-Means

Yuchi Kanzawa

Shibaura Institute of Technology, Koto 135-8548 Tokyo, Japan
kanzawa@sic.shibaura-it.ac.jp

Abstract. In this paper, the quadratic regularized and standard fuzzy *c*-means clustering algorithms (qFCM and sFCM) are generalized with respect to hard *c*-means (HCM) regularization. First, qFCM is generalized from quadratic regularization to power regularization. The relation between this generalization and sFCM is then compared to the relation between other pairs of methods from the perspective of HCM regularization, and, based on this comparison, sFCM is generalized through the addition of a fuzzification parameter. In this process, we see that other methods can be constructed by combining HCM and a regularization term that can either be weighted by data-cluster dissimilarity or not. Furthermore, we see numerically that the existence or nonexistence of this weighting determines the property of these methods' classification rules for an extremely large datum. We also note that the problem of non-convergence in some methods can be avoided through further modification.

Keywords: fuzzy *c*-means clustering, regularization.

1 Introduction

The hard *c*-means (HCM) clustering algorithm [1] splits datasets into well-separated clusters by minimizing the sum of squared distances between data and cluster centers. This concept has been extended to fuzzy clustering, in which datum membership is shared among all cluster centers rather than restricted to a single cluster. To derive fuzzy clustering, the objective function of HCM is transformed into nonlinear functions. Specifically, Dunn's algorithm replaces linear membership weights with squared ones, and creates cluster centers based on weighted means [2]. Bezdek generalized Dunn's method to use the power of membership as weights [3], resulting in what is commonly known as the fuzzy *c*-means (FCM) algorithm. Pal and Bezdek [4] suggested taking the exponent from 1.5 to 2.5. To distinguish this algorithm from the many variants that have since been proposed, we hereafter refer to it as the standard FCM (sFCM).

Another fuzzy approach to cluster analysis is the regularization of the objective function of HCM. Recognizing that HCM is singular, and that a proper cluster cannot be obtained by the Lagrangian multiplier method, Miyamoto

V. Torra et al. (Eds.): MDAI 2013, LNAI 8234, pp. 152–165, 2013.

and Mukaidono introduced a regularization term (an entropy term [5] or a quadratic term [6]) with a positive parameter into its objective function, resulting in entropy-regularized FCM (eFCM) and quadratic-regularized FCM (qFCM). Honda and Ichihashi proposed another fuzzy approach to use nonlinear membership weights with entropy [7] to create FCM^{se}.

In this paper, qFCM and sFCM are generalized from the unified perspective of HCM regularization. First, qFCM is generalized from quadratic regularization to power-regularization. This generalization is motivated from how Bezdek generalized sFCM for use with fuzzification weights other than quadratic function [3], the fixed weight used by Dunn [2]. This generalization is useful because another value of the exponent than two has the possibility to improve its clustering accuracy than qFCM, which is similar to Pal and Bezdek [4] suggested for sFCM taking the exponent from 1.5 to 2.5. Next, the relation between sFCM and pFCM is compared to the relation between eFCM and FCM^{se} with respect to HCM regularization, and sFCM is generalized using an additional fuzzification parameter. In this process, we will see that all of the methods considered in this paper can be constructed by combining HCM and a particular regularization term which can either be weighted by data-cluster dissimilarity or not. Whether the term is weighted in this way determines the classification rule for an extremely large datum: a weighted regularization term yields a fuzzy membership value, while an unweighted regularization term yields a crisp membership value. This prospective is significant because it provides an unified view of classification rule for many variants of fuzzy clustering. We will also see how both FCM^{se} and generalized sFCM can become unstable and non-convergent, and how these deficiencies can be addressed. Since the motivation of this paper is mainly methodological similar to [5]–[7], only simple illustrative examples are shown as [5]–[7] by which we can see the properties of classification rule of the proposed methods.

The rest of this paper is organized as follows: in section 2, notation and conventional methods are introduced; in section 3, basic concepts are presented, along with the proposed generalizations of qFCM and sFCM; in section 4, some illustrative examples are provided; and in section 5, some concluding remarks are offered.

2 Preliminaries

2.1 Notation and Hard c-Means

Let $X = \{x_k \in \mathbb{R}^p \mid k \in \{1, \cdots, N\}\}$ be a dataset of p-dimensional point. The membership by which x_i belongs to the i-th cluster is denoted by $u_{i,k}$ ($i \in \{1, \cdots, C\}, k \in \{1, \cdots, N\}$) and the set of $u_{i,k}$ is denoted by u, also known as the partition matrix. The cluster center set is denoted by $v = \{v_i \mid v_i \in \mathbb{R}^p, i \in \{1, \cdots, C\}\}$. The squared Euclidean distance between the k-th datum and the i-th cluster center is denoted by $d_{i,k} = \|x_k - v_i\|_2^2$. HCM is obtained by solving the following optimization problem:

$$\underset{u,v}{\text{minimize}} \sum_{i=1}^{C} \sum_{k=1}^{N} u_{i,k} d_{i,k}, \tag{1}$$

$$\text{subject to } \sum_{i=1}^{C} u_{i,k} = 1. \tag{2}$$

and the updating equations of the memberships and the cluster centers are given as

$$u_{i,k} = \begin{cases} 1 & (x_k \text{ belongs to the } i\text{-th cluster}), \\ 0 & (\text{otherwise}), \end{cases} \tag{3}$$

$$v_i = (\sum_{k=1}^{N} u_{i,k} x_k) / (\sum_{k=1}^{N} u_{i,k}). \tag{4}$$

The algorithm is a 2-step iteration consisting of the calculation of memberships $u_{i,k}$ and cluster centers v_i [1].

The standard Fuzzy c-means (sFCM) is obtained by solving the following optimization problem:

$$\underset{u,v}{\text{minimize}} \sum_{i=1}^{C} \sum_{k=1}^{N} u_{i,k}^{m} d_{i,k} \tag{5}$$

subject to Eq. (2), where m is an additional weighting exponent. If $m = 1$, sFCM is reduced to HCM. The larger the m, the fuzzier the memberships will be, so m can be considered the fuzzification parameter. sFCM was first proposed by Dunn [2] using a fuzzification parameter fixed at $m = 2$, and was extended by Bezdek [3] for $m > 1$. An iterative algorithm is used to derive a clustering partition. From the necessary conditions for optimality, new cluster centers are derived from the weighted centers using

$$v_i = (\sum_{k=1}^{N} u_{i,k}^{m} x_k) / (\sum_{k=1}^{N} u_{i,k}^{m}), \tag{6}$$

and memberships are calculated under the constraint of Eq. (2) using

$$u_{i,k} = \begin{cases} 1/(d_{i,k}/d_{j,k})^{1/(m-1)} & (I_k = \emptyset), \\ 1/|I_k| & (I_k \neq \emptyset, i \in I_k), \\ 0 & (I_k \neq \emptyset, i \in I_k), \end{cases} \tag{7}$$

where I_k is a set of indices such that $d_{i,k} = 0$. This classification function has a maximum ($u_i(x)$=1) for the cluster center v_i, and $u_i(x)$ moves toward $1/C$ as $\|x\| \to +\infty$. sFCM will be generalized in a later section. Hereafter, all conventional methods are presented only with the corresponding optimization problems and the property of their classification rules.

Another approach to fuzzifying membership is regularization of the objective function of HCM, as accomplished by Miyamoto and Mukaidono through the introduction of a regularization term with positive parameter λ into the objective function. Using the entropy term [5] or the quadratic term [6], respectively, the entropy-regularized FCM (eFCM) and quadratic regularized FCM (qFCM) are defined as

$$\underset{u,v}{\text{minimize}} \sum_{i=1}^{C}\sum_{k=1}^{N} u_{i,k}d_{i,k} + \lambda^{-1}\sum_{i=1}^{C}\sum_{k=1}^{N} u_{i,k}\log(u_{i,k}), \tag{8}$$

$$\underset{u,v}{\text{minimize}} \sum_{i=1}^{C}\sum_{k=1}^{N} u_{i,k}d_{i,k} + \lambda^{-1}\sum_{i=1}^{C}\sum_{k=1}^{N} u_{i,k}^2, \tag{9}$$

subject to Eq. (2). The classification function of eFCM has features differing from those of sFCM [5]. Notably, $u_i(x)$ does not have the maximum value $u_i(x) = 1$ on the cluster center, and $u_i(x)$ moves toward 1 as $\|x\| \to +\infty$, so the maximum membership value may be given for a data-point that is not particularly close to the cluster center. In contrast, the classification function of qFCM is piecewise linear [6]. If a center is sufficiently far from x and another center is nearer to x than v_i, then $u_i(x) = 0$. Note that eFCM will be used as a point of comparison for other methods introduced in this paper. qFCM will be generalized later in this paper.

As further another fuzzification approach, Honda and Ichihashi proposed nonlinear membership weights other than the $u_{i,k}^m$ used in sFCM [7]. With the entropy term, the optimization problem of FCM for nonlinear membership weights is

$$\underset{u,v}{\text{minimize}} \sum_{i=1}^{C}\sum_{k=1}^{N}(u_{i,k} + \lambda^{-1}u_{i,k}\log(u_{i,k}))d_{i,k} \tag{10}$$

subject to Eq. (2). This method is referred to as FCM$^{\text{se}}$. The classification function of FCM$^{\text{se}}$ has the feature that the maximum value of membership ($u_i = 1$) is not assigned to cluster centers (as with eFCM), and u_i moves toward $1/C$ as $\|x\| \to +\infty$ (as with sFCM). FCM$^{\text{se}}$ will be used to motivate the proposed methods in comparison to eFCM with respect to HCM regularization.

3 Proposed Methods

3.1 Basic Concept

In this paper, qFCM and sFCM are generalized from the unified perspective of HCM regularization. First, qFCM is generalized from quadratic regularization to power-regularization, yielding

$$\underset{u,v}{\text{minimize}} \sum_{i=1}^{C}\sum_{k=1}^{N} u_{i,k}d_{i,k} + \lambda^{-1}\sum_{i=1}^{C}\sum_{k=1}^{N} u_{i,k}^m. \tag{11}$$

Note that the second term in the objective function is extended from the quadratic for membership $u_{i,k}$ in qFCM to m-th power. The algorithm derived in the next subsection is referred to as power-regularized fuzzy c-means (pFCM). If $m = 2$, of course, pFCM reduces to qFCM. This generalization is similar to Bezdek's generalization of sFCM for fuzzification parameter $m > 1$ [3] from Dunn's method with the fuzzification parameter fixed to $m = 2$ [2].

Next, we see an interpretation of the relation between eFCM and FCM$^{\mathrm{se}}$ that differs from the one originally given in [7]. The objective function of FCM$^{\mathrm{se}}$, Eq. (10), is equivalently described as

$$\underset{u,v}{\text{minimize}} \sum_{i=1}^{C}\sum_{k=1}^{N} u_{i,k}d_{i,k} + \lambda^{-1}\sum_{i=1}^{C}\sum_{k=1}^{N} u_{i,k}\log(u_{i,k})d_{i,k}. \tag{12}$$

Contrasting this formula and the objective function of eFCM, Eq. (8), we note that the first term of both FCM$^{\mathrm{se}}$ and eFCM is the objective function of HCM given by Eq. (1), and the second term is the regularization by entropy, which for eFCM is weighted only by λ, In eFCM, the singular situation in which only crisp memberships are obtained in HCM is regularized by the negative entropy with an optimal value of $u_{i,k} = 1/C$. Combined with HCM and the negative entropy, optimal membership is determined by the balance of the crisp-power of HCM and the fuzzy-power of the negative entropy. This balance is controlled by the fuzzification parameter λ. Applying this interpretation to FCM$^{\mathrm{se}}$, both the fuzzification parameter λ and data-cluster dissimilarity $d_{i,k}$ contribute to the balance of crisp- and fuzzy-powers. Thus, an extremely large $\|x\|$ has an extremely fuzzy membership based on the strength of the fuzzy-power of regularizer, whereas in eFCM, an extremely large $\|x\|$ has a very crisp membership, since the regularizer is weighted only by constant value of λ.

This comparison also applies to sFCM and pFCM. The objective function of pFCM given by Eq. (11), is equivalently described with constraint (2) as

$$\text{Eq. (11)} \Leftrightarrow \underset{u,v}{\text{minimize}} \sum_{i=1}^{C}\sum_{k=1}^{N} u_{i,k}d_{i,k} + \lambda^{-1}\sum_{i=1}^{C}\sum_{k=1}^{N} u_{i,k}^{m} - \lambda^{-1}N \tag{13}$$

$$\Leftrightarrow \underset{u,v}{\text{minimize}} \sum_{i=1}^{C}\sum_{k=1}^{N} u_{i,k}d_{i,k} + \lambda^{-1}\sum_{i=1}^{C}\sum_{k=1}^{N} u_{i,k}^{m} - \lambda^{-1}\sum_{k=1}^{N}\sum_{i=1}^{C} u_{i,k} \tag{14}$$

$$\Leftrightarrow \underset{u,v}{\text{minimize}} \sum_{i=1}^{C}\sum_{k=1}^{N} u_{i,k}d_{i,k} + \lambda^{-1}\sum_{i=1}^{C}\sum_{k=1}^{N} (u_{i,k}^{m} - u_{i,k}). \tag{15}$$

The objective function of sFCM given by Eq. (5), is equivalently described as

$$\text{Eq. (5)} \Leftrightarrow \underset{u,v}{\text{minimize}} \sum_{i=1}^{C}\sum_{k=1}^{N} u_{i,k}d_{i,k} + \sum_{i=1}^{C}\sum_{k=1}^{N} (u_{i,k}^{m} - u_{i,k})d_{i,k}. \tag{16}$$

Comparing Eqs. (15) and (16), the first term of both pFCM and sFCM is the objective function of HCM given by Eq. (1), and the second term is the regularization by $\sum_{i=1}^{C}\sum_{k=1}^{N}(u_{i,k}^{m} - u_{i,k})$ in which the regularizer of pFCM is weighted

only by λ, while the regularizer of sFCM is weighted only by data-cluster dissimilarity $d_{i,k}$. Therefore, just as the regularizer of FCM$^{\text{se}}$ is weighted not only by $d_{i,k}$ but also by λ, sFCM can be generalized into adopting another fuzzification parameter λ, as expressed by

$$\underset{u,v}{\text{minimize}} \sum_{i=1}^{C}\sum_{k=1}^{N} u_{i,k}d_{i,k} + \lambda^{-1}\sum_{i=1}^{C}\sum_{k=1}^{N}(u_{i,k}^{m} - u_{i,k})d_{i,k}. \tag{17}$$

The algorithm based on this objective function, which is derived in the next subsection, is referred to as generalized standard fuzzy c-means (gsFCM). Note that if $\lambda = 1$, gsFCM is reduced to sFCM. In pFCM, the singularity in HCM is regularized by the second term with the optimal value of $u_{i,k} = 1/C$. Combined with HCM and the regularizer, the optimal membership is determined by the balance of the crisp-power of HCM and the fuzzy-power of the regularizer. This balance is controlled by the fuzzification parameter λ. Applying this interpretation to gsFCM, data-cluster dissimilarity $d_{i,k}$ contributes to the balance of crisp- and fuzzy-powers. Thus, an extremely large $\|x\|$ (data-cluster dissimilarity is also extremely large) has an extremely fuzzy membership, whereas for pFCM, an extremely large $\|x\|$ has an extremely crisp membership because the regularizer is parametrized only with constant value of λ.

Based on these points, four methods in FCM—namely, gsFCM (including sFCM), eFCM, FCM$^{\text{se}}$, and pFCM (including qFCM)—are constructed by combining HCM and a regularization term, where the regularizer weighted by data-cluster dissimilarity produces gsFCM or FCM$^{\text{se}}$, and the regularizer not weighted by data-cluster dissimilarity produces eFCM or qFCM. It will be shown that this difference in weightedness determines whether the classification for an extremely large $\|x\|$ is fuzzy or crisp: specifically, we will see that gsFCM and FCM$^{\text{se}}$, in which the regularizer is weighted by data-cluster dissimilarity, both yields fuzzy membership for an extremely large $\|x\|$, whereas eFCM and qFCM, in which the regularizer is not weighted by data-cluster dissimilarity, both yields crisp membership for an extremely large $\|x\|$. These considerations are summarized in Table 1.

Table 1. Fuzzy c-Means Framework from Regularization View of Hard c-Means

		Weight by Dissimilarity between Data and Cluster($d_{i,k}$)	
		With	Without
Regularization Term	$u_{i,k}^{m} - u_{i,k}$	gsFCM (sFCM)	pFCM (qFCM)
	$u_{i,k}\log(u_{i,k})$	FCM$^{\text{se}}$	eFCM
Classification Rule for $\|x\| \to +\infty$		$1/C$	1 or 0

3.2 Power-Regularized Fuzzy c-Means

pFCM is obtained by solving the optimization problem (11) subject to Eq. (2) and $u_{i,k} \geq 0$. The optimal cluster center is derived in the same manner as HCM, eFCM, qFCM and FCM$^{\mathrm{se}}$, according to Eq. (4).

The optimal membership is derived from the following Karsh-Kuhn-Tucker (KKT) conditions:

$$d_{i,k} + m\lambda^{-1}u_{i,k}^{m-1} - \gamma_k - \delta_{i,k} = 0, \tag{18}$$

$$\delta_{i,k}u_{i,k} = 0, \tag{19}$$

$$u_{i,k} \geq 0, \tag{20}$$

$$\delta_{i,k} \geq 0, \tag{21}$$

$$\sum_{i=1}^{C} u_{i,k} = 1, \tag{22}$$

where $(\gamma, \delta) = (\gamma_1, \cdots, \gamma_N, \delta_{1,1}, \cdots, \delta_{C,N})$ is the KKT vector. From Eq. (18), we have

$$u_{i,k}^{m-1} = \frac{\lambda}{m}(\gamma_k - d_{i,k} + \delta_{i,k}). \tag{23}$$

Eq. (19) implies that $\delta_{i,k} = 0$ or $u_{i,k} = 0$. If $u_{i,k} = 0$, Eqs. (21) and (23) imply that $\delta_{i,k}$ is described both by γ_k and $d_{i,k}$ as $\delta_{i,k} = d_{i,k} - \gamma_k \geq 0 \Leftrightarrow d_{i,k} \geq \gamma_k$. If $\delta_{i,k} = 0$, Eqs. (20) and (23) imply that $\gamma_k \geq d_{i,k}$, and that in this case, $u_{i,k}$ is calculated by

$$u_{i,k} = \left(\frac{\lambda}{m}(\gamma_k - d_{i,k})\right)^{\frac{1}{m-1}}. \tag{24}$$

Based on the above, $u_{i,k}$ can be described as

$$u_{i,k} = \begin{cases} \left(\frac{\lambda}{m}(\gamma_k - d_{i,k})\right)^{\frac{1}{m-1}} & (\gamma_k \geq d_{i,k}), \\ 0 & (\gamma_k < d_{i,k}). \end{cases} \tag{25}$$

Since $u_{i,k}$ is not decreasing for γ_k and satisfies

$$\lim_{\gamma_k \to +\infty} u_{i,k}(\gamma_k) = +\infty, \quad \lim_{\gamma_k \to -\infty} u_{i,k}(\gamma_k) = 0, \tag{26}$$

there exists a unique γ_k satisfying condition (22), which implies that we have the unique optimal solution $u_{i,k}$ for the optimization problem (18–22), where $u_{i,k}(\gamma_k)$ stands for the value $u_{i,k}$ depending on γ_k (See Fig. 1). It is difficult, however, to calculate the optimal value of γ_k directly, so we utilize the bisection method as follows. First, we establish $\gamma_k = \min_{1 \leq i \leq C}\{d_{i,k}\}$ as a lower bound of γ_k satisfying condition (22), since $u_{i,k}(\gamma_k) = 0$ for all $i \in \{1, \cdots, C\}$. Next, we establish $\gamma_k = \max_{1 \leq i \leq C}\{d_{i,k}\} + \frac{m\lambda^{-1}}{C^{m-1}}$ as an upper bound of γ_k satisfying condition (22),

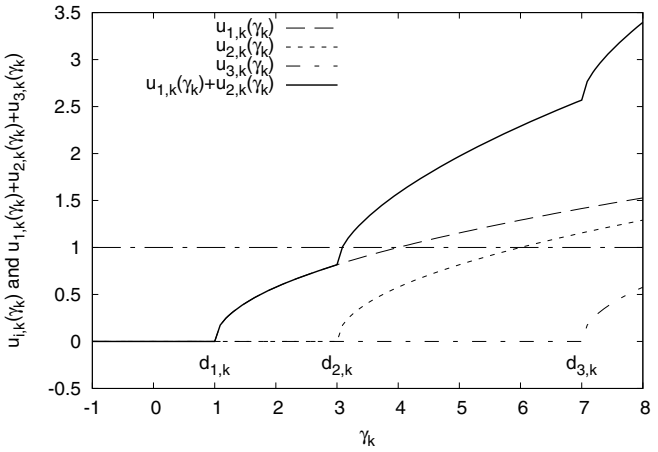


Fig. 1. $u_{i,k}(\gamma_k)$ with $(m, \lambda) = (3.0, 1.0)$, $d_{1,k} = 1$, $d_{2,k} = 3$, and $d_{3,k} = 7$: γ_k is determined as $\sum_{i=1}^{3} u_{i,k}(\gamma_k) = 1$

since $(\frac{\lambda}{m}(\gamma_k - \max_{1\le i\le C}\{d_{i,k}\}))^{\frac{1}{m-1}} = 1/C \Leftrightarrow \gamma_k = \max_{1\le i\le C}\{d_{i,k}\} + \frac{m\lambda^{-1}}{C^{m-1}}$ $\Rightarrow u_{i,k}(\gamma_k) \ge 1/C$ for all $i \in \{1, \cdots, C\}$, which implies $\sum_{i=1}^{C} u_{i,k}(\gamma_k) \ge 1$. Using these lower and upper bounds, the value of γ_k satisfying condition (22) is obtained using following algorithm:

Algorithm 1

STEP 1. *Let* γ_k^- *and* γ_k^+ *be* $\min_{1\le i\le C}\{d_{i,k}\}$ *and* $\max_{1\le i\le C}\{d_{i,k}\} + \frac{m\lambda^{-1}}{C^{m-1}}$*, respectively.*

STEP 2. *Let* $\tilde{\gamma}_k$ *be* $(\gamma_k^- + \gamma_k^+)/2$*. If* $|\gamma_k^- - \gamma_k^+|$*is sufficiently small, terminate this algorithm and let the optimal* γ_k *be* $\tilde{\gamma}_k$*.*

STEP 3. *If* $\sum_{i=1}^{C} u_{i,k}(\tilde{\gamma}_k) > 1$*, let* $\gamma_k^+ = \tilde{\gamma}_k$*. Otherwise, let* $\gamma_k^- = \tilde{\gamma}_k$*. Go to Step. 2.*

With the resulting value of γ_k, optimal membership is described by Eq. (25).

Based on these points, we propose the following power-regularized fuzzy c-means algorithm (pFCM):

Algorithm 2 (pFCM)

STEP 1. *Give the number of cluster* C *and the fuzzification parameter* (m, λ)*, and set the initial cluster centers set* v*.*

STEP 2. *Calculate* γ_k *by Algorithm 1.*

STEP 3. *Calculate* u *by Eq. (25).*

STEP 4. *Calculate* v *by Eq. (4).*

STEP 5. *Check the stopping criterion for* (γ, u, v)*. If the criterion is not satisfied, go to Step. 2.*

3.3 Generalized Standard Fuzzy c-Means

gsFCM is obtained by solving the optimization problem (17) subject to Eq. (2) and $u_{i,k} \ge 0$. The zero point of the derivative of the objective function with respect to v_i yields the updating equation for v_i,

$$v_i = \frac{\sum_{k=1}^{N}((1-\lambda^{-1})u_{i,k} + \lambda^{-1}u_{i,k}^m)x_k}{\sum_{k=1}^{N}((1-\lambda^{-1})u_{i,k} + \lambda^{-1}u_{i,k}^m)}. \tag{27}$$

Given the KKT conditions of the optimal solution of memberships,

$$(1-\lambda^{-1})d_{i,k} + m\lambda^{-1}d_{i,k}u_{i,k}^{m-1} - \gamma_k - \delta_{i,k} = 0, \tag{28}$$

$$\delta_{i,k}u_{i,k} = 0, \tag{29}$$

$$u_{i,k} \geq 0, \tag{30}$$

$$\delta_{i,k} \geq 0, \tag{31}$$

$$\sum_{i=1}^{C} u_{i,k} = 1, \tag{32}$$

the updating equation of the membership is derived in a way similar to that of pFCM, yielding the following algorithm for gsFCM:

Algorithm 3 (gsFCM)

Step 1. *Give the number of cluster C and fuzzification parameter (m, λ), and set the initial cluster centers v.*

Step 2. *Calculate γ_k by the following sub-algorithm:*

(a) Let γ_k^- and γ_k^+ be $(1-\lambda^{-1})\min_{1\leq i\leq C}\{d_{i,k}\}$ and $\lambda^{-1}m\max_{1\leq i\leq C}\{d_{i,k}\}(1/C^{m-1} + (\lambda-1)/m)$, respectively.

(b) Let $\tilde{\gamma}_k$ be $(\gamma_k^- + \gamma_k^+)/2$. If $|\gamma_k^- - \gamma_k^+|$ is sufficiently small, terminate this algorithm and let the optimal γ_k be $\tilde{\gamma}_k$.

(c) If $\sum_{i=1}^{C} u_{i,k}(\tilde{\gamma}_k) > 1$, let $\gamma_k^+ = \tilde{\gamma}_k$. Otherwise, let $\gamma_k^- = \tilde{\gamma}_k$. Go to Step. 2b.

Step 3. *Calculate the memberships as*

$$u_{i,k} = \begin{cases} (\frac{\lambda\gamma_k}{md_{i,k}} - \frac{\lambda-1}{m})^{1/(m-1)} & (\gamma_k \geq (1-\lambda^{-1})d_{i,k}), \\ 0 & (\gamma_k \leq (1-\lambda^{-1})d_{i,k}). \end{cases} \tag{33}$$

Step 4. *Calculate v by Eq. (27).*

Step 5. *Check the stopping criterion for (γ, u, v). If the criterion is not satisfied, go to Step. 2.*

3.4 Modification of gsFCM and FCM$^{\mathbf{se}}$

In deriving the updating equation of v_i of gsFCM, the following optimization problem is solved:

$$\underset{v}{\text{minimize}} \sum_{k=1}^{N}((1-\lambda^{-1})u_{i,k} + \lambda^{-1})u_{i,k}^m d_{i,k}. \tag{34}$$

Though the coefficient of $\|v_i\|_2^2$, $\sum_{k=1}^{N}((1-\lambda^{-1})u_{i,k} + \lambda^{-1}u_{i,k}^m)$, must be positive for v_i to have an optimal value, it may be non-positive depending on the given

fuzzification parameter λ and the updating value of the membership $u_{i,k}$. In this case, v_i has no optimal value, and the derived updating equation Eq.(27) does not minimize the objective function. gsFCM in this case will prove to be unstable and non-convergent. This problem occurs also in FCM$^{\mathrm{se}}$, since the coefficient of $\|v_i\|_2^2$ of FCM$^{\mathrm{se}}$, $\sum_{k=1}^{N}(u_{i,k}+\lambda^{-1})u_{i,k}\log(u_{i,k}))$ may also be non-positive.

One way to avoid such cases in gsFCM is to modify the objective function by adding a positive value to the coefficient of $d_{i,k}$, $((1-\lambda^{-1})u_{i,k}+\lambda^{-1}u_{i,k}^m)$. Since this coefficient has minimal value $\lambda^{-1}m^{-\frac{1}{m-1}}(1-\lambda)^{\frac{m}{m-1}}(m^{-1}-1)$, the modification is given by

$$\begin{aligned}\underset{u,v}{\text{minimize}} &\sum_{i=1}^{C}\sum_{k=1}^{N}u_{i,k}d_{i,k}\\ &+\lambda^{-1}\sum_{i=1}^{C}\sum_{k=1}^{N}(u_{i,k}^m-u_{i,k}+m^{-\frac{1}{m-1}}(1-\lambda)^{\frac{m}{m-1}}(1-m^{-1}))d_{i,k}. \quad (35)\end{aligned}$$

This modified objective function always yields the optimal cluster centers

$$v_i=\frac{\sum_{k=1}^{N}((1-\lambda^{-1})u_{i,k}+\lambda^{-1}(u_{i,k}^m+m^{-\frac{1}{m-1}}(1-\lambda)^{\frac{m}{m-1}}(1-m^{-1})))x_k}{\sum_{k=1}^{N}((1-\lambda^{-1})u_{i,k}+\lambda^{-1}(u_{i,k}^m+m^{-\frac{1}{m-1}}(1-\lambda)^{\frac{m}{m-1}}(1-m^{-1})))}, \quad (36)$$

and the same membership as Eq. (33).

Similarly, the objective function of FCM$^{\mathrm{se}}$ can be modified as

$$\underset{u,v}{\text{minimize}}\sum_{i=1}^{C}\sum_{k=1}^{N}(u_{i,k}+\lambda^{-1}u_{i,k}\log(u_{i,k})+\exp(\lambda-1))d_{i,k} \quad (37)$$

yielding optimal cluster centers

$$v_i=\frac{\sum_{k=1}^{N}(u_{i,k}+\lambda^{-1}(u_{i,k}\log(u_{i,k})+\exp(-\lambda-1)))x_k}{\sum_{k=1}^{N}(u_{i,k}+\lambda^{-1}(u_{i,k}\log(u_{i,k})+\exp(-\lambda-1)))} \quad (38)$$

and the same membership as the original FCM$^{\mathrm{se}}$.

4 Numerical Example

In this section, illustrative examples are provided for the sake of comparing the characteristic features of the proposed fuzzification with conventional algorithms. The first set of examples use an artificial triangle-shaped dataset consisting of 400 points in two-dimensional space (Fig. 2). Partitioning this dataset into 4 clusters via FCM algorithms with different types of fuzzification, which follows the experimental mannar in [7], reveals the difference of classification rules among different types of fuzzification. The derived fuzzy classification functions are shown in Figs. 3–6, with the gray scale indicating the maximum membership value, i.e., the degree of membership in the nearest cluster indicated by

the circles. For eFCM, $u_i(x)$ moves toward 1 as $\|x\| \to +\infty$ (see Fig. 3), while for FCM^{se}, $u_i(x)$ moves toward $1/C$ as $\|x\| \to +\infty$ (see Fig. 4). In the results of the proposed methods (Figs. 5 and 6), gsFCM with $\lambda = 1$ and pFCM with $(m, \lambda) = (2, 2)$ coincide with sFCM and qFCM, respectively. The larger the fuzzification parameter m and the smaller the fuzzification parameter λ become, the fuzzier the membership will be.

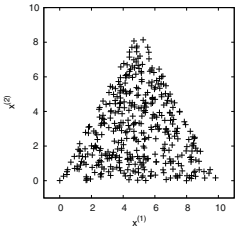


Fig. 2. Triangle-shaped Dataset

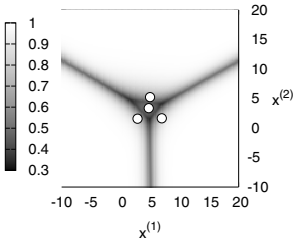


Fig. 3. eFCM with $\lambda = 0.2$

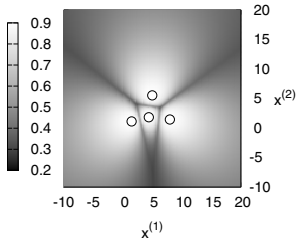
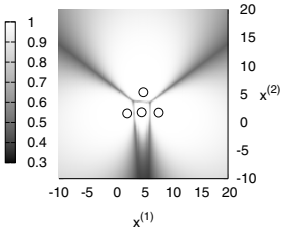
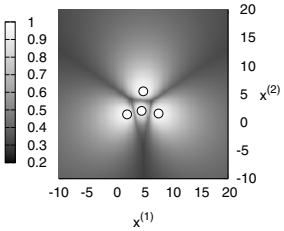


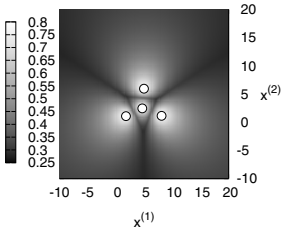
Fig. 4. FCM^{se} with $\lambda = 2.4$



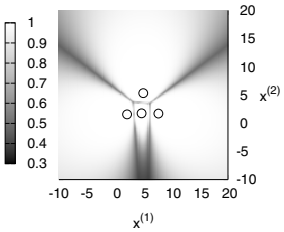
(a) $(m, \lambda) = (1.1, 0.85)$



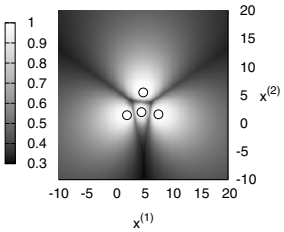
(b) $(m, \lambda) = (1.5, 0.85)$



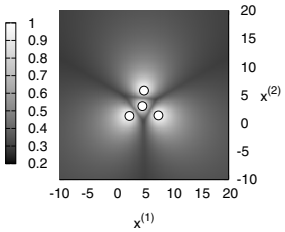
(c) $(m, \lambda) = (2.0, 0.85)$



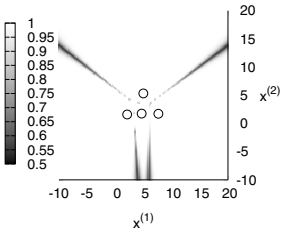
(d) $(m, \lambda) = (1.1, 1.0)$



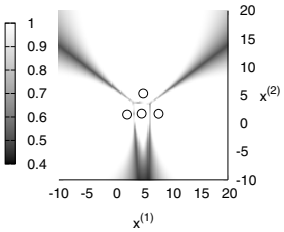
(e) $(m, \lambda) = (1.5, 1.0)$



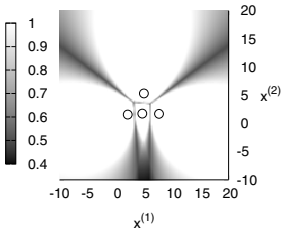
(f) $(m, \lambda) = (2.0, 1.0)$



(g) $(m, \lambda) = (1.1, 8.0)$

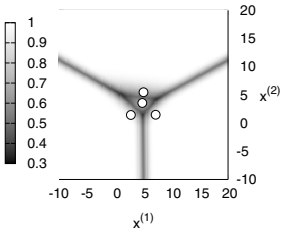


(h) $(m, \lambda) = (1.5, 8.0)$

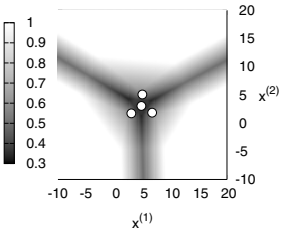


(i) $(m, \lambda) = (2.0, 8.0)$

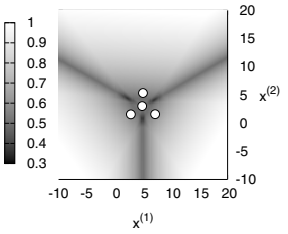
Fig. 5. gsFCM including sFCM



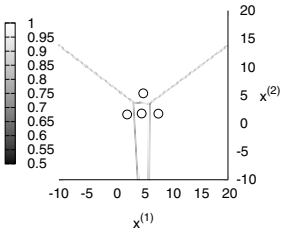
(a) $(m, \lambda) = (1.3, 0.07)$



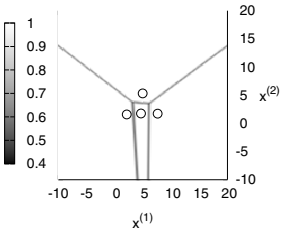
(b) $(m, \lambda) = (2.0, 0.07)$



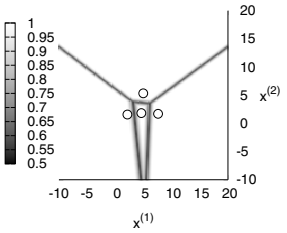
(c) $(m, \lambda) = (5.0, 0.07)$



(d) $(m, \lambda) = (1.3, 1.0)$



(e) $(m, \lambda) = (2.0, 1.0)$



(f) $(m, \lambda) = (5.0, 1.0)$

Fig. 6. pFCM including qFCM

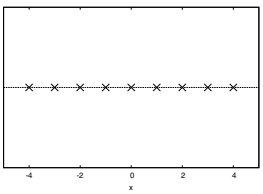


Fig. 7. Line-Shaped Dataset

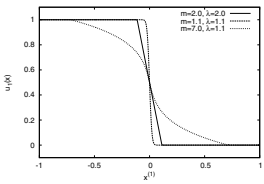


Fig. 8. $u_1(x)$ of pFCM for Line-Data

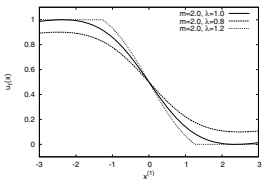


Fig. 9. $u_1(x)$ of gsFCM for Line-Data

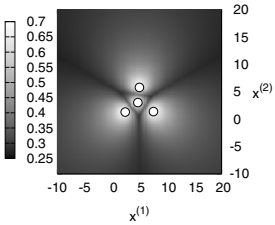


Fig. 10. modified gsFCM with $(m, \lambda) = (2.0, 0.8)$

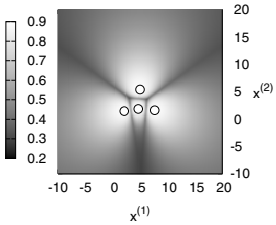


Fig. 11. modified FCM$^{\text{se}}$ with $\lambda = 2.3$

The next set of examples show the shape of the classification functions of gsFCM and pFCM near the clusters' border for an artificial, line-shaped dataset consisting of 9 points in two-dimensional space (Fig. 7) partitioned into two clusters. The derived fuzzy classification functions with a given set of fuzzification

parameters are shown in Figs. 8 and 9. In pFCM (Fig. 8), the classification function is generally "s"-shaped in $0 < u_i(x) < 1$, where only the case $m = 2.0$ (qFCM) is piecewise-linear. In gsFCM (Fig. 9), we see the following three cases: (1) If $\lambda < 1$, $u_i(x)$ is greater than 0 and less than 1; (2) If $\lambda = 1$, there are isolated points of x satisfying $u_i(x) \in \{0, 1\}$; and (3) If $\lambda > 1$, there are regions of x satisfying $u_i(x) \in \{0, 1\}$.

The final set of examples illustrate the modified gsFCM and modified FCM$^{\mathrm{se}}$, and make use of the same triangle-dataset shown in Fig. 2. Note that the reason gsFCM for $(m, \lambda) = (2.0, 0.8)$ and FCM$^{\mathrm{se}}$ for $\lambda = 2.4$ do not converge is that the coefficients of $\|v_i\|_2^2$, $\sum_{k=1}^{N}((1 - \lambda^{-1})u_{i,k} + \lambda^{-1}u_{i,k}^m)$ for gsFCM and $\sum_{k=1}^{N}(u_{i,k} + \lambda^{-1}u_{i,k}\log(u_{i,k}))$ for FCM$^{\mathrm{se}}$ are not positive. For the same parameters, respectively, the modified gsFCM and the modified FCM$^{\mathrm{se}}$ do converge, and their classification functions, shown in Figs. 10 and 11, are similar to the cases of gsFCM with $(m, \lambda) = (2.0, 0.85)$ and FCM$^{\mathrm{se}}$ with $\lambda = 2.3$, as shown in Figs. 5(c) and 4, respectively.

5 Conclusion

In this paper, qFCM and sFCM were generalized from the unified perspective of HCM regularization. First, qFCM was generalized from quadratic regularization to power-regularization. Next, the relation between sFCM and pFCM was compared with the relation between eFCM and FCM$^{\mathrm{se}}$ in regards to HCM regularization, and sFCM was generalized with an additional fuzzification parameter. Through this process, we saw that all subsequent methods could be constructed by combining HCM and a particular regularization term, where each regularizer yields two methods: on in which the regularizer is weighted by data-cluster dissimilarity and one in which it is not. We observed numerically that the existence or nonexistence of this weighting determines whether the membership of an extremely large datum will be fuzzy or crisp, respectively. This unified view can be used to investigate FCM variants other than those presented here, and for constructing further FCM variants in the future.

In future work the classification function for gsFCM and pFCM, observed numerically in this paper, will be analyzed theoretically, as was done for conventional methods in [5] and [6]. Further generalization of FCM variants should then resolve the following proposition:

Proposition 1. *If a function $\sum_{i=1}^{C} f(u_{i,k})$ has a minimal point $u_{i,k} = 1/C$ subject to Eq. (2), then, the classification function of the obtained clustering algorithm solving the optimization problem*

$$\underset{u,v}{\text{minimize}} \sum_{i=1}^{C}\sum_{k=1}^{N} u_{i,k}d_{i,k} + \lambda^{-1}\sum_{i=1}^{C}\sum_{k=1}^{N} f(u_{i,k}) \tag{39}$$

has the property that membership approaches crispness for $\|x\| \to +\infty$*, and the optimization problem*

$$\underset{u,v}{\text{minimize}} \sum_{i=1}^{C}\sum_{k=1}^{N} u_{i,k}d_{i,k} + \lambda^{-1}\sum_{i=1}^{C}\sum_{k=1}^{N} f(u_{i,k})d_{i,k} \tag{40}$$

has the property that membership approaches $1/C$ *for* $\|x\| \to +\infty$.

Acknowledgment. This work has partly been supported by the Grant-in-Aid for Scientific Research, Japan Society for the Promotion of Science, No. 00298176.

References

1. MacQueen, J.B.: Some Methods of Classification and Analysis of Multivariate Observations. In: Proc. 5th Berkeley Symposium on Math. Stat. and Prob., pp. 281–297 (1967)
2. Dunn, J.: A Fuzzy Relative of the Isodata Process and Its Use in Detecting Compact, Well-Separated Clusters. Journal of Cybernetics 3(3), 32–57 (1973)
3. Bezdek, J.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
4. Pal, N.R., Bezdek, J.C.: On Cluster Validity for Fuzzy c-Means Model. IEEE Trans. Fuzzy Syst. 1, 370–379 (1995)
5. Miyamoto, S., Mukaidono, M.: Fuzzy c-Means as a Regularization and Maximum Entropy Approach. In: Proc. 7th Int. Fuzzy Systems Association World Congress (IFSA 1997), vol. 2, pp. 86–92 (1997)
6. Miyamoto, S., Umayahara, K.: Fuzzy Clustering by Quadratic Regularization. In: Proc. 1998 IEEE Int. Conf. Fuzzy Syst., pp. 1394–1399 (1998)
7. Honda, K., Ichihashi, H.: A Regularization Approach to Fuzzy Clustering with Nonlinear Membership Weights. JACIII 11(1), 28–34 (2007)

Semi-supervised Sequential Kernel Regression Models with Pairwise Constraints

Hengjin Tang[1] and Sadaaki Miyamoto[2]

[1] Graduate School of Systems and Information Engineering
University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki 305-8573, Japan
s1230167@u.tsukuba.ac.jp
[2] Department of Risk Engineering, Faculty of Systems and Information Engineering
University of Tsukuba, 1-1-1 Tennodai, Tsukuba, Ibaraki 305-8573, Japan
miyamoto@risk.tsukuba.ac.jp

Abstract. Regression analysis has a long history and switching regression models is a derived form that can output multiple clusters and regression models. Semi-supervision is also useful technique for improving accuracy of regression analysis. However, there is one problem: the results have a strong dependency on the predefined number of clusters. To avoid these drawbacks, we proposed semi-supervised sequential regression models which we call SSSeRM that are related to the algorithm of sequential extractions. In sequential extractions process, one cluster is extracted at a time using a method of noise-detection, and the number of clusters are determinate by automatically. In this paper, we extend the capability of SSSeRM for handling non-linear structures by using kernel methods. Kernel methods can handle non-linear data and we propose two kernel regression algorithms (sequential kernel regression models and semi-supervised sequential kernel regression models) which can output clusters and regression models without defining cluster number. We compare these methods with the ordinary kernel switching regression models and semi-supervised kernel switching regression models and show the effectiveness of the proposed method by using numerical examples.

Keywords: kernel regression, switching regression models, semi-supervised clustering, pairwise constraints, sequential clustering.

1 Introduction

Regression analysis is a statistical technique and has a long history [1–3]. Its basic model is to estimate one regression model that describes relationships between a dependent variable and one or more independent variables. However, it often occurs that describing the whole data by using one regression model cannot capture characteristics of data appropriately. In such cases, typical method is to classify data into multiple classes and do regression analysis in each class respectively, and this is known as c-regression problem which is used in many real applications. The famous method of c-regression for classification is Switching Regression Models (SRM) [4, 5]. Another useful method for data analysis is

V. Torra et al. (Eds.): MDAI 2013, LNAI 8234, pp. 166–178, 2013.

semi-supervised clustering [6–8] whereby we can add prior information of data set to clustering procedure.

There is an important problem related to those algorithms, that is, the number of clusters must be defined before the algorithms run and this number can be a sensitive factor to the clustering results. To solve this problem, "sequential extraction" algorithms have been developed by several researchers [9–11] and one method [11] proposed by one of the authors is related to noise clustering [12, 13]. The advantage of "sequential extraction" is that the algorithm can output clusters without setting predefined cluster number.

In our previous work [14], we proposed Semi-Supervised Sequential Regression Models (SSSeRM) which consist SRM, semi-supervision, and "sequential extraction". In SSSeRM, we can use of prior information and have multiple regression models automatically.

However, it is difficult to handle data with non-linear structures by SSSeKRM. For this view, we focus on kernel methods [15–17]. Kernel methods were used in various fields and in this paper, we compare four algorithms related to kernel methods by numerical examples. Four algorithms are called Kernel Switching Regression Models (KSRM), Semi-Supervised Kernel Switching Regression Models (SSKSRM), Sequential Kernel Regression Models (SeKRM), and Semi-Supervised Sequential Kernel Regression Models (SSSeKRM). New algorithms we proposed in this paper are SeKRM and SSSeKRM.

The rest of this paper is organized as follows: we describe each algorithm in Section 2, numerical examples are given in Section 3, and the paper finally concludes in Section 4.

2 Algorithms

We first define some notations. We assume data set $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)$ in which $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \boldsymbol{R}^p$ are data of the independent variable $\boldsymbol{x}$, $y_1, \ldots, y_n \in \boldsymbol{R}$ are those of the dependent variable y, and n is the number of data. I_n is $n \times n$ dimensional identity matrix. The number of clusters is denoted by c, and $C^{(i)}(i = 1, \ldots, c)$ means the number i cluster. Moreover, u_{ki} is the membership grade of $(\boldsymbol{x}_k, y_k)$ belonging to $C^{(i)}$ and we denote membership matrix $U = (u_{ki})(k = 1, \ldots, n, i = 1, \ldots, c)$.

2.1 Kernel Swtiching Regression Models

Kernel Switching Regression Models (KSRM) are based on Kernel Regression (KR) and Switching Regression Models (SRM). KR is the combination of ridge regression and kernel methods and the objective function of KR is following:

$$J_{KR}(\boldsymbol{\alpha}) = (\boldsymbol{y} - K\boldsymbol{\alpha})^T(\boldsymbol{y} - K\boldsymbol{\alpha}) + \lambda\boldsymbol{\alpha}^T K\boldsymbol{\alpha} \tag{1}$$

where K is the kernel matrix, α is the regression parameter, and λ is the regularization parameter. K is defined as follows:

$$K = \begin{pmatrix} k(\boldsymbol{x_1}, \boldsymbol{x_1}) & k(\boldsymbol{x_2}, \boldsymbol{x_1}) & \dots & k(\boldsymbol{x_n}, \boldsymbol{x_1}) \\ k(\boldsymbol{x_1}, \boldsymbol{x_2}) & k(\boldsymbol{x_2}, \boldsymbol{x_2}) & \dots & k(\boldsymbol{x_n}, \boldsymbol{x_2}) \\ \vdots & \vdots & \ddots & \vdots \\ k(\boldsymbol{x_1}, \boldsymbol{x_n}) & k(\boldsymbol{x_2}, \boldsymbol{x_n}) & \dots & k(\boldsymbol{x_n}, \boldsymbol{x_n}) \end{pmatrix} \tag{2}$$

where K is a positive definite matrix. There are various types of kernels and we use the Gaussian kernel

$$k(\boldsymbol{x}, \boldsymbol{x'}) = \exp(-\beta \|\boldsymbol{x} - \boldsymbol{x'}\|^2) \tag{3}$$

in this paper. The optimal solutions for $\boldsymbol{\alpha}$ and output function $f(\boldsymbol{x}; \boldsymbol{\alpha})$ are calculated as follows:

$$\hat{\boldsymbol{\alpha}} = (K + \lambda I_n)^{-1} \boldsymbol{y} \tag{4}$$

$$f(\boldsymbol{x}; \hat{\boldsymbol{\alpha}}) = \boldsymbol{y}^T (K + \lambda I_n)^{-1} \begin{pmatrix} k(\boldsymbol{x}, \boldsymbol{x_1}) \\ k(\boldsymbol{x}, \boldsymbol{x_2}) \\ \vdots \\ k(\boldsymbol{x}, \boldsymbol{x_n}) \end{pmatrix} \tag{5}$$

Next, we introduce Switching Regression Models (SRM) [4, 5]. SRM are very useful for real applications since they can output multiple clusters and regression models simultaneously. The aim of SRM is to determine the c regression models. In this paper, we assume that data have multiple non-linear structures, so we consider the combination of KR and SRM and call them Kernel Switching Regression Models (KSRM). The aim of KSRM is to output c kernel regression models:

$$y = f_{(i)}(\boldsymbol{x}; \boldsymbol{\alpha}^{(i)}) + e_i, \quad i = 1, \dots, c. \tag{6}$$

and the objective function of kernel switching regression models uses the next equation:

$$J_{KSRM}(U, \alpha) = \sum_{k=1}^{n} \sum_{i=1}^{c} u_{ki} (y_k - f_{(i)}(\boldsymbol{x_k}; \boldsymbol{\alpha}^{(i)}))^2. \tag{7}$$

We use alternate optimization for KSRM since there are two parameters U and $\boldsymbol{\alpha}$ that we optimize. The algorithm of kernel switching regression models (KSRM) is the following one:

Procedure: Kernel Switching Regression Models
KSRM1: Set the initial value U.
KSRM2: Calculate regression parameter $\boldsymbol{\alpha}^{(i)}$ and output function $f_{(i)}(\boldsymbol{x}; \boldsymbol{\alpha}^{(i)})$ of the corresponding clusters.
KSRM3: Calculate membership matrix U.
KSRM4: If the clusters are convergent, stop; else go to **KSRM2**.
End of KSRM

The optimal solutions for α and U are as follows:

$$\boldsymbol{\alpha}^{(i)} = (K^{(i)} + \lambda I_{n^{(i)}})^{-1}\boldsymbol{y}^{(i)} \tag{8}$$

$$f_{(i)}(\boldsymbol{x};\boldsymbol{\alpha}^{(i)}) = (\boldsymbol{y}^{(i)})^T(K^{(i)} + \lambda I_{n^{(i)}})^{-1}\begin{pmatrix} k(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{1}}^{(i)}) \\ k(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{2}}^{(i)}) \\ \vdots \\ k(\boldsymbol{x}, \boldsymbol{x}_{\boldsymbol{n^{(i)}}}^{(i)}) \end{pmatrix} \tag{9}$$

$$u_{ki} = 1 \iff \boldsymbol{\alpha}^{(i)} = \arg\min_{\boldsymbol{\alpha}^{(l)}}(y_k - f_{(l)}(\boldsymbol{x_k};\boldsymbol{\alpha}^{(l)}))^2 \tag{10}$$

$$u_{kj} = 0, \quad j \neq i \tag{11}$$

The calculations of $\boldsymbol{\alpha}^{(i)}$ and $f_{(i)}(\boldsymbol{x};\boldsymbol{\alpha}^{(i)})$ in (8) and (9) are different from those of the ordinary kernel regression in (4) and (5). In equations (8) and (9), $n^{(i)}$ is the number of data set related to $C^{(i)}$, that means

$$n^{(1)} + n^{(2)} + \ldots n^{(c)} = n. \tag{12}$$

$I_{n^{(i)}}$ is $n^{(i)} \times n^{(i)}$ dimensional identity matrix. $K^{(i)}$ is $n^{(i)} \times n^{(i)}$ dimensional kernel matrix, $\boldsymbol{x}_{\boldsymbol{k}}^{(i)}$ and $\boldsymbol{y}^{(i)}$ $(k \in (1, 2, \ldots, n^{(i)}))$ are generated by sorting data set in $C^{(i)}$:

$$\boldsymbol{x}_{\boldsymbol{1}}^{(i)}, \boldsymbol{x}_{\boldsymbol{2}}^{(i)}, \ldots, \boldsymbol{x}_{\boldsymbol{n^{(i)}}}^{(i)} \in \boldsymbol{R}^p \tag{13}$$

$$\boldsymbol{y}^{(i)} = (y_1^{(i)}, y_2^{(i)}, \ldots, y_{n^{(i)}}^{(i)})^T \in \boldsymbol{R}^{n^{(i)}} \tag{14}$$

For example, if $n = 4$, $C = 2$, $(u_{11}, u_{21}, u_{31}, u_{41}) = (1, 0, 1, 0)$, and $(u_{12}, u_{22}, u_{32}, u_{42}) = (0, 1, 0, 1)$, then $n^{(1)} = 2$, $n^{(2)} = 2$, $(\boldsymbol{x}_{\boldsymbol{1}}^{(1)}, \boldsymbol{x}_{\boldsymbol{2}}^{(1)}) = (\boldsymbol{x_1}, \boldsymbol{x_3})$, $(\boldsymbol{x}_{\boldsymbol{1}}^{(2)}, \boldsymbol{x}_{\boldsymbol{2}}^{(2)}) = (\boldsymbol{x_2}, \boldsymbol{x_4})$, $\boldsymbol{y}^{(1)} = (y_1, y_3)^T$, $\boldsymbol{y}^{(2)} = (y_2, y_4)^T$, and $K^{(1)}$, $K^{(2)}$ are calculated as follows:

$$K^{(1)} = \begin{pmatrix} k(\boldsymbol{x}_{\boldsymbol{1}}^{(1)}, \boldsymbol{x}_{\boldsymbol{1}}^{(1)})\ k(\boldsymbol{x}_{\boldsymbol{2}}^{(1)}, \boldsymbol{x}_{\boldsymbol{1}}^{(1)}) \\ k(\boldsymbol{x}_{\boldsymbol{1}}^{(1)}, \boldsymbol{x}_{\boldsymbol{2}}^{(1)})\ k(\boldsymbol{x}_{\boldsymbol{2}}^{(1)}, \boldsymbol{x}_{\boldsymbol{2}}^{(1)}) \end{pmatrix}, K^{(2)} = \begin{pmatrix} k(\boldsymbol{x}_{\boldsymbol{1}}^{(2)}, \boldsymbol{x}_{\boldsymbol{1}}^{(2)})\ k(\boldsymbol{x}_{\boldsymbol{2}}^{(2)}, \boldsymbol{x}_{\boldsymbol{1}}^{(2)}) \\ k(\boldsymbol{x}_{\boldsymbol{1}}^{(2)}, \boldsymbol{x}_{\boldsymbol{2}}^{(2)})\ k(\boldsymbol{x}_{\boldsymbol{2}}^{(2)}, \boldsymbol{x}_{\boldsymbol{2}}^{(2)}) \end{pmatrix}.$$

2.2 Semi-Supervised Kernel Switching Regression Models

Semi-supervised clustering are famous methods for adding prior information and one method is to use pairwise constraints [6–8]. Pairwise constraints consist of must-link and cannot-link and each of them represents the following relationship:

must-link: two objects should be in the same cluster.
cannot-link: two objects should not be in the same cluster.

In this paper, we use pairwise constraints for semi-supervision. We apply COP K-means [6] to kernel switching regression models (KSRM) and call them Semi-Supervised Kernel Switching Regression Models (SSKSRM). In the same way

as COP K-means, the pairwise constraints are not violated during the clustering process in this paper. The algorithm of semi-supervised kernel switching regression models (SSKSRM) is as follows:

Procedure: Semi-Supervised Kernel Switching Regression Models
SSKSRM1: Set the initial value U.
SSKSRM2: Calculate regression parameter $\boldsymbol{\alpha}^{(i)}$ and output function $f_{(i)}(\boldsymbol{x}; \boldsymbol{\alpha}^{(i)})$ of the corresponding clusters.
SSKSRM3: Calculate membership matrix U such that *violation of pairwise constraints* is false. If no such cluster exists, fail.
SSKSRM4: If the clusters are convergent, stop; else go to **SSKSRM2**.
End of SSKSRM

The difference from KSRM is about updating cluster assignments in **SSKSRM3**. If U violates pairwise constraints, we continue to search another U that satisfy all pairwise constraints. If there is no such U, the algorithm stops and partition output is empty. The algorithm of *checking violations of pairwise constraints* in SSKSRM is as follows:

Procedure: checking violations of pairwise constraints in SSKSRM
1: For all (x, x') in must-link, if $x \in C^{(i)}$, $x' \in C^{(j)}$ $(i \neq j)$, then return **true**.
2: For all (x, x') in cannot-link, if $x, x' \in C^{(i)}$, then return **true**.
3: Else return **false**.
End

2.3 Sequential Kernel Regression Models

One of the authors has proposed different algorithms for sequential extraction of clusters [11] and we developed some derivative methods in other works [14, 18, 19]. In these algorithms, one cluster is extracted at a time. The extraction process continues until no sufficient data exist.

Sequential Kernel Regression Models (SeKRM) use the next objective function:

$$J_{SeKRM}(U, \alpha^{(s)}) = \sum_{k=1}^{n} u_{ks}(y_k - f_{(s)}(\boldsymbol{x_k}; \boldsymbol{\alpha}^{(s)}))^2 + \sum_{k=1}^{n} u_{k0}\delta. \tag{15}$$

Note that there are only two clusters: u_{ks} is the membership belonging to the number s cluster extracted by SeKRM and u_{k0} is the membership belonging to the noise cluster 0; $\delta > 0$ is a parameter which means every object has a constant dissimilarity δ from the noise cluster. This algorithm applies a variation of noise clustering [12, 13] to extract regression models sequentially.

The optimal solution of U is calculated as follows:

$$(u_{ks}, u_{k0}) = \begin{cases} (1,0), (y_k - f_{(s)}(\boldsymbol{x_k}; \boldsymbol{\alpha}^{(s)}))^2 \leq \delta) \\ (0,1), (y_k - f_{(s)}(\boldsymbol{x_k}; \boldsymbol{\alpha}^{(s)}))^2 > \delta) \end{cases} \tag{16}$$

and the optimal solution $\alpha^{(s)}$ for regression models is calculated as same as that in KSRM.

X is assumed as a data set which we aim to analyze. We apply sequential clustering to kernel switching regression models, and call it sequential kernel regression models (SeKRM).

Procedure: Sequential Kernel Regression Models
SeKRM1: Set the initial data set $X^{(0)} = X$, $s = 1$, the initial value U.
SeKRM2: Calculate regression parameter $\boldsymbol{\alpha}^{(s)}$ and output function $f_{(s)}(\boldsymbol{x}; \boldsymbol{\alpha}^{(s)})$ of the corresponding clusters.
SeKRM3: Calculate membership matrix U.
SeKRM4: If the clusters are convergent, stop and extract cluster $C^{(s)}$ that belongs to the elements with $u_{ks} = 1$; else go to **SeKRM2**.
SeKRM5: Let $X^{(s)} = X^{(s-1)} - C^{(s)}$. If $X^{(s)}$ does not have sufficient elements to extract one more cluster, stop; otherwise go to **SeKRM2**.
End of SeKRM

2.4 Semi-supervised Sequential Kernel Regression Models

Semi-Supervised Sequential Kernel Regression Models (SSSeKRM) are the combinations of semi-supervision and SeKRM which are explained in Section 2.2 and Section 2.3. If there are prior information, the performance of SeKRM is more likely to become better by using prior information. The objective function of SSSeKRM is same as SeKRM in Section 2.3. The difference is *checking violations of pairwise constraints* about membership U in the process. The algorithm of SSSeKRM is as follows:

Procedure: Semi-Supervised Sequential Kernel Regression Models
SSSeKRM1: Set the initial data set $X^{(0)} = X$, $s = 1$, the initial value U.
SSSeKRM2: Calculate regression parameter $\boldsymbol{\alpha}^{(s)}$ and output function $f_{(s)}(\boldsymbol{x}; \boldsymbol{\alpha}^{(s)})$ of the corresponding clusters.
SSSeKRM3: Calculate membership matrix U such that *checking violations of pairwise constraints* is false. If no such cluster exists, fail.
SSSeKRM4: If the clusters are convergent, stop and extract cluster $C^{(s)}$ that belongs to the elements with $u_{ks} = 1$; else go to **SSSeKRM2**.
SSSeKRM5: Let $X^{(s)} = X^{(s-1)} - C^{(s)}$. If $X^{(s)}$ does not have sufficient elements to extract one more cluster, stop; otherwise go to **SSSeKRM2**.
End of SSSeKRM

We note that *checking violations of pairwise constraints* in SSSeKRM should be different from SSKSRM in **Section 2.2**, which is as follows:

Procedure: checking violations of pairwise constraints in SSKSRM
1: For all (x, x') in must-link, repeat 1.1–1.2.
1.1: If $x \in C^{(0)}$ and $x' \in C^{(s)}$, return **true**.
1.2: If $x \in C^{(s)}$ and $x' \in C^{(0)}$, return **true**.
2: For all (x, x') in cannot-link, if $x, x' \in C^{(s)}$, then return **true**.
3: Else return **false**.
End

To summarize, we do not judge $x, x' \in C^{(0)}$ where (x, x') in cannot-link to be the violation of constraints, since other clusters can subsequently be extracted from the noise cluster $C^{(0)}$.

3 Experiments

We show numerical examples of clustering for an artificial data set. The purpose to show an example is to clearly show differences among the four methods (KSRM, SSKSRM, SeKRM, SSSeKRM) and for this purpose two-dimensional data is used, since we can view differences at a glance. This artificial dataset contains two clusters with non-linear structure.

Figure 1 shows the results using KSRM where two clusters are assumed. Figures 2 and 3 show overall results and the sequentially extracted clusters using SeKRM. Figures 4 and 5 show SSKSRM (must-link) and SSKSRM (cannot-link) where two clusters are assumed. Figures $6-8$ show overall results and the sequentially extracted clusters using SSSeKRM (must-link). Figures $9-11$ show overall results and the sequentially extracted clusters using SSSeKRM (cannot-link). Must-link is represented as bullet ($\bullet$) in Figures $4, 6-8$ and cannot-link is represented as triangle ($\triangle$, $\bigtriangledown$) in Figures $5, 9-11$. For verifying the effect of semi-supervision, we use two different settings both of must-link and cannot-link, In Figures $4-6$, 9, left figure and right figure have different prior information of semi-supervision. Prior information of must-link in Figure 4 (left) and Figure 6 (left), Figure 4 (right) and Figure 6 (right), and prior information of cannot-link in Figure 5 (left) and Figure 9 (left), Figure 5 (right) and Figure 9 (right) are same.

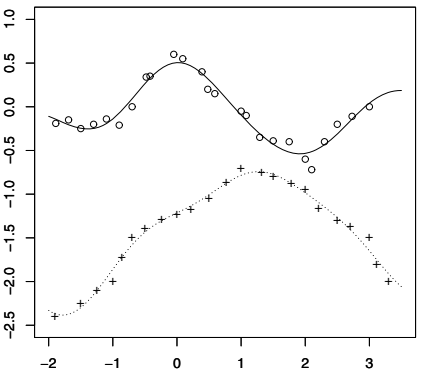


Fig. 1. Two kernel regression models using kernel switching regression models (KSRM), where two clusters are assumed

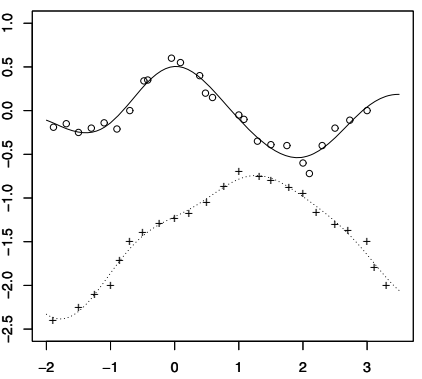


Fig. 2. Overall results of sequential kernel regression models (SeKRM), where $\circ$ and solid line represent the first extracted cluster, and $+$ and dashed line represent second extracted cluster

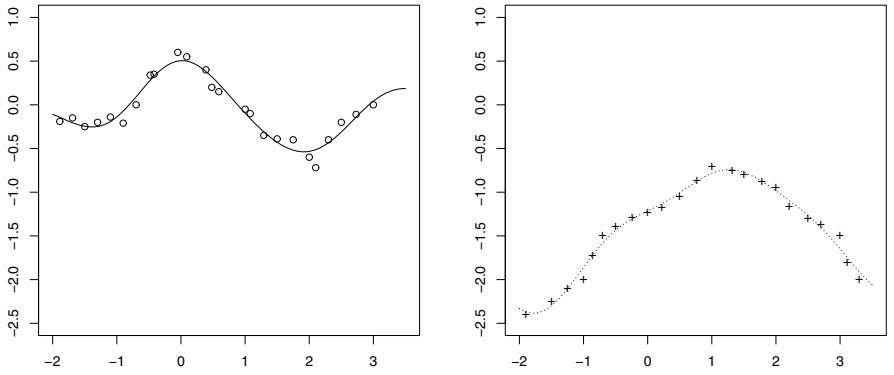


Fig. 3. First (left) and second (right) extracted cluster of sequential kernel regression models (SeKRM)

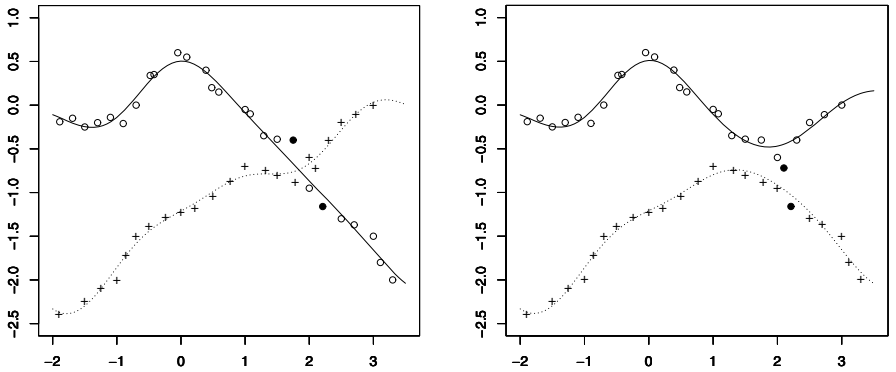


Fig. 4. Two kernel regression models using semi-supervised kernel regression models (SSKSRM, must-link), where two clusters are assumed, • represents must-link, and ∘, •, and solid line represent the first cluster, and + and dashed line represent second extracted cluster (left figure), • is in the second cluster and the rest of it is as same as left figure (right figure)

In all figures, solid line and dashed line represent regression curves of cluster 1 and cluster 2 respectively. Circle (∘) is in cluster 1 and plus (+) is in cluster 2. Bullet (•) is included in cluster 1 (Figures 4 (left) and 6 (left)), and in cluster 2 (Figures 4 (right) and 6 (right)). Triangle(up) (△) is in cluster 1 and triangle(down) (▽) is in cluster 2. In Figures $4-11$, we set kernel parameter $\beta = 0.5$ and noise parameter $\delta = 0.0625$.

From those figures, we find two points about each kernel algorithms. One is that the results of non-sequential algorithms (KSRM and SSKSRM (Figures 1,4,5)) and sequential algorithms (SeKRM and SSSeKRM (Figures 2,6,9))

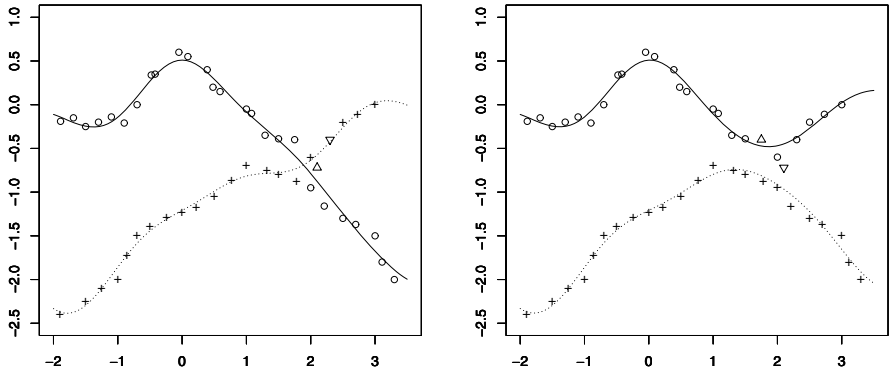


Fig. 5. Two kernel regression models using semi-supervised kernel regression models (SSKSRM, cannot-link), where two clusters are assumed, $\triangle$ and $\bigtriangledown$ represent cannot-link, and $\circ$, $\triangle$, and solid line represent the first cluster, and $+$, $\bigtriangledown$, and dashed line represent the second cluster

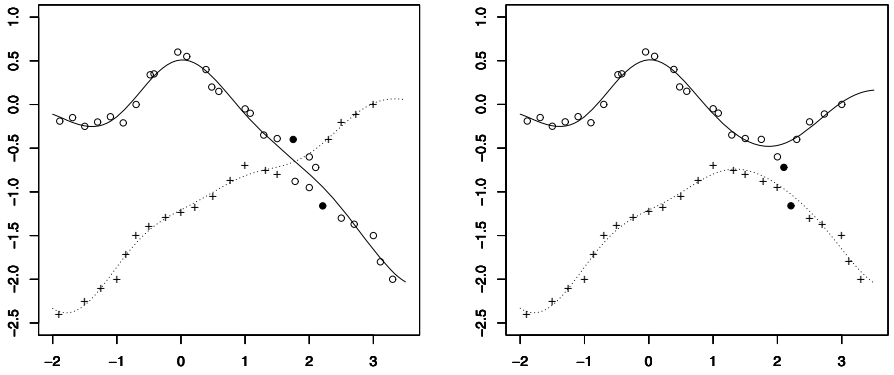


Fig. 6. Overall results of semi-supervised sequential kernel regression models (SSSeKRM, must-link), where $\circ$, $\bullet$, and solid line represent the first extracted cluster, and $+$ and dashed line represent the second extracted cluster (left figure), $\bullet$ is in the second cluster and the rest of it is as same as left figure (right figure)

are almost same (comparisons between KSRM (Figure 1) and SeKRM (Figure 2), SSKSRM (must-link) (Figure 4) and SSSeKRM (must-link) (Figure 6), and SSKSRM (cannot-link) (Figure 5) and SSSeKRM (cannot-link) (Figure 9)). The other is that the outputs (clusters and regression models) can be modified by adding pairwise constraints (the difference between KSRM (Figure 1) and SSKSRM (Figures 4 and 5), or SeKRM (Figure 2) and SSSeKRM (Figures 6 and 9)).

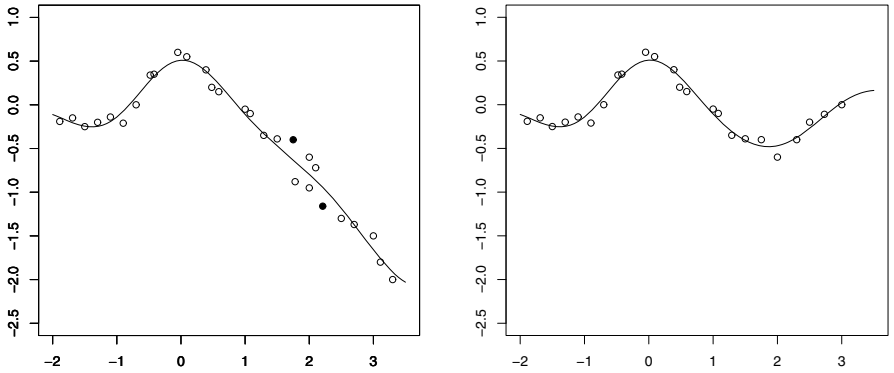


Fig. 7. First extracted cluster of semi-supervised sequential kernel regression models (SSSeKRM, must-link)

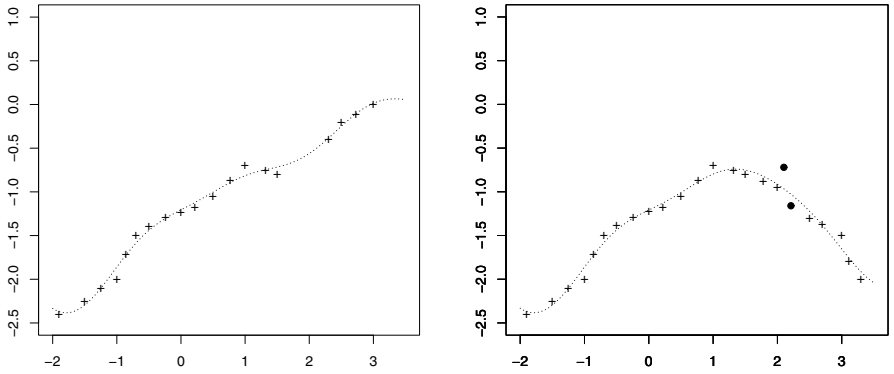


Fig. 8. Second extracted cluster of semi-supervised sequential kernel regression models (SSSeKRM, must-link)

To summarize, our proposed algorithms SeKRM and SSSeKRM can handle non-linear structure without predefining cluster number and SSSeKRM can handle prior information by adding pairwise constraints.

4 Conclusions

We have developed four algorithms (Kernel Switching Regression Models (KSRM), Semi-Supervised Kernel Regression Models (SSKSRM), Sequential Regression Models (SeKRM), and Semi-Supervised Sequential Regression Models (SSSeKRM)) and compared them by numerical examples.

From the experiments, we find two points about kernel methods: one is that the results of sequential kernel algorithms (SeKRM and SSSeKRM) are almost

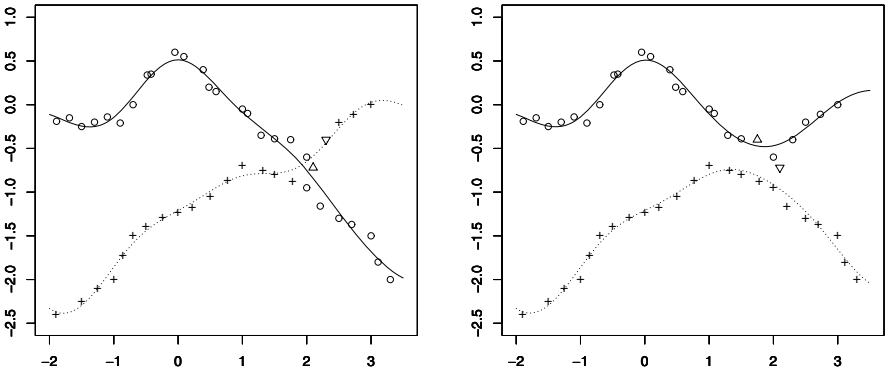


Fig. 9. Overall results of semi-supervised sequential kernel regression models (SSSeKRM, cannot-link), where ∘, △, and solid line represent the first extracted cluster, and +, ▽, and dashed line represent the second extracted cluster

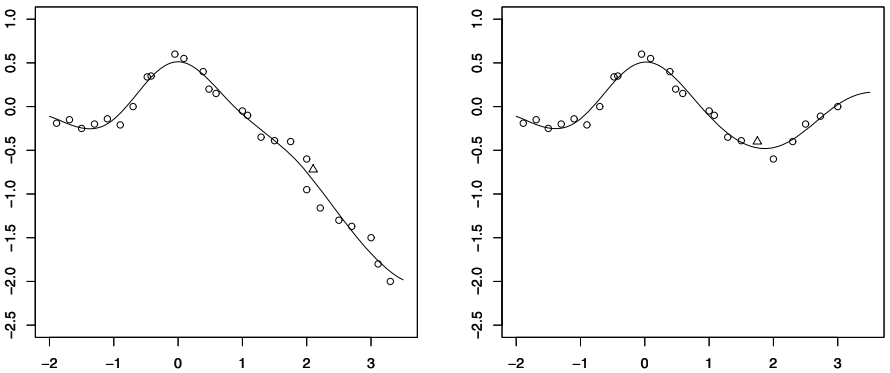


Fig. 10. First extracted cluster of semi-supervised sequential kernel regression models (SSSeKRM, cannot-link)

the same as non-sequential methods (KSRM and SSKSRM) and we can obtain clusters and regression models automatically; the other is that we can also use prior information by adding semi-supervisions (pairwise constraints) to modify results in sequential kernel methods.

Generally, real world problems have many data with many dimensions and complex structures. As a future work, we will apply our algorithms to those data. Additionally, we also plan to extend SSSeKRM by using penalty functions as pairwise constraints.

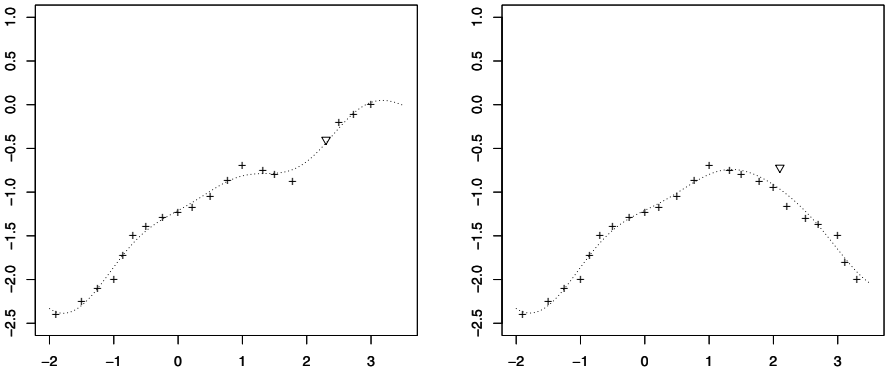


Fig. 11. Second extracted cluster of semi-supervised sequential kernel regression models (SSSeKRM, cannot-link)

Acknowledgment. This work has partly been supported by the Grant-in-Aid for Scientific Research, Japan Society for the Promotion of Science, No.23500269.

References

1. Galton, F.: Typical laws of heredity. Nature 15, 492–495, 512–514, 532–533 (1877)
2. Yule, G.U.: On the Theory of Correlation. Journal of the Royal Statistical Society 60(4), 812–854 (1897)
3. Pearson, K., Yule, G.U., Blanchard, N., Lee, A.: The Law of Ancestral Heredity. Biometrika 2(2), 211–236 (1903)
4. Quandt, R.E.: A New Approach to Estimating Switching Regressions. Journal of the American Statistical Association 67, 306–310 (1972)
5. Goldfeld, S.M., Quandt, R.E.: Techniques for Estimating Switching Regressions. In: Goldfeld, S.M., Quandt, R.E. (eds.) Studies in Nonlinear Estimation, Ballinger, Cambridge, Massachusetts, pp. 3–35 (1976)
6. Wagstaff, K., Cardie, C., Rogers, S., Schröedl, S.: Constrained K-means Clustering with Background Knowledge. In: Proceedings of the Eighteenth International Conference on Machine Learning, pp. 577–584 (2001)
7. Basu, S., Bilenko, M., Mooney, R.J.: A Probabilistic Framework for Semi-Supervised Clustering. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 59–68 (2004)
8. Chapelle, O., Schölkopf, B., Zien, A.: Semi-Supervised Learning. The MIT Press, Cambridge (2006)
9. Mirkin, B.: The Iterative Extraction Approach to Clustering. Lecture Notes in Computational Science and Engineering, vol. 58, pp. 153–179. Springer, New York (2007)
10. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, 4th edn. Academic Press, Massachusetts (2008)
11. Miyamoto, S., Kuroda, Y., Arai, K.: Algorithms for Sequential Extraction of Clusters by Possibilistic Method and Comparison with Mountain Clustering. Journal of Advanced Computational Intelligence and Intelligent Informatics 12(5), 448–453 (2008)

12. Davé, R.N., Krishnapuram, R.: Robust clustering methods: a unified view. IEEE Transactions on Fuzzy Systems 5(2), 270–293 (1997)
13. Davé, R.N., Sen, S.: On Generalizing the Noise Clustering Algorithms. In: Proceedings of the Seventh IFSA World Congress, vol. 3, pp. 205–210 (1997)
14. Tang, H., Miyamoto, S.: Sequential Regression Models with Pairwise Constraints Using Noise Clusters. Journal of Advanced Computational Intelligence and Intelligent Informatics 16(7), 814–818 (2012)
15. Girolami, M.: Mercer Kernel-Based Clustering in Feature Space. IEEE Transactions on Neutral Networks 13(3), 780–784 (2002)
16. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press (2004)
17. Kernel-Machines. Org, `http://kernel-machines.org`
18. Tang, H., Miyamoto, S.: Algorithms in Sequential Fuzzy Regression Models Based on Least Absolute Deviations. In: Torra, V., Narukawa, Y., Daumas, M. (eds.) MDAI 2010. LNCS (LNAI), vol. 6408, pp. 129–139. Springer, Heidelberg (2010)
19. Tang, H., Miyamoto, S.: Sequential Extraction of Fuzzy Regression Models: Least Squares and Least Absolute Deviations. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 19(suppl.1), 53–63 (2011)

Query Optimization Strategies in Similarity-Based Databases*

Petr Krajca and Vilem Vychodil

DAMOL (Data Analysis and Modeling Laboratory)
Dept. Computer Science, Palacky University, Olomouc
17. listopadu 12, CZ–77146 Olomouc, Czech Republic
petr.krajca@upol.cz, vychodil@acm.org

Abstract. We deal with algorithmic aspects and implementation issues of query execution in relational similarity-based databases. We are concerned with a generalized relational model of data in which queries can be matched to degrees taken from scales represented by complete residuated lattices. The main contribution of this paper are optimization techniques for efficient evaluation of queries involving similarity-based restrictions. In addition, we present experimental evaluation of the proposed techniques showing their efficiency compared to naive approaches.

Keywords: domain similarities, fuzzy logic, monotone queries, query execution, relational model of data, residuated lattices.

1 Introduction and Related Work

In this paper, we deal with similarity-based queries and imperfect matches which are treated in a generalization of the Codd model of data [8, 12, 24] which results by considering complete residuated lattices as structures representing degrees of matches. In our previous work [2–4], we have investigated the model from the point of view of data representation, querying, and similarity-based functional dependencies in data. So far, we have not considered important issues related to efficiency of the model, namely, in context of query execution. This paper deals with issues related to efficiency and presents preliminary results.

In contrast to the classic Codd model of data, the model we are concerned with allows users to formulate queries which can be matched to degrees, leaving the ordinary yes/no matches particular cases. Analogously, the model allows users to formulate soft constraints which are allowed to be satisfied to degrees (for instance, a constraint can be violated a little but not too much). Such concepts are appealing if users query relational databases containing data defined on domains whose values may be compared (by rational observers) according to their similarity or closeness. In that case, users may be interested not only in the

* P. Krajca is supported by grant no. P103/11/1456 of the Czech Science Foundation; V. Vychodil is supported by project reg. no. CZ.1.07/2.3.00/20.0059 of the European Social Fund in the Czech Republic.

V. Torra et al. (Eds.): MDAI 2013, LNAI 8234, pp. 179–191, 2013.

ordinary exact matches but also in imperfect matches which take the similarities into account. Typical examples of queries involving similarity are "Show cars with engines similar to V8 360-hp 5.7L", "Show high quality wines sold for approximately $150", "Show patients with symptoms similar to dry cough", and the like. An important aspect of the similarity-based querying which stems from our model is that it can provide answers in cases where the ordinary counterparts of queries yield no answer. Indeed, in many practical situations, it may happen that an ordinary query is not matched at all (e.g., "Show cars sold for $15,000") but its similarity-based counterpart (e.g., "Show cars sold for approximately $15,000") is matched by existing data to high degrees (e.g., by "cars offered for slightly less than $15,000"). Needless to say, this qualitative extension of the ordinary relational queries improves a user-machine interaction and is especially interesting in supporting a decision process (e.g., helping answer questions like "Which car in the price category around $15,000 should I buy?").

In order to apply the model, there is a need to have efficient ways to execute (interpret) similarity-based queries. Of course, the similarity-based queries described above can be mimicked in an ordinary relational database and implemented for instance by ordinary SQL queries [2] but the main drawback of this approach is its efficiency. In fact, the present RDBMSs lack the ability to optimize such queries which leads to inefficient query executions involving full table scans suitable only for small data. In this paper, we show ways to improve the naive executions of similarity-based queries and outline their implementation in a software prototype of our query language RESIQL [22].

Related to our approach are various approaches to relational databases with ranking and explicit scores (degrees) assigned to tuples in relations. For instance, there are substantial results in probabilistic databases [6, 9–11, 18] which unlike our model aim at processing uncertain data (in our model, data are certain but are allowed to match queries to degrees). Various ranking approaches were proposed on top of the classic relational model (RM), most notably RankSQL [23], see also [20] for a survey of approaches. Our approach differs in the way in which it incorporates the ranking into the relational model—instead of developing an extension on top of the classic RM, we develop the model using a more general metamathematics under which the ranks and similarities on domains naturally emerge, see Section 2 for details. Our paper is related to and exploits some ideas from the influential paper [15] on monotone query execution since the type of queries we consider in this paper are in fact monotone. There have been many results concerned with "fuzzy data" which started with [5, 25] and can be seen as extensions of the RM from the viewpoint of fuzzy logics in the wide sense, dealing with fuzzy sets stored in databases. In contrast, we do not consider "fuzzy data" (in fact, we do not impose any restrictions on domains) and our approach is more connected to fuzzy logics in the narrow sense [17].

This paper is organized as follows. In Section 2 we briefly introduce the generalized model and in Section 3 we describe the proposed optimization techniques and make some implementation notes. Furthermore, in Section 4 we present an experimental evaluation of the proposed optimizations.

2 Generalized Relational Model of Data

We outline here the foundations of our model and introduce notions necessary for understanding of the basic type of queries considered in this paper and the optimization techniques, details can be found in [3, 4].

Our model can be seen as a generalization of the classic RM which results by substituting the two-element Boolean algebra which is the implicit structure of yes/no matches (in fact, truth degrees assigned to formulas) in the classic RM by a more general structure, namely a (complete) residuated lattice [16]. Hence, our model departs from the yes/no matches and allows general "degrees of matches" upon which we build the generalized relational model. In the classic RM, the concept of a relation on a relation scheme R (a finite set of attributes), which is considered as a finite subset of a Cartesian product $\prod_{y\in R} D_y$ of domains D_y of attributes $y \in R$ can be identified with an indicator function

$$\mathcal{D}\colon \textstyle\prod_{y\in R} D_y \to \{0,1\} \tag{1}$$

so that for only finitely many tuples $r \in \prod_{y\in R} D_y$ we have $\mathcal{D}(r) = 1$. If $\mathcal{D}$ is viewed as a result of query Q, then $\mathcal{D}(r) = 1$ is interpreted so that "the tuple r matches the query Q". In our model, we replace $\{0,1\}$ by a set L of degrees which is assumed to be equipped with a partial order $\leq$ so that $\langle L, \leq\rangle$ is a complete lattice, i.e., an arbitrary subset of L has its infimum (greatest lower bound) and supremum (least upper bound) in L. We adhere to the *comparative meaning* of degrees from L (higher degrees represent better matches) as it is usual in fuzzy logics in the narrow sense (FLns), see [14, 17, 19]. Under this assumption, we may replace (1) by

$$\mathcal{D}\colon \textstyle\prod_{y\in R} D_y \to L \tag{2}$$

so that for only finitely many tuples $r \in \prod_{y\in R} D_y$ we have $\mathcal{D}(r) \neq 0$. Clearly, (2) is a map which assigns to each r a value $\mathcal{D}(r)$ from L, we call the value *the rank of r in $\mathcal{D}$* and if $\mathcal{D}$ is interpreted as a result of a query Q, then $\mathcal{D}(r)$ is the *degree to which r matches the query Q*. The notion of a relation on a relation scheme which appears in the ordinary RM can be then seen as a particular case of (2) for $L = \{0,1\}$ with its natural ordering (i.e., $0 < 1$).

Furthermore, the lattice of degrees should be equipped with operations to aggregate degrees. Such operations and in particular (truth functions of) general conjunctions appear in our model as we consider counterparts to relational operations like the natural join. Indeed, in the ordinary RM, for relations $\mathcal{D}_1$ and $\mathcal{D}_2$ on relation schemes $R \cup S$ and $S \cup T$ such that R, S, T are pairwise disjoint, we consider the natural join of $\mathcal{D}_1$ and $\mathcal{D}_2$ as a relation on $R \cup S \cup T$, denoted by $\mathcal{D}_1 \bowtie \mathcal{D}_2$ which consists of concatenation of all joinable tuples from $\mathcal{D}_1$ and $\mathcal{D}_2$. Identifying the relations with their indicator functions as in (1) and using the usual notation for tuple concatenation (i.e., rs stands for the set-theoretic union of maps r and s, see [24]), we have $(\mathcal{D}_1 \bowtie \mathcal{D}_2)(rst) = 1$ iff $\mathcal{D}_1(rs) = 1$ *and* $\mathcal{D}_2(st) = 1$. Therefore, we may rewrite the natural join as follows

$$(\mathcal{D}_1 \bowtie \mathcal{D}_2)(rst) = \mathcal{D}_1(rs) \otimes \mathcal{D}_2(st), \tag{3}$$

where $\otimes$ is a binary operation $\otimes\colon \{0,1\}^2 \to \{0,1\}$ which coincides with the truth function of the logical connective "conjunction" in the usual sense (i.e., $1 \otimes 1 = 1$ and $1 \otimes 0 = 0 \otimes 1 = 0 \otimes 0 = 0$). Thus, for considering analogues of natural joins in our model, we need a reasonable generalization of $\otimes$. A reasonable choice is a binary operation $\otimes\colon L^2 \to L$ which is commutative, associative, neutral with respect to 1 (full match), and is distributive over arbitrary suprema, i.e.,

$$a \otimes \bigvee_{i \in I} b_i = \bigvee_{i \in I} (a \otimes b_i) \tag{4}$$

holds true for any $a \in L$ and all $b_i \in L$ ($i \in I$). As a consequence, $\otimes$ is *monotone* which is a desirable property since then better results of subqueries (e.g., $\mathcal{D}_1$ and $\mathcal{D}_2$) yield better results of composed queries whose results are computed by $\otimes$ as in case of (3). As it is well known, (4) together with the fact that $\langle L, \otimes, 1\rangle$ is a commutative monoid is equivalent to stating that for $\otimes$ there exists a (uniquely given) binary operation $\to\colon L^2 \to L$ satisfying the following *adjointness property*:

$$a \otimes b \leq c \quad \text{iff} \quad a \leq b \to c \tag{5}$$

for all $a, b, c \in L$. Recall that the adjointness of $\otimes$ and $\to$ is a crucial property of structures of degrees used in FLns, see [1, 16, 19]. Altogether, our structure of degrees which replaces the two-element Boolean algebra shall be a (complete) residuated lattice $\mathbf{L} = \langle L, \wedge, \vee, \otimes, \to, 0, 1\rangle$ (with $a \leq b$ iff $a \wedge b = a$ as usual). Note that we have justified the presence of $\otimes$ by the need to have a reasonable generalization of a natural join in our model. Analogously, one can say that $\to$ is crucial for expressing a "graded containment" which is essential, e.g., for expressing queries like "all As are Bs" which involve universal quantification.

There are wide benefits of using complete residuated lattices as structures for degrees of matches. First, the structures are reasonably strong (the adjointness ensures that $\mathbf{L}$ and $\otimes$ and $\to$ have reasonable properties). Second, the class of residuated lattices is large and includes popular t-norm based structures [21] defined of the real unit interval, finite structures (finite scales of degrees including, e.g., the well-known Likert scale), and various nonlinear structures. Third, with residuated lattices we get reasonable logical background for our model. As a consequence, database instances can be seen as safe interpretations of (many-sorted) predicate languages [7, 19], predicate formulas (with free variables) can be seen as prescribing queries in our model, and evaluation of the formulas in structures can be seen as a way of query evaluation, see [4] for details.

Following the previous arguments, the basic notion which appears in our model and which replaces the ordinary notion of a relation on a relation scheme is introduced as follows.

Definition 1 (ranked data tables). Let $\mathbf{L}$ be a complete residuated lattice, $R \subseteq Y$ be a finite set of attributes (a relation scheme). Then, any map $\mathcal{D}$ of the from (2) such that for only finitely many tuples $r \in \prod_{y \in R} D_y$ we have $\mathcal{D}(r) \neq 0$ is called a *ranked data table* (an RDT).

In order to be able to express similarity-based queries, we assume that each domain D_y is equipped with a *similarity* $\mathbf{L}$-relation [1], i.e., a map $\approx_y\colon D_y \times D_y \to$

L which assigns to each pair of values $d_1, d_2 \in D_y$ a degree $d_1 \approx_y d_2$ to which d_1 is similar to d_2. We assume that each $\approx_y$ is at least *reflexive* ($d \approx_y d = 1$ for all $d \in D_y$) and *symmetric* ($d_1 \approx_y d_2 = d_2 \approx_y d_1$ for all $d_1, d_2 \in D_y$). Therefore, we sometimes call RDTs ranked data tables *over domains with similarities* to emphasize the presence of similarities in our model.

This is the basic framework in which we present the topics of efficient query execution. Let us stress again that unlike the approaches which build a ranking system on top of the classic RM as [20, 23], we introduce a generalized model (from the FLns perspective) which is in our opinion a conceptually clean way to cope with issues related to ranking and imperfect matches.

3 Query Execution

In this paper, we consider algebraic queries consisting of arbitrary combinations of operations which are counterparts to the classic restrictions (selections), projections, and joins which are explained in detail in the following subsections. The operations represent an important fragment of monotone operations that appear in our model and correspond to the most-widely used operations in relational query languages [13, 24].

Our goal is, given a query which can be seen as a term (a relational algebra expression as in [24]) consisting of (i) symbols for restrictions, joins, and projections and (ii) relation symbols (i.e., names of ranked data tables), describe execution of the query in a database instance (interpreting relation symbols by concrete RDTs and providing similarities on domains) so that tuples matching the query are listed in a *descending order according to their ranks.* The rationale is obvious—users who query the database want to see the best matches first and may want to stop searching in the results if either a desirable result is found or no desirable result is found after seeing a predefined number of best results. Thus, the basic principle of querying in our model is essentially the same as in the approach to monotone query evaluation proposed by Fagin and it is tempting to exploit the algorithm described in [15] (Fagin algorithm).

Note that [15] deals primarily with combination of queries obtained from independent subsystems. From our point of view, it deals with combinations of atomic subqueries. Our situation is technically more involved since we want to allow an arbitrary nesting of relational expressions. As a result, query execution in our model can be seen as a recursive application of a modification of the Fagin algorithm. Moreover, in order to apply the Fagin algorithm for similarity-based restrictions, we need to devise ways to efficiently list elements of domains similar to a given value (again, in a descending order according to their similarity). This and related issues are outlined in the following subsections.

3.1 Similarity-Based Restrictions

We consider the following operation of a similarity-based restriction: For an RDT $\mathcal{D}$ on R, attribute $y \in R$, and $d \in D_y$, we define a *similarity-based restriction* $\sigma_{y \approx d}(\mathcal{D})$ of $\mathcal{D}$ by $y \approx d$ by

$$\big(\sigma_{y\approx d}(\mathcal{D})\big)(r) = \mathcal{D}(r) \otimes (r(y) \approx_y d), \tag{6}$$

for all tuples $r \in \prod_{y\in R} D_y$. Hence, if $\mathcal{D}$ is a result of query Q, the rank given by (6) is a degree to which "r matches Q and its y-value is similar to d". Clearly, $\sigma_{y\approx d}(\cdots)$ is a counterpart to the restriction (selection) which appears in the classic RM and utilizes domain similarities instead of domain equalities. In the proposed language RESIQL [22], the corresponding query is written as Q `WHERE` $(y$ `~~` $d)$ with Q being a relational expression, optionally followed by a clause `TOP` k, meaning that only k best matches should be shown.

Note that both $\mathcal{D}(r)$ (for any r) and $r(y) \approx_y d$ (for any r) can be seen as two subqueries which are aggregated by $\otimes$. The operation $\otimes$ which appears in (6) is monotone and strict in sense of [15], i.e., we may utilize the Fagin algorithm to list tuples with the highest k ranks if we can supply adequate functions implementing the "sorted access" and "random access" for both the subqueries. In a more detail, in case of $\mathcal{D}$, the "sorted access" means listing one by one in descending order based on rank, the tuples r such that $\mathcal{D}(r) > 0$. If $\mathcal{D}$ is stored in a database, this can be efficiently done based on indexing tuples in $\mathcal{D}$ by ranks (e.g., by an ordinary B-tree index). The "random access" in case of $\mathcal{D}$ means for any r, we can retrieve $\mathcal{D}(r)$. Again, this can be efficiently done for a stored $\mathcal{D}$ based on a primary key (and the associated index). If $\mathcal{D}$ results from a subquery, we assume that both the sorted and random accesses are provided by evaluation of the subquery.

Considering the subquery which involves comparing similarity of y-values or tuples r with a fixed value d, we may argue as follows: The "random access" is tantamount to computing values of $\approx_y$ which we assume are supplied along with the data. Hence, the random access is trivial (and depends on the definition supplied by users). On the contrary, the "sorted access" which means listing for a given d one by one in descending order based on similarity with d, the values from the domain D_y (which appear in the RDT $\mathcal{D}$), represents a serious issue. Indeed, the virtue of our model is that it gives users quite a large freedom to choose similarity relations which best fit their needs (we have postulated just reflexivity and symmetry) which from the computational point of view can be seen as an obstacle for providing optimization on the general level since we cannot make specific assumptions on properties of the user-defined similarities. Nevertheless, we try to give a description of the "sorted access" for the most common scenarios which may occur.

Basically, there are two major types of domains with similarities deserving our attention—the domains of ordinal data (values that are naturally ordered) and domains of nominal data (finitely many distinct values). For both the cases, we now focus on the issue of obtaining a predefined number of values which are most similar to a specified value which is the core of the "sorted access".

Domains of Ordinal Data For domains of ordinal data, we can consider a general family of similarities for which we can implement an efficient "sorted access" by an algorithm which involves traversing data table with two cursors. Our approach is based on the following property of similarities:

Definition 2. Let $\lhd$ be a strict total order on D. Similarity $\approx\colon D \times D \to L$ is called *monotone with respect to* $\lhd$ if, for all elements $d_1, d_2, d_3 \in D$ such that $d_1 \lhd d_2 \lhd d_3$ we have $d_1 \approx d_3 \leq d_1 \approx d_2 \wedge d_2 \approx d_3$.

The monotony with respect to $\lhd$ is a desirable property which can be exploited to speed up the database access using B-tree indexes. On domains which are subsets of real numbers, one can define similarities monotone with respect to the genuine strict ordering of reals by computing the absolute difference between values, and subsequently, map the difference to L using an antitone scaling function. In a more general setting, one can check the following:

Theorem 1. *Let $f : \mathbb{R} \to \mathbb{R}$ be injective and monotone (or antitone) and let $s\colon \mathbb{R} \to L$ be antitone. Then, $\approx$ defined by*

$$d_1 \approx d_2 = s\big(|f(d_1) - f(d_2)|\big) \tag{7}$$

is monotone w.r.t. $<$. □

The algorithm can be now described as follows:

Input: Assume we have an index over attributes $y_1, \ldots, y_n$ of RDT $\mathcal{D}$ so that the values of y_n are indexed based on a total strict order $\lhd$; assume we are given domain elements $d_1 \in D_{y_1}, \ldots, d_n \in D_{y_n}$, and a similarity $\approx_{y_n}$ which is monotone with respect to $\lhd$. Furthermore, assume that $\mathbf{L}$ is linearly ordered.

Output: The result is a given number of tuples which are the best matches with respect to the following condition which generalizes the condition in (6):

$$(y_1 = d_1) \otimes \cdots \otimes (y_{n-1} = d_{n-1}) \otimes (y_n \approx_{y_n} d_n), \tag{8}$$

i.e., each output tuple r satisfies $r(y_1) = d_1, \ldots, r(y_{n-1}) = d_{n-1}$ (here $=$ denote identities on domains which are equal to 1 for d_1 and d_2 iff d_1 and d_2 are identical) and such tuples are listed one by one in descending order based on $r(y_n) \approx_{y_n} d_n$.

Initialization: The ranked data table $\mathcal{D}$ is traversed with two cursors, one moving forward and the second moving backward. First, both cursors are moved to a first tuple fully satisfying condition (8), i.e., satisfying condition:

$$(y_1 = d_1) \otimes \cdots \otimes (y_{n-1} = d_{n-1}) \otimes (y_n = d_n), \tag{9}$$

If no such tuple exists, both cursors are moved to the next closest tuple, w.r.t. the order given by attributes $y_1, \ldots, y_n$. Afterwards, the backward moving cursor moves to a preceding tuple.

Computation: Tuples are fetched using both cursors. During that, it is necessary to decide between cursors which tuple shall be returned to ensure that tuples are returned in the descending order according to their ranks. The fetch operation evaluates the similarity-based condition $y_n \approx_{y_n} d_n$ for the current tuple of each cursor and the tuple having the highest rank is returned (i.e., the tuple and the similarity degree are appended to the output) and the corresponding cursor is

moved to the next tuple in its direction. If the condition $y_n \approx_{y_n} d_n$ evaluates to zero, or if the cursor cannot move to the next tuple because there are no more tuples in its direction, it is no longer considered as a valid cursor and only the remaining cursor is used.

Termination: The computation phase terminates if either a predefined number of tuples has been returned or if both cursors are considered invalid.

Proof (Correctness of the algorithm). The algorithm terminates after finitely many steps. The monotony from Definition 2 ensures that the two cursors moving in the opposite directions cannot "skip" a tuple with a value which is more similar to d_n than any of the listed values because this would mean that the RDT $\mathcal{D}$ contains values $d_1 \lhd d_2 \lhd d_3$ such that $d_1 \approx d_3 \not\leq d_1 \approx d_2$ or $d_1 \approx d_3 \not\leq d_2 \approx d_3$, violating the monotony. Hence, if a tuple such that $y_n \approx_{y_n} d_n = 0$ is reached, it is certain that in the given direction is no tuple with a nonzero rank. □

Remark 1. Let us note that the algorithm can handle more complex conditions than (8). Namely, the similarity $\approx_{y_n}$ can be replaced by some other types of **L**-relations of domains. For instance, one can use $<$, $\leq$, and $\geq$ (with the usual bivalent interpretation) or a more complex expression preserving monotony of the similarity, e.g., $(y_n \approx_{y_n} d_n) \vee (y_n < d_n)$ which may be understood as prescribing a degree to which "the value of y_n does not exceed d_n too much".

Domains of Nominal Data Domains of nominal data can be viewed as relatively small finite sets (compared to typical sizes of RDTs) of values and the similarities among their values are provided by users as an enumeration (typically, the similarities are given by an expert in a particular domain).

Exploiting the finiteness of such domains, we can represent similarities on the domains by RDTs. In a more detail, for $\approx_y$ on a finite domain D_y with $|D_y| = n$ we can consider an RDT $\mathcal{D}_y$ on relation scheme $\{y, y'\}$ over domains $D_{y'} = D_y$ such that $\mathcal{D}_y(r) = r(y) \approx_y r(y')$, i.e., it is an RDT where rank indicates similarity of two values from the given domain. Observe that $\mathcal{D}_y$ is indeed an RDT since it contains finitely many (at most n^2 or $n(n-1)/2$ if the reflexivity and symmetry of $\approx_y$ are exploited) tuples with nonzero ranks. Therefore, $\mathcal{D}_y$ can be seen as a *materialized similarity relation* $\approx_y$.

Now, an efficient execution of a similarity-based restriction like (6) can be reduced to an efficient execution of a natural join. Indeed, we can write

$$\sigma_{y \approx d}(\mathcal{D}) = \mathcal{D} \bowtie \pi_{\{y\}}(\sigma_{y'=d}(\mathcal{D}_y)), \tag{10}$$

where $\sigma_{y'=d}(\mathcal{D}_y)$ denotes an ordinary equality-based restriction to tuples from $\mathcal{D}_y$ whose y'-values are exactly d, $\pi_{\{y\}}(\cdots)$ is a projection onto the attribute y as in the usual sense (the ranks are preserved, cf. Section 3.3), and $\bowtie$ is defined as in (3). Thus, the similarity-based restriction is expressed by a natural join (which is in fact a semijoin because y belongs to the relation scheme of $\mathcal{D}$).

For readers familiar with RESIQL [22], we note here how this particular approach can be used in the language. For instance, for a domain of "car engines"

(consisting of finitely many engine types), we create an RDT named *engine_sim* corresponding to $\mathcal{D}_y$ above and define an operator `~~engine` which specifies the similarity on the domain as follows:

```
CREATE OPERATOR engine ~~engine engine AS
  RANK FROM engine_sim WHERE value1 = $1 ∧ value2 = $2
       OTHERWISE 0.0::rank
  RETURNS rank;
```

Note that in the above statement, the `RANK FROM`-clause is a scalar expression [22] for retrieving the rank of a tuple in a ranked data tables. Retrieving similarity of two individual values is fast because the tuple encoding similarity of the values is found based on an index. In order to optimize restrictions based on similarity, our implementation of the language transforms queries like

```
RETRIEVE cars WHERE engine ~~engine 'V8 360-hp 5.7L' TOP 5;
```

to equivalent queries of the following form:

```
RETRIEVE cars
  NATURAL JOIN [value2 AS engine FROM
                engine_sim WHERE value1 = 'V8 360-hp 5.7L']
  TOP 5;
```

which are then efficiently executed. Note that the previous `RETRIEVE`-statement is in fact a direct application of (10) which is formalized in RESIQL.

3.2 Natural Joins

In the previous section, we have seen that efficient execution of certain similarity-based restrictions depends on efficient processing of natural joins. A *natural join of* $\mathcal{D}_1$ *and* $\mathcal{D}_2$ in our model is introduced as in (3) with $\mathcal{D}_1$ and $\mathcal{D}_2$ being arbitrary RDTs and $\otimes$ taken from $\mathbf{L}$. In RESIQL, natural joins (of relational expressions Q_1 and Q_2) are expressed as Q_1 `NATURAL JOIN` Q_2.

The algorithm for efficient execution of natural joins is based on modification of the Fagin algorithm. In order to make the computation efficient, we assume as in the ordinary case that both $\mathcal{D}_1$ and $\mathcal{D}_2$ have an index on the set of all common attributes. Under this assumptions, we may use a modification of the Fagin algorithm which consists of three phases, cf. [15]:

Sorted Access Phase: Retrieve tuples from $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively, in the descending order according to their ranks. Denote the sets of retrieved tuples by R_1 and R_2, respectively. Continue with enlarging R_1 and R_2 until

$$J = \{\langle rs, st\rangle \,|\, rs \in R_1 \text{ and } st \in R_2\} \tag{11}$$

has at least k elements.

Random Access Phase: For each $rs \in R_1$ retrieve all values $\mathcal{D}_2(st) > 0$ and store rst; For each $st \in R_2$ retrieve all values $\mathcal{D}_1(rs) > 0$ and store rst.

Computation Phase: Take the set S of all tuples rst stored in the previous step, sort the set according to $\mathcal{D}_1(rs) \otimes \mathcal{D}_2(st)$ and output first k records.

Remark 2. Note that the proposed algorithm differs from the basic Fagin algorithm mainly in considering pairs of joinable tuples (in the sorted access phase) instead of the same objects which are retrieved from both $\mathcal{D}_1$ and $\mathcal{D}_2$ as it is in [15]. The random access phase is adjusted accordingly and the computation phase remains the same as in [15]. Using the same argument as in [15], one can show based on upwards closed collections of tuples that the algorithm is correct. We postpone further analysis to an extended version of this paper.

3.3 Projections

If $\mathcal{D}$ is an RDT on T, the *projection* $\pi_R(\mathcal{D})$ of $\mathcal{D}$ onto $R \subseteq T$ is defined by

$$(\pi_R(\mathcal{D}))(r) = \bigvee \{ \mathcal{D}(rs) \mid s \in \textstyle\prod_{y \in T \setminus R} D_y \} \tag{12}$$

for each tuple $r \in \prod_{y \in R} D_y$. In this case, the efficient implementation of the operation is straightforward: Retrieve tuples rs from $\mathcal{D}$ in the descending order according to their ranks and if r has not been seen in any previous step, then output r with rank $\mathcal{D}(rs)$. If $\mathbf{L}$ is linear, the algorithm is correct. For nonlinear $\mathbf{L}$ the algorithm must be adjusted to properly compute a supremum of incomparable degrees (we omit details here).

3.4 Notes on Further Optimizations

By combination of similarity-based restrictions, natural joins, and projections, we can derive various similarity-based operations including similarity-based joins, semijoins, and closures. As a consequence, the optimizations we have introduced for these three fundamental operations can be utilized for the derived operations.

Furthermore, the query execution can be improved by applying laws represented by rewriting rules to simplify algebraic expressions in much the same way as the ordinary RDBMS. The key difference is that not all laws that hold in the ordinary RM can be applied in our model since the underlying logic is weaker than the Boolean logic. Nevertheless, a lot of important rules used to simplify queries are still valid. For instance, $\pi_S(\sigma_{y \approx d}(\mathcal{D})) = \sigma_{y \approx d}(\pi_S(\mathcal{D}))$ provided that $\mathcal{D}$ is an RDT on R and $y \in S \subseteq R$. Analogously, $\sigma_{y \approx d}(\mathcal{D}_1 \bowtie \mathcal{D}_2) = \sigma_{y \approx d}(\mathcal{D}_1) \bowtie \mathcal{D}_2$ whenever $\mathcal{D}_2$ is an RDT on R_2 and $y \notin R_2$. Many of the important laws are consequences of (4) and thus the adjointness property which justifies our selection of residuated lattices as the structures of ranks.

4 Experimental Evaluation and Conclusions

We have developed an experimental implementation of RESIQL in Java which incorporates optimizations outlined in this paper. Their efficiency can be viewed from two angles: (1) from the viewpoint of the overall query execution time, and

Table 1. Average number of fetched tuples

	index scan		full table scan		actual result	
dataset	fetched tuples	std. dev.	fetched tuples	std. dev.	tuples	std. dev.
cars	191.3	169.3	4,707.0	0.0	186.3	169.3
wine quality	776.7	566.4	5,320.0	0.0	772.0	566.1
bank	835.5	1,559.8	45,211.0	0.0	830.5	1559.8
adult	6,519.9	4,297.8	48,813.0	0.0	6,514.9	4297.9

Table 2. Average time to process similarity-based query (in milliseconds)

	index scan		full table scan	
dataset	time	std. dev.	time	std. dev.
cars	4.57	2.75	14.88	2.86
wine quality	8.27	5.07	17.23	2.57
bank	11.52	16.24	139.49	8.85
adult	39.99	24.65	165.61	15.69

(2) from the viewpoint of the number of tuples that have to be fetched from the physical database file. The second point of view is important since reading of data from physical files is usually the most demanding operation in real database management systems.

To assess the proposed algorithms from these two viewpoints, we have prepared a set of experiments using real-world datasets from the UCI Machine Learning Repository (wine quality, bank, adult) and our own dataset (cars). For one attribute in each dataset we had defined a nontrivial similarity and run one thousand random similarity-based queries. All experiments were performed twice—with and without an index over utilized attributes which had forced the database system to use our index scan algorithm and the naive full table scan algorithm, respectively.

Summary of the results is presented in Table 1 and Table 2. Apparently, the naive table scan algorithm is outperformed by our index scan algorithm both in terms of the overall execution time and also in terms of the number of tuples fetched from physical files, which, in fact, is very close to the number of tuples in the result set (see Table 1, third group of columns).

Conclusions. We have proposed algorithms for efficient execution of similarity-based queries in a generalized relation model of data which supports imperfect matches. We have focused mainly on algorithmic aspects connected to similarities defined on domains of ordinal and nominal data. The algorithms were proposed so that query results are obtained consecutively by ranks in the descending order without the need to make full tables scans and exploiting the usual B-tree indexes. The positive impact on the query execution performance has been demonstrated by experiments. This paper is an initial study of algorithmic issues in the model which will be continued in the future.

References

1. Belohlavek, R.: Fuzzy Relational Systems: Foundations and Principles. Kluwer Academic Publishers, Norwell (2002)
2. Belohlavek, R., Opichal, S., Vychodil, V.: Relational algebra for ranked tables with similarities: Properties and implementation. In: Berthold, M.R., Shawe-Taylor, J., Lavrac, N. (eds.) IDA 2007. LNCS, vol. 4723, pp. 140–151. Springer, Heidelberg (2007)
3. Bělohlávek, R., Vychodil, V.: Data tables with similarity relations: Functional dependencies, complete rules and non-redundant bases. In: Li Lee, M., Tan, K.-L., Wuwongse, V. (eds.) DASFAA 2006. LNCS, vol. 3882, pp. 644–658. Springer, Heidelberg (2006)
4. Belohlavek, R., Vychodil, V.: Query systems in similarity-based databases: logical foundations, expressive power, and completeness. In: ACM Symposium on Applied Computing (SAC), pp. 1648–1655. ACM (2010)
5. Buckles, B.P., Petry, F.E.: A fuzzy representation of data for relational databases. Fuzzy Sets and Systems 7(3), 213–226 (1982)
6. Cavallo, R., Pittarelli, M.: The theory of probabilistic databases. In: Proceedings of the 13th International Conference on Very Large Data Bases, VLDB 1987, pp. 71–81. Morgan Kaufmann Publishers Inc., San Francisco (1987)
7. Cintula, P., Hájek, P.: Triangular norm based predicate fuzzy logics. Fuzzy Sets and Systems 161, 311–346 (2010)
8. Codd, E.F.: A relational model of data for large shared data banks. Communications of the ACM 26, 64–69 (1983)
9. Dalvi, N., Ré, C., Suciu, D.: Probabilistic databases: diamonds in the dirt. Commun. ACM 52, 86–94 (2009)
10. Dalvi, N., Suciu, D.: Efficient query evaluation on probabilistic databases. The VLDB Journal 16, 523–544 (2007)
11. Dalvi, N., Suciu, D.: Management of probabilistic data: foundations and challenges. In: Proc. ACM PODS 2007, pp. 1–12. ACM, New York (2007)
12. Date, C.J., Darwen, H.: Databases, Types, and The Relational Model: The Third Manifesto, 3rd edn. Addison-Wesley (2006)
13. Date, C.J.: Database in Depth: Relational Theory for Practitioners: The Relational Model for Practitioners, 1st edn. O'Reilly Media (2005)
14. Esteva, F., Godo, L.: Monoidal t-norm based logic: towards a logic for left-continuous t-norms. Fuzzy Sets and Systems 124(3), 271–288 (2001)
15. Fagin, R.: Combining fuzzy information from multiple systems. J. Comput. Syst. Sci. 58(1), 83–99 (1999)
16. Goguen, J.A.: The logic of inexact concepts. Synthese 19, 325–373 (1979)
17. Gottwald, S.: Mathematical fuzzy logics. Bull. Symb. Logic 14(2), 210–239 (2008)
18. Gupta, R., Sarawagi, S.: Creating probabilistic databases from information extraction models. In: Proceedings of the 32nd International Conference on Very large Data Bases, VLDB 2006, pp. 965–976. VLDB Endowment (2006)
19. Hájek, P.: Metamathematics of Fuzzy Logic. Kluwer Academic Publishers, Dordrecht (1998)
20. Ilyas, I.F., Beskales, G., Soliman, M.A.: A survey of top-k query processing techniques in relational database systems. ACM Comp. Surv. 40(4), 11:1–11:58 (2008)
21. Klement, E.P., Mesiar, R., Pap, E.: Triangular Norms, 1st edn. Springer (2000)
22. Krajca, P., Vychodil, V.: Foundations of relational similarity-based query language RESIQL. In: Proc. 2013 IEEE Symposium on Foundations of Computational Intelligence (FOCI), pp. 15–23. IEEE (2013)

23. Li, C., Chang, K.C.C., Ilyas, I.F., Song, S.: Ranksql: query algebra and optimization for relational top-k queries. In: Proc. 2005 ACM SIGMOD, pp. 131–142 (2005)
24. Maier, D.: The Theory of Relational Databases. Computer Science Press (1983)
25. Prade, H., Testemale, C.: Generalizing database relational algebra for the treatment of incomplete or uncertain information and vague queries. Information Sciences 34(2), 115–143 (1984)

Variables for Controlling Cluster Sizes on Fuzzy *c*-Means

Yoshiyuki Komazaki[1] and Sadaaki Miyamoto[2]

[1] Master's Program in Risk Engineering,
University of Tsukuba
Ibaraki 305-8573, Japan
s1220596@u.tsukuba.ac.jp
[2] Department of Risk Engineering
University of Tsukuba
Ibaraki 305-8573, Japan
miyamoto@risk.tsukuba.ac.jp

Abstract. The fuzzy *c*-means proposed by Dunn and Bezdek is one of the most popular methods of fuzzy clustering. Clusters obtained by the fuzzy *c*-means are in the Voronoi sets when crisp reallocation rule is applied. This means that a part of a larger cluster may be assigned to a smaller one when there are clusters of different sizes. Therefore, some methods using variables for controlling cluster sizes have been proposed. In this paper, we study their theoretical properties and compare them using numerical examples.

1 Introduction

Fuzzy clustering means a method of clustering with fuzzy membership function for clusters. Fuzzy *c*-means proposed by Dunn [1] and Bezdek [2] is the most popular one, which we call here the standard fuzzy *c*-means (SFCM). SFCM has a simple objective function, and thus it has been studied by many authors and many different methods of fuzzy clustering have been proposed.

A major drawback to SFCM clustering is that it tends to make clusters of equal sizes. Namely, a part of a large cluster is misclassified as one of a smaller cluster if volumes of clusters are out of balance. Therefore some approaches using variables controlling cluster sizes have been proposed for tackling such a problem, and we discuss three methods here. One is derived from a modified entropy-based fuzzy *c*-means [3]. Another is a fuzzy extension of the maximum likelihood procedure [4], and the third is fuzzy *c*-means proposed by Ichihashi et al. [5], whose results are expected to be similar to those of the Gaussian mixture model.

All of these methods can solve the problem of cluster sizes. Nevertheless, there is no comparative study of these methods from theoretical viewpoint, and these methods are still open to discuss. The purpose of this paper is to study theoretical properties of these methods. We discuss them based on classifier functions [6] and thus our conclusions have generality.

V. Torra et al. (Eds.): MDAI 2013, LNAI 8234, pp. 192–203, 2013.

We first show SFCM as a basic algorithm of fuzzy c-means and three methods using variables for controlling cluster sizes in Section 2. Further, we show theoretical properties of these methods based on classifier functions in Section 3. We apply these methods to illustrative examples and show effectiveness and these properties with brief interpretations in Section 4. Finally, Section 5 concludes the paper.

2 Fuzzy c-Means with Cluster Sizes

In this section, we show the standard fuzzy c-means (SFCM) introduced by Dunn [1] and Bezdek [2] and algorithms with variables for cluster sizes [3,4,5].

2.1 Fuzzy c-Means

Let $X = \{x_1, \ldots, x_n\}$ be a set of objects for clustering. They are points in the p-dimensional Euclidean space $\mathcal{R}^p$. Let $V = \{v_1, \ldots, v_c\}$ be a set of centers of cluster i and let $U = (u_{ik})$ be an $c \times n$ matrix of fuzzy membership of x_k to cluster i. x_k and v_i are both p-dimensional vectors, i.e., $x_k = (x_k^1, \ldots, x_k^p)^T$ and $v_i = (v_i^1, \ldots, v_i^p)^T$.

SFCM is based on minimization of the following objective function:

$$J_{sfcm} = \sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m d_{ik}, \tag{1}$$

where d_{ik} is dissimilarity between x_k and v_i; m is fuzzy parameter which is larger than 1. Note that the objective function is obviously equal to that of k-means if the fuzzy parameter m is 1. The constraint of U is

$$\mathcal{U} = \{(u_{ik}) : u_{ik} \in [0, 1], \sum_{i=1}^{c} u_{ik} = 1, \forall k\}. \tag{2}$$

Unless noted otherwise, d_{ik} is the squared Euclidean norm:

$$d_{ik} = \|x_k - v_i\|^2 = \sum_{l=1}^{p} (x_k^l - v_i^l)^2. \tag{3}$$

The following iterative algorithm for minimizers J_{sfcm} is used.

Step 1. Generate c initial values for centroids V.
Step 2. Calculate optimal U that minimizes J_{sfcm} .
Step 3. Calculate optimal V that minimizes J_{sfcm} .
Step 4. If (U, V) is convergent, stop; else return to Step 2.

The optimal solutions of Step 2 and Step 3 are given by the Lagrangian multiplier method.

$$u_{ik} = \frac{\left(\frac{1}{d_{ik}}\right)^{\frac{1}{m-1}}}{\sum_{j=1}^{c}\left(\frac{1}{d_{jk}}\right)^{\frac{1}{m-1}}} \tag{4}$$

$$v_i = \frac{\sum_{k=1}^{n}(u_{ik})^m x_k}{\sum_{k=1}^{n}(u_{ik})^m} \tag{5}$$

Note that eq.(4) excludes the case when $d_{ik} = 0$ holds. If this is the case, then $u_{ik} = 1$ and $u_{jk} = 0$ $(\forall j \neq i)$.

In Step 4 we judge that the solution is convergent when U or V is unchanged.

2.2 Variables for Controlling Cluster Sizes

SFCM with crisp reallocation by the maximum membership rule may fail to divide accurately if there are unbalanced clusters like those in Fig.1 in Section 4. In the case of Fig.1, even if each centroids are at center of each circle, about 4.0 percent area of the left side of larger cluster must be assigned to the smaller one when crisp reallocation rule is applied. Therefore, three methods using variables for controlling cluster sizes have been proposed [3,4,5] for tackling such a problem.

The objective functions proposed in [3],[4] and [5], respectively, are as follows,

$$J_{fcma} = \sum_{i=1}^{c}\sum_{k=1}^{n}(\alpha_i)^{1-m}(u_{ik})^m d_{ik} \tag{6}$$

$$J_{pfcm} = \sum_{i=1}^{c}\sum_{k=1}^{n}(u_{ik})^m\{d_{ik} - \lambda\log(\alpha_i)\} \tag{7}$$

$$J_{efca} = \sum_{i=1}^{c}\sum_{k=1}^{n}u_{ik}\left\{d_{ik} + \lambda\log\left(\frac{u_{ik}}{\alpha_i}\right)\right\}, \tag{8}$$

where $A = (\alpha_1, \ldots, \alpha_c)$ is a variable for controlling cluster sizes, and λ is a positive parameter. The constraint for A is

$$\mathcal{A} = \left\{A = (\alpha_1, \ldots, \alpha_i) : \sum_{j=1}^{c}\alpha_j = 1; \alpha \geq 0, 1 \leq i \leq c\right\}. \tag{9}$$

Let us denote these three algorithms using the above objective functions as FCMA, PFCM and EFCA respectively. J_{fcma} has three variables U, V, and A, hence the following algorithm with three steps should be used.

Step1. Generate c initial values for V and A.
Step2. Calculate optimal U that minimizes J_{fcma} .
Step3. Calculate optimal V that minimizes J_{fcma} .
Step4. Calculate optimal A that minimizes J_{fcma} .
Step5. If (U, V, A) is convergent, stop; else return to Step2. .

PFCM and EFCA also use the same algorithm. The optimal solutions of each steps can be computed by the Lagrangian multiplier method.

Solutions for J_{fcma}

$$u_{ik} = \frac{\alpha_i \left(\frac{1}{d_{ik}}\right)^{\frac{1}{m-1}}}{\sum_{j=1}^{c} \alpha_j \left(\frac{1}{d_{jk}}\right)^{\frac{1}{m-1}}} \tag{10}$$

$$v_i = \frac{\sum_{k=1}^{n} (u_{ik})^m x_k}{\sum_{k=1}^{n} (u_{ik})^m} \tag{11}$$

$$\alpha_i = \frac{(\sum_{k=1}^{n} (u_{ik})^m d_{ik})^{\frac{1}{m}}}{\sum_{i=1}^{c} (\sum_{k=1}^{n} (u_{ik})^m d_{ik})^{\frac{1}{m}}} \tag{12}$$

Solutions for J_{pfcm}

$$u_{ik} = \frac{\left(\frac{1}{d_{ik} - \lambda \log \alpha_i}\right)^{\frac{1}{m-1}}}{\sum_{j=1}^{c} (\frac{1}{d_{jk} - \lambda \log \alpha_j})^{\frac{1}{m-1}}} \tag{13}$$

$$v_i = \frac{\sum_{k=1}^{n} (u_{ik})^m x_k}{\sum_{k=1}^{n} (u_{ik})^m} \tag{14}$$

$$\alpha_i = \frac{\sum_{k=1}^{n} (u_{ik})^m}{\sum_{i=1}^{c} \sum_{k=1}^{n} (u_{ik})^m} \tag{15}$$

Solutions for J_{efca}

$$u_{ik} = \frac{\alpha_i \exp(-\frac{d_{ik}}{\lambda})}{\sum_{j=1}^{c} \alpha_j \exp(-\frac{d_{jk}}{\lambda})} \tag{16}$$

$$v_i = \frac{\sum_{k=1}^{n} u_{ik} x_k}{\sum_{k=1}^{n} u_{ik}} \tag{17}$$

$$\alpha_i = \frac{\sum_{k=1}^{n} u_{ik}}{n} \tag{18}$$

3 Classifier Function

After finishing clustering, we are able to set a value of membership to a new object by classifier function. In the case of SFCM, the following is considered [6].

$$U_i^s(x) = \frac{\left(\frac{1}{d(v_i,x)}\right)^{\frac{1}{m-1}}}{\sum_{j=1}^{c} \left(\frac{1}{d(v_j,x)}\right)^{\frac{1}{m-1}}}. \tag{19}$$

This function is simply derived from the optimal solution of u_{ik}, where v_i ($i = 1, \ldots, c$) are the converged centroids. A classifier function helps us to consider the theoretical properties of clustering because it is defined in the whole space.

We can convert the result of fuzzy clustering to crisp clusters by regarding an object having the maximum value of membership to cluster i as a member of cluster i.

Now, a region of cluster i in SFCM is represented as the following.

$$U_i^s(x) > U_j^s(x) \tag{20}$$

$$\Leftrightarrow \left(\frac{1}{d(v_i,x)}\right)^{\frac{1}{m-1}} > \left(\frac{1}{d(v_j,x)}\right)^{\frac{1}{m-1}} \tag{21}$$

$$\Leftrightarrow d(v_i,x) < d(v_j,x) \tag{22}$$

Hence, the region of cluster i is

$$R_i = \{x \in \mathcal{R}^p \ : \ d(v_i,x) < d(v_j,x),\ j \neq i\} \tag{23}$$

It shows that the result of SFCM makes the Voronoi regions whose representative point is v_i. Now, as x approaches infinity in a region of cluster i, we obtain

$$\lim_{\|x\|\to\infty} U_i^s(x) = \frac{1}{c}. \tag{24}$$

In this way, we make characteristics of method clear by analyzing its classifier function. The classifier function of three methods using variables controlling size of clusters is the following.

$$U_i^a(x) = \frac{\alpha_i \left(\frac{1}{d(v_i,x)}\right)^{\frac{1}{m-1}}}{\sum_{j=1}^c \alpha_j \left(\frac{1}{d(v_j,x)}\right)^{\frac{1}{m-1}}} \tag{25}$$

$$U_i^p(x) = \frac{\left(\frac{1}{d(v_i,x)-\lambda\log\alpha_i}\right)^{\frac{1}{m-1}}}{\sum_{j=1}^c \left(\frac{1}{d(v_j,x)-\lambda\log\alpha_j}\right)^{\frac{1}{m-1}}} \tag{26}$$

$$U_i^e(x) = \frac{\alpha_i \exp\left(-\frac{d(v_i,x)}{\lambda}\right)}{\sum_{j=1}^c \alpha_j \exp\left(-\frac{d(v_j,x)}{\lambda}\right)} \tag{27}$$

The next propositions show theoretical properties of these classifier functions.

Proposition 1. *As x approaches infinity in an unbounded region R_i, then*

$$\lim_{\|x\|\to\infty} U_i^a(x) = \alpha_i \tag{28}$$

$$\lim_{\|x\|\to\infty} U_i^p(x) = \frac{1}{c} \tag{29}$$

$$\lim_{\|x\|\to\infty} U_i^e(x) = 1 \tag{30}$$

is obtained.

These can be confirmed visually by Fig.2 in Section 4.

Proposition 2. *As x approaches v_i, u_{ik} approaches unity in FCMA, however it doesn't approach unity in PFCM or EFCA, namely,*

$$\lim_{x \to v_i} U_i^a(x) = 1 \tag{31}$$

$$\lim_{x \to v_i} U_i^p(x) = \frac{1}{1 + C_p} < 1 \tag{32}$$

$$\lim_{x \to v_i} U_i^e(x) = \frac{1}{1 + C_e} < 1, \tag{33}$$

where

$$C_p = \sum_{j=1, j \neq i}^{c} \left(\frac{\lambda \log \alpha_i}{d(v_j, x) - \lambda \log \alpha_j} \right)^{\frac{1}{m-1}} \tag{34}$$

$$C_e = \alpha_i^{-1} \sum_{j=1, j \neq i}^{c} \alpha_j \exp\left(-\frac{d(v_j, x)}{\lambda} \right). \tag{35}$$

The proofs of Proposition 1 and 2 are obvious and thus the detail is omitted.

Proposition 3. *The region of cluster i is multiplicatively weighted Voronoi region[7] in FCMA, and locally additively weighted Voronoi in EFCA and PFCM. Each representative point of the regions is v_i $(i = 1, \ldots, c)$. Multiplicatively weighted Voronoi region i is defined as*

$$R_i = \left\{ x \in \mathcal{R}^p \ : \ \frac{d(v_i, x)}{w_i} < \frac{d(v_j, x)}{w_j}, \ j \neq i \right\}, \tag{36}$$

and additively weighted Voronoi region i is defined as

$$R_i = \{ x \in \mathcal{R}^p \ : \ d(v_i, x) - w_i < d(v_j, x) - w_j, \ j \neq i \}, \tag{37}$$

where $w_i > 0$ $(i = 1, \ldots, c)$ are weights of the region i.

Proof. Each boundary between cluster i and cluster j given by $U_i(x) = U_j(x)$ is as follows.

FCMA

$$\begin{aligned} & U_i^a(x) = U_j^a(x) \\ \Leftrightarrow \ & \alpha_i \left(\frac{1}{d(v_i, x)} \right)^{\frac{1}{m-1}} = \alpha_j \left(\frac{1}{d(v_j, x)} \right)^{\frac{1}{m-1}} \\ \Leftrightarrow \ & \alpha_i^{m-1} \frac{1}{d(v_i, x)} = \alpha_j^{m-1} \frac{1}{d(v_j, x)} \\ \Leftrightarrow \ & \frac{d(v_j, x)}{\alpha_i^{m-1}} = \frac{d(v_j, x)}{\alpha_j^{m-1}} \end{aligned} \tag{38}$$

PFCM

$$
\begin{aligned}
& U_i^p(x) = U_j^p(x) \\
\Leftrightarrow & \left(\frac{1}{d(v_i,x) - \lambda \log \alpha_i}\right)^{\frac{1}{m-1}} = \left(\frac{1}{d(v_j,x) - \lambda \log \alpha_j}\right)^{\frac{1}{m-1}} \\
\Leftrightarrow & d(v_i,x) - \lambda \log \alpha_i = d(v_j,x) - \lambda \log \alpha_j \\
\Leftrightarrow & d(v_i,x) - \lambda \log \frac{1}{\alpha_j} = d(v_j,x) - \lambda \log \frac{1}{\alpha_i}
\end{aligned} \tag{39}
$$

EFCA

$$
\begin{aligned}
& U_i^e(x) = U_j^e(x) \\
\Leftrightarrow & \alpha_i \exp\left(-\frac{d(v_i,x)}{\lambda}\right) = \alpha_j \exp\left(-\frac{d(v_j,x)}{\lambda}\right) \\
\Leftrightarrow & \log \alpha_i - \frac{d(v_i,x)}{\lambda} = \log \alpha_j - \frac{d(v_j,x)}{\lambda} \\
\Leftrightarrow & d(v_i,x) - \lambda \log \frac{1}{\alpha_j} = d(v_j,x) - \lambda \log \frac{1}{\alpha_i}
\end{aligned} \tag{40}
$$

The above indicates that FCMA makes multiplicatively weighted Voronoi region with weights α_i^{m-1} , while PFCM and EFCM make locally additively weighted Voronoi region with weights $\lambda \log(1/\alpha_i)$ (for cluster $j \neq i$). 'Locally' means that a weight of a region is dependent on a pair of clusters, in other words, the weight of a region between region i and j is different from the weight of the region considering between regions i and k.

Note that these propositions imply that the region of cluster i ($i = \arg\max_i \alpha_i$) is infinite while the region of cluster j ($j = 1, \ldots, c, j \neq i$) is finite in FCMA. Additionally, the boundary is locally linear (hyper-plane) when the dissimilarity function d is defined as the squared Euclidean norm, while boundary is locally hyperbolic when d is defined as the Euclidean norm in PFCM or EFCM.

4 Numerical Examples

The purpose of this paper is to give theoretical properties of methods with variables for controlling cluster sizes, hence we show only the result of simple illustrative examples in this section, and omit the result of applying to real examples.

4.1 First Data Set

Figure 1 is an artificially generated data set with two groups: one has 20 objects randomly in a circle with the radius of 1.0 , the other has 180 objects randomly

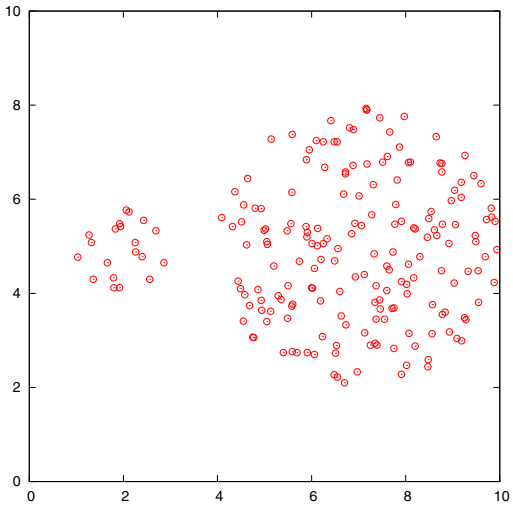


Fig. 1. Artificially generated data set with two groups: one has 20 objects in circle with the radius of 1.0, the other has 180 objects in circle with the radius of 3.0 and the distance between the centers of two circles is 5.0

in a circle with the radius of 3.0 and the distance between the centers of two circles is 5.0.

Figure 2 shows the results of clustering the data set as shown in Fig.1 ($c = 2$) with SFCM, FCMA, PFCM and EFCA, respectively, and with $\lambda = 5.0, m = 1.6$. In the figure, the objects of two clusters are displayed in small squares or small circles, and the centroids are cross marks. The contours denote the membership value, and increment is 0.1. Solid line in the contours, which shows the membership value is 0.5, indicates the boundary between two clusters. This data set has two clusters, which are small and large. SFCM makes a Voronoi diagram when the maximum membership rule is applied, thus a part of large cluster is misclassified as a part of smaller cluster as shown in Fig.2**(a)** while three methods consider these cluster sizes and succeed in having good clusters as shown in Fig.2**(b)-(d)**.

Centroid Inside and Outside of Its Region. PFCM and EFCA represent cluster sizes by additive weights, while FCMA represents them by multiplicative weights, whereby PFCM and EFCA may output an odd result: there is no centroid in its region. Such a result is shown when $d(v_i, v_j) < |\lambda \log(\alpha_i/\alpha_j)|$. Figure 3 shows the results of clustering data in Fig.1 by PFCM and EFCA,

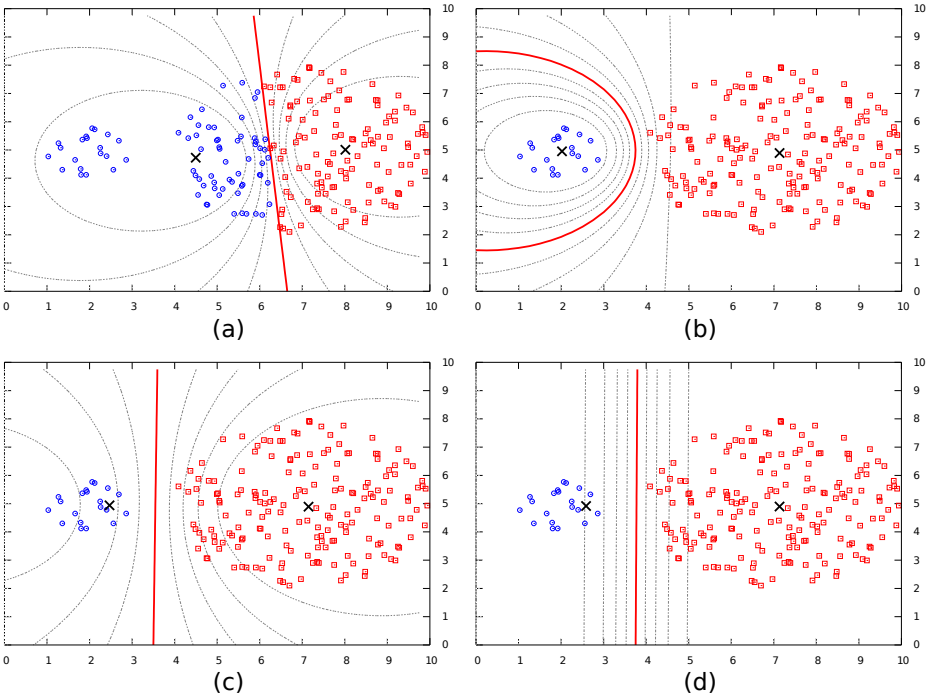


Fig. 2. Clusters when **(a)**SFCM, **(b)**FCMA, **(c)**PFCM and **(d)**EFCA were applied to the data set shown as Fig.1. A part of larger cluster is misclassified as a part of larger cluster in SFCM while FCMA, PFCM and EFCA succeed in having good clusters.

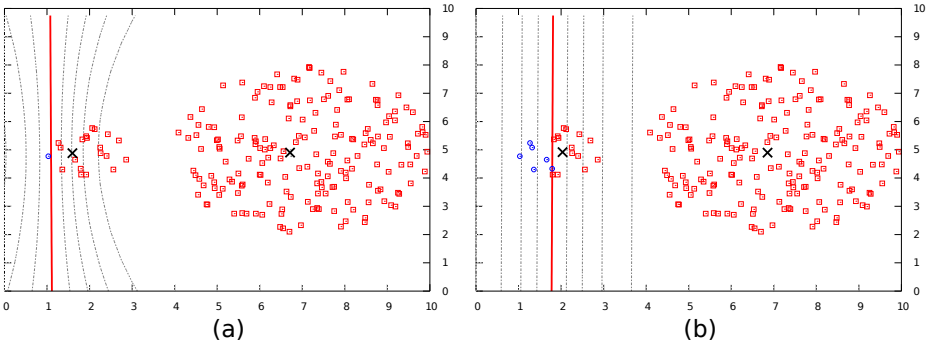


Fig. 3. Clusters when **(a)**PFCM and **(b)**EFCM with too large λ were applied to the data set shown as Fig.1. The centroid of smaller cluster is out of its region.

where $\lambda = 7.75$ and $\lambda = 8.20$ respectively. In these case, $|\lambda \log(\alpha_i/\alpha_j)| = 31.29$, $d(v_i, v_j) = 26.18$ in Fig.3**(a)**, and $|\lambda \log(\alpha_i/\alpha_j)| = 25.54$, $d(v_i, v_j) = 23.14$ in Fig.3**(b)**, therefore the centroid of a smaller cluster is in the region of a larger cluster. Note that FCMA doesn't output such results because multiplicative weights are used, however it is not flexible since it has only one parameter m.

4.2 Second Data Set

Figure 4 shows an artificially generated data set with three groups: one has 180 objects randomly in a circle with the radius of 3.0 , another has 80 objects randomly in a circle with the radius of 2.0 and the other has 20 objects randomly in a circle with the radius of 1.0.

Figure 5 shows the results of clustering the data set in Fig.4 ($c = 3$). The contours denote the membership value of the largest cluster. This data set has three clusters, which are small, medium and large. In this case, no matter what value of m or initial V, SFCM fails in a good classification: a centroid in small group side is pulled by larger cluster since SFCM tries to make clusters equally as shown in Fig.5**(a)**. On the other hand, the three methods are able to succeed as shown in Fig.5**(b)-(d)**. These results indicate that these methods may work well when there are three ore more clusters.

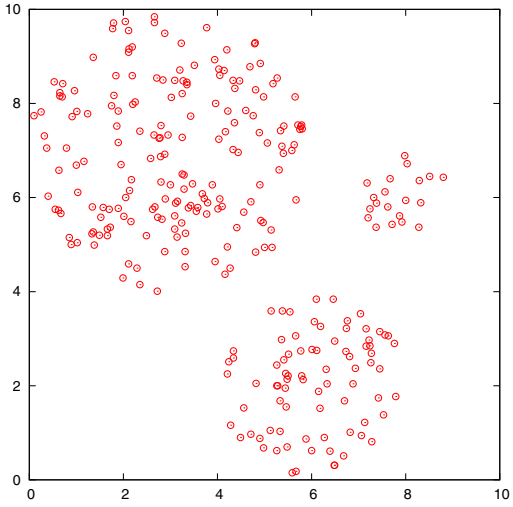


Fig. 4. Artificially generated data set with three groups: one has 180 objects in circle with the radius of 3.0, another has 80 objects in circle with the radius of 2 and the other has 20 objects in circle with the radius of 1.0

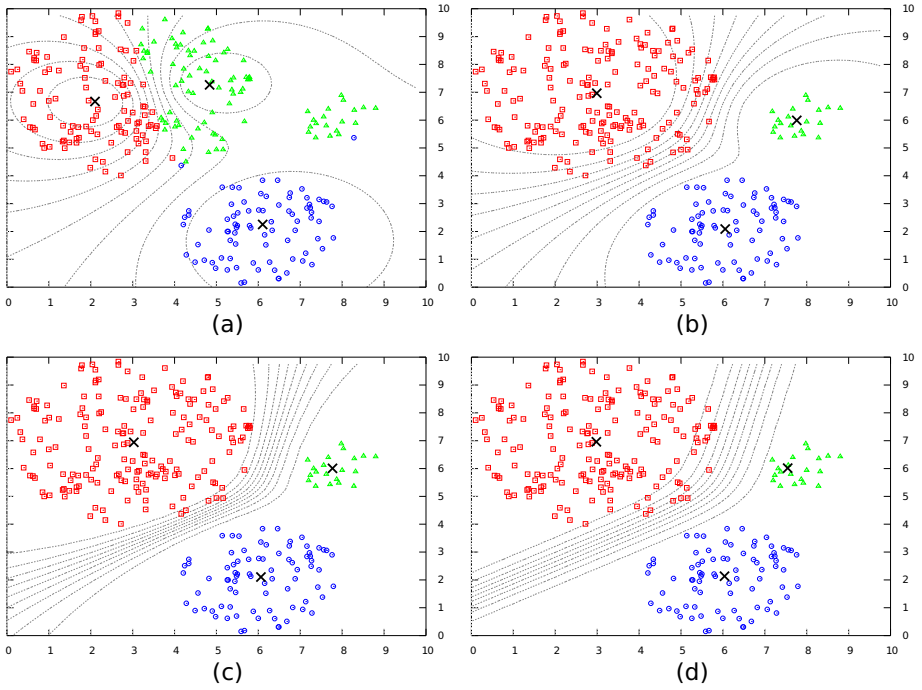


Fig. 5. Clusters when **(a)**SFCM, **(b)**FCMA, **(c)**PFCM and **(d)**EFCA were applied to the data set shown as Fig.4. The centroid of small cluster side is pulled by large cluster in SFCM while FCMA, PFCM and EFCA succeed in having good clusters.

5 Conclusion

In this paper, we described three methods with variables for controlling cluster sizes, and showed their theoretical properties using their classifier functions. Furthermore we applied these methods to illustrative examples and showed that these methods worked well. Each of the methods outputted different results though all of these methods were able to handle the cluster sizes.

From a practical viewpoint, the terms of covariance variables within clusters should also be used [6,5] with appropriate parameters. However we omitted discussion of this topic in this paper for simplicity. Besides, there are rooms for further discussion of the combination of the kernel method or the addition of constraints in semi-supervised clustering and comparison with other approaches, for example conditional FCM [8,9], whose constraint of membership is continuously updated during clustering, for our future works.

Acknowledgment. This work has partly been supported by the Grant-in-Aid for Scientific Research, Japan Society for the Promotion of Science, No. 23500269.

References

1. Dunn, J.C.: A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. Cybernetics and Systems 3(3), 32–57 (1973)
2. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Kluwer Academic Publishers, Norwell (1981)
3. Miyamoto, S., Kurosawa, N.: Controlling cluster volume sizes in fuzzy c-means clustering. In: SCIS and ISIS, Yokohama, Japan, pp. 1–4 (September 2004)
4. Yang, M.S.: On a class of fuzzy classification maximum likelihood procedures. Fuzzy Sets Syst. 57(3), 365–375 (1993)
5. Ichihashi, H., Honda, K., Tani, N.: Gaussian mixture pdf approximation and fuzzy c-means clustering with entropy regularization. In: Proc. of the 4th Asian Fuzzy System Symposium, pp. 217–221 (2000)
6. Miyamoto, S., Ichihashi, H., Honda, K.: Algorithms for fuzzy clustering. Springer, Heidelberg (2008)
7. Boots, B.N.: Weighting thiessen polygons. Economic Geography, 248–259 (1980)
8. Noordam, J., Van Den Broek, W., Buydens, L.: Multivariate image segmentation with cluster size insensitive fuzzy c-means. Chemometrics and Intelligent Laboratory Systems 64(1), 65–78 (2002)
9. Lai, Y., Huang, P., Lin, P.: An integrity-based fuzzy c-means method resolving cluster size sensitivity problem. In: 2010 International Conference on Machine Learning and Cybernetics (ICMLC), vol. 5, pp. 2712–2717. IEEE (2010)

On Sequential Cluster Extraction Based on L_1-Regularized Possibilistic Non-metric Model

Yukihiro Hamasuna[1] and Yasunori Endo[2]

[1] Department of Informatics, School of Science and Engineering,
Kinki University,
Kowakae 3-4-1, Higashi-osaka, Osaka, 577-8502, Japan
yhama@info.kindai.ac.jp
[2] Faculty of Engineering, Information and Systems,
University of Tsukuba, Tennodai 1-1-1, Tsukuba, Ibaraki, 305-8573, Japan
endo@risk.tsukuba.ac.jp

Abstract. The fuzzy non-metric model is one of the clustering methods in which the membership grade of each datum to each cluster is calculated directly from dissimilarities between data. The cluster center which is referred to as representative of cluster is not used in fuzzy non-metric model. This paper discusses a new possibilistic approach for non-metric model from the viewpoint of being in the cluster. In the previous study, new possibilistic clustering and its variant have been proposed by using L_1-regularization. These possibilistic clustering methods with L_1-regularization induce a change in the membership function. Two types of non-metric model based on possibilistic approach named L_1-regularized possibilistic non-metric model are proposed in this paper. Next, the way of sequential extraction algorithm is also discussed. Moreover, the results of sequential extraction based on proposed methods are shown.

Keywords: possibilistic clustering, non-metric model, L_1-regularization, sequential cluster extraction.

1 Introduction

The aim of data analysis is to discover important structures from massive and complex databases. Clustering is one of the data analysis method which divides a set of objects into some groups called clusters. Objects classified in same cluster are considered similar, while objects classified in different cluster are considered dissimilar. Fuzzy c-means clustering (FCM) is the most well-known clustering method [1–3]. Possibilistic clustering (PCM) is also well-known as one of the useful methods as well as FCM. Because PCM is robust against noise and outliers which negatively affect clustering results[4]. The robustness for noise or outliers is essential for clustering methods to be useful in real world applications [5]. A procedure of sequential cluster extraction has been proposed by using this drawback [5, 6]. A proposal of algorithms extracting "one cluster at a time" is based on the idea of noise clustering [7]. Sequential cluster extraction does not

V. Torra et al. (Eds.): MDAI 2013, LNAI 8234, pp. 204–213, 2013.

need to determine the number of clusters in advance. This advantage is quite important for massive and complex data sets to detect dense cluster.

The fuzzy non-metric model (FNM) is also one of the clustering method in which the membership grade of each datum to each cluster is calculated directly from dissimilarities between data [8]. The cluster center which is referred to as representative of cluster is not used in FNM. Then, data space need not necessarily be Euclidean space. Therefore, relational data is handled in FNM and other relational clustering methods such as Ref. [10]. Some studies for handling relational data have been discussed [11, 12].

A constraint for membership grade is considered in FCM, while it is not considered in PCM. In order to obtain nontrivial solutions, particular additional terms with respect to membership grade are considered in PCM. L_1-regularization is well-known as useful technique and applied to induce the sparseness, that is, small variables are calculated as zero [13]. In the field of clustering, sparse possibilistic clustering method has been proposed by introducing L_1-regularization [14]. This method induces the sparseness with calculating the small membership grade as zero. This means that it induces the sparseness from the viewpoint of not being in the cluster. It should be also considered that the sparseness for being in the field of clustering. From that sense, crisp possibilistic c-means clustering (CPCM) has been proposed and described its classification function [15]. The way of sequential cluster extraction by CPCM has also been proposed.

In this paper, we will propose two types of non-metric model based on possibilistic approach with L_1-regularization named L_1-regularized possibilistic non-metric model (L_1PNM) from the viewpoint of handling relational data and constructing sequential extraction algorithm. This paper is organized as follows: In section 2, we introduce some symbols and fuzzy non-metric model. In section 3, we propose two types of L_1-regularized possibilistic non-metric model (L_1PNM). In section 4, we show the algorithm of sequential extraction. In section 5, we show the results of sequential extraction based on proposed method. In section 6, we conclude this paper.

2 Preparation

A set of objects to be clustered is given and denoted by $X = \{x_1, \ldots, x_n\}$ in which x_k $(k = 1, \ldots, n)$ is an object. In most cases, $x_1, \ldots, x_n$ are p-dimensional vectors $\Re^p$, that is, a datum $x_k \in \Re^p$. A cluster is denoted as $C_i (i = 1, \ldots, c)$. A membership grade of x_k belonging to C_i and a partition matrix is also denoted as u_{ki} and $U = (u_{ki})_{1 \leq k \leq n,\ 1 \leq i \leq c}$.

2.1 Fuzzy Non-metric Model

Fuzzy non-metric model (FNM) and entropy based FNM (EFNM) are based on optimizing an objective function under the constraint for membership grade [8, 9].

We consider following two objective functions J and J_e.

$$J(U) = \sum_{i=1}^{c}\sum_{k=1}^{n}\sum_{t=1}^{n}(u_{ki})^m(u_{ti})^m r_{kt},$$
$$J_e(U) = \sum_{i=1}^{c}\sum_{k=1}^{n}\sum_{t=1}^{n}u_{ki}u_{ti}r_{kt} + \lambda\sum_{i=1}^{c}\sum_{k=1}^{n}u_{ki}\log u_{ki}.$$

here, $m > 1.0$ and $\lambda > 0.0$ is fuzzification parameters and r_{kt} means a dissimilarity measure between x_k and x_t. One of the examples of r_{kt} is the squared L_2-norm between data:

$$r_{kt} = \|x_k - x_t\|^2.$$

J is the objective function of FNM [8] and J_e is the one of EFNM [9].
Probabilistic constraint for FNM and EFNM is as follows:

$$\mathcal{U}_f = \left\{(u_{ki}) : u_{ki} \in [0,1],\ \sum_{i=1}^{c}u_{ki} = 1,\ {}^\forall k\right\}.$$

2.2 Algorithm of Fuzzy Non-metric Model

The algorithm of FNM and EFNM is described as Algorithm 1.

3 L_1-Regularized Possibilistic Non-metric Model

3.1 Objective Function and Optimal Solution

The objective functions of L_1-regularized possibilistic non-metric model (L_1PNM) and entropy based method (EL_1PNM) are based on the one of FNM and EFNM. We consider following objective function for L_1PNM:

$$J_{lp}(U) = \sum_{i=1}^{c}\sum_{k=1}^{n}\sum_{t=1}^{n}(u_{ki})^m(u_{ti})^m r_{kt} + \gamma\sum_{i=1}^{c}\sum_{k=1}^{n}|1-u_{ki}|$$

$m > 1.0$ and $\gamma > 0.0$ are the parameters of L_1PNM. The condition $\mathcal{U}_p$ for FNM is written as follows:

$$\mathcal{U}_p = \left\{(u_{ki}) \ :\ u_{ki} \in [0,1],\quad {}^\forall k\right\}, \tag{1}$$

where, we have omitted original constraint $0 < \sum_{k=1}^{n}u_{ki} \leq n$. d_{ki} is as follows:

$$d_{ki} = \sum_{t=1}^{n}(u_{ki})^m r_{kt}. \tag{2}$$

The main problem of constructing the algorithm of L_1PNM is how to derive the optimal solution of u_{ki}. Each membership u_{ki} could be solved separately

Algorithm 1. Algorithm of FNM and EFNM

STEP1 Set initial values for u_{ki} and parameters.

STEP2 Calculate d_{ki} for FNM as follows:

$$d_{ki} = \sum_{t=1}^{n} (u_{ki})^m r_{kt}$$

and for EFNM as follows:

$$d_{ki} = \sum_{t=1}^{n} u_{ki} r_{kt}$$

STEP3 Calculate $u_{ki} \in U$ for FNM as follows:

$$u_{ki} = \frac{\left(\frac{1}{d_{ki}}\right)^{\frac{1}{m-1}}}{\sum_{l=1}^{c} \left(\frac{1}{d_{kl}}\right)^{\frac{1}{m-1}}}$$

and for EFNM as follows:

$$u_{ki} = \frac{\exp\left(-d_{ki}/\lambda\right)}{\sum_{l=1}^{c} \exp\left(-d_{kl}/\lambda\right)}$$

STEP4 If convergence criterion is satisfied, stop. Otherwise go back to **STEP2**.

in L_1PNM procedure because of the condition $\mathcal{U}_p$. First, we will consider the following semi-objective function:

$$J_{lp}^{ki}(u_{ki}) = (u_{ki})^m d_{ki} + \gamma \left|1 - u_{ki}\right|.$$

We will decompose $1 - u_{ki} = \xi^+ - \xi^-$, in order to obtain partial derivatives with respect to u_{ki} where all element of ξ^+ and ξ^- are nonnegative. The semi-objective function is rewritten by using decomposition method [16] as follows:

$$J_{lp}^{ki}(u_{ki}) = (u_{ki})^m d_{ki} + \gamma \left(\xi^+ + \xi^-\right).$$

Constraints are as follows:

$$1 - u_{ki} \leq \xi^+, \quad 1 - u_{ki} \geq -\xi^-, \quad \xi^+, \xi^- \geq 0.$$

Introducing the Lagrange multipliers β^+, β^-, ψ^+, and $\psi^- \geq 0$, Lagrangian L^{lp} is as follows:

$$\begin{aligned} L^{lp} = {} & (u_{ki})^m d_{ki} + \gamma \left(\xi^+ + \xi^-\right) \\ & + \beta^+ \left(1 - u_{ki} - \xi^+\right) + \beta^- \left(-1 + u_{ki} - \xi^-\right) - \psi^+ \xi^+ - \psi^- \xi^-. \end{aligned} \tag{3}$$

From $\frac{\partial L^{lp}}{\partial \xi^+} = 0$ and $\frac{\partial L^{lp}}{\partial \xi^-} = 0$,

$$\gamma - \beta^+ - \psi^+ = 0, \quad \gamma - \beta^- - \psi^- = 0. \tag{4}$$

Since $\psi^+,\ \psi^- \geq 0$, conditions $0 \leq \beta^+ \leq \gamma$ and $0 \leq \beta^- \leq \gamma$ are obtained from (4). Substituting (4) into (3), the Lagrangian L^{lp} is simplified as follows:

$$L^{lp} = (u_{ki})^m d_{ki} + \beta(1 - u_{ki}). \tag{5}$$

Here, $\beta = \beta^+ - \beta^-$ and satisfies $-\gamma \leq \beta \leq \gamma$.
From $\frac{\partial L^{lp}}{\partial u_{ki}} = 0$,

$$u_{ki} = \left(\frac{\beta}{md_{ki}}\right)^{\frac{1}{m-1}}. \tag{6}$$

Substituting (6) to (5), the Lagrangian dual problem is written as follows:

$$L_d^{lp} = \left(\frac{\beta}{md_{ki}}\right)^{\frac{m}{m-1}} d_{ki} + \beta\left\{1 - \left(\frac{\beta}{md_{ki}}\right)^{\frac{1}{m-1}}\right\}.$$

From $\frac{\partial L_d^{lp}}{\partial \beta} = 0$, this dual problem is solved as,

$$\beta = md_{ki}. \tag{7}$$

The optimal solution of primal problem is derived by considering (6), (7) and $-\gamma \leq \beta \leq \gamma$. In the case of $\beta < 0$ is not realized since md_{ki} is always positive. Second, the case of $0 \leq \beta \leq \gamma$, the optimal solution is $u_{ki} = 1$ since $\beta = md_{ki}$. Third, the case of $\gamma < \beta$, the optimal solution is $u_{ki} = \left(\frac{\gamma}{md_{ki}}\right)^{\frac{1}{m-1}}$. Finally, the optimal solution for u_{ki} of L_1PNM is derived as follows:

$$u_{ki} = \begin{cases} 1 & 0 \leq d_{ki} \leq \frac{\gamma}{m} \\ \left(\frac{\gamma}{md_{ki}}\right)^{\frac{1}{m-1}} & \frac{\gamma}{m} < d_{ki} \end{cases} \tag{8}$$

3.2 Entropy Based L_1PNM

Next, we will consider the objective function of entropy based L_1-regularized possibilistic non-metric model (EL_1PNM). We consider the following objective function for EL_1PNM:

$$J_{elp}(U) = \sum_{i=1}^{c}\sum_{k=1}^{n}\sum_{t=1}^{n} u_{ki}u_{ti}r_{kt} + \lambda\sum_{i=1}^{c}\sum_{k=1}^{n} u_{ki}\left(\log u_{ki} - 1\right) + \gamma\sum_{i=1}^{c}\sum_{k=1}^{n}|1 - u_{ki}|.$$

here, $\lambda > 0.0$ and $\gamma > 0.0$ are the parameters of EL_1PNM. The constraint for u_{ki} remain the same as (1). d_{ki} is as follows:

$$d_{ki} = \sum_{t=1}^{n} u_{ki}r_{kt}. \tag{9}$$

In order to derive the optimal solution of u_{ki} for EL_1PNM, we will consider the following semi-objective function as well as L_1PNM:

$$J_{elp}^{ki}(u_{ki}) = u_{ki}d_{ki} + \lambda u_{ki}\left(\log u_{ki} - 1\right) + \gamma\left|1 - u_{ki}\right|.$$

As the same procedure of L_1PNM, the Lagrangian L^{elp} is simplified as follows:

$$L^{elp} = u_{ki}d_{ki} + \lambda u_{ki}\left(\log u_{ki} - 1\right) + \beta(1 - u_{ki}). \tag{10}$$

Here, $-\gamma \leq \beta \leq \gamma$ is considered.
From $\frac{\partial L^{elp}}{\partial u_{ki}} = 0$,

$$u_{ki} = \exp\left(-\frac{d_{ki} - \beta}{\lambda}\right). \tag{11}$$

Substituting above to L^{elp}, the Lagrangian dual problem is written as follows:

$$L_d^{elp} = \beta - \lambda \exp\left(-\frac{d_{ki} - \beta}{\lambda}\right).$$

From $\frac{\partial L_d^{elp}}{\partial \beta} = 0$, this dual problem is solved as,

$$\beta = d_{ki}. \tag{12}$$

The optimal solution of primal problem is derived as the same procedure of L_1PNM. In the case of $\beta < 0$ is not realized since d_{ki} is always positive. Second, the case of $0 \leq \beta \leq \gamma$, the optimal solution is $u_{ki} = 1$ since $\beta = d_{ki}$. Third, the case of $\gamma < \beta$, the optimal solution is $u_{ki} = \exp\left(-\frac{d_{ki}-\gamma}{\lambda}\right)$. Finally, the optimal solution for u_{ki} of EL_1PNM is derived as follows:

$$u_{ki} = \begin{cases} 1 & 0 \leq d_{ki} \leq \gamma \\ \exp\left(-\dfrac{d_{ki} - \gamma}{\lambda}\right) & \gamma < d_{ki} \end{cases} \tag{13}$$

3.3 Algorithm of Proposed Method

The algorithm of L_1PNM is described as Algorithm 2. Eqs. **A**, **B** used in each algorithm follow Table 1.

Algorithm 2. Algorithm of L_1PNM

L_1**PNM1** Set initial values and parameters.
L_1**PNM2** Calculate d_{ki} by using Equation **A**.
L_1**PNM3** Calculate $u_{ki} \in U$ by using Equation **B**.
L_1**PNM4** If convergence criterion is satisfied, stop. Otherwise go back to L_1**PNM2**.

Table 1. The dissimilarity d_{ki} and optimal solution of $u_{ki}$for L_1PNM and EL_1PNM

Algorithm	Eq. **A**	Eq. **B**
L_1PNM	(2)	(8)
EL_1PNM	(9)	(13)

4 Sequential Extraction Algorithm

The objective function of PCM can be minimized separately because probabilistic constraint used in FCM is not considered [5]. This implies the drawback that the cluster centers calculated by PCM would be completely the same. The sequential extraction procedure is constructed by considering this drawback. The basis of this procedure has been already proposed [5] and discussed [6]. We consider the sequential extraction algorithm by L_1PNM. The datum has small d_{ki} allocated near the data has small r_{kt}. Then, the membership grade of such data could be calculated as $u_{ki} = 1$. These data should be considered in one cluster which satisfies $u_{ki} = 1$. Therefore, L_1PNM can extract one cluster at a time by minimizing the objective function in the case of $c = 1$. The algorithm of sequential extraction by L_1PNM is described as Algorithm 3.

Algorithm 3. Sequential cluster extraction algorithm based on L_1PNM

STEP 1 Give X, initial values u_{ki} and parameters m or λ and γ.
STEP 2 Repeat L_1PNM algorithm with $c = 1$ until convergence criterion is satisfied.
STEP 3 Extract $\{x_k \mid u_{ki} = 1\}$ from X.
STEP 4 If $X = \emptyset$ or convergence criterion is satisfied, stop. Otherwise, give initial values and go back to **STEP2**.

5 Numerical Examples

We show the numerical examples of sequential extraction with butterfly data set and polaris data set described in Figs. 1 and 2. The butterfly data set consists of 15 data point and two attributes and should be classified into two clusters. The polaris data set which consists of 51 data point and two attributes should be classified into three clusters.

These figures are results of conventional entropy based FCM with $\lambda = 0.5$ [2]. In these figures, $\bullet$, $\times$, $+$ are clusters and $\star$ means cluster centers.

We set the fuzzification parameter $\lambda = 100$ used in EL_1RPNM. First, we show the results of butterfly data set by sequential cluster extraction with $\gamma = 20.0$ and $\gamma = 30.0$ described Figs. 3 and 4, respectively. Next, we show the results of polaris data set by sequential extraction with $\gamma = 70.0$, $\gamma = 80.0$, $\gamma = 90.0$, and $\gamma = 100.0$ described in Figs. 5, 6, 7, and 8, respectively.

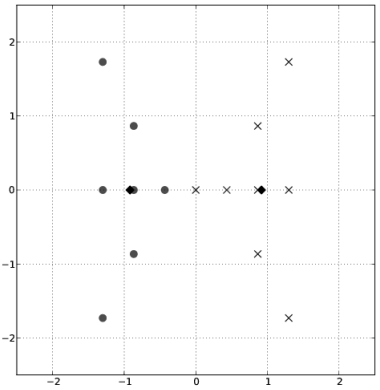


Fig. 1. Butterfly data ($n = 15$, $p = 2$) should be classified into two clusters

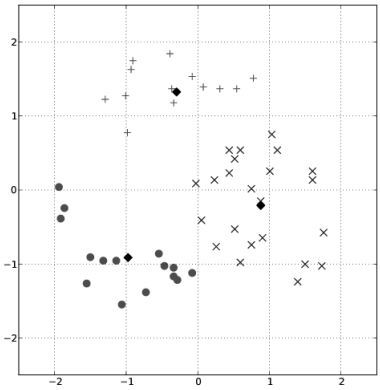


Fig. 2. Polaris data ($n = 51$, $p = 2$) should be classified into three clusters

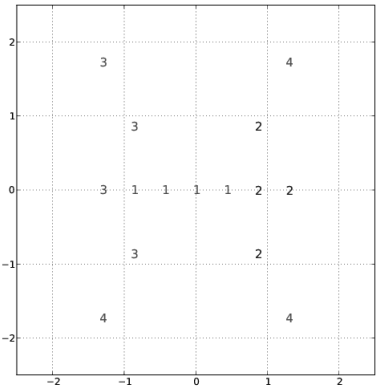


Fig. 3. Sequential cluster extraction by EL_1PNM with $\lambda = 100.0$, $\gamma = 20.0$, number of extracted cluster is 4

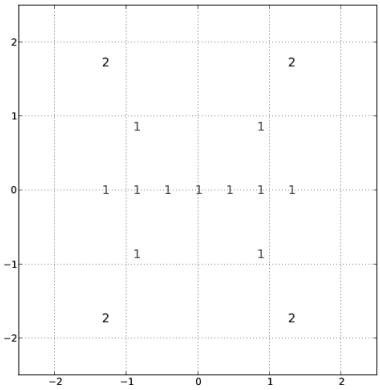


Fig. 4. Sequential cluster extraction by EL_1PNM with $\lambda = 100.0$, $\gamma = 30.0$, number of extracted cluster is 2

The value displayed on each data point means the order of extracting clusters. These results shows that the large λ induces the broad area which satisfies $u_{ki} = 1$ and extracts small number of clusters. It is verified that the sequential extraction algorithm by L_1PNM and EL_1PNM are strongly depended on initial values and fail to extract suitable clusters from these results and other experimental results. Therefore, the other sequential clustering methods which is robust for initial values have to be considered for handling the data set which consists of only dissimilarities between data and relational data.

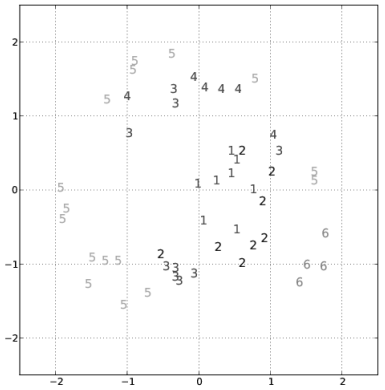


Fig. 5. Sequential cluster extraction by EL_1PNM with $\lambda = 100.0$, $\gamma = 70.0$, number of extracted cluster is 6

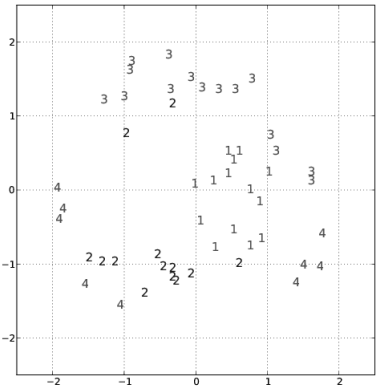


Fig. 6. Sequential cluster extraction by EL_1PNM with $\lambda = 100.0$, $\gamma = 80.0$, number of extracted cluster is 4

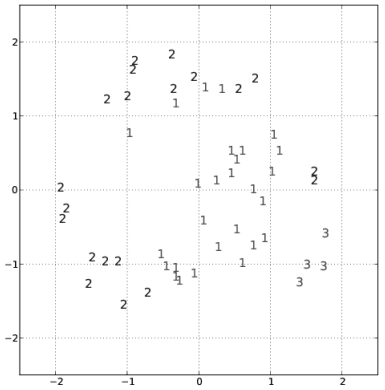


Fig. 7. Sequential cluster extraction by EL_1PNM with $\lambda = 100.0$, $\gamma = 90.0$, number of extracted cluster is 3

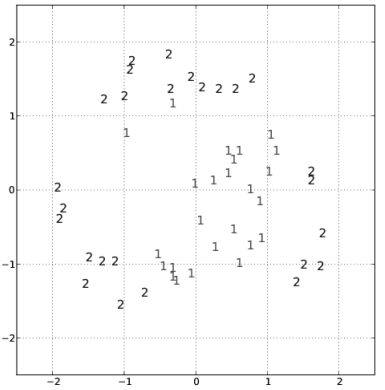


Fig. 8. Sequential cluster extraction by EL_1PNM with $\lambda = 100.0$, $\gamma = 100.0$, number of extracted cluster is 2

6 Conclusions

In this paper, we have proposed L_1-regularized possibilistic non-metric model (L_1PNM) and entropy based L_1PNM (EL_1PNM). We have moreover shown the results of sequential cluster extraction based on proposed method.

In future works, we will consider other sequential clustering methods which is robust for initial values for handling data set which only consists of dissimilarities between data or relational data.

References

1. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
2. Miyamoto, S., Mukaidono, M.: Fuzzy c-means as a regularization and maximum entropy approach. In: Proc. of the 7th International Fuzzy Systems Association World Congress (IFSA 1997), vol. 2, pp. 86–92 (1997)
3. Miyamoto, S., Ichihashi, H., Honda, K.: Algorithms for Fuzzy Clustering. STUDFUZZ, vol. 229. Springer, Heidelberg (2008)
4. Krishnapuram, R., Keller, J.M.: A possibilistic approach to clustering. IEEE Transactions on Fuzzy Systems 1(2), 98–110 (1993)
5. Davé, R.N., Krishnapuram, R.: Robust clustering methods: A unified view. IEEE Transactions on Fuzzy Systems 5(2), 270–293 (1997)
6. Miyamoto, S., Kuroda, Y., Arai, K.: Algorithms for Sequential Extraction of Clusters by Possibilistic Method and Comparison with Mountain Clustering. Journal of Advanced Computational Intelligence and Intelligent Informatics (JACIII) 12(5), 448–453 (2008)
7. Davé, R.N.: Characterization and detection of noise in clustering. Pattern Recognition Letters 12(11), 657–664 (1991)
8. Roubens, M.: Pattern classification problems and fuzzy sets. Fuzzy Sets and Systems 1, 239–253 (1978)
9. Endo, Y.: On Entropy Based Fuzzy Non Metric Model – Proposal, Kernelization and Pairwise Constraints –. Journal of Advanced Computational Intelligence and Intelligent Informatics (JACIII) 16(1), 169–173 (2012)
10. Ruspini, E.: Numerical methods for fuzzy clustering. Information Science 2(3), 319–350 (1970)
11. Hathaway, R.J., Davenport, J.W., Bezdek, J.C.: Relational Duals of the c-Means Clustering Algorithms. Pattern Recognition 22(2), 205–212 (1989)
12. Hathaway, R.J., Bezdek, J.C.: Nerf c-Means: Non-Euclidean Relational Fuzzy Clustering. Pattern Recognition 27(3), 429–437 (1994)
13. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society Series B 58(1), 267–288 (1996)
14. Inokuchi, R., Miyamoto, S.: Sparse Possibilistic Clustering with L1 Regularization. In: Proc. of The 2007 IEEE International Conference on Granular Computing (GrC 2007), pp. 442–445 (2007)
15. Hamasuna, Y., Endo, Y.: On Sparse Possibilistic Clustering with Crispness – Classification Function and Sequential Extraction. In: Joint 6th International Conference on Soft Computing and Intelligent Systems and 12th International Symposium on Advanced Intelligent Systems (SCIS & ISIS 2012), pp. 1801–1806 (2012)
16. Tsuda, K., Kudo, T.: Clustering graphs by weighted substructure mining. In: Proc. of the 23rd International Conference on Machine Learning (ICML), pp. 953–9960 (2006)

Fast Implementations of Markov Clustering for Protein Sequence Grouping*

László Szilágyi[1,2] and Sándor Miklos Szilágyi[3]

[1] Sapientia - Hungarian Science University of Transylvania,
Faculty of Technical and Human Science, Tîrgu-Mureş, Romania
lalo@ms.sapientia.ro
[2] Budapest University of Technology and Economics, Department of Control Engineering and Information Technology, Budapest, Hungary
[3] Petru Maior University of Tîrgu-Mureş, Romania

Abstract. Two efficient versions of a Markov clustering algorithm are proposed, suitable for fast and accurate grouping of protein sequences. First, the essence of the matrix splitting approach consists in optimal reordering of rows and columns in the similarity matrix after every iteration, transforming it into a matrix with several compact blocks along the diagonal, and zero similarities outside the blocks. These blocks are treated separately in later iterations, thus significantly reducing the overall computational load. Alternately, a special sparse matrix architecture is employed to represent the similarity matrix of the Markov clustering algorithm, which also helps getting rid of a severe amount of unnecessary computations. The proposed algorithms were tested to classify sequences of protein databases like SCOP95. The proposed solutions achieve a speed-up factor in the range 15-300 compared to the conventionally implemented Markov clustering, depending on input data size and parameter settings, without damaging the partition accuracy. The convergence is usually reached in 40-50 iterations. Combining the two proposed approaches brings us close to the 1000 times speed-up ratio.

Keywords: Markov clustering, bioinformatics, protein sequence classification, unsupervised classification.

1 Introduction

By definition, protein families represent groups of molecules with relevant sequence similarity [3]. Establishing protein families in large databases is one of the fundamental goals of functional genomics. A successful classification may contribute to the delineation of functional diversity of homologous proteins, and can provide valuable evolutionary insights as well [5]. Members of such protein families may serve similar or identical biological purposes [9]. Identifying these

* This work was supported by the Hungarian National Research Funds (OTKA) under grant no. PD103921, the Hungarian Academy of Science through the János Bolyai Fellowship Program.

V. Torra et al. (Eds.): MDAI 2013, LNAI 8234, pp. 214–225, 2013.

families is generally performed by clustering algorithms [6], supported by pairwise similarity or dissimilarity measures. Well established properties of some proteins in the family may be reliably transferred to other members whose functions are not well known [8].

TRIBE-MCL is an efficient clustering method proposed for protein sequence classification [5], based on Markov chain theory [4]. It assigns a graph structure to the protein database such a way that each protein has a corresponding node, while initial edge weights in the graph represent computed pairwise similarity values, obtained via BLAST search methods [1]. Clusters are then obtained by alternately applying two matrix operations called inflation and expansion.

In this paper we introduce two efficient approaches aimed to accelerate the execution speed of the algorithm, without damaging the outcome of the clusters. The first proposed approach optimizes the execution via splitting the similarity matrix into several smallers ones once the graph has been disintegrated into isolated subgraphs. The second one uses a special sparse matrix structure to model the similarity matrix, reducing the computational burden by eliminating the unnecessary computations with zeros. Further on, these two approaches are combined in a third one, which will be formulated after the numerical tests.

The remainder of this paper is structured as follows: Section 2 takes into account the functional details of the TRIBE-MCL algorithm. Section 3 presents the details of the proposed efficient TRIBE-MCL algorithms. Section 4 evaluates and discusses the efficiency of the proposed method. Section 5 presents the conclusions and gives some hints for further research.

2 Background

TRIBE-MCL is an iterative algorithm, which operates on a directional graph. Each of the n nodes of the graph represents a protein sequence from the set we wish to cluster, while each edge length S_{ij}, $i, j = 1 \dots n$, shows the similarity between protein sequences of index i and j, respectively. Edge lengths are stored in the $n \times n$ similarity matrix S. Initial edge lengths usually come from pairwise sequence alignment. During the iterations, S behaves as a column stochastic matrix, whose elements represent probabilities of transitions (evolution).

The TRIBE-MCL algorithm consists of two main operations, namely the inflation and expansion, which are repeated alternately until a convergence is reached, that is, the similarity matrix becomes invariant during a cycle:

1. Inflation has the main goal to differentiate among connections within the graph, favoring more likely direct walks along the graph in the detriment of less likely walks. It is computed via taking each element of the similarity matrix to the power of $r > 1$. The strength of this differentiation is controlled by the so called inflation rate r: large values express the preference of likely walks more severely, causing sudden ruptures within the graph, possibly not in the ideal place. Low inflation rates are more likely to yield smooth partitions, but the convergence may become rather slow.

2. Expansion operation is intended to reveal possible longer walks along the graph, to emphasize changes within the protein structures that happened in two or more evolutionary steps. Expansion in achieved via matrix multiplication, by taking similarity matrix S to the second power.

Auxiliary computations are also included in each iteration, in order to maintain the similarity matrix S as a symmetric column stochastic matrix.

Clusters are defined as connected subgraphs within the graph described by the similarity matrix, so a stable state of the similarity matrix means that the clusters don't change their contents during an iteration.

In a previous paper [14], we have proposed a series of generalizations of the conventional version of the TRIBE-MCL algorithm [5], e.g. time-variant inflation rate, generalized inflation scheme, singleton filter, etc. These changes brought slight improvements to the accuracy and efficiency of the algorithm.

3 Methods

In this paper we introduce two implementations of the TRIBE-MCL algorithm, with the aim of seriously reducing its computational load, without harming the accuracy of classification. We will test the proposed method on the proteins of the SCOP95 database.

3.1 The SCOP95 Database

The SCOP (Structural Classification of Proteins) database [12] contains protein sequences in order of tens of thousands, hierarchically classified into classes, folds, superfamilies and families [2]. The SCOP95 database involved in this study, is a subset of SCOP (version 1.69), which contains 11944 proteins, exhibiting a maximum similarity of 95% among each other. Pairwise similarity and distance matrices (BLAST [1], Smith-Waterman [13], Needleman-Wunsch [10], PRIDE [7], etc.) are available at the Protein Classification Benchmark Collection [11]. In this study we employ BLAST similarity measures, because that one suppresses low similarities, thus contributing to computational load reduction.

3.2 Matrix Splitting

Most of the computational load of the algorithm is caused by the matrix multiplication, which has a theoretical complexity of $\mathcal{O}(n^3)$. In order to reduce runtime, it would be beneficial at any time of the execution, to separate those proteins which no longer have any influence upon the others. This idea we employed in the previous paper [14], where we proposed to exclude the rows and columns of singletons from the similarity matrix in each iteration. This way we achieved $30\% - 50\%$ reduction of the overall processing time, depending on the percentage of singletons within the data.

In the following, we will formulate a more optimal separation scheme of clusters. Let us denote by Σ the initial set of proteins, which is intended to be

Data: similarity matrix $S = [s_{ij}]$ with $1 \leq i, j \leq n$
Result: reordering buffer R, number of clusters q, indexes of first elements of clusters $Q_1 \dots Q_q$

$m \leftarrow 0$;
$q \leftarrow 0$;
$M \leftarrow \Phi$;
while $m < n$ **do**
  Find smallest $i \in \{1, 2, ...n\}$ such that $i \notin M$;
  $m \leftarrow m + 1$;
  $R_m \leftarrow i$;
  $M \leftarrow M \cup \{i\}$;
  $q \leftarrow q + 1$;
  $Q_q \leftarrow m$;
  fifo.push(i);
  while fifo *not empty* **do**
    l = fifo.pop();
    for *each* $j \in \{1, 2, ...n\}$ *with* $j \notin M$ *and* $s_{lj} > 0$ **do**
      $m \leftarrow m + 1$;
      $R_m \leftarrow j$;
      $M \leftarrow M \cup \{j\}$;
      fifo.push(j);
    end
  end
end
$Q_{q+1} \leftarrow n + 1$;

Algorithm 1. The subgraph identification function

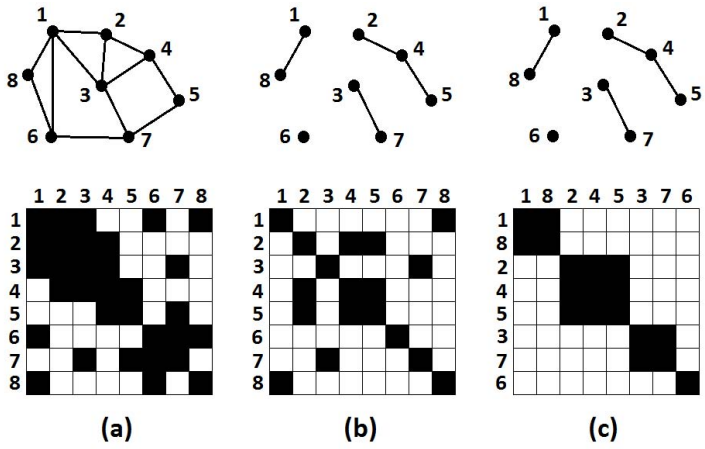


Fig. 1. Permutation of columns and rows: (a) an initial graph with several connections and the corresponding similarity matrix; (b) after a certain amount of iterations the graph breaks into pieces; (c) reordering the rows and columns in the matrix makes the similarity matrix contain non-zero blocks along the diagonal. At this given matrix splitting, the reordering buffer contains $R = [1, 8, 2, 4, 5, 3, 7, 6]$, the number of isolated subgraphs is $q = 4$, while the stored indexes of first elements are $Q = [1, 3, 6, 8]$.

classified. At any iteration t, we may look for isolated subgraphs in the graph represented by the similarity matrix S. Whenever we find a subset of proteins $\Sigma_1 \subset \Sigma$, corresponding to a connected subgraph isolated from the rest of the proteins ($s_{ij} = 0$, $\forall i \in \Sigma_1$ and $j \in \Sigma \setminus \Sigma_1$), in further iterations we may treat the proteins of Σ_1 separately from the others, because the rows and columns of S corresponding to these proteins will not interact with any other rows and columns. If we reorder all rows and columns of the similarity matrix S such a way, that isolated subgraphs are placed in consecutive rows and columns, we will have a similarity matrix formed by small square shaped blocks of nonzero elements placed along the main diagonal, and all other elements of the matrix will be zero.

In order to implement this idea, we need to define a reordering buffer R of size n, which will contain the permuted protein indexes corresponding to isolated subgraphs in the graph represented by S. Further on, we need a group buffer Q to store the indexes of initial elements of protein groups within the reordering buffer. The latter buffer will need a time-variant size of storage (denoted by q), but it will never exceed the limit of n items. Algorithm 1 presents the procedure of localizing isolated subgraphs within the graph. In this procedure, M represents the set of graph nodes already found during the process. The procedure sequentially looks for seed nodes which were not yet found and occupies the isolated subgraph using existing connections between nodes. Figure 1 exhibits the outcome of a column and row reordering, splitting an 8×8 matrix into four small matrices.

Having the isolated groups of nodes separated, we may reformulate the operations performed within each iteration as follows. For each square block along the diagonal of reordered matrix S, that is, for each $b \in \{1, 2, \ldots, q\}$, we consider the subset of proteins in the connected subgraph $\Sigma_b = \{R_{Q_b}, R_{Q_b+1}, \ldots, R_{Q_{b+1}-1}\}$ assuming that $Q_{q+1} = n + 1$, and then

– inflation is computed as:

$$s_{\alpha\beta}^{(\text{new})} = \left(s_{\alpha\beta}^{(\text{old})}\right)^r \qquad \forall \alpha, \beta \in \Sigma_b, \tag{1}$$

– expansion is given by the formula:

$$s_{\alpha\beta}^{(\text{new})} = \sum_{\gamma \in \Sigma_b} s_{\alpha\gamma}^{(\text{old})} s_{\gamma\beta}^{(\text{old})} \qquad \forall \alpha, \beta \in \Sigma_b, \tag{2}$$

– normalization is given by:

$$s_{\alpha\beta}^{(\text{new})} = s_{\alpha\beta}^{(\text{old})} \left(\sum_{\gamma \in \Sigma_b} s_{\gamma\beta}^{(\text{old})} \right)^{-1} \qquad \forall \alpha, \beta \in \Sigma_b, \tag{3}$$

– symmetry is approximated as:

$$s_{\alpha\beta}^{(\text{new})} = s_{\beta\alpha}^{(\text{new})} = \sqrt{s_{\alpha\beta}^{(\text{old})} s_{\beta\alpha}^{(\text{old})}} \tag{4}$$

$\forall \alpha, \beta \in \Sigma_b$, $\alpha < \beta$. After symmetrization, similarity values below ε are reduced to 0.

The proposed matrix splitting algorithm is summarized in Algorithm 2. Two parameters need to be set at the beginning: the inflation rate $r > 1$ and threshold ε around 10^{-3}. Generally 30-50 iterations are needed for a stable convergence. After 15 iterations most of the clusters are in their final form.

3.3 Sparse Matrix

The sparse matrix is a memory saving representation for matrices which contain a low amount of non-zero values. The sparse matrix stores only the non-zero values together with its coordinates (row and column). In our case, a non-zero element in the similarity matrix requires at least twice more bytes than an element of an two-dimensional array. Whenever using matrices of low density, employing sparse matrices will reduce the necessary storage space.

Sparse matrices also contribute to the efficiency of the algorithm. While computing the normalization of a column, zero elements are not added to the sum, thus reducing the number of additions. In fact, a zero element can only change to non-zero during the expansion. But also in case of the expansion, zero elements in the input do not affect the outcome of any element of the output matrix.

In a conventional sparse matrix structure, the non-zero elements of each column are stocked in a chained list, ordered by row coordinate. Thus the sparse matrix has an array of list head pointers, each one pointing to the first non-zero element of the corresponding column. Each non-zero element is represented by the structure $(row, value, next)$. The latter variable in the structure is a pointer to the next non-zero element in the column.

In a conventional sparse matrix, the inflation operation requires a single parsing of each column and thus the power computation is only performed for non-zero elements. The normalization needs to parse twice each column: first it computes the sum of each column and then it divides all non-zero elements by the sum of the column. Assuring matrix symmetry is more complicated, because it requires searching for the transposed for each non-zero element.

Expansion requires a new sparse matrix for the output. During the computation of the expanded matrix, the elements of each column are determined in such an order, that new non-zero elements are always placed at the end of the list. That is why, it is worth to have a pointer to the tail of the column list as well (see Fig. 2). Further on, as expansion is computed right after having made the similarity matrix symmetric, we may approximate the element s_{ij} as:

$$s_{ij}^{(\text{new})} = \sum_{k=1}^{n} s_{ik} s_{kj} \approx \sum_{k=1}^{n} s_{ik} s_{jk} \quad , \tag{1}$$

which is easier to compute as columns are way easier to parse than rows in this data structure.

Data: similarity matrix $S = [s_{ij}]$ with $1 \leq i, j \leq n$
Result: same similarity matrix S
$m \leftarrow n$; $q \leftarrow 1$; $M \leftarrow \Sigma$; $Q_q \leftarrow 1$; $Q_{q+1} \leftarrow n + 1$;
repeat
  for $b \in \{1, 2, ...q\}$ **do**
    $\Sigma_b \leftarrow \{R_{Q_b}, R_{Q_b+1}, \ldots, R_{Q_{b+1}-1}\}$;
    `Inflation`;
    for $\alpha, \beta \in \Sigma_b$ **do**
      $s_{\alpha\beta} \leftarrow s^r_{\alpha\beta}$;
    end
    `Normalization`;
    $S' \leftarrow \mathbf{0}$;
    for $\beta \in \Sigma_b$ **do**
      $z \leftarrow 0$;
      for $\gamma \in \Sigma_b$ **do**
        $z \leftarrow z + s_{\gamma\beta}$;
      end
      for $\alpha \in \Sigma_b$ **do**
        $s'_{\alpha\beta} \leftarrow s_{\alpha\beta}/z$;
      end
    end
    `Symmetry`;
    for $\alpha, \beta \in \Sigma_b$ *with* $\alpha < \beta$ **do**
      $z \leftarrow \sqrt{s'_{\alpha\beta} s'_{\beta\alpha}}$;
      if $z < \varepsilon$ **then**
        $z \leftarrow 0$;
      end
      $s_{\alpha\beta} \leftarrow z$; $s_{\beta\alpha} \leftarrow z$;
    end
    `Normalization again, as above`;
    `Expansion`;
    $S \leftarrow \mathbf{0}$;
    for $\alpha, \beta \in \Sigma_b$ **do**
      $z \leftarrow 0$;
      for $\gamma \in \Sigma_b$ **do**
        $z \leftarrow z + s'_{\alpha\gamma} s'_{\gamma\beta}$;
      end
      $s_{\alpha\beta} \leftarrow z$;
    end
  end
  `Call Subgraph Identification function`;
until *convergence occurs*;

Algorithm 2. Algorithm for Tribe-MCL via matrix splitting. S and S' are two instances of the similarity matrix, necessary for the correct handling of data

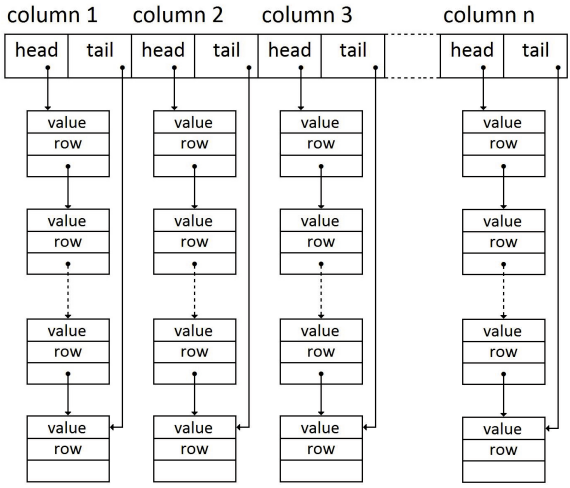


Fig. 2. Data structure used by the sparse matrix implementation

4 Results and Discussion

The main goal of protein clustering is to reveal hidden similarities among proteins. When evaluating the accuracy of the output, one can count the number of mixed clusters (those which contain proteins from two or more different families) and their cardinality. We have shown in the previous work [14], that the inflation rate is the main factor to influence the amount of mixed clusters. The approach proposed here computes exactly the same partitions as the conventional TRIBE-MCL, in a more efficient way. That is why the evaluation of accuracy is unnecessary in this study. The reader interested in accuracy details is referred to [14].

We have employed the proposed algorithms to classify either the whole set of 11944 proteins in the SCOP database, or selected subsets. At the selection of subsets, whole families were chosen from the hierarchical data structure, in order to keep all connections of each selected protein. The hierarchical structure of the SCOP database was only used to select input data and verify the final partition accuracy. Partitioning only uses the pairwise similarity data.

Fig. 3 summarizes some efficiency tests performed on a set of 908 proteins (all families from SCOP95 which have 11 to 14 proteins): varying the inflation rates between 1.3 and 2.0, the duration of each iteration was recorded and plot in this figure. In case of the matrix splitting approach, after only 4-6 iterations completed, the large connected block within the similarity graph is broken into small subgraphs, enabling us to compute subsequent iterations on very small matrices. Late iterations are performed approximately 1000 times quicker. Although the computation load stabilizes at a low level after the initial few iterations, the convergence of the output data requires around 40-50 cycles. Without this proposed efficient scheme, all iterations would need the same amount of computations as

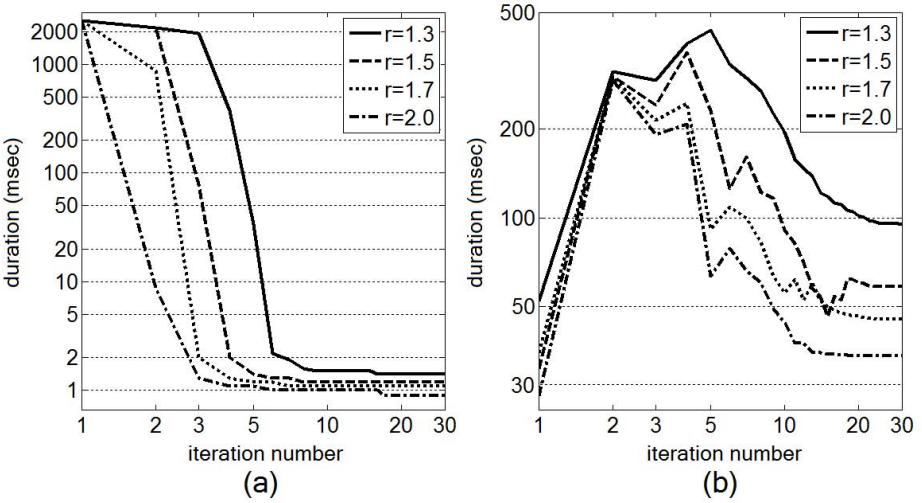


Fig. 3. The duration of the first 50 iterations, using the proposed method at various inflation rates, to classify 913 proteins from SCOP95: (a) matrix splitting approach; (b) sparse matrix implementation

the first one. This way we are able to approximate the speed-up ratio reached via fragmenting the similarity matrix. On the other hand, the sparse matrix implementation provides more efficiently computed initial loops, but the late iterations will require more computations than the matrix splitting approach. It is also visible that the duration of loops initially rises in the case of sparse matrix representation, which happens due to the growing amount of non-zero elements is the similarity matrix. After having performed 10-15 slower iterations, the duration of later iterations stabilizes at a low level.

The above remarked trends are also visible in Fig. 4, which presents efficiency results of the proposed methods on various data sets, using a fixed inflation rate $r = 1.5$. Data sets involved in the tests reported here were chosen as all protein families with cardinality between 10-18 (1795 proteins), 10-20 (2106 proteins), 8-30 (3887 proteins), 5-50 (6522 proteins), 3-99 (8920 proteins), and whole SCOP95 database (11944 proteins). All efficiency tests were run on PC with quad core Intel i7 processor running at 3.4GHz frequency.

Let us remark some trends identified from Figs. 3-4:

1. In every case, we needed a few iterations to break the similarity graph into several small isolated subgraphs. The larger the input data set, the more iterations are necessary. Using an inflation rate fixed at a reasonable value ($r = 1.5$) with the matrix splitting approach, a set of 1000 proteins requires 3 slow loops at the beginning, while at 5000 proteins, the fourth iteration is slow as well. One can expect that 10^5 proteins will need no more than 6-7 slow iterations. The trend of longer initial iterations is similar at the sparse matrix version, but it last a longer number of iterations.

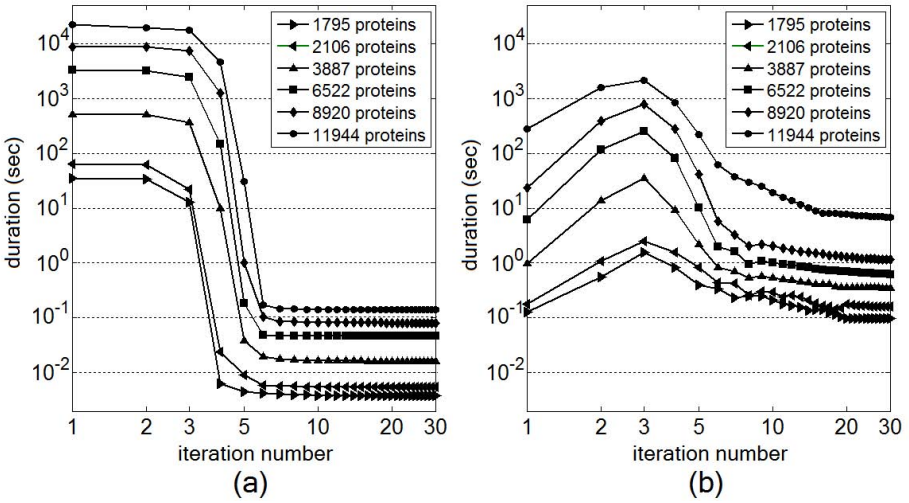


Fig. 4. The duration of the first 50 iterations, using the proposed method with various input data sets, plotted on logarithmic scale, using inflation rate $r = 1.5$: (a) matrix splitting approach; (b) sparse matrix implementation

2. Choosing a larger inflation rate reduces the number of slow iterations. However, it is not recommended to use very high inflation rates, because they yield small clusters in the output, which will hardly reveal any biologically relevant protein similarities.
3. Even though larger number of initial, longer lasting loops are performed by the sparse matrix version, this approach has the better overall runtime, because these initial loops have lower computational burden than the first loops of the matrix splitting approach. This is visible in Fig. 4, where the scales on the vertical axis of the two graphs (a) and (b) are identical.
4. Theoretically both approaches perform the same computations. If the input data set is the same, and the algorithm parameters are set equally, both approaches will lead to the same partition. Further on, we may also assert that after any number of iterations, the current partition of the two approaches are theoretically equivalent.
5. Based on the above assumption, we may combine the two approaches to provide a third, even more efficient one, which performs the initial iterations using the sparse matrix approach and switches to matrix splitting version thereafter, always using the version which performs the iterations quicker. Switching is performed when the largest connected subgraph is smaller than 5% of the total number of graph nodes. Table 1 refers to this switching method as combined approach.

Table 1 gives us a summary of speed-up ratios reached on input data of various sizes, at different inflation rates. These values were computed against the performance of the conventional TRIBE-MCL algorithm, which computes

Table 1. Speed-up ratios reached by the proposed efficient execution scheme

Number of proteins	Inflation rate	Speed-up ratio		
		Matrix splitting	Sparse matrix	Combined
908	1.3	17.87	18.65	80.4
908	1.5	26.16	30.96	150.9
908	1.7	36.67	40.36	184.1
908	2.0	49.07	52.24	339.4
1795	1.5	21.29	189.0	705.5
2106	1.5	21.48	212.7	773.1
3877	1.5	18.39	320.4	423.4
6522	1.5	18.03	327.4	356.7
8920	1.5	16.84	278.7	297.9
11944	1.5	17.30	197.7	224.9

Table 2. Amount of proteins in mixed clusters, out of 11944

Inflation rate	Proteins in mixed clusters at the level of				Total
	classes	folds	superfamilies	families	
1.30	446	245	123	1237	2051
1.35	110	89	118	771	1088
1.40	29	50	51	507	637
1.45	0	35	39	448	522
1.50	0	8	13	356	377
1.55	0	0	10	239	249
1.65	0	0	10	184	194
1.75	0	0	0	97	97
1.85	0	0	0	31	31
2.00	0	0	0	19	19
2.10	0	0	0	3	3
2.35	0	0	0	0	0

the whole similarity matrix in every iteration, encoded in a two-dimensional array. Even higher speed-up ratios could be reached using parallel computing.

The proposed efficient implementations enabled us to perform several tests on the whole SCOP95 database, to evaluate the amount of obtained mixed clusters depending on the algorithm's parameters. Mixed clusters are clusters where proteins from different families are present. We can further distinguish mixtures at the level of classes, folds, superfamilies, and families. For example, a cluster mixed at the level of folds contains proteins from different folds but all its proteins are from the same class. Table 2 presents the amount of proteins situated in mixed clusters for various values of the inflation rate. All these tests were run for threshold value $\varepsilon = 10^{-3}$.

As it was expected, the number of proteins in mixed clusters decreases as the inflation rate grows. Mixtures at the level of classes, folds, superfamilies and families vanish at $r = 1.44$, $r = 1.52$, $r = 1.73$, and $r = 2.35$, respectively.

5 Conclusions

In this paper we have proposed two efficient implementation schemes and a combined third efficient procedure for the graph-based TRIBE MCL clustering method, a useful tool in protein sequence classification. With these novel formulations, late iterations of the algorithm are performed up to thousands times quicker, and the overall runtime becomes shorter by 2-3 orders of magnitude, than in the conventional case. This speed-up is achieved without any damage of the classification accuracy.

References

1. Altschul, S.F., Madden, T.L., Schaffen, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J.: Gapped BLAST and PSI-BLAST: a new generation of protein database search program. Nucleic Acids Res. 25, 3389–3402 (1997)
2. Andreeva, A., Howorth, D., Chadonia, J.M., Brenner, S.E., Hubbard, T.J.P., Chothia, C., Murzin, A.G.: Data growth and its impact on the SCOP database: new developments. Nucleic Acids Res. 36, D419–D425 (2008)
3. Dayhoff, M.O.: The origin and evolution of protein superfamilies. Fed. Proc. 35, 2132–2138 (1976)
4. Eddy, S.R.: Profile hidden Markov models. Bioinformatics 14, 755–763 (1998)
5. Enright, A.J., van Dongen, S., Ouzounis, C.A.: An efficient algorithm for large-scale detection of protein families. Nucleic Acids Res. 30, 1575–1584 (2002)
6. Everitt, B.S., Landau, S., Leese, M., Stahl, D.: Cluster Analysis, 5th edn. John Wiley & Sons, Chichester (2011)
7. Gáspári, Z., Vlahovicek, K., Pongor, S.: Efficient recognition of folds in protein 3D structures by the improved PRIDE algorithm. Bioinformatics 21, 3322–3323 (2005)
8. Heger, A., Holm, L.: Towards a covering set of protein family profiles. Prog. Biophys. Mol. Bio. 73, 321–337 (2000)
9. Hegyi, H., Gerstein, M.: The relationship between protein structure and function: a comprehensive survey with application to the yeast genome. J. Mol. Biol. 288, 147–164 (1999)
10. Needleman, S.B., Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol. 48, 443–453 (1970)
11. Protein Classification Benchmark Collection, `http://net.icgeb.org/benchmark`
12. Structural Classification of Proteins database, `http://scop.mrc-lmb.cam.ac.uk/scop`
13. Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. J. Mol. Biol. 147, 195–197 (1981)
14. Szilágyi, L., Medvés, L., Szilágyi, S.M.: A modified Markov clustering approach to unsupervised classification of protein sequences. Neurocomputing 73, 2332–2345 (2010)

The Property of χ^2_{01}-Concordance for Bayesian Confirmation Measures

Robert Susmaga and Izabela Szczęch

Institute of Computing Science, Poznań Univesity of Technology,
Piotrowo 2, 60-965 Poznań, Poland

Abstract. The paper considers evaluation of rules with particular interestingness measures being Bayesian confirmation measures. It analyses the measures with regard to their agreement with a statistically significant dependency between the evidence and the hypothesis. As it turns out, many popular confirmation measures were not defined to possess such a form of agreement. As a result, even in situations when there is only a weak dependency in data, measures could indicate strong confirmation (or disconfirmation), encouraging the user to take some unjustified actions. The paper employs a χ^2-based coefficient allowing to assess the level of dependency between the evidence and hypothesis in experimental data. A method of quantifying the level of agreement (concordance) between this coefficient and the measure being analysed is introduced. Experimental results for 12 popular confirmation measures are additionally visualised with scatter-plots and histograms.

Keywords: Interestingness measures, Bayesian confirmation, statistical dependency.

1 Introduction

Regardless of the application domain, a crucial step in discovering knowledge from data is the evaluation of induced patterns [2,10,17,23]. Evaluation of patterns in form of *if-then* rules is often done using quantitative measures of interest (e.g. rule support, confidence, gain, lift) [10,23]. Among such interestingness measures, an important role is played by a group called *Bayesian confirmation measures*. Generally, they express the degree to which a rule's premise (also referred to as the conditional part or evidence) confirms its conclusion (also referred to as the decision part or hypothesis) [5,9]. To narrow down the field of available confirmation measures, various properties of such measures are introduced and analysed. Popular properties of confirmation measures include monotonicity property, Ex_1 property and its generalization to weak Ex_1, logicality L property and its generalization to weak L, and a group of symmetry properties (for a survey refer to [5,7,12,13]).

Let us stress that the property analysis becomes much more complex when we assume that it is conducted upon data that may be error-prone. But in practice, the existence of possible data errors is a real phenomenon and must be taken into

V. Torra et al. (Eds.): MDAI 2013, LNAI 8234, pp. 226–236, 2013.

account, so that insignificant, accidental conclusions could be eliminated [14]. Unfortunately, at times the popular confirmation measures may indicate strong confirmation or strong disconfirmation, while there is only a weak dependency in data [22]. Such indications are potentially dangerous, since they may lead to unjustified, and thus inappropriate, user actions. To examine this aspect of the confirmation measures, the paper assesses the significance of the dependency between the evidence and the hypothesis in experimental data, and introduces a method of quantifying the level of agreement (referred to as concordance) between this assessment and the measure being analysed.

The rest of the paper is organized as follows. Section 2 describes the concept of Bayesian confirmation and defines popular measures. An overview of common measure properties is presented in Section 3. Section 4 discusses hazards of using the confirmation measures under observational errors, including methodology aimed at assessing the (χ^2-based) level of dependency between the evidence and the hypothesis in data. Moreover, it introduces concordance between the χ^2-based coefficient and confirmation measures. Last but not least, it provides experimental evaluations of the selected confirmation measures. Final remarks and conclusions are contained in Section 5.

2 Bayesian Confirmation Measures

In this paper, we consider evaluation of patterns represented in the form of rules. The starting point for such rule induction process (rule mining) is a sample of a larger reality, often represented in the form of a data table. Formally, a data table (dataset) is a pair $S = (U, A)$, where U is a non-empty finite set of objects, called the *universe*, and A is a non-empty finite set of *attributes* providing descriptions to the objects.

A rule induced from the dataset consists of a *premise* "*if* E" (referring to an existing piece of evidence, E) and a *conclusion* "*then* H" (referring to a hypothesised piece of evidence, H). Below, we shall use the common, shortened denotation $E \to H$ (read as "*if* E, *then* H").

To evaluate the patterns induced from datasets with respect to their relevance and utility, quantitative interestingness measures have been proposed and analysed [10]. This paper concentrates on a group of interestingness measures called Bayesian confirmation measures. They quantify the degree to which the evidence in the rule's premise E provides support *for* or *against* the hypothesised piece of evidence in the rule's conclusion H [9].

In the context of a particular dataset, the relation between E and H may be quantified by four non-negative frequencies a, b, c and d, briefly represented in a 2×2 contingency table (Table 1). As an illustration, let us recall a popular folk statement that "*all ravens are black*", formalized as a rule "*if x is a raven, then x is black*", often used by Hempel [15]. Regarding that rule, the frequencies may be interpreted as follows: a is the number of black ravens, b is the number of black non-ravens, c is the number of non-black ravens, and d is the number of non-black non-ravens. Observe that a, b, c and d can thus be used to estimate

probabilities: e.g. the probability of the premise is expressed as $P(E) = (a+c)/n$, the conditional probability of the conclusion given the premise is $P(H|E) = P(H \cap E)/P(E) = a/(a+c)$, and so on.

Table 1. An exemplary contingency table of the rule's premise and conclusion

	H	$\neg H$	Σ
E	a	c	$a+c$
$\neg E$	b	d	$b+d$
Σ	$a+b$	$c+d$	n

The group of confirmation measures that we shall present and analyse consists of interestingness measures that satisfy the property of Bayesian confirmation. Formally, for a rule $E \rightarrow H$, an interestingness measure $c(H, E)$ has the property of Bayesian confirmation when it satisfies the following conditions:

$$c(H,E) \begin{cases} > 0 \text{ when } P(H|E) > P(H) & \text{(confirmation)}, \\ = 0 \text{ when } P(H|E) = P(H) & \text{(neutrality)}, \\ < 0 \text{ when } P(H|E) < P(H) & \text{(disconfirmation)}. \end{cases} \tag{1}$$

Thus, the confirmation is interpreted as an increase in the probability of the conclusion H provided by the premise E (similarly for the neutrality and the disconfirmation).

Let us stress that the list of alternative, non-equivalent measures of Bayesian confirmation is quite large [5,8]. The commonly used confirmation measures are presented in Table 2 (for brevity, some definitions are only formulated for two of the main defined situations: confirmation and disconfirmation; in the case of neutrality their values default to zero).

3 Properties of Bayesian Confirmation Measures

To discriminate between interestingness measures and help to choose a suitable one for a particular application, many properties have been proposed and compared in the literature [7,10,17,11]. Properties group the measures according to similarities in their behaviour. Among commonly used properties of confirmation measures there are such properties as:

- *Property* M, ensuring monotonic dependency of a measure on the number of objects satisfying (supporting) or not the premise and/or the conclusion of the rule [12,23], so that the measure is non-decreasing with respect to a and d, and non-increasing with respect to b and c. Thus, e.g. arrival of new objects supporting the rule (or counterexamples, respectively) to the dataset cannot lower (increase) the value of the measure.
- *Property* Ex_1, and its generalization *weak* Ex_1, assuring that any conclusively confirmatory rule is assigned a higher value than any rule which is not

Table 2. Popular confirmation measures

$D(H,E) = P(H\|E) - P(H) = \frac{a}{a+c} - \frac{a+b}{n}$	[6]
$M(H,E) = P(E\|H) - P(E) = \frac{a}{a+b} - \frac{a+c}{n}$	[18]
$S(H,E) = P(H\|E) - P(H\|\neg E) = \frac{a}{a+c} - \frac{b}{b+d}$	[4]
$N(H,E) = P(E\|H) - P(E\|\neg H) = \frac{a}{a+b} - \frac{c}{c+d}$	[19]
$C(H,E) = P(E \wedge H) - P(E)P(H) = \frac{a}{n} - \frac{(a+c)(a+b)}{n^2}$	[3]
$F(H,E) = \frac{P(E\|H) - P(E\|\neg H)}{P(E\|H) + P(E\|\neg H)} = \frac{ad-bc}{ad+bc+2ac}$	[16]
$Z(H,E) = \begin{cases} 1 - \frac{P(\neg H\|E)}{P(\neg H)} = \frac{ad-bc}{(a+c)(c+d)} & \text{in case of confirmation} \\ \frac{P(H\|E)}{P(H)} - 1 = \frac{ad-bc}{(a+c)(a+b)} & \text{in case of disconfirmation} \end{cases}$	[5]
$A(H,E) = \begin{cases} \frac{P(E\|H) - P(E)}{1 - P(E)} = \frac{ad-bc}{(a+b)(b+d)} & \text{in case of confirmation} \\ \frac{P(H) - P(H\|\neg E)}{1 - P(H)} = \frac{ad-bc}{(b+d)(c+d)} & \text{in case of disconfirmation} \end{cases}$	[13]

conclusively confirmatory, and any conclusively disconfirmatory rule is assigned a lower value than any rule which is not conclusively disconfirmatory [5,13].

– *Logicality* L, and its generalization *weak* L, indicating conditions under which measures should obtain their maximal/minimal values [5,9,13]. Another property closely related to L, Ex_1 and their generalizations is *maximality/minimality* proposed in [11].

Searching for measures that possess property Ex_1, Crupi et al. [5] have proposed measure $Z(H,E)$. Later, as its likelihoodist counterpart, measure $A(H,E)$ has been proposed in [13] (for definitions see Table 2). It has been proved in [13] that neither measure $Z(H,E)$ nor $A(H,E)$ satisfies weak Ex_1, however new measures enjoying weak Ex_1 can be derived from $Z(H,E)$ and $A(H,E)$. They are denoted as $c_1(H,E)$, $c_2(H,E)$, $c_3(H,E)$, and $c_4(H,E)$ (for definitions see Table 3; brevity comments similar to that of Table 2 apply here). Measures $c_1(H,E)$ and $c_2(H,E)$ are defined using parameters α and β, where $\alpha+\beta = 1$ and $\alpha > 0$, $\beta > 0$. Observe that parameters α and β can be used to closen the new measure to $Z(H,E)$ or $A(H,E)$, i.e. to Bayesian or likelihoodist inspirations.

Table 3. Derived confirmation measures

$c_1(H,E) =$	$\begin{cases} \alpha + \beta A(H,E) & \text{in case of confirmation when } c = 0 \\ \alpha Z(H,E) & \text{in case of confirmation when } c > 0 \\ \alpha Z(H,E) & \text{in case of disconfirmation when } a > 0 \\ -\alpha + \beta A(H,E) & \text{in case of disconfirmation when } a = 0 \end{cases}$
$c_2(H,E) =$	$\begin{cases} \alpha + \beta Z(H,E) & \text{in case of confirmation when } b = 0 \\ \alpha A(H,E) & \text{in case of confirmation when } b > 0 \\ \alpha A(H,E) & \text{in case of disconfirmation when } d > 0 \\ -\alpha + \beta Z(H,E) & \text{in case of disconfirmation when } d = 0 \end{cases}$
$c_3(H,E) =$	$\begin{cases} A(H,E)Z(H,E) & \text{in case of confirmation} \\ -A(H,E)Z(H,E) & \text{in case of disconfirmation} \end{cases}$
$c_4(H,E) =$	$\begin{cases} min(A(H,E), Z(H,E)) & \text{in case of confirmation} \\ max(A(H,E), Z(H,E)) & \text{in case of disconfirmation} \end{cases}$

4 Using Bayesian Confirmation Measures in Error-Prone Situations

4.1 The Property of Concordance

In real-life situations the existence of possible errors must be taken into account. Thus, we should look for a statistically significant dependency between the evidence and the hypothesis, which may be quantified and measured with different tools. A good and popular one is the two-dimensional χ^2 test, often used to test for the independence of two discrete-valued variables. The popular alternatives to this test include the Cramer's V coefficient, the Yule's Q coefficient or the Fisher coefficient [20].

For 2×2-sized contingency tables, of the form $\left[\begin{smallmatrix} a & c \\ b & d \end{smallmatrix}\right]$, as used in defining confirmation measures, a coefficient $\chi^2_0 = \frac{n(ad-bc)^2}{(a+b)(c+d)(a+c)(b+d)}$ is defined. This coefficient is approximately χ^2-distributed and ranges from 0 to n. To make it n-independent, it is scaled down (divided) by n, producing a value belonging to the interval $[0, 1]$. This version of the coefficient will be further referred to as the "scaled-down χ^2_0" and denoted as χ^2_{01}.

In practice, two potentially unfavourable situations can concern the confirmation measure applied to a contingency table created from error-prone data:

- the value of $c(H,E)$ indicates either weak confirmation or weak disconfirmation, while there is a strong dependency between the evidence and the hypothesis,
- the value of $c(H,E)$ indicates either strong confirmation or strong disconfirmation, while there is only a weak dependency between the evidence and the hypothesis.

To counteract those, there arises a need to evaluate the concordance between confirmation measures and statistical significance of the evidence-hypothesis dependency. For such an evaluation to be useful, it should provide continuous measurements, the higher the more the measure $c(H, E)$ 'agrees' with the level of dependency between the evidence and the hypothesis. This evaluation may be performed using different statistical tools, and in this study we use linear Pearson correlation between $|c(H, E)|$ (the absolute value of $c(H, E)$) and χ^2_{01}, denoted as $r(|c(H, E)|, \chi^2_{01})$. Taking $|c(H, E)|$ into account (thus ignoring the sign of $c(H, E)$) is essential, as it is the absolute value of the confirmation measure, and not its sign, that determines the 'strength' of $c(H, E)$ (i.e. the degree to which the premise of a rule evaluated by the measure confirms or disconfirms its conclusion). Potential alternatives to the linear Pearson correlation include the Spearman rank correlation coefficient [21] or mutual information measures [1].

What is specific about the property of concordance is that it is a representative of continuous-type properties: it can be quantified as the agreement with the level of dependency between E and H.

The relation between χ^2_{01} coefficient and a given confirmation measure $c(H, E)$ may be additionally visualized, which is easily done with a scatter-plot of $c(H, E)$ against χ^2_{01}. Each such scatter-plot will fit a 2×1-sized rectangular envelope, with its axes ranging from -1 to $+1$ (horizontal, $c(H, E)$) and from 0 to 1 (vertical, χ^2_{01}), as illustrated in Figure 1, with lighter and darker regions and graded transitions between them. Given a measure $c(H, E)$, the points of the $c(H, E)$-versus-χ^2_{01} scatter-plot should possibly occupy the darker regions of the figure, while possibly avoiding any of the lighter ones.

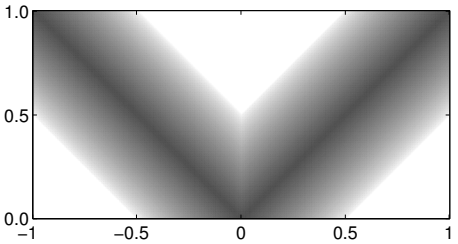


Fig. 1. The desirable (darker) and undesirable (lighter) regions of the $c(H, E)$-versus-χ^2_{01} scatter-plot of $c(H, E)$

4.2 The Experimental Set-Up

Given $n > 0$ (the total number of observations), the dataset is generated as the set of all possible $\left[\begin{smallmatrix} a & c \\ b & d \end{smallmatrix}\right]$ contingency tables satisfying $a + b + c + d = n$. The set is thus exhaustive and non-redundant (i.e. it contains exactly one copy of each contingency table satisfying the above condition).

The exact number t of tables in the set is $t = (n+1)(n+2)(n+3)/6$. This value grows quickly, although polynomially, not exponentially; e.g. the number of all tables for $n = 128$ equals $t = 366145$. Unfortunately, the number t can become considerable: for n about 1000 (a typical number of objects in a benchmark classification data set) t exceeds hundreds of millions.

After having set the total number of observations n to 128, the following operations were performed:

- the exhaustive and non-redundant set of $\left[\begin{smallmatrix} a & c \\ b & d \end{smallmatrix}\right]$ contingency tables satisfying $a + b + c + d = n$ was generated,
- the values of the 12 selected confirmation measures (with $c_1(H, E)$ and $c_2(H, E)$ defined for $\alpha = \beta = 0.5$) for all the generated tables were calculated,
- the values of the χ^2_{01} coefficient for all the generated tables were computed,
- the correlations between the absolute values of each of 12 selected confirmation measures and the χ^2_{01} coefficient (i.e. concordances) were established.

Similar steps (but with n decreased to 32 to facilitate the rendering process) led to the charts, i.e. scatter-plots of $c(H, E)$ against χ^2_{01} (Figure 2) and so called triple-region histograms of $c(H, E)$ (Figure 3). The triple-region histograms show the distribution of the measure, with each bar additionally displaying the number of points situated above (upper white region), on (dark region) or below (lower white region) the $|c(H, E)| = \chi^2_{01}$ line. Characteristically, the size of the lower region always exceeds considerably the size of the upper region, while the dark region is only a thin, horizontal strip (with the notable exception of $c_3(H, E)$, for which only the dark region exists).

Table 4. The coefficients of the χ^2_{01}-concordance of the 12 selected confirmation measures

$c(H,E)$	$r(\lvert c(H,E)\rvert, \chi^2_{01})$	$c(H,E)$	$r(\lvert c(H,E)\rvert, \chi^2_{01})$
$D(H,E)$	0.713	$Z(H,E)$	0.694
$M(H,E)$	0.713	$A(H,E)$	0.694
$S(H,E)$	0.912	$c1(H,E)$	0.697
$N(H,E)$	0.912	$c2(H,E)$	0.697
$C(H,E)$	0.908	$c3(H,E)$	1.000
$F(H,E)$	0.711	$c4(H,E)$	0.957

4.3 The Experimental Results

The conducted experiments revealed interesting results of both generic and specific nature [22]. The following remarks concern the χ^2_{01}-concordance (as quantified by the Pearson correlation coefficient r) of the measures (see Table 4 and Figures 2 and 3):

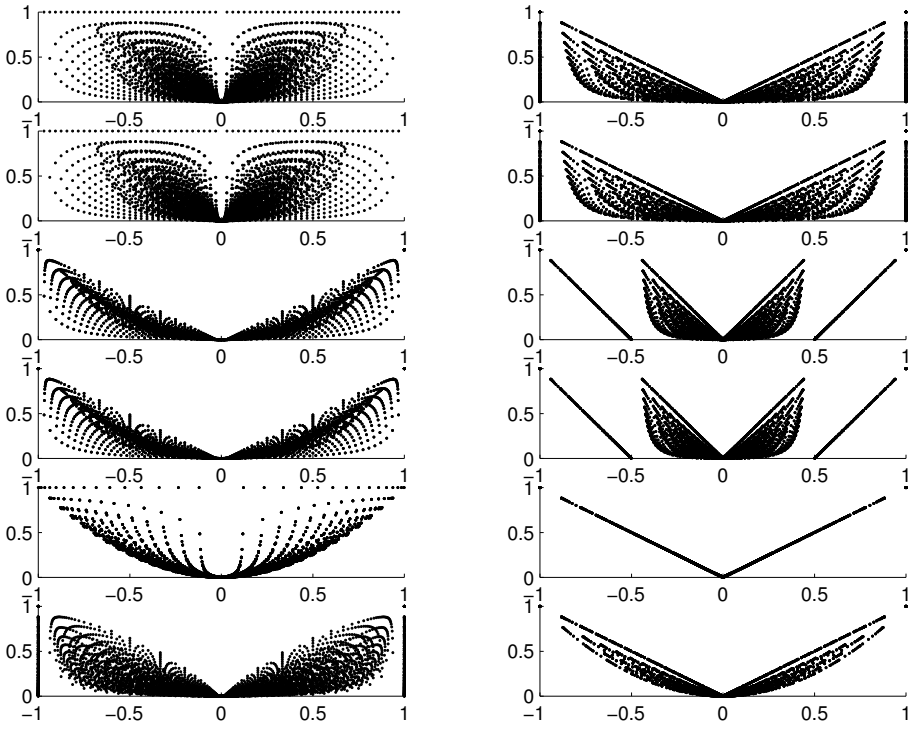


Fig. 2. Scatter-plots of the 12 selected confirmation measures against χ^2_{01} (left-hand column: measures $D(H,E)$, $M(H,E)$, $S(H,E)$, $N(H,E)$, $C(H,E)$, $F(H,E)$; right-hand column: measures $Z(H,E)$, $A(H,E)$, $c_1(H,E)$, $c_2(H,E)$, $c_3(H,E)$, $c_4(H,E)$; $c_1(H,E)$ and $c_2(H,E)$ defined with $\alpha = \beta = 0.5$)

- measure $c_3(H,E)$ enjoys an ideal χ^2_{01}-concordance, which is due to the fact that $|c_3(H,E)| = \chi^2_{01}$,
- the concordance of the other measures ranges from 0.957 ($c_4(H,E)$) down to 0.694 ($Z(H,E)$ and $A(H,E)$), in result of which all of them can be referred to as approximately concordant,
- the absolute values of the approximately concordant measures tend to exceed those of χ^2_{01}.

A conclusion is that not all of the measures possess ideal concordance. The less concordant measures should thus be used with some care, especially when applied to real-life, error-prone data, as the may express either strong confirmation or strong disconfirmation in statistically insignificant situations.

It is especially interesting that measures $c_1(H,E)$ and $c_2(H,E)$, which depend on the value of the α parameter, i.e. the free parameter that is used to define these measures (the β parameter is, on the other hand, constrained, as $\beta = 1-\alpha$), evince varying shapes of their corresponding scatter-plots, see Figure 4. This will necessarily influence their correlations with the χ^2_{01} coefficient.

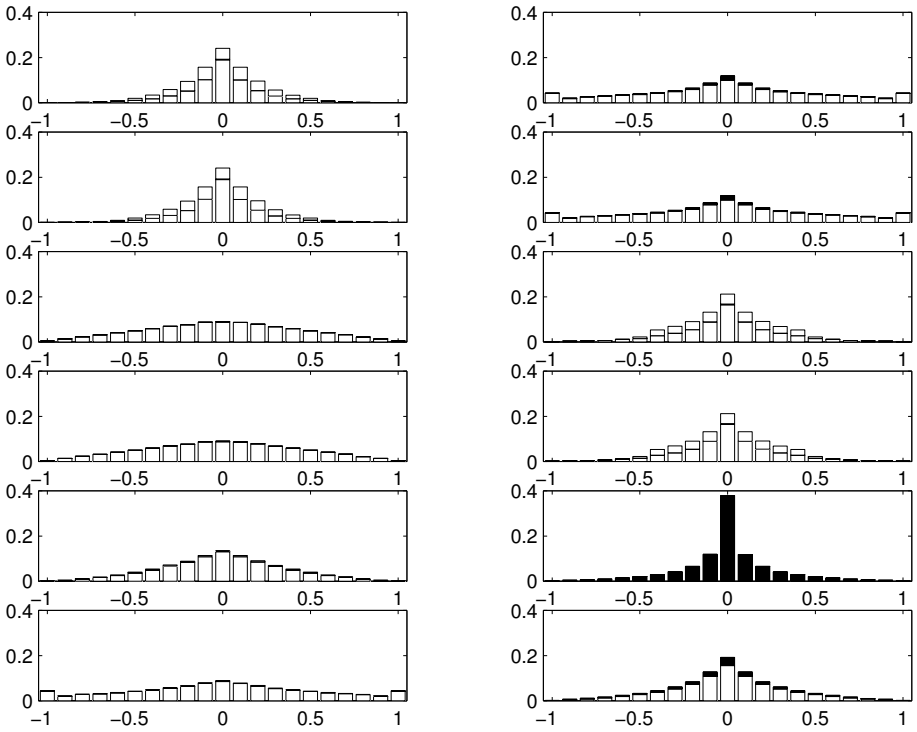


Fig. 3. Triple-region histograms of the 12 selected confirmation measures $c(H, E)$ in relation to χ^2_{01} (left-hand column: measures $D(H, E)$, $M(H, E)$, $S(H, E)$, $N(H, E)$, $C(H, E)$, $F(H, E)$; right-hand column: measures $Z(H, E)$, $A(H, E)$, $c_1(H, E)$, $c_2(H, E)$, $c_3(H, E)$, $c_4(H, E)$; $c_1(H, E)$ and $c_2(H, E)$ defined with $\alpha = \beta = 0.5$)

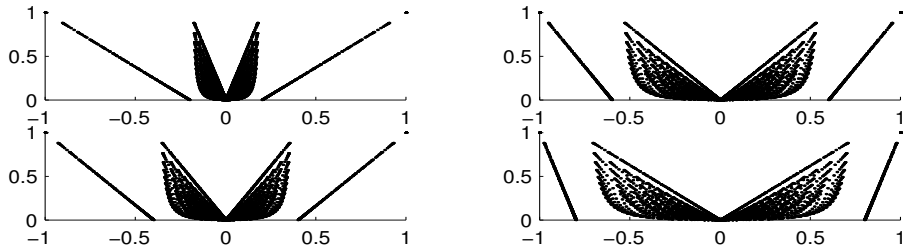


Fig. 4. Scatter-plots of measures $c_1(H, E)$ and $c_2(H, E)$ against χ^2_{01}, defined for various values of α (left-hand column: $\alpha = 0.2$, $\alpha = 0.4$; right-hand column: $\alpha = 0.6$, $\alpha = 0.8$), see Figure 2 for $\alpha = 0.5$

Because, by definition, most values of these measures belong to the interval $(-\alpha, +\alpha)$, see Figure 4 (more details can be found in [22]), their concordances are then also changed accordingly, see Figure 5. This means that the α parameter can be directly used to control this aspect of these two measures. In particular,

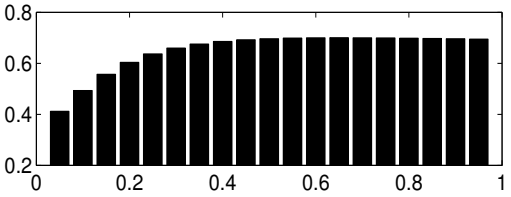


Fig. 5. The concordances of $c_1(H, E)$ and $c_2(H, E)$, as influenced by the changing α

when $\alpha \to 1.0$, measures $c_1(H, E)$ and $c_2(H, E)$ approach measures $Z(H, E)$ and $A(H, E)$, respectively, in which case they also acquire their corresponding concordances (which is, in both cases, 0.694).

For more detailed analyses of these (and other) properties of the confirmation measures see [22].

5 Conclusions

The paper considers Bayesian confirmation measures, which have become the subject of numerous, intensive studies. What is characteristic of these studies is that virtually all of them were confined to environments that had been explicitly or implicitly assumed to be free from observational errors. In real-life situations, however, the existence of such errors must be taken into account and properly approached. This goal is in this paper accomplished with the χ^2 test, commonly used to examine for the dependence between two discrete-valued variables.

The actual amount of how concordant a confirmation measure is with the level of dependency between the evidence and the hypothesis is quantified with the Pearson correlation coefficient between the measure and an introduced χ^2_{01} coefficient. The relations between the measures and χ^2_{01} are additionally illustrated by scatter-plots and specialized, triple-region histograms.

The general conclusion is that most measures possess rather high, although not ideal, concordance. The scatter-plots and the triple-region histograms of these measures reveal particular situations in which they express either strong confirmation or strong disconfirmation in statistically insignificant situations. This means that they should be used with special care in error-prone environments. Interestingly enough, the concordance of the parametrized confirmation measures, $c_1(H, E)$ and $c_2(H, E)$, is influenced by the parameters used in their definitions, so it may be controlled to some extent. Measure $c_3(H, E)$, a notable exception amongst the 12 selected confirmation measures, enjoys full concordance, so its indications may assumed to be safest in this particular respect.

Acknowledgment. The work has been supported by the Polish Ministry of Science and Higher Education and a local statutory grant (DS).

References

1. Bell, C.: Mutual information and maximal correlation as measure dependence. The Annals of Mathematical Statistics 33, 587–595 (1962)
2. Brzeziński, D., Stefanowski, J.: Accuracy updated ensemble for data streams with concept drift. In: Proceedings of the 6th International Conference on Hybrid Artificial Intelligent Systems, pp. 155–163 (2011)
3. Carnap, R.: Logical Foundations of Probability, 2nd edn. University of Chicago Press (1962)
4. Christensen, D.: Measuring confirmation. Journal of Philosophy 96, 437–461 (1999)
5. Crupi, V., Tentori, K., Gonzalez, M.: On bayesian measures of evidential support: Theoretical and empirical issues. Philosophy of Science 74, 229–252 (2007)
6. Eells, E.: Rational Decision and Causality. Cambridge University Press, Cambridge (1982)
7. Eells, E., Fitelson, B.: Symmetries and asymmetries in evidential support. Philosophical Studies 107(2), 129–142 (2002)
8. Fitelson, B.: The plurality of bayesian measures of confirmation and the problem of measure sensitivity. Philosophy of Science 66, 362–378 (1999)
9. Fitelson, B.: Studies in Bayesian Confirmation Theory. Ph.D. thesis, University of Wisconsin, Madison (2001)
10. Geng, L., Hamilton, H.: Interestingness measures for data mining: A survey. ACM Computing Surveys 38(3) (2006)
11. Glass, D.H.: Confirmation measures of association rule interestingness. Knowlegde Based Systems 44, 65–77 (2013)
12. Greco, S., Pawlak, Z., Słowiński, R.: Can bayesian confirmation measures be useful for rough set decision rules? Engineering Applications of Artifficial Intelligence 17, 345–361 (2004)
13. Greco, S., Słowiński, R., Szczęch, I.: Properties of rule interestingness measures and alternative approaches to normalization of measures. Information Sciences 216, 1–16 (2012)
14. Hastie, T., Tibshirani, R., Friedman, J.: Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer (2003)
15. Hempel, C.: Studies in the logic of confirmation (i). Mind 54, 1–26 (1945)
16. Kemeny, J., Oppenheim, P.: Degrees of factual support. Philosophy of Science 19, 307–324 (1952)
17. Lenca, P., Meyer, P., Vaillant, B., Lallich, S.: On selecting interestingness measures for association rules: User oriented description and multiple criteria decision aid. European Journal of Operational Research 184(2), 610–626 (2008)
18. Mortimer, H.: The Logic of Induction. Prentice-Hall, Paramus (1988)
19. Nozick, R.: Philosophical Explanations. Clarendon Press, Oxford (1981)
20. Rayner, J., Best, D.: A Contingency Table Approach to Nonparametric Testing. Taylor & Francis Group (2001)
21. Spearman, C.E.: The proof and measurement of association between two things. American Journal of Psychology 15, 72–101 (1904)
22. Susmaga, R., Szczęch, I.: Statistical significance of bayesian confirmation measures. Tech. rep., RA-010/12, Poznań University of Technology (2012)
23. Szczęch, I.: Multicriteria attractiveness evaluation of decision and association rules. In: Peters, J.F., Skowron, A., Wolski, M., Chakraborty, M.K., Wu, W.Z. (eds.) Transactions on Rough Sets X. LNCS, vol. 5656, pp. 197–274. Springer, Heidelberg (2009)

Permutability of Fuzzy Consequence Operators Induced by Fuzzy Relations

Neus Carmona[1], Jorge Elorza[1], Jordi Recasens[2], and Jean Bragard[1]

[1] Departamento de Física y Matemática Aplicada,
Facultad de Ciencias, Universidad de Navarra, Pamplona, Spain
ncarmona@alumni.unav.es, {jelorza,jbragard}@unav.es
[2] Secció Matemàtiques i Informàtica, ETS Arquitectura del Vallès,
Universitat Politècnica de Catalunya, Sant Cugat del Vallès, Spain
j.recasens@upc.edu

Abstract. In this paper we study the permutability of the composition of fuzzy consequence operators when they are induced by fuzzy relations using the usual Zadeh's compositional rule. In particular, we study the case of fuzzy indistinguishability operators and fuzzy preorders. We study the connection between the permutability of the fuzzy relations and the permutability of their induced fuzzy operators.

Keywords: Permutability, Indistinguishability operator, Fuzzy Preorder, Fuzzy Consequence Operator.

1 Introduction

Composition of fuzzy operators often appears in fields like fuzzy mathematical morphology or approximate reasoning. In fuzzy mathematical morphology, fuzzy operators are used as morphological filters for image processing [7,8]. In approximate reasoning, fuzzy consequence operators perform the role of deriving consequences from certain premises and relations [6,9,10,12]. These two fields are closely related and several results can be transfered from one field to the other. We refer the reader to [11] for further details. In a previous paper [5] we studied the composition of fuzzy consequence operators and fuzzy interior operators in a general context. This paper continues the work we started there. We study the case of fuzzy operators induced by fuzzy relations. The aim is to connect the permutability of the generating relations with the permutability of the induced operators.

We will focus on the case of fuzzy operators induced by fuzzy indistinguishability operators and fuzzy preorders through Zadeh's compositional rule. These cases are particularly important since the induced fuzzy operators are fuzzy consequence operators. Our paper is organized as follows:

In Section 2 we set the framework. We recall the main definitions and results that will be used throughout the paper and we state some results and definitions from our previous work that will also be needed.

In Section 3 we first study the permutability of general fuzzy preorders and then we focus on the case of fuzzy indistinguishability operators.

V. Torra et al. (Eds.): MDAI 2013, LNAI 8234, pp. 237–247, 2013.

In Section 4 we relate the permutability of fuzzy relations with the permutability of their respective Zadeh's induced operators. We study the case of fuzzy consequence operators induced by fuzzy preorders and fuzzy indistinguishabilities.

In Section 5 we analyze another approach to permutability of fuzzy preorders.

Finally, in Section 6 we present the conclusions.

2 Preliminars

The structure $\langle L, \wedge, \vee, *, \rightarrow, 0, 1\rangle$ is said to be a complete residuated lattice in the sense of Bělohlávek [2] when:

1. $\langle L, \wedge, \vee, 0, 1\rangle$ is a complete commutative lattice, where 0 denotes the least element and 1 denotes the greatest one
2. $(L, *)$ is a commutative monoid i.e. $*$ is associative, commutative and with neutral element 1,
3. the operations $*$ and $\rightarrow$ satisfy the adjointness property:

$$x * y \leq z \quad \Leftrightarrow \quad x \leq y \rightarrow z$$

where $\leq$ denotes the lattice ordering.

The following holds for every complete commutative residuated lattice:

Proposition 1. *[2] Let $\langle L, \wedge, \vee, *, \rightarrow, 0, 1\rangle$ be a complete commutative residuated lattice. The following conditions hold for each index set I:*

1. $x * \bigvee_{i\in I} y_i = \bigvee_{i\in I}(x * y_i)$
2. $x * \bigwedge_{i\in I} y_i \leq \bigwedge_{i\in I}(x * y_i)$

We will work in the usual $\langle [0,1], \wedge, \vee, *, \rightarrow, 0, 1\rangle$ where $\wedge$ and $\vee$ are the usual infimum and supremum, $*$ is a left continuous t-norm and $\rightarrow$ is the residuum of $*$ defined for $\forall a, b \in X$ as $a \rightarrow b = \sup\{\gamma \in [0,1] \mid a * \gamma \leq b\}$.

In this paper, X will be a non-empty classical universal set and $[0,1]^X$ will be the set of all fuzzy subsets of X with truth values in $[0,1]$. Ω' will denote the set of fuzzy operators defined from $[0,1]^X$ to $[0,1]^X$ and Γ' will denote the set of fuzzy binary relations defined in X.

Definition 1. *[12] A fuzzy operator $C \in \Omega'$ is called a **fuzzy consequence operator** or **fuzzy closure operator** (FCO for short) when it satisfies for all $\mu, \nu \in [0,1]^X$:*

1. *Inclusion $\mu \subseteq C(\mu)$*
2. *Monotony $\mu \subseteq \nu \Rightarrow C(\mu) \subseteq C(\nu)$*
3. *Idempotence $C(C(\mu)) = C(\mu)$*

Ω will denote the set of all FCO.

The inclusion of fuzzy subsets is given by the pointwise order, i.e. $\mu \subseteq \nu$ if and only if $\mu(x) \leq \nu(x)$ for all $x \in X$.

Given two fuzzy operators C_1, C_2 we say that $C_1 \leq C_2$ if $C_1(\mu) \subseteq C_2(\mu)$ for all $\mu \in [0,1]^X$.

Definition 2. *A **fuzzy relation** on a set X is a map $R : X \times X \longrightarrow [0,1]$. A fuzzy relation on X is said to be:*

(R) Reflexive if $R(x,x) = 1 \quad \forall x \in X$
(S) Symmetric if $R(x,y) = R(y,x) \quad \forall x,y \in X$
(T) $*$-Transitive if $R(x,y) * R(y,z) \leq R(x,z) \quad \forall x,y,z \in X$

*A fuzzy relation satisfying (R) and (T) is called a **fuzzy preorder**. If it also satisfies (S), then it is called a fuzzy similarity or **indistinguishability operator**. Given $R,S \in \Gamma'$, we say that $R \leq S$ if and only if $R(x,y) \leq S(x,y)$ for all $x,y \in X$.*

For a given fuzzy relation $R \in \Gamma'$, a fuzzy subset μ of X is called **$*$-compatible** with R if $\mu(x) * R(x,y) \leq \mu(y)$ for all $x,y \in X$.

Definition 3. *[4] Let g be a fuzzy operator and R a fuzzy relation. We will say that g is **$*$-concordant with R** if all the subsets from the image of g are $*$-compatible with R.*

Every fuzzy relation induces a fuzzy operator using Zadeh's compositional product:

Definition 4. *Let $R \in \Gamma'$ be a fuzzy relation on X. The fuzzy operator induced by R through Zadeh's compositional rule is defined by*

$$C_R^*(\mu)(x) = \sup_{w \in X} \{\mu(w) * R(w,x)\} \tag{1}$$

Every fuzzy operator also induces a fuzzy relation:

Definition 5. *Let C be a fuzzy operator in Ω'. The fuzzy relation induced by C is given by*

$$R_C(x,y) = C(\{x\})(y) \tag{2}$$

where $\{x\}$ denotes the singleton x.

It is well-known [10] that for any fuzzy relation R, $R_{C_R^*} = R$. However, in general conditions $C_{R_C}^* \neq C$.

In a previous paper [4] we extended the operator induced through Zadeh's compositional rule to a more general one which is generated through the same rule but using a fuzzy operator and a fuzzy relation. The generator operator performs a selection among the fuzzy subsets before applying the operator induced through Zadeh's rule.

Definition 6. *Let $g \in \Omega'$ be a fuzzy operator and let $R \in \Gamma'$ be a fuzzy relation on X. We define the operator C_R^g induced by g and R as*

$$C_R^g(\mu)(x) = \sup_{w \in X} \{g(\mu)(w) * R(w,x)\} \tag{3}$$

R and g are called the generators of C_R^g.

3 Permutability of Fuzzy Preorders and Fuzzy Indistinguishability Operators

Once we have set the framework, let us recall the concept of permutability between fuzzy relations. Permutability of fuzzy relations is considered with the sup-$*$ product.

Definition 7. *Let $R,S \in \Gamma'$ be fuzzy relations on a set X and $*$ a t-norm. The **sup-$*$ composition** of R and S is the fuzzy relation defined for all $x,y \in X$ by*

$$R \circ S(x,y) = \sup_{w \in X} \{R(x,w) * S(w,y)\} \tag{4}$$

Definition 8. *Let $R,S \in \Gamma'$ be fuzzy relations. We say that R and S are **permutable** or that R and S permute if $R \circ S = S \circ R$ where $\circ$ is the sup-$*$ composition.*

Permutability of preorders is closely related to the transitive closure of a fuzzy relation. The transitive closure of a fuzzy relation R is the smallest upper approximation of R which is $*$-transitive [1]. More precisely,

Definition 9. *Let R be a fuzzy relation. We define the **transitive closure $\overline{R}$ of** R as the fuzzy relation given by*

$$\overline{R} = \inf_{\substack{S \in \widehat{\Gamma} \\ R \leq S}} \{S\} \tag{5}$$

where $\widehat{\Gamma}$ denotes the set of all $*$-transitive fuzzy relations in X.

The explicit formula for the transitive closure is given by $\overline{R} = \sup_{n \in \mathbb{N}} R^n$ where the power of R is defined using the sup-$*$ product. It is easy to see that $\overline{R}$ is transitive [1]. The $*$-transitive closure also preserves reflexivity and symmetry. Hence, the transitive closure of a reflexive fuzzy relation is fuzzy preorder and the transitive closure of a reflexive and symmetric relation is an indistinguishability operator.

It was proved in [14] that two $*$-indistinguishability operators defined on a finite set X permute if and only if $E \circ F$ is an $*$-indistinguishability operator. In this case, $E \circ F = \overline{\max(E,F)}$. We extend this result to general fuzzy preorders and any set X, finite or not.

Lemma 1. *Let R and P be two fuzzy $*$-preorders on a set X. Then, $R \circ P \leq \overline{\max(R,P)}$.*

Proof.

$$R \circ P \leq \max(R,P) \circ \max(R,P) \leq \sup_{n \in \mathbb{N}} (\max(R,P))^n = \overline{\max(R,P)}.$$

□

Theorem 1. *Let R and P be two fuzzy $*$-preorders on X. Then, R and P are permutable if and only if $R \circ P$ and $P \circ R$ are fuzzy $*$-preorders. In this case, $R \circ P$ coincides with the $*$-transitive closure $\overline{\max(R,P)}$ of $\max(R,P)$.*

Proof. Assume first that $R \circ P = P \circ R$ and let us show that they are fuzzy preorders.

reflexivity: $R \circ P(x,x) = \sup_{w \in X}\{R(x,w) * P(w,x)\} \geq R(x,x) * P(x,x) = 1$

$*$-transitivity: Since R is $*$-transitive, $\sup_{w \in X}\{R(x,w) * R(w,y)\} \leq R(x,y)$.

The same holds for P. Thus,

$$\begin{aligned}
R \circ P(x,y) * R \circ P(y,z) &= \sup_{w \in X}\{R(x,w) * P(w,y)\} * \sup_{h \in X}\{R(y,h) * P(h,z)\} \\
&= \sup_{w,h \in X}\{R(x,w) * P(w,y) * R(y,h) * P(h,z)\} \\
&\leq \sup_{w,h \in X}\{R(x,w) * (P \circ R)(w,h) * P(h,z)\} \\
&= \sup_{w,h \in X}\{R(x,w) * (R \circ P)(w,h) * P(h,z)\} \\
&= \sup_{w,h,y \in X}\{R(x,w) * R(w,y) * P(y,h) * P(h,z)\} \\
&= \sup_{y \in X}\{\sup_{w \in X}\{R(x,w) * R(w,y)\} * \sup_{h \in X}\{P(y,h) * P(h,z)\}\} \\
&\leq \sup_{y \in X}\{R(x,y) * P(y,z)\} \;=\; R \circ P(x,z).
\end{aligned}$$

Since $R \circ P \geq R$ and $R \circ P \geq P$ we have that $R \circ P \geq \max(R,P)$. As $R \circ P$ is a fuzzy preorder, it follows that $R \circ P \geq \overline{\max(R,P)}$. From Lemma 1, we get $R \circ P = \overline{\max(R,P)} = P \circ R$.

Conversely, assume that $R \circ P$ and $P \circ R$ are fuzzy $*$-preorders. A similar argument than the used above proves that both of them are greater than or equal to the $*$-transitive closure of their maximum $\overline{\max(R,P)}$. From Lemma 1, $R \circ P = \overline{\max(R,P)} = P \circ R$. Hence, R and P permute. □

Notice that it is not deduced from the previous results that if $R \circ P$ is a fuzzy preorder then R and P permute. We need both compositions to be fuzzy preorders in order to find permutability between them. Let us present an example to illustrate that the composition in one direction is not enough to ensure permutability between fuzzy preorders.

Example 1. Let Q and R be the following min-preorders. Notice that $R \circ Q$ is also a min-preorder but $Q \circ R$ is not. R and P do not permute.

$$Q = \begin{pmatrix} 1 & 0.4 & 0.5 \\ 0.6 & 1 & 0.5 \\ 0.3 & 0.3 & 1 \end{pmatrix} \qquad R \circ Q = \begin{pmatrix} 1 & 0.4 & 0.6 \\ 0.7 & 1 & 0.75 \\ 0.4 & 0.4 & 1 \end{pmatrix}$$

$$R = \begin{pmatrix} 1 & 0.3 & 0.6 \\ 0.7 & 1 & 0.75 \\ 0.4 & 0.3 & 1 \end{pmatrix} \qquad Q \circ R = \begin{pmatrix} 1 & 0.4 & 0.6 \\ 0.7 & 1 & 0.75 \\ 0.4 & 0.3 & 1 \end{pmatrix}$$

Now, let us focus on the case indistinguishability operators.

Corollary 1. *Let E and F be two $*$-indistinguishability operators on X. Then, E and F are permutable if and only if $E \circ F$ is a $*$-indistinguishability operator. In this case, $E \circ F$ coincides with the $*$-transitive closure* $\max(E,F)$ *of* $\max(E,F)$.

Proof. Since E and F are fuzzy preorders, Theorem 1 ensures that they permute if and only if $E \circ F = \max(E,F) = F \circ E$. Since $\max(E,F)$ is reflexive and symmetric, $\max(E,F)$ is an indistinguishability operator. □

4 Permutability of Fuzzy Operators Induced by Fuzzy Relations

We are ready to study permutability between fuzzy operators. We consider the usual composition.

Definition 10. *Let C, C' be fuzzy operators. We say that C and C' are* ***permutable*** *or that C and C' permute if $C \circ C' = C' \circ C$ where $\circ$ denotes the usual composition.*

When the operators are induced by fuzzy relations, composition of fuzzy operators can be described as follows.

Proposition 2. *Let R, S be two fuzzy relations and let C^*_R and C^*_S be the corresponding fuzzy operators induced through Zadeh's rule. Then,*

$$C^*_R \circ C^*_S = C^*_{S \circ R} \tag{6}$$

where $S \circ R$ denotes the sup-$*$ product composition of fuzzy relations.

Proof. For all $\mu \in [0,1]^X$ and all $x \in X$ we have

$$\begin{aligned} C^*_R \circ C^*_S(\mu)(x) \;=\; C^*_R(C^*_S(\mu))(x) \;&=\; \sup_{w \in X} \{\, C^*_S(\mu)(w) \,*\, R(w,x) \,\} \\ &=\; \sup_{w \in X} \{\, \sup_{z \in X} \{\, \mu(z) \,*\, S(z,w) \} \,*\, R(w,x) \,\} \\ &=\; \sup_{w,z \in X} \{\, \mu(z) \,*\, S(z,w) \,*\, R(w,x) \,\} \\ &=\; \sup_{z \in X} \{\, \mu(z) \,*\, \sup_{w \in X} \{\, S(z,w) \,*\, R(w,x) \,\} \,\} \\ &=\; \sup_{z \in X} \{\, \mu(z) \,*\, S \circ R(z,x) \,\} \;=\; C^*_{S \circ R}(\mu)(x). \end{aligned}$$

□

Remark 1. Notice that Definition 6 and the proof of Proposition 2 provide several forms to write composition of induced fuzzy operators:

$$C^*_R \circ C^*_S \;=\; C^*_{S \circ R} \;=\; C_R^{C^*_S} \tag{7}$$

They will be used to characterize permutability.

We are ready to focus on the case when the relation is a fuzzy preorder or indistinguishability. It is well known that fuzzy operators induced from fuzzy relations through Zadeh's compositional product are fuzzy consequence operators if and only if the relation is a fuzzy preorder [9].

Proposition 3. *[9] Let $R \in \Gamma'$ be a fuzzy relation. Then, C_R^* is a FCO if and only if R is a fuzzy preorder.*

Notice here, that not every FCO can be writen in the form C_R^* for a certain fuzzy preorder.

Proposition 4. *[9] Let R, P be fuzzy preorders, then $C_R^* = C_P^*$ if and only if $R = P$.*

In a previous paper [5] we established the following characterization of the permutability of fuzzy consequence operators:

Theorem 2. *[5] Let C, C' be fuzzy consequence operators. Then, C and C' permute if and only if $C \circ C'$ and $C' \circ C$ are fuzzy consequence operators. In this case, $C \circ C' = \overline{\max(C, C')}$.*

Here, $\overline{\max(C,C')}$ denotes the closure of the operator $\max(C,C')$. That is, the smallest FCO which is greater than or equal to $\max(C,C')$. This concept was first defined for general lattices [15] and later introduced by Zadeh in the fuzzy context.

Definition 11. *Let $C : [0,1]^X \longrightarrow [0,1]^X$ be a fuzzy operator. We define the **fuzzy closure $\overline{C}$ of** C as the fuzzy operator given by*

$$\overline{C} = \inf_{\substack{\phi \in \Omega \\ C \leq \phi}} \{\phi\} \tag{8}$$

It is natural to think that permutability of fuzzy preorders is connected to the permutability of their consequences. The relation between permutability of fuzzy preorders and permutability of their induced consequence operators can be summarized in the following theorem.

Theorem 3. *Let R, P be fuzzy preorders. Then,*

$$C_R^* \circ C_P^* = C_P^* \circ C_R^* \quad \Leftrightarrow \quad R \circ P = P \circ R$$

Proof. Assume that $C_R^* \circ C_P^* = C_P^* \circ C_R^*$. From Theorem 2 it follows that both $C_{P \circ R}^*$ and $C_{R \circ P}^*$ are FCO. According to Propositions 3 and 4, $P \circ R$ and $R \circ P$ are fuzzy preorders and $P \circ R = R \circ P$ thus R and P permute.

Conversely, assume $R \circ P = P \circ R$. By Theorem 1 both are fuzzy preorders, thus $C_{P \circ R}^*$ and $C_{R \circ P}^*$ are FCO. According to Theorem 2, C_R^* and C_P^* permute. □

Corollary 2. *If $P \circ R$ and $R \circ P$ are fuzzy preorders, then R and P permute and their consequences also permute.*

Let us focus on the particular case of the fuzzy consequence operators induced by indistinguishability operators. If the relation is an indistinguishability operator, the induced fuzzy operator behaves especially well. We refer the interested reader to [13] for further details.

Proposition 5. *Let E be a fuzzy indistinguishability operator and let C_E^* be the fuzzy operator induced through Zadeh's compositional rule. Then,*

1. C^*_E *is a fuzzy consequence operator.*
2. $C^*_E(\bigcup_{i\in I}\mu_i) = \bigcup_{i\in I} C^*_E(\mu_i)$ *for any index set* I *and all* $\mu_i \in [0,1]^X$.
3. $C^*_E(\{x\})(y) = C^*_E(\{y\})(x)$ *for all* $x, y \in X$ *where* $\{x\}$ *denotes the singleton of* x.
4. $C^*_E(\alpha * \mu) = \alpha * C^*_E(\mu)$ *for any constant* $\alpha \in [0,1]$ *and* $\mu \in [0,1]^X$.

Proposition 6. *There is a bijection between the set of* $*$*-indistinguishability operators and the set of fuzzy operators satisfying the conditions of Proposition 5.*

Corollary 3. *Let* $C \in \Omega'$ *be a fuzzy operator satisfying all the properties of Proposition 5. Then, there exists a fuzzy indistinguishability relation* E *such that* $C^*_E = C$.

Differently from general fuzzy preorders, every operator satisfying conditions of Proposition 5 can be writen in the form C^*_E for a certain indistinguishability E.

Corollary 4. *Let* C^*_E, C^*_F *be fuzzy operators satisfying all the properties form Proposition 5. Then,*

$$C^*_E = C^*_F \Leftrightarrow E = F$$

Even if C^*_E and C^*_F do not permute, their composition always satisfy the following properties.

Proposition 7. *Let* E, F *be indistinguishability operators. Then,* $C^*_{E\circ F}$ *satisfies properties 2, 4 of Proposition 5. Moreover, it satisfies the inclusion and monotony properties from the definition of FCO.*

Proof. Since both C^*_E and C^*_F satify properties 2 and 4, it follows that

$$C^*_E(C^*_F(\bigcup_{i\in I}\mu_i)) = C^*_E(\bigcup_{i\in I} C^*_F(\mu_i)) = \bigcup_{i\in I} C^*_E(C^*_F(\mu_i))$$

for any index set I and all $\mu_i \in [0,1]^X$ and

$$C^*_E(C^*_F(\alpha * \mu)) = C^*_E(\alpha * C^*_F(\mu)) = \alpha * C^*_E(C^*_F(\mu))$$

for any constant $\alpha \in [0,1]$ and $\mu \in [0,1]^X$. □

Permutability of C^*_E and C^*_F can be characterized as follows:

Theorem 4. *Let* E, F *be* $*$*-indistinguishability operators. Then, their consequences* C^*_E *and* C^*_F *permute if and only if* $E \circ F$ *is an indistinguishability operator.*

Proof. It directly follows from Corollary 1 and Theorem 3. □

Corollary 5. *Let* C, C' *be fuzzy operators satifying all the conditions of Proposition 5. Then,* C *and* C' *permute if and only if* $C \circ C'$ *also satisfies all these conditions.*

5 Another Approach to Permutability of Fuzzy Preorders

We have shown that two preorders R and P permute if and only if their consequences permute. For that, we need both $R \circ P$ and $P \circ R$ to be fuzzy preorders. For indistinguishability operators, the symmetric property facilitates the way. We need just to find that one of the compositions is an indistinguishability operator to get both of them. In this section, we study permutability of general fuzzy preorders from a different approach. We will need some results from [4]. The proofs can be found there.

Theorem 5. *Let $R \in \Gamma'$ be a reflexive fuzzy relation and let $g \in \Omega'$ be a FCO. If g is $*$-concordant with R, the operator C_R^g induced by g and R is also a FCO.*

Proposition 8. *Let R, P be fuzzy preorders and let C_R^* and C_P^* be their respective induced FCO. If C_R^* is $*$-concordant with P and C_P^* is $*$-concordant with R, then P and R permute.*

Proof. It directly follows from Theorems 5 and 3. □

We would like to show under which conditions two fuzzy preorders permute. For that, let us recall the definition of the fuzzy the preorder generated for a fuzzy operator C that we introduced in [4]:

Definition 12. *Let C be a fuzzy operator in Ω'. The fuzzy relation R_c^c induced by C is given by*

$$R_c^c(x,y) = \inf_{\mu \in [0,1]^X} \{C(\mu)(x) \to C(\mu)(y)\} \tag{9}$$

The following theorem is adapted from [3]:

Theorem 6. *Let $\{\mu_i\}_{i \in I} \subseteq [0,1]^X$ be an arbitrary family of fuzzy subsets. Then,*

$$R(x,y) = \inf_{i \in I} \{\mu_i(x) \to \mu_i(y)\} \tag{10}$$

is the largest fuzzy preorder for which every fuzzy subset of the family $\{\mu_i\}_{i \in I}$ is $*$-compatible with.

Notice that $\{\mu_i\}_{i \in I}$ is also $*$-compatible with S for every fuzzy relation S smaller than or equal to (10).

According to Theorem 6, the fuzzy relation R_c^c induced by a fuzzy operator C defined in (9) gives an upper bound which is sufficient for an operator and a relation to be $*$-concordant. In fact, if a fuzzy relation S is smaller than or equal to R_C^C for a certain fuzzy operator C, every fuzzy subset of the image of C will be compatible with S.

Proposition 9. *Let S be a fuzzy relation such that $S \leq R_c^c$ for a certain $C \in \Omega'$. Then, C is $*$-concordant with S.*

Corollary 6. *Let R, P be fuzzy preorders and let C_R^* and C_P^* be their respective induced FCO. If*

$$R \leq R_{c_P^*}^{c_P^*} \quad \text{and} \quad P \leq R_{c_R^*}^{c_R^*}$$

then, R and P permute.

6 Conclusions

In this paper we have proved that given two fuzzy $*$-preorders R and P on a general set they permute if and only if their compositions in both directions are fuzzy preorders, that is, if $R \circ P$ and $P \circ R$ are fuzzy perorders. In this case, their composition is the $*$-transitive closure of their maximum. If the preorders are $*$-indistinguishability operators, we only require one of the compositions to be an indistinguishability operator in order to ensure that they permute.

We have also shown that composition of fuzzy operators induced by fuzzy relations can be described in different ways. More precisely, $C_R^* \circ C_S^* = C_{S \circ R}^* = C_R^{C_S^*}$.

We have proved that for any pair of fuzzy preorders, they permute if and only if their consequences also do. Therefore, C_R^* and C_S^* permute if and only if $R \circ P$ and $P \circ R$ are fuzzy perorders. If R and P are indistinguishability operators, their consequences permute if and only if the composition of the relations in one direction is an indistinguishability operator.

Finally, we have given an alternative approach to permutability for general fuzzy preorders. We have connected permutability of fuzzy preorders to the crossed concordance between their consequences. Finally, we have found an upper bound which is sufficient for two fuzzy preorders to permute.

Acknowledgement. We ackowledge the partial support of the project FIS2011-28820-C02-02 from the Spanish Government and N.C. acknowledges the financial support of the "Asociación de Amigos de la Universidad de Navarra".

References

1. Bandler, W.,, Kohout, J.: Special properties, closures and interiors of crisp and fuzzy relations. Fuzzy Sets and Systems 26, 317–331 (1988)
2. Bělohlávek, R.: Fuzzy Relational Systems: Foundations and Principles. Ifsr International Series on Systems Science and Engineering, vol. 20. Kluwer Academic/Plenum Publishers, New York (2002)
3. Bodenhofer, U., De Cock, M., Kerre, E.E.: Openings and closures of fuzzy preorderings: theoretical basics and applications to fuzzy rule-based systems. International Journal of General Systems 32(4), 343–360 (2003) Special Issue: Issue Fuzzy Logic in Systems Modelling
4. Carmona, N., Elorza, J., Recasens, J., Bragard, J.: On the induction of new fuzzy relations, new fuzzy operators and their aggregation. In: Bustince, H., Fernandez, J., Mesiar, R., Calvo, T. (eds.) Aggregation Functions in Theory and in Practise. AISC, vol. 228, pp. 309–322. Springer, Heidelberg (2013)
5. Carmona, N., Elorza, J., Recasens, J., Bragard, J.: Permutability of fuzzy consequence operators and fuzzy interior operators. In: Bielza, C., Salmerón, A., Alonso-Betanzos, A., Hidalgo, J.I., Martínez, L., Troncoso, A., Corchado, E., Corchado, J.M. (eds.) CAEPIA 2013. LNCS (LNAI), vol. 8109, pp. 62–69. Springer, Heidelberg (2013)
6. Castro, J.L., Trillas, E.: Tarski's fuzzy consequences. In: Proceedings of the International Fuzzy Engineering Symposium 1991, vol. 1, pp. 70–81 (1991)
7. De Baets, B., Kerre, E., Gupta, M.: The fundamentals of fuzzy mathematical morphology part 1: Basic concepts. International Journal of General Systems 23(2) (1995)

8. Deng, T.Q., Heijmans, H.J.A.M.: Grey-scale morphology based on fuzzy logic. J. Math. Imaging Vision 16, 155–171 (2002)
9. Elorza, J., Burillo, P.: On the relation of fuzzy preorders and fuzzy consequence operators. International Journal of Uncertainty, Fuzziness and Knowledge-based Systems 7(3), 219–234 (1999)
10. Elorza, J., Burillo, P.: Connecting fuzzy preorders, fuzzy consequence operators and fuzzy closure and co-closure systems. Fuzzy Sets and Systems 139(3), 601–613 (2003)
11. Elorza, J., et al.: On the relation between fuzzy closing morphological operators, fuzzy consequence operators induced by fuzzy preorders and fuzzy closure and co-closure systems. Fuzzy Sets and Systems 218, 73–89 (2013)
12. Pavelka, J.: On Fuzzy Logic I. Zeitschr. f. Math. Logik und Grundlagen d. Math. 25, 45–52 (1979)
13. Recasens, J.: Indistinguishability operators and approximate reasoning. In: Recasens, J. (ed.) Indistinguishability Operators. STUDFUZZ, vol. 260, pp. 189–199. Springer, Heidelberg (2010)
14. Recasens, J.: Permutable indistinguishability operators, perfect vague groups and fuzzy subgroups. Information Sciences 196, 129–142 (2012)
15. Ward, M.: The closure operators of a lattice. Annals of Mathematics 43(2), 191–196 (1940)

Fuzzy Multisets in Granular Hierarchical Structures Generated from Free Monoids

Tetsuya Murai[1], Sadaaki Miyamoto[2], Masahiro Inuiguchi[3], Yasuo Kudo[4], and Seiki Akama[5]

[1] Graduate School of Information Science and Technologies, Hokkaido University
[2] Graduate School of Systems and Information Engineering, University of Tsukuba
[3] Graduate School of Engineering Science, Osaka University
[4] College of Information and Systems, Muroran Institute of Technology
[5] C-Republic

Abstract. This paper focuses on the two definitions of fuzzy multisets by Yager and Minamoto, respectively, and examines their difference in the framework of granular hierarchical structures generated from free monoids. Then we can conclude that, in order to define the basic order on the set of multisets on interval $(0, 1]$, the Yager definition adopts the one induced just from the range $\mathbb{N}$, the set of natural numbers, while the Miyamoto definition uses one generated from both the domain $(0, 1]$ and the range $\mathbb{N}$ through the notion of cuts.

1 Introduction

The theory of multisets[1, 5, 6, 16] and its extensions to fuzzy and rough multisets[2, 4, 8, 10, 11, 15, 18] is now widely agreed as one of important tools in the areas of computer science[9, 17], database and data mining[2, 7], and decision making[3, 8, 14].

There are many definitions of fuzzy multisets and, among them, the two definitions by Yager[18] and Miyamoto[10], respectively, are elementary ones. In order to examine the difference of the two definitions in the same framework, in this paper, we reformulate the two definitions in a unified way in granular hierarchical structures proposed by the authors[12].

2 Preliminaries

In this section, we briefly describe some elementary concepts and symbols which we use in later sections. Let U be a universal set in what follows,

2.1 Free Monoids

The basic structure we use in this paper is well known as a free monoid. A *free monoid* is a tuple $\langle U^*, \bullet, \varepsilon \rangle$, which is generated from U using the concatenation operation $\bullet$ with the empty string ε as the identity element with respect to $\bullet$.

V. Torra et al. (Eds.): MDAI 2013, LNAI 8234, pp. 248–259, 2013.

Of course, the operation $\bullet$ is associative and ε is the unit element with respect to $\bullet$. Every element in U^* is called a *finite sequence* or *string*. Symbol $\bullet$ is often abbreviated unless confusion arises as like $st = s \bullet t$.

There are several ways of constructing free monoids and, among them, we adopt here the definition as a direct sum of Cartesian products:

Definition 1. $U^* \stackrel{\text{def}}{=} \sum_{n\in\mathbb{N}} U^n$, *where let* $U^0 = \{\varepsilon\}$.

For elements in U^*, say $(x_0, \ldots, x_{n-1}) \in U^n$ and $(y_0, \ldots, y_{m-1}) \in U^m$ $(n, m \in \mathbb{N})$, operation of concatenation $\bullet$ is defined by

$$(x_0, \ldots, x_{n-1}) \bullet (y_0, \ldots, y_{m-1}) \stackrel{\text{def}}{=} (x_0, \ldots, x_{n-1}, y_0, \ldots, y_{m-1}) \in U^{n+m}.$$

The operation $\bullet$ is obviously associative and ε is the unit element with respect to $\bullet$. By the definition, $(x_0, \ldots, x_{n-1}) = x_0 \bullet \cdots \bullet x_{n-1}$ $(= x_0 \cdots x_{n-1})$. Let $x^k = (\overbrace{x, \cdots, x}^{k})$ for $k \in \mathbb{N}^+ = \mathbb{N} \setminus \{0\}$ and $x^0 = \varepsilon$.

In this paper, we adopt the following definition of substrings, which is the same as the concept of subsequences of the usual numerical sequences.

Definition 2. *Given strings* $s = x_0 \cdots x_{n-1}$ *and* $t = y_0 \cdots y_{m-1}$ *(*$x_0, \ldots, x_{n-1}$, $y_0, \ldots, y_{m-1} \in U$*), where* $n, m \in \mathbb{N}$, $n = \{0, 1, \ldots, n-1\}$ *and* $m = \{0, 1, \ldots, m-1\}$, s *is said to be a* substring *of* t*, denoted* $s \leq^* t$*, just in case there exists an order-preserving injection* $\varphi : n \to m$ *such that* $x_k = y_{\varphi(k)}$ *for every* $k \in n$.

The relation $\leq^*$ is a partial order on U^*, thus a structure $\langle U^*, \leq^* \rangle$ is a partially ordered set. In general, however, it does not form a lattice because there do not necessarily exist the join and meet of two strings.

2.2 Finite Naïve Subsets

The power set $\mathcal{P}(U)$ of U is defined by

$$\mathcal{P}(U) \stackrel{\text{def}}{=} \{X \mid X \subseteq U\},$$

where $\subseteq$ is the usual inclusion relation, $\mathcal{P}(U)$ is well-known to form a Boolean algebra with intersection $\cap$, union $\cup$, complement $^{\mathrm{C}}$ and the least and greatest elements $\emptyset$ and U.

Every subset in $\mathcal{P}(U)$ is usually identified with its corresponding element in the following set of putting

$$2^U \stackrel{\text{def}}{=} \{\chi \mid \chi : U \to 2\},$$

where $2 = \{0, 1\}$. In fact, 2^U forms a Boolean algebra and an isomorphism φ from $\mathcal{P}(U)$ to 2^U as a Boolean algebra is given by, for a subset $X \in \mathcal{P}(U)$,

$$\varphi(X)(x) = \begin{cases} 1, & \text{if } x \in X, \\ 0, & \text{otherwise.} \end{cases}$$

and thus we identify $\mathcal{P}(U)$ with 2^U. The mapping $\varphi(X)$ is often called the *characteristic function* of X.

In this paper, we define the set of *naïve subsets* by

Definition 3. $P(U) \stackrel{\text{def}}{=} \{X \in \mathcal{P}(U) \mid |X| < \infty\}$. *where* $|\cdot|$ *denotes the cardinality.*

In general, $P(U)$ is not necessarily a Boolean algebra but a distributive lattice. When U is finite, we have $P(U) = \mathcal{P}(U)$ and thus $P(U)$ is a Boolean algebra..

2.3 Finite Fuzzy Sets

Let $I = [0, 1]$. Following Zadeh[19], the founder of fuzzy set theory, by a *fuzzy set* on U, we mean a mapping $\mu : U \to I$, which is usually called *membership function* and the class of fuzzy sets on U is just the following set of putting:

$$I^U = \{\mu \mid \mu : U \to I\}.$$

In this paper, we deal with its finitery subclass:

Definition 4. $F(U) \stackrel{\text{def}}{=} \{\mu \in I^U \mid |\mu^{-1}(I^+)| < \infty\}$, *where* $I^+ = (0, 1]$.

We call every element in $F(U)$ a *finite fuzzy set*, or simply *fuzzy set* unless confusion arises, on U.

A partial order (inclusion relation) on $F(U)$ is derived from the natural order $\leq_I$ on I by

$$\mu \subseteq_F \mu' \stackrel{\text{def}}{\Leftrightarrow} \forall x \in U(\mu(x) \leq_I \mu'(x)),$$

and then, its compatible meet (intersection) and join (union) of two fuzzy sets are defined in a pointwise way, by, for every $x \in U$,

$$(\mu \cap_F \mu')(x) \stackrel{\text{def}}{=} \min_I\{\mu(x), \mu'(x)\},$$
$$(\mu \cup_F \mu')(x) \stackrel{\text{def}}{=} \max_I\{\mu(x), \mu'(x)\}$$

for fuzzy sets $\mu, \mu' \in F(U)$, where $\min_I$ and $\max_I$ denote the maximum and minimum elements, respectively, with respect to the order $\leq_I$.

$F(U)$ has the least element 0_F with respect to $\leq_F$ defined by $0_F(x) = 0$ for every $x \in U$ and $F(U)$ is a distributive lattice with the least element 0_F. When U is finite, $F(U)$ has the greatest element 1_F defined by $1_F(x) = 1$ for every $x \in U$ and $F(U)$ and complement ${}^{\mathrm{C}} : F(U) \to F(U)$ defined by $\mu^{\mathrm{C}}(x) = 1 - \mu(x)$ for every $x \in U$. $F(U)$ is known to form a pseudo-Boolean algebra.

2.4 Finite Multisets

By a *multiset* on U, we mean a mapping $\tilde{\chi} : U \to \mathbb{N}$, which is usually called *count function* and the class of multisets on U is just the following set of putting:

$$\mathbb{N}^U = \{\tilde{\chi} \mid \tilde{\chi} : U \to \mathbb{N}\}.$$

In this paper, we deal with its finitery subclass:

Definition 5. $M(U) \stackrel{\text{def}}{=} \{\tilde{\chi} \in \mathbb{N}^U \mid |\tilde{\chi}^{-1}(\mathbb{N}^+)| < \infty\}$, *where* $\mathbb{N}^+ = \mathbb{N} \setminus \{0\} = \{1, 2, \ldots\}$.

We call every element in $M(U)$ a *finite multiset*, or simply *multiset* unless confusion arises, on U.

A partial order (inclusion relation) on $M(U)$ is derived from the natural order $\leq_{\mathbb{N}}$ on $\mathbb{N}$ by

$$\tilde{\chi} \subseteq_M \tilde{\chi}' \stackrel{\text{def}}{\Leftrightarrow} \forall x \in U(\tilde{\chi}(x) \leq_{\mathbb{N}} \tilde{\chi}'(x)),$$

and then, its compatible meet (intersection) and join (union) of two multisets are derived in a pointwise way by, for every $x \in U$,

$$(\tilde{\chi} \cap_M \tilde{\chi}')(x) \stackrel{\text{def}}{=} \min_{\mathbb{N}}\{\tilde{\chi}(x), \tilde{\chi}'(x)\},$$
$$(\tilde{\chi} \cup_M \tilde{\chi}')(x) \stackrel{\text{def}}{=} \max_{\mathbb{N}}\{\tilde{\chi}(x), \tilde{\chi}'(x)\}$$

for multisets $\tilde{\chi}$, $\tilde{\chi}' \in M(U)$, where $\min_{\mathbb{N}}$ and $\max_{\mathbb{N}}$ denote the maximum and minimum elements, respectively, with respect to the order $\leq_{\mathbb{N}}$.

With respect to $\leq_M$, $M(U)$ has the least element $\tilde{0}$ defined by $\tilde{0}(x) = 0$ for every $x \in U$, but it does not have the greatest element and so, in general, we cannot give a natural definition of a complement-like operation on $M(U)$. Thus a structure $\langle M(U), \cap_M, \cup_M, \tilde{0}\rangle$ is a distributive lattice with the least element $\tilde{0}$. Similarly, from the natural operations + (sum) and × (product), we also define the following addition and multiplication between multisets $\tilde{\chi}$, $\tilde{\chi}'$ by

$$(\tilde{\chi} +_M \tilde{\chi}')(x) = \tilde{\chi}(x) + \tilde{\chi}'(x),$$
$$(\tilde{\chi} \times_M \tilde{\chi}')(x) = \tilde{\chi}(x) \times \tilde{\chi}'(x),$$

respectively. Each of $+_M$ and $\times_M$ satisfies distributivity with $\cap_M$ or $\cup_M$, respectively, and so both $\langle M(U), \cap_M, \cup_M, +_M\rangle$ and $\langle M(U) \setminus \{\tilde{0}\}, \cap_M, \cup_M, \times_M\rangle$ are distributive-lattice-ordered commutative monoids.

2.5 Finite Fuzzy Multisets

Yager[18] defined a *fuzzy multiset* on U as the following mapping

$$\tilde{\mu} : U \to M(I^+),$$

where $M(I^+)$ is the set of finite multisets on I^+. Note that non-belongingness is not counted so the range is the multisets on $I^+ = I \setminus \{0\}$. Thus the set of fuzzy multisets is the following set of putting:

$$M(I^+)^U = \{\tilde{\mu} \mid \tilde{\mu} : U \to M(I^+)\}.$$

In this paper, we deal with its finitery subclass:

Definition 6. $FM(U) \stackrel{\text{def}}{=} \{\tilde{\mu} \in M(I^+)^U \mid |\tilde{\mu}^{-1}(M(I^+) \setminus \{\tilde{0}\})| < \infty\}$.

We call every element in $FM(U)$ a *finite fuzzy multiset*, or simply *fuzzy multiset* unless confusion arises, on U. Also we call every element in $M(I^+)$ a *fuzzy multigrade*.

Yager[18] proposed a definition of an inclusion relation (partial order) on $FM(U)$ naturally derived from the standard one on $M(I^+)$ in an essentially similar way that partial orders on $F(U)$ and $M(U)$ are derived from the ones on I^+ and $\mathbb{N}$, respectively.

Definition 7 (Yager order[18]). *For fuzzy multigrades* $\tilde{g}$, $\tilde{g}' \in M(I^+)$,

$$\tilde{g} \leq_{YM} \tilde{g}' \overset{\text{def}}{\Leftrightarrow} \forall \alpha \in I^+ (\tilde{g}(\alpha) \leq_{\mathbb{N}} \tilde{g}'(\alpha)),$$

which we call the Yager order *on* $M(I^+)$ *in this paper.*

Then it is extended to the following inclusion on $FM(U)$, which we also call the *Yager inclusion* on $FM(U)$:

Definition 8 (Yager inclution[18]). *For fuzzy multisets* $\tilde{\mu}$, $\tilde{\mu}' \in FM(U)$,

$$\begin{aligned} \tilde{\mu} \subseteq_{YFM} \tilde{\mu}' &\overset{\text{def}}{\Leftrightarrow} \forall x \in U \ (\tilde{\mu}(x) \leq_{YM} \tilde{\mu}'(x)) \\ &\Leftrightarrow \forall x \in U \ \forall \alpha \in I^+ (\tilde{\mu}(x)(\alpha) \leq_{\mathbb{N}} \tilde{\mu}'(x)(\alpha)). \end{aligned}$$

Note that the Yager order and thus inclusion are defined only using the natural order on $\mathbb{N}$.

Example 1

$$\begin{aligned} \{\{1/0.5, 2/1\}/x, \{1/0.2\}/y\} = \tilde{\mu} \subseteq_{YFM} \tilde{\mu'} = \{\{1/0.5, 5/1\}/x, \{1/0.2, 1/1\}/y\}, \\ \{\{1/0.5\}/x, \{1/0.2\}/y\} = \tilde{\mu} \not\subseteq_{YFM} \tilde{\mu'} = \{\{1/0.8\}/x, \{1/0.6\}/y\}. \end{aligned}$$

The Yager inclusion has the infimum and supremum of two fuzzy multisets and thus its compatible operations of intersection and union are respectively introduced in a pointwise way: for fuzzy multisets $\tilde{\mu}$, $\tilde{\mu}' \in FM(U)$

$$\begin{aligned} (\tilde{\mu} \cap_{YFM} \tilde{\mu}')(x) &\overset{\text{def}}{=} \tilde{\mu}(x) \cap_M \tilde{\mu}'(x), \\ (\tilde{\mu} \cup_{YFM} \tilde{\mu}')(x) &\overset{\text{def}}{=} \tilde{\mu}(x) \cup_M \tilde{\mu}'(x), \end{aligned}$$

for every $x \in U$.

Miyamoto[10] pointed out that, although the class of the usual fuzzy sets can be embedded into the class of fuzzy multisets, the usual fuzzy set inclusion and operations are not compatible with the ones for fuzzy multisets. In fact, $F(U)$ can be embedded into $FM(U)$ by the following injection $\mathcal{E}$: for $\mu \in F(U)$

$$\mathcal{E}(\mu)(x)(\alpha) = \begin{cases} 1, & \text{if } \alpha = \mu(x), \\ 0, & \text{otherwise,} \end{cases}$$

for any $x \in U$ and $\alpha \in I^+$. Two orders, however, are not compatible with each other.

Example 2. For $\mu = \{0.5/x, 0.2/y\}$ and $\mu' = \{0.8/x, 0.6/y\}$, we have $\mu \subseteq_F \mu'$. But $\{\{1/0.5\}/x, \{1/0.2\}/y\} = \mathcal{E}(\mu) \not\subseteq_{YFM} \mathcal{E}(\mu') = \{\{1/0.8\}/x, \{1/0.6\}/y\}$ as shown in the previous example.

By this incompatibility, the subclass $\mathcal{E}(F(U))$ is closed neither with respect to $\cap_{YFM}$ nor to $\cup_{YFM}$. The reason is that, while, in general, the domain U of $M(U)$ is neutral from a structural point of view of U, the domain I^+ of $M(I^+)$ has its original linear order $\leq_{I^+}$, which is not considered in the definitions of Yager order and inclusion for fuzzy multisets.

3 Granular Hierarchical Structures

We proposed in [12] *granular hierarchical structures* of finite naïve subsets and multisets where they are derived from free monoids and homomorphisms. Then, we can represent finite naïve subsets and multisets as equivalence classes of strings with respect to some appropriate equivalence relation based on a homomorphism. In what follows, we assume the following function

$$ct^* : U \times U^* \to \mathbb{N},$$

which gives us the number of symbol $x \in U$ appearing in string $s \in U^*$.

3.1 U^* and $M(U)$

Firstly, a mapping $m : U^* \to M(U)$ defined by

$$m(s)(x) \stackrel{\text{def}}{=} ct^*(x, s)$$

is a subjective homomorphism where concatenation $\bullet$ is preserved as addition $+_M$:

$$m(\varepsilon) = \tilde{0},$$
$$m(s \bullet s') = m(s) +_M m(s'),$$

for $s, s' \in U^*$. Then an equivalence relation $\sim_{m^U}$, or simply $\sim_m$, on U^* is naturally defined by

$$s \sim_m s' \Leftrightarrow m(s) = m(s')$$

and $M(U)$ is isomorphic to $U^*/\sim_m$ as a monoid. Operation $\bullet_m$ on $U^*/\sim_m$ naturally induced from U^*, in a well-defined way, by

$$[s]_m \bullet_m [s']_m = [t]_m \text{ s.t. } t \in m^{-1}(m(s) +_M m(s')),$$

is commutative, where s,$s' \in U^*$ and $[s]_m$ denotes the equivalence class of s with respect to $\sim_m$. Further two operations are induced from U^* by

$$[s]_m \cap_m [s']_m = [t]_m \text{ s.t. } t \in m^{-1}(m(s) \cap_M m(s')),$$
$$[s]_m \cup_m [s']_m = [t]_m \text{ s.t. } t \in m^{-1}(m(s) \cup_M m(s')),$$

respectively, and we can easily show structure $\langle U^*/\sim_m, \bullet_m, \cap_m, \cup_m, [\varepsilon]_m\rangle$ forms a distributive-lattice-ordered commutative monoid with the least element $[\varepsilon]_m$. Hence we have

$$M(U) \cong U^*/\sim_m \text{ (as a distributive-lattice-ordered commutative monoid).}$$

3.2 $M(U)$ and $P(U)$

Secondly, a mapping $p : M(U) \to P(U)$ defined by

$$p(\tilde{\chi})(x) \stackrel{\text{def}}{=} min_{\mathbb{N}}\{1, \tilde{\chi}(x)\}$$

is a subjective homomorphism where the least element, multiset intersection and union are preserved:

$$\begin{aligned} p(\tilde{0}) &= \emptyset, \\ p(\tilde{\chi} \cap_M \tilde{\chi}') &= p(\tilde{\chi}) \cap p(\tilde{\chi}'), \\ p(\tilde{\chi} \cup_M \tilde{\chi}') &= p(\tilde{\chi}) \cup p(\tilde{\chi}'), \end{aligned}$$

for $\tilde{\chi}, \tilde{\chi}' \in M(U)$. Then we can introduce an equivalence relation $\sim_{p^U}$, or simply $\sim_p$, on $M(U)$ naturally defined by

$$\tilde{\chi} \sim_p \tilde{\chi}' \Leftrightarrow p(\tilde{\chi}) = p(\tilde{\chi}')$$

and $P(U)$ is isomorphic to $M(U)/\sim_p$ as a distributive lattice by the lattice isomorphism theorem. Here note that we also have

$$p(\tilde{\chi} +_M \tilde{\chi}') = p(\tilde{\chi}) \cup p(\tilde{\chi}'),$$

which means that addition $+_M$ together with $\cup_M$ collapses to set union $\cup$. Then we can similarly define meet $\cap_p$, join $\cup_p$, and $0_p = [\tilde{0}]_p$ is the least element of $M(U)/\sim_p$. Now we can easily show structure $\langle M(U)/\sim_p, \cap_p, \cup_p, 0_p, \rangle$ forms a distributive lattice with the least element and we easily have

$$P(U) \cong M(U)/\sim_p \text{ (as a distributive lattice).}$$

When U is finite, we can further introduce complement ${}^{\mathrm{C_P}}$ on $M(U)/\sim_p$ by

$$([\tilde{\chi}]_p)^{\mathrm{C_P}} = [\tilde{\chi}']_p \text{ s.t. } \tilde{\chi}' \in p^{-1}(p(\tilde{\chi})^{\mathrm{C}}),$$

where $\tilde{\chi}$, $\tilde{\chi}' \in M(U)$ and $[\tilde{\chi}]_p$ denotes the equivalence class of $\tilde{\chi}$ with respect to $\sim_p$. and the greatest element of $M(U)/\sim_p$ is also defined by

$$1_p = [\tilde{\chi}]_p \text{ s.t. } \tilde{\chi} \in p^{-1}(U).$$

Now we can easily show structure $\langle M(U)/\sim_p, \cap_p, \cup_p, {}^{\mathrm{C_P}}, 0_p, 1_p\rangle$ forms a Boolean algebra and we have

$$P(U) \cong M(U)/\sim_p \text{ (as a Boolean algebra).}$$

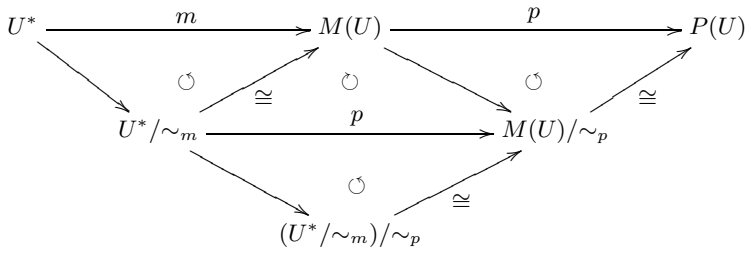


Fig. 1. Granular hierarchical structure generated from free monoid

3.3 U^* and $P(U)$

Finally, by composing the above two homomorphisms m and p, we have another homomorphism

$$p \circ m : U^* \to P(U)$$

and the following isomorphism:

$$P(U) \cong (U^*/\sim_m)/\sim_p \text{ (as a Boolean algebra or distributive lattice).}$$

3.4 Granular Hierarchical Structures

Thus we have the granular hierarchical structures shown in Figure 1. Those three isomorphisms introduced in this subsection give us the following results:

Theorem 1 (Murai et al.[12])

1. *For any multiset $\tilde{\chi} \in M(U)$, there exists a string $s \in U^*$ such that $\tilde{\chi} = [s]_m$.*
2. *For any naïve subset $X \in P(U)$, there exists a multiset $\tilde{\chi} \in M(U)$ such that $X = [\tilde{\chi}]_p$.*
3. *For any naïve subset $X \in P(U)$, there exists a string $s \in U^*$ such that $X = [[s]_m]_p$.*

Example 3. For universes U and V, given a mapping $f : U \to V$, we can extend its domain and range to the sets of strings, finite multisets, and finite naïve subsets, respectively, in the following steps.

1. $f^* : U^* \to V^*$ such that, for a string $s = x_1 \cdots x_n \in U^*$,
$$f^*(s) = f^*(x_1 \cdots x_n) = f(x_1) \cdots f(x_n) \text{ and } f^*(\varepsilon) = \varepsilon.$$
2. $f^m : M(U) \to M(V)$ such that, for a finite multiset $\tilde{\chi} = [s]_{m^U}$,
$$f^m(\tilde{\chi}) = f^m([s]_{m^U}) = [f^*(s)]_{m^V}.$$
3. $f^p : P(U) \to P(V)$ such that, for a finite naïve subset $X = [\tilde{\chi}]_{p^U} = [[s]_{m^U}]_{p^U}$,
$$f^p(X) = f^p([\tilde{\chi}]_{p^U}) = [f^m(\tilde{\chi})]_{p^V} = [f^m([s]_{m^U})]_{p^V} = [[f^*(s)]_{m^V}]_{p^V}.$$

These extensions are illustrated as the commutative diagram in Figure 2.

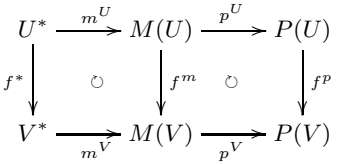


Fig. 2. Extension of mapping in granular hierarchical structure

4 Fuzzy Multisets in Granular Hierarchical Structures

In this section, we try to define fuzzy multisets as families of a kind of α-cuts ($\alpha \in I$) in granular hierarchical structures.

4.1 Extension of α-Cuts

First we remark the usual α-cut of a fuzzy set is represented by composition of mappings. Let us define a mapping $\bar{\alpha} : I \to 2$ by, for $g \in I$,

$$\bar{\alpha}(g) = \begin{cases} 1, & \text{if } g \geq \alpha, \\ 0, & \text{otherwise.} \end{cases}$$

Then an α-cut of fuzzy set $\mu \in M(U)$ is represented by the composition $\bar{\alpha} \circ \mu \in P(U)$.

We extend the above mapping to fuzzy multigrade cases by the way described in Example 3. However, we need a slight modification on $\bar{\alpha}$ because non-belongingness is usually not counted in both crisp and fuzzy multisets (cf. [18, 10]). So we replace the domain by I^+ and adopt ε instead of 0 in the range. That is, the mapping $\bar{\alpha} : I^+ \to \{1, \varepsilon\}$ is defined by, for any $\alpha \in I^+$,

$$\bar{\alpha}(g) = \begin{cases} 1, & \text{if } g \geq \alpha \\ \varepsilon, & \text{otherwise,} \end{cases}$$

where we use the same symbol $\bar{\alpha}$ unless confusion arises.

Now we extend the mapping $\bar{\alpha}$ using Example 3.

1. $\bar{\alpha}^* : (I^+)^* \to \{1, \varepsilon\}^*$ such that, for $g_1, \ldots, g_n \in I^+$,
$$\bar{\alpha}^*(g_1 \cdots g_n) = \bar{\alpha}(g_1) \cdots \bar{\alpha}(g_n)$$
where note that $\{1, \varepsilon\}^* = \{1^k \mid k \in \mathbb{N}\} = \{1^0 (= \varepsilon), 1^1, 1^2, \cdots\} \cong \mathbb{N}$, so $\bar{\alpha}^*$ is a finite multiset on $(I^+)^*$.
2. $\bar{\alpha}^M : M(I^+) \to M(\{1, \varepsilon\})$ such that, for $[g_1 \cdots g_n]_{m^{I^+}} \in M(I^+)$
$$\bar{\alpha}^M([g_1 \cdots g_n]_{m^{I^+}}) = [\bar{\alpha}^*(g_1 \cdots g_n)]_{m^{\{1,\varepsilon\}}} = [\bar{\alpha}(g_1) \cdots \bar{\alpha}(g_n)]_{m^{\{1,\varepsilon\}}},$$
where note that $M(\{1, \varepsilon\}) = \{[\iota_1 \cdots \iota_n]_m \mid \iota_k \in \{1, \varepsilon\}\} = \{[1^k]_m \mid k \in \mathbb{N}\} = \{[1^0]_m, [1^1]_m, [1^2]_m, \ldots\} \cong \mathbb{N}$, which means $\bar{\alpha}^M$ is a finite multiset on $M(I^+)$.

Definition 9 (α-cuts of fuzzy multisets). *For a fuzzy multiset $\tilde{\mu} \in FM(U)$ and an $\alpha \in I^+$, the α-cut of fuzzy multiset $\tilde{\mu}$ is the composition $\bar{\alpha}^M \circ \tilde{\mu}$.*

By this usage, the α-cut of fuzzy multiset is a multiset on U.

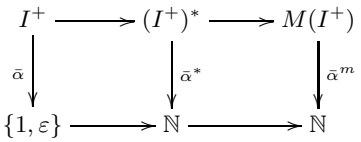


Fig. 3. Extensions of an abstract α-cut in granular hierarchical structure

4.2 Miyamoto Order and Inclusion

Using the above mapping $\bar{\alpha}^M$, we can define the new partial order between fuzzy multigrades.

Definition 10 (Miyamoto order[10]). *For fuzzy multigrade* $\tilde{g}$, $\tilde{g}' \in M(I^+)$,

$$\tilde{g} \leq_{MM} \tilde{g}' \overset{\text{def}}{\Longleftrightarrow} \forall \alpha \in I^+ \, (\bar{\alpha}^M(\tilde{g}) \leq_{\mathbb{N}} \bar{\alpha}^M(\tilde{g}')).$$

$\leq_{MM}$ is a partial order on $M(I^+)$, and we call this order the *Miyamoto order* on $M(I^+)$. Then, it can be naturally extended to the new inclusion relation on $FM(U)$.

Definition 11 (Miyamoto inclusion[10]). *For fuzzy multisets* $\tilde{\mu}$, $\tilde{\mu}' \in FM(U)$.

$$\begin{aligned} \tilde{\mu} \subseteq_{MFM} \tilde{\mu}' &\overset{\text{def}}{\Longleftrightarrow} \forall x \in U (\tilde{\mu}(x) \leq_{MM} \tilde{\mu}'(x)) \\ &\Longleftrightarrow \forall x \in U \, \forall \alpha \in I^+ \, (\bar{\alpha}^M \circ \tilde{\mu}(x) \leq_{\mathbb{N}} \bar{\alpha}^M \circ \tilde{\mu'}(x)) \\ &\Longleftrightarrow \forall x \in U \, \forall \alpha \in I^+ \, (\bar{\alpha}^M(\tilde{\mu}(x)) \leq_{\mathbb{N}} \bar{\alpha}^M(\tilde{\mu'}(x))). \end{aligned}$$

We call this inclusion relation the *Miyamoto inclusion* on $FM(U)$ because it is just the same one he defined using sequences of decreasing order (Miyamoto[10]).

Example 4. Fuzzy multigrades $\tilde{g} = [0.5 \bullet 1 \bullet 0.2]_{m^{I^+}}$ and $\tilde{g}' = [1 \bullet 0.2 \bullet 0.5 \bullet 0.5]_{m^{I^+}}$ obviously satisfies the Yager order, that is, $\tilde{g} \leq_{YM} \tilde{g}'$, because

$$\begin{aligned} 1 = \tilde{g}(0.2) &\leq_{\mathbb{N}} \tilde{g}'(0.2) = 1, \\ 1 = \tilde{g}(0.5) &\leq_{\mathbb{N}} \tilde{g}'(0.5) = 2, \\ 1 = \tilde{g}(1) &\leq_{\mathbb{N}} \tilde{g}'(1) = 1. \end{aligned}$$

Next, let us apply $\bar{\alpha}^M$ to the above $\tilde{g}$, $\tilde{g}'$, then we have

$$\begin{aligned} &\text{for } 0 < \alpha \leq 0.2, \quad 3 = \bar{\alpha}^M([0.5 \bullet 1 \bullet 0.2]_{m^{I^+}}) \leq_{\mathbb{N}} \bar{\alpha}^M([1 \bullet 0.2 \bullet 0.5 \bullet 0.5]_{m^{I^+}}) = 4, \\ &\text{for } 0.2 < \alpha \leq 0.5, \; 2 = \bar{\alpha}^M([0.5 \bullet 1 \bullet 0.2]_{m^{I^+}}) \leq_{\mathbb{N}} \bar{\alpha}^M([1 \bullet 0.2 \bullet 0.5 \bullet 0.5]_{m^{I^+}}) = 3, \\ &\text{for } 0.5 < \alpha \leq 1, \quad 1 = \bar{\alpha}^M([0.5 \bullet 1 \bullet 0.2]_{m^{I^+}}) \leq_{\mathbb{N}} \bar{\alpha}^M([1 \bullet 0.2 \bullet 0.5 \bullet 0.5]_{m^{I^+}}) = 1. \end{aligned}$$

and thus they satisfy the Miyamoto order $\tilde{g} \leq_{MM} \tilde{g}'$. We have the same result if we adopt sequences of decreasing order as representative elements of equivalence classes:

$$\begin{aligned} &\text{for } 0 < \alpha \leq 0.2, \quad 3 = \bar{\alpha}^M([1 \bullet 0.5 \bullet 0.2]_{m^{I^+}}) \leq_{\mathbb{N}} \bar{\alpha}^M([1 \bullet 0.5 \bullet 0.5 \bullet 0.2]_{m^{I^+}}) = 4, \\ &\text{for } 0.2 < \alpha \leq 0.5, \; 2 = \bar{\alpha}^M([1 \bullet 0.5 \bullet 0.2]_{m^{I^+}}) \leq_{\mathbb{N}} \bar{\alpha}^M([1 \bullet 0.5 \bullet 0.5 \bullet 0.2]_{m^{I^+}}) = 3, \\ &\text{for } 0.5 < \alpha \leq 1, \quad 1 = \bar{\alpha}^M([1 \bullet 0.5 \bullet 0.2]_{m^{I^+}}) \leq_{\mathbb{N}} \bar{\alpha}^M([1 \bullet 0.5 \bullet 0.5 \bullet 0.2]_{m^{I^+}}) = 1. \end{aligned}$$

Example 5. Fuzzy multigrades $\tilde{g} = [0.5]_{m^{I^+}}$, $\tilde{g}' = [1]_{m^{I^+}}$ does not satisfy the Yager order that is, $\tilde{g} \not\leq_{YM} \tilde{g}'$, because

$$1 = \tilde{g}(0.5) \leq_{\mathbb{N}} \tilde{g}'(0.5) = 0,$$
$$0 = \tilde{g}(1) \leq_{\mathbb{N}} \tilde{g}'(1) = 1.$$

But, by applying $bar\alpha^M$ to them, we have

$$\text{for } 0 < \alpha \leq 0.5,\ 1 = \bar{\alpha}^M([0.5]_{m^{I^+}}) \leq_{\mathbb{N}} \bar{\alpha}^M([0.5]_{m^{I^+}}) = 1,$$
$$\text{for } 0.5 < \alpha \leq 1,\ 0 = \bar{\alpha}^M([0.5]_{m^{I^+}}) \leq_{\mathbb{N}} \bar{\alpha}^M([1]_{m^{I^+}}) = 1.$$

and thus they satisfy the Miyamoto order $\tilde{g} \leq_{MM} \tilde{g}'$.

5 Concluding Remarks

Through the examination in this paper, when defining the basic order on the set $M(I^+)$, we have found that the Yager definition adopts the one induced just from the range $\mathbb{N}$, while the Miyamoto definition uses one generated from both the domain I^+ and the range $\mathbb{N}$ through the notion of cuts. Such choice would depend on the context of given problems.

We plan to describe the decomposition theorem for fuzzy multisets and applications to decision making in future tasks. The idea in this paper also can be applied to rough multisets as illustrated in Figure 4 because the rough set operators $[R]$ and $\langle R\rangle$ generated from an equivalence relation R on U (Pawlak[13]) are mappings on $P(U)$. Further it can be extended to rough fuzzy multisets and other hybridizations of those extended naïve set concepts.

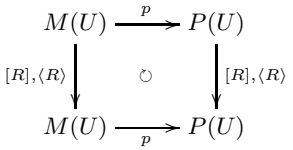


Fig. 4. Rough multisets in a granular hierarchical structure

Acknowledgments. This research was partially supported by Grant-in-Aid No. 24650099 for Challenging Exploratory Research of the Japan Society for the Promotion of Science.

References

[1] Blizard, W.D.: Multiset theory. Notre Dame Journal of Formal Logic 30, 36–66 (1989)

[2] Chan, C.-C.: Learning rules from very large databases using rough multisets. In: Peters, J.F., Skowron, A., Grzymała-Busse, J.W., Kostek, B.z., Swiniarski, R.W., Szczuka, M.S. (eds.) Transactions on Rough Sets I. LNCS, vol. 3100, pp. 59–77. Springer, Heidelberg (2004)

[3] Chen, Y.-K., Liao, H.-C.: An investigation on selection of simplified aggregate production planning strategies using MADM approaches. Int. J. of Production Research 41, 3359–3374 (2003)
[4] Girish, K.P., Sunil, J.J.: Rough multisets and information multisystems. In: Advances in Decision Sciences, vol. 2011 (2011)
[5] Jena, S.P., Ghosh, S.K., Tripathy, B.K.: On the theory of bags and lists. Information Sciences 132, 241–254 (2001)
[6] Knuth, D.E.: The Art of Computer Programming, vol. 2. Addison-Wesley, Reading (1969)
[7] Lamperti, G., Melchiori, M., Zanella, M.: On multisets in database systems. In: Calude, C.S., Pun, G., Rozenberg, G., Salomaa, A. (eds.) Multiset Processing. LNCS, vol. 2235, pp. 147–215. Springer, Heidelberg (2001)
[8] Li, B.: Fuzzy bags and applications. Fuzzy Sets and Systems 34, 61–71 (1990)
[9] Miyamoto, S.: Fuzzy multisets and fuzzy clustering of documents. In: Proc. of 10th IEEE Int. Conf. on Fuzzy Systems, pp. 1539–1542 (2001)
[10] Miyamoto, S.: Generalizations of multisets and rough approximations. International Journal of Intelligent Systems 19, 639–652 (2004)
[11] Miyamoto, S.: Different generalizations of bags. Annals of Operations Research 195, 221–236 (2012)
[12] Murai, T., Miyamoto, S., Inuiguchi, M., Akama, S.: Granular Hierarchical Structures of Finite Naïve Subsets and Multisets. Int. J. Reasoning-based Intelligent Systems 4(3), 118–128 (2012)
[13] Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning about Data. Kluwer, Dordrecht (1991)
[14] Rebaï, A., Martel, J.-M.: A fuzzy bag approach to choosing the "best" multiattributed potential actions in a multiple judgement and non cardinal data context. Fuzzy Sets and Systems 87, 159–166 (1997)
[15] Rochester, D., Bosc, P.: The set of fuzzy relative integers and fuzzy bags. Int. J. of Intelligent Systems 24, 677–694 (2009)
[16] Singh, D., Ibrahim, A.M., Bello, A., Yohanna, T., Singh, J.N.: A systematization of fundamentals of multisets. Lecturas Matemáticas 29, 33–48 (2008)
[17] Syropoulos, A.: Mathematics of multisets. In: Calude, C.S., Pun, G., Rozenberg, G., Salomaa, A. (eds.) Multiset Processing. LNCS, vol. 2235, pp. 347–358. Springer, Heidelberg (2001)
[18] Yager, R.R.: On the Theory of Bags. Int. J. General Systems 13, 23–37 (1986)
[19] Zadeh, L.A.: Fuzzy Sets. Information and Control 8(3), 338–353 (1965)

Landmark Selection for Isometric Feature Mapping Based on Mixed-Integer Optimization

Carlotta Orsenigo and Carlo Vercellis

Dept. of Management, Economics and Industrial Engineering, Politecnico di Milano, Via Lambruschini 4b, 20156 Milano, Italy

Abstract. Isometric feature mapping (Isomap) demonstrated noteworthy performance for nonlinear dimensionality reduction in a wide range of application domains. To improve the scalability of the algorithm a fast variant, called Landmark Isomap (L-Isomap), has been proposed in which time-consuming computations are performed on a subset of points referred to as landmarks. In this paper we present a novel method for landmark selection to be framed within the L-Isomap procedure. It is based on a mixed-integer problem aimed at finding a set of landmarks which are representative of dense regions of points in the input space which mostly contain samples of the same class. The optimization model is solved by a heuristic algorithm based on Lagrangian relaxation with subgradient method. Computational experiments performed on benchmark data sets highlighted the effectiveness of the proposed landmark selection algorithm which, combined with L-Isomap, provided promising results in terms of classification accuracy and computational effort.

Keywords: Isometric feature mapping, landmark selection, mixed-integer optimization, classification.

1 Introduction

Dimensionality reduction is aimed at finding meaningful low-dimensional representations of high dimensional data. As a preprocessing step it plays a prominent role for both unsupervised tasks, such as clustering or data visualization, and supervised learning.

For dimensionality reduction several linear and nonlinear approaches have been proposed. Classical methods addressing linear data projections are Principal Component Analysis [1] and Metric Multidimensional Scaling [2]. Within the family of nonlinear techniques manifold learning algorithms have recently drawn great interest, being able to handle data with intrinsic nonlinear structures. Representative methods include Isometric Feature Mapping [3], Locally Linear Embedding [4] and Laplacian Eigenmaps [5].

Manifold learning techniques try to unveil the low-dimensional manifold, embedded in the high-dimensional Euclidean space, along which data are supposed to lie. Given a set of m points $\mathcal{S}_m = \{\mathbf{x}_i,\ i \in \mathcal{M} = \{1, 2, \ldots, m\}\} \subset \Re^n$ which lie on or close to a nonlinear manifold M of unknown dimension d,

V. Torra et al. (Eds.): MDAI 2013, LNAI 8234, pp. 260–271, 2013.

with $d \ll n$, they aim at finding a function $f : M \to \Re^d$ mapping $\mathcal{S}_m$ into $\mathcal{D}_m = \{\mathbf{z}_i,\ i \in \mathcal{M} = \{1, 2, \ldots, m\}\} \subset \Re^d$ such that some geometrical properties are preserved in the projection. In the context of classification manifold learning methods have been successfully applied for diversified purposes, such as neuroimaging data analysis [6], banks' rating prediction [7], face and speech recognition [8,9]. An empirical comparison of these techniques for microarray data classification is presented in [10].

Isometric feature mapping (Isomap) finds an embedding which attempts to preserve the global geometrical properties of the data in the low-dimensional space. To this aim, it computes the geodesic distance between each pair of points, defined as the length of the shortest path between the corresponding vertices in a weighted neighborhood graph, and derives the data projection by the eigen-decomposition of the $m \times m$ matrix of squared geodesic distances. When the number of samples increases, however, computing all the shortest paths and the spectral decomposition of the full distance matrix may be too time-expensive, limiting the scalability of the algorithm. To overcome this drawback a fast variant called Landmark Isomap (L-Isomap) was proposed in [11], in which the geodesic distances are computed between the m points and a subset of l distinguished samples, indicated as landmarks. Multidimensional scaling is then applied to the resulting $l \times m$ distance matrix to find the landmarks embedding, and a fixed linear transformation is finally used to project the remaining points in the d-dimensional space.

Design the set of landmarks to be framed within the L-Isomap algorithm is still an open question. Some authors resorted to clustering methods, such as self-organizing map [12] and fuzzy c-means [13], or to weighting schemes defined on the distance between points and their neighbors [14]. An interesting approach based on integer optimization was developed in [15] and effectively applied for analysing protein interactions [16]. It relies on the approximate solution of a minimum set covering problem, which finds a minimum set of landmarks whose neighborhoods cover the entire set of points.

In this paper we present a novel method for landmark selection based on mixed-integer optimization. The proposed model identifies a set of l points achieving an optimal trade-off between two distinct objectives. From one side, it searches for points endowed with highly cohesive neighborhoods. From the other side, it favors the selection of close samples whose neighborhoods mostly contain points of the same class. The combination of these two objectives is aimed at finding a set of landmarks which are representative of dense regions of points in the input space which are also homogeneous in terms of class membership. The optimization problem is solved by a heuristic algorithm based on Lagrangian relaxation with subgradient method. Experiments conducted to evaluate the usefulness of the new model for landmark selection highlighted the potential of the proposed method which, combined with L-Isomap, exhibited promising performances in terms of classification accuracy and computational effort compared to the original Isomap algorithm.

The remainder of the paper is organized as follows. Section 2 offers a brief overview of isometric feature mapping and of its fast L-Isomap variant. Section 3 describes the mixed-integer problem proposed for landmark selection. Section 4 illustrates the experimental settings and the computational results concerning the classification of six benchmark data sets. Conclusions and future extensions are discussed in section 5.

2 Isometric Feature Mapping

Isometric feature mapping (Isomap) represents an extension of metric multidimensional scaling (MDS) to nonlinear manifolds. Unlike MDS which builds low-dimensional representations based on the Euclidean distance among points, Isomap tries to preserve the global geometric properties of the data by finding an embedding in which the geodesic distance between two points in the input space is as close as possible to the Euclidean distance between their projections in the target space. The geodesic distance is defined by the length of the shortest curve connecting two points on the underlying manifold, which is generally unknown in advance. For this reason, the geodesic distance between two points is approximated by the shortest path computed between the corresponding vertices in a weighted neighborhood graph, whose nodes represent data points and edges neighborhood relations. The embedding in the low d-dimensional space is therefore obtained by performing the singular value decomposition of the matrix of squared geodesic distances.

Within the original Isomap algorithm two different criteria were proposed for building the neighborhood graph [3]. According to the first, which is the one adopted in this study, two nodes are connected by an edge if one of them is among the k nearest neighbors of the other. As an alternative, an edge is added between two nodes if their Euclidean distance is smaller than a given threshold ε; this second rule is tantamount to defining for each point a neighborhood composed by all the samples lying within a ε-radius hypersphere. In both cases a weighted neighborhood graph is obtained in which the weight of an edge equals the Euclidean distance between its endpoints.

The Isomap algorithm can be summarized as follows.

Procedure. Isomap($\mathcal{S}_m, d, k$ or ε)

1. Build the neighborhood graph by connecting each point of $\mathcal{S}_m$ to at most k nearest neighbors (or to the points lying within a ε-radius hypersphere).
2. Compute the matrix $\mathbf{G}$ of the geodesic distances estimated by the length the shortest paths between each pair of vertices in the neighborhood graph.
3. Find the embedding in the low d-dimensional space by applying multidimensional scaling. To this aim, first compute the square m-dimensional matrix $\mathbf{K} = -\mathbf{HSH}/2$, where $\mathbf{S}$ is the matrix of squared geodesic distances and $\mathbf{H}$ is the centering matrix of size m. Then, consider the d-dimensional diagonal matrix $\mathbf{\Lambda}$ composed by the first d largest eigenvalues of $\mathbf{K}$ and the $m \times d$ matrix $\mathbf{V}$ of associated eigenvectors. Finally, find the embedding of the points

as $\mathbf{Z} = \mathbf{V}\mathbf{\Lambda}^{1/2}$, where $\mathbf{Z}$ is a $m \times d$ matrix whose rows are the projections $\mathbf{z}_i, i \in \mathcal{M}$, of the points in the low d-dimensional space.

Isometric feature mapping has been successfully applied to the analysis of several high-dimensional real world and artificial data sets. When the number of points increases, however, the algorithm may turn out to be too expensive in terms of computational effort. From one side, it requires to find the matrix $\mathbf{G}$ of the geodesic distances between each pair of nodes in the neighborhood graph, with a time complexity of $O\left(km^2 \log m\right)$ if the Dijkstra's algorithm is used. From the other side, it performs the eigendecomposition of the full $m \times m$ matrix $\mathbf{K}$ with a complexity of $O\left(m^3\right)$.

To overcome these inefficiencies, a fast extension indicated as Landmark Isomap (L-Isomap) was presented in [11]. In the proposed variant, l data points are first designated to be the landmarks or prototypes. Then, the shortest paths from each point to the landmarks are computed and classical MDS is applied to the resulting $l \times m$ distance matrix, in order to obtain the landmarks embedding in the low-dimensional space. The projections of the remaining points are finally derived by an affine linear transformation of their squared distances to the landmarks. As observed in [17], for a d-dimensional embedding at least $d+1$ prototypes must be chosen. In practice, it is recommended to select rather more landmarks than the strict minimum to ensure stability, even though substantial computational savings are obtained when $l \ll m$.

In this paper we propose a novel method for isolating landmark points based on mixed-integer optimization. In particular, once the neighborhood graph is built we solve a mixed-integer problem whose solution returns a set of prototypes which are subsequently fed into the L-Isomap algorithm for the data embedding. Before describing the optimization model, however, we are required to provide some final details about the Isomap implementation concerning the estimate of d, the choice of the parameters regulating the neighborhood size and the out-of sample extension.

2.1 Parameters Selection

Evaluating the dimensionality d of the projection space is still an open issue for which no dominant techniques currently exist. The most straightforward way, suggested in [3] and used in the present work, is to consider the curve of residual variance and select the dimension at which the curve flattens. The residual variance is defined by $1 - R^2\left(\mathbf{G}, \mathbf{D}\right)$, where $\mathbf{G}$ is the geodesic distance matrix, $\mathbf{D}$ is the matrix of Euclidean distances in the projection space and R is the linear correlation coefficient over all entries of $\mathbf{G}$ and $\mathbf{D}$.

In the Isomap procedure the shortest path between each pair of vertices can be computed only if the neighborhood graph is connected. This requirement makes the algorithm highly sensitive to the choice of its parameters. In particular, too small values of k or ε may create holes in the manifold and disconnect the graph. On the contrary, too large values may cause inappropriate connections among distinct folds and may result into misleading projections, especially for data

sets affected by noise or outliers. To overcome these drawbacks and to obtain a connected graph without unnaturally increasing the value of k we resorted to the linkage paradigm proposed in [18], for which the separate subgraphs are joined by computing a minimum spanning tree among the points that best represent the centroids of the single components.

2.2 Out-of-Sample Extension

For classification tasks, as those addressed in this study, the use of manifold learning techniques for dimensionality reduction requires an out-of-sample embedding method to project new data points in the low-dimensional space. To this aim one may resort to general regression neural networks [19] or to multi-output kernel ridge regression (KRR) [20], which represents a generalization of linear ridge regression based on kernel functions.

KRR can achieve an ideal trade-off between bias and variance of the estimates leading to a more precise approximation of the mapping, and has proven to be rather effective in combination with a supervised variant of Isomap compared to regression neural networks [18]. For this reason we applied multi-output kernel ridge regression based on the radial basis function kernel for out-of-sample data projections.

3 Mixed-Integer Optimization for Landmark Selection

This section provides a description of the mixed-integer problem for landmark selection and of the heuristic procedure adopted for its solution.

The proposed method identifies a subset of points achieving an optimal trade-off between two distinct objectives. From one side, it searches for points endowed with highly cohesive neighborhoods, in order to retain as landmarks samples which are intimately related to their k nearest neighbors. From the other side, it favors the selection of close points whose neighborhoods contain, for the most part, samples of the same class. The combination of these two objectives is aimed at finding a set of landmarks which are representative of dense regions of points in the input space which are also homogeneous in terms of class membership.

Let $\mathcal{L}$ be the index set of landmarks and $\mathcal{K}_i$ the index set of the k nearest neighbors of $\mathbf{x}_i$, $i \in \mathcal{M}$, including also i. Let $\bar{y}_i$ denote the estimated class of neighborhood $\mathcal{K}_i$, defined as the class label of the majority of its points, and $\mathcal{I}$ the index set of the pair of points whose neighborhoods have the same estimated class: $\mathcal{I} = \{(i,j) \in \mathcal{M}, i < j : \bar{y}_i = \bar{y}_j\}$. Finally, define the family of binary variables

$$p_i = \begin{cases} 1 & \text{if } i \in \mathcal{L} \\ 0 & \text{otherwise} \end{cases}, \; i \in \mathcal{M} \tag{1}$$

each taking the value 1 if and only if the corresponding point represents a landmark.

To determine the set $\mathcal{L}$ we formulate the following optimization problem

$$\max_{\mathbf{p},\mathbf{q}} \quad \delta \sum_{i\in\mathcal{M}} c_i p_i - (1-\delta) \sum_{(i,j)\in\mathcal{I}} d_{ij} q_{ij} \tag{LS}$$

$$\text{s. to} \quad q_{ij} \geq p_i + p_j - 1 \quad (i,j) \in \mathcal{I} \tag{2}$$

$$p_i \in \{0,1\} \quad i \in \mathcal{M}, \quad q_{ij} \geq 0 \quad (i,j) \in \mathcal{I}$$

where d_{ij} is the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ in the input space, c_i is the cohesion coefficient of neighborhood $\mathcal{K}_i$, defined as $c_i = \left(\sum_{r,h\in\mathcal{K}_i, r<h} d_{rh}\right)^{-1}$, $i \in \mathcal{M}$, and $\delta \in (0,1)$ is a parameter controlling the trade-off between the objective function terms. Notice that the continuous variable q_{ij} takes the value 1 only when $\mathbf{x}_i$ and $\mathbf{x}_j$ are both selected, in light of the constraints (2). In all the other cases, $q_{ij} = 0$ due to the maximization of the objective function. Furthermore, the parameter δ exerts a relevant influence on the cardinality l of the set $\mathcal{L}$. Indeed, for a fixed k the number of selected landmarks generally increases as δ approaches 1.

Model LS can be solved to optimality by means of standard mixed-integer programming algorithms. When large scale instances are considered, however, exact methods are computationally expensive and heuristic procedures are required. Among the most effective approaches we focused on Lagrangian relaxation with subgradient optimization [21].

To this aim, we relaxed constraints (2) by means of the nonnegative Lagrangian multipliers λ_{ij}, $(i,j) \in \mathcal{I}$. As observed in [22], by imposing the condition $\lambda_{ij} \leq (1-\delta)\, d_{ij}$ on each multiplier it is possible to set $q_{ij} = 0$, $(i,j) \in \mathcal{I}$, and obtain the following simplified Lagrangian problem

$$L(\boldsymbol{\lambda}) = \max\left\{ \sum_{i\in\mathcal{M}} f_i p_i + \sum_{(i,j)\in\mathcal{I}} \lambda_{ij} : p_i \in \{0,1\} \quad i \in \mathcal{M} \right\}, \tag{LR}$$

where

$$f_i = \delta c_i - \sum_{i,j\in\mathcal{M},\, \bar{y}_i=\bar{y}_j, 1\leq j<i} \lambda_{ji} - \sum_{i,j\in\mathcal{M},\, \bar{y}_i=\bar{y}_j, i<j\leq m} \lambda_{ij} \tag{3}$$

is the reduced cost associated to $\mathbf{x}_i$. For a given multiplier vector $\boldsymbol{\lambda}$ an optimal solution to LR is given by $p_i(\boldsymbol{\lambda}) = 1$ if $f_i > 0$ and $p_i(\boldsymbol{\lambda}) = 0$ otherwise. The Lagrangian dual problem LD associated to LR consists of finding a vector $\boldsymbol{\lambda}^*$ which minimizes the upper bound $L(\boldsymbol{\lambda})$:

$$\min\left\{ L(\boldsymbol{\lambda}) : 0 \leq \lambda_{ij} \leq (1-\delta)\, d_{ij} \quad (i,j) \in \mathcal{I} \right\}. \tag{LD}$$

Since finding $\boldsymbol{\lambda}^*$ is generally time-consuming for large scale instances, it is advisable to determine a near-optimal solution to LD by using the subgradient method, which generates a sequence $\{\boldsymbol{\lambda}^0, \boldsymbol{\lambda}^1, \ldots\}$ of nonnegative multiplier vectors where $\boldsymbol{\lambda}^0$ is arbitrarily defined.

The subgradient method can be summarized as follows.

Procedure. Subgradient method(LB,T,$\alpha_{\min}$)

1. Define the starting vector $\boldsymbol{\lambda}^0$ and set $UB = L\left(\boldsymbol{\lambda}^0\right)$. Set $t = 0$ and $\alpha = 10^{-2}$.
2. Compute the current optimal solution of the Lagrangian problem $\tilde{\mathbf{z}}\left(\boldsymbol{\lambda}^t\right)$ and the upper bound $L\left(\boldsymbol{\lambda}^t\right)$. Set $UB = \min\left\{UB, L\left(\boldsymbol{\lambda}^t\right)\right\}$.
3. Compute the subgradient vector $\mathbf{s}\left(\boldsymbol{\lambda}^t\right)$ for the current solution $\tilde{\mathbf{z}}\left(\boldsymbol{\lambda}^t\right)$. If $\mathbf{s}\left(\boldsymbol{\lambda}^t\right) = \mathbf{0}$ output $\boldsymbol{\lambda}^t$ and stop (in this case $\tilde{\mathbf{z}}\left(\boldsymbol{\lambda}^t\right)$ is an optimal solution). Otherwise, compute a new Lagrangian multiplier vector

$$\boldsymbol{\lambda}^{t+1} = \min\left\{\max\left\{0, \boldsymbol{\lambda}^t + \alpha\frac{LB - L\left(\boldsymbol{\lambda}^t\right)}{\left\|\mathbf{s}\left(\boldsymbol{\lambda}^t\right)\right\|^2}\mathbf{s}\left(\boldsymbol{\lambda}^t\right)\right\}, (1-\delta)\,\mathbf{E}\right\} \tag{4}$$

   where $\alpha > 0$ is a given step-size parameter and $\mathbf{E}$ is the matrix of Euclidean distances in the input space.
4. If UB has not improved in the last T iterations with the current value of $\alpha > 0$ set $\alpha = 0.5\alpha$. If $\alpha \leq \alpha_{\min}$ output $\boldsymbol{\lambda}^t$ and stop. Otherwise, set $t = t+1$ and return to Step 2.

In our implementation the starting multiplier vector was set to $\boldsymbol{\lambda}^0 = \mathbf{0}$ whereas the lower bound LB on the optimal solution, provided in input, was fixed to the objective function value of problem LS obtained by letting $p_i = 1$, $i \in \mathcal{M}$. Finally, we set $T = 20$ and $\alpha_{\min} = 5 \cdot 10^{-4}$.

4 Experiments

To investigate the performance of the proposed landmark selection method we performed computational experiments concerning the classification of six benchmark data sets. Our purpose was to determine whether the fast variant based on L-Isomap with the LS model may be effectively used for dimensionality reduction when it is combined with a classical supervised learning algorithm represented by the 1-nearest neighbor classifier (1NN). In particular we intended to compare, in terms of classification accuracy, the quality of the data embedding when the projections were built on the entire set of data, as in the original Isomap procedure, and on the subset of distinguished landmarks.

Four alternative methods were considered in our tests. The first was given by the 1NN classifier directly applied to each data set in the original input space. The second resorted to the classical Isomap algorithm for dimensionality reduction. The last two methods, denoted as L-Isomap$_{LS}$ and L-Isomap$_{SC}$, implemented the fast variant of Isomap based, respectively, on the LS and the set covering (SC) problem for landmark selection, where the SC problem was solved by means of the greedy algorithm described in [16]. Notice that for Isomap-based methods 1NN was used to classify the projection of each data set in the low d-dimensional space.

The experimental framework comprised two main phases. First, the intrinsic dimensionality d of the manifold was estimated for each data set.

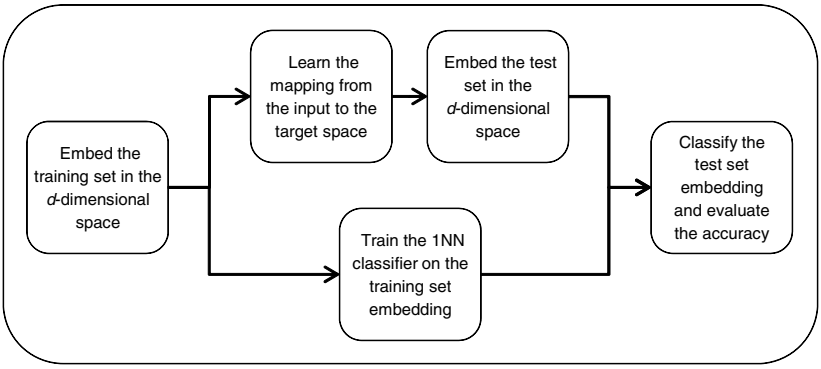


Fig. 1. Embedding and classification scheme for each pair of training and test set within cross-validation for Isomap-based methods.

Then cross-validation was applied in order to evaluate the classification accuracy. For methods based on dimensionality reduction three additional steps, depicted in Figure 1, were performed within cross-validation, for each pair of training and test set. Specifically, the d-dimensional embedding of the training data was first computed. Then, the function mapping the training set into its projection was approximated by multi-output kernel ridge regression and was used to find the low-dimensional representation of the test set. Finally, the 1NN classifier was trained on the projected training set and its accuracy was evaluated on the embedded test set.

Experiments were performed on six data sets publicly available from the UCI Machine Learning Repository [23]: Spectf Heart (Spectf), Musk, Red Wine Quality (Wine), Madelon, Waveform Database Generator (Waveform) and Page Blocks Classification (Pageblocks). As indicated in Table 1, these data sets differ in terms of number of points and attributes and were chosen with the aim of analyzing the effect of landmark selection on both small and larger sets of data. Before the tests a preprocessing step was conducted in which duplicate samples were removed and all attributes were standardized. The final size of each data set is provided in Table 1.

To automatically estimate the dimensionality d of the embedding space we considered the curve of residual variance generated by Isomap and searched for the dimension at which the curve ceased to decrease significantly. The intrinsic dimensionality d evaluated for each data set is indicated in Table 1.

The most promising parameters of each method were empirically found by minimizing the classification error rate in a preliminary 3-fold cross-validation run. In order to limit the grid search for Isomap-based methods, the number k of nearest neighbors was varied in the interval $[2, 1/10m]$ with a step size of 2. For L-Isomap$_{LS}$ the parameter δ regulating the trade-off between the objective function terms took values in the interval $[0.2, 0.8]$ with step 0.2. Finally, for KRR the regularization coefficient was fixed to 10^j, $j \in [-3, -1]$, and the radial basis function kernel parameter to 10^j, $j \in [-5, -3]$.

Table 1. Description of the data sets. The last column indicates the dimensionality of the projection space estimated for each data set.

Data set	Points	Attributes	Classes	d
Spectf	267	44	2	4
Musk	476	166	2	8
Wine	1359	11	6	6
Madelon	2600	500	2	4
Waveform	5000	40	3	10
Pageblocks	5406	10	5	8

The classification accuracy was evaluated by means of ten times stratified 3-fold cross-validation by using the best parameters identified in the exploratory run. To guarantee a fair comparison the same folds for training and testing were used for all methods. The computational results are detailed in Table 2 where, for each method, the first row reports the average classification accuracy and the second the corresponding standard deviation. For L-Isomap algorithms the value in the third row indicates the average percentage of training points selected as landmarks and used for data embedding.

From the results of Table 2 some empirical conclusions can be drawn. Resorting to dimensionality reduction by means of Isomap and its variants induced an improvement in accuracy with respect to the base case represented by 1NN applied in the original input space. Not surprisingly, the best performance was most often achieved by the Isomap-based classifier, for which the low-dimensional representations were built on the whole set of training samples.

As indicated in Table 2, the proposed L-Isomap$_{LS}$ algorithm dominated its counterpart based on the minimum set covering problem in terms of prediction accuracy. Despite the fast extensions both resorted to a restricted percentage of points for data embedding, on some data sets L-Isomap$_{LS}$ provided better results with an even smaller number of landmarks. This remarkable behavior may be ascribed to two main reasons. The landmark selection model in L-Isomap$_{LS}$ uses the class labels of the points to define the estimated class of each neighborhood. This may positively affect the quality of the data projection and enhance the performance in the subsequent classification task. Moreover, by means of the δ parameter L-Isomap$_{LS}$ is capable of properly tuning the number of distinguished landmarks. This may represent a potential advantage over L-Isomap$_{SC}$ especially when dimensionality reduction is applied to data sets affected by noise or outliers.

Compared to the original Isomap algorithm, L-Isomap$_{LS}$ exhibited a mild degradation of the classification performance, except for the Wine data set on which it provided the highest prediction accuracy. The slight loss in accuracy, however, was counterbalanced by the computational saving entailed by L-Isomap$_{LS}$ for projecting the data in the reduced space. Benefits in terms of computational effort clearly grew with the size of the data set. As an example,

Table 2. Classification results of ten times 3-fold cross-validation. For each data set, the average accuracy is indicated in the first row. The standard deviation and the average percentage of training points selected as landmarks are reported in the second and the third row, respectively.

	Method			
Data set	1NN	Isomap	L-Isomap$_{SC}$	L-Isomap$_{LS}$
Spectf	0.691	0.818	0.798	0.813
	0.021	0.016	0.015	0.013
			20.2	9.7
Musk	0.841	0.886	0.861	0.876
	0.010	0.008	0.010	0.011
			18.4	23.6
Wine	0.508	0.527	0.515	0.533
	0.011	0.008	0.011	0.008
			9.1	2.5
Madelon	0.537	0.644	0.580	0.619
	0.007	0.005	0.006	0.003
			14.5	20.3
Waveform	0.729	0.776	0.754	0.770
	0.004	0.002	0.002	0.002
			6.4	5.1
Pageblocks	0.946	0.966	0.954	0.961
	0.001	0.001	0.001	0.001
			6.5	10.8

the two largest sets of data, Waveform and Pageblocks, were projected in approximately 40 seconds by Isomap and in less than 5 seconds by L-Isomap$_{LS}$. These preliminary results open the way to the use of L-Isomap$_{LS}$ as a valuable alternative to Isomap for the embedding of large data sets, being able to produce effective low-dimensional representations with considerable savings in the computing time.

5 Conclusions and Future Extensions

The paper presents a novel method for landmark selection to be framed within the Isomap algorithm for nonlinear dimensionality reduction in the context of classification. It relies on a mixed-integer problem aimed at finding a set of landmarks which are representative of dense regions of points in the input space which mostly contain samples of the same class. The optimization model is solved via an approximation algorithm based on Lagrangian relaxation with subgradient method. Computational experiments performed on six data sets empirically

demonstrated the potential of the proposed method in terms of classification accuracy and computational effort.

In light of the promising results on the benchmark data sets the present study could be extended in several directions. First, it would be worthwhile to extend the evaluation of the proposed method to massive data sets containing several thousands of samples. Furthermore, other optimization models for landmark selection could be developed and investigated. Finally, it would be useful to analyse the effectiveness of alternative heuristic procedures for solving the proposed mixed-integer problem.

References

1. Jolliffe, I.T.: Principal component analysis. Springer, New York (1986)
2. Cox, T.F., Cox, M.A.A.: Multidimensional Scaling. Chapman and Hall, London (1994)
3. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323 (2000)
4. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326 (2000)
5. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation 15, 1373–1396 (2003)
6. Park, H.: ISOMAP induced manifold embedding and its application to Alzheimer's disease and mild cognitive impairment. Neuroscience Letters 513, 141–145 (2012)
7. Orsenigo, C., Vercellis, C.: Linear versus nonlinear dimensionality reduction for banks' credit rating prediction. Knowledge-Based Systems 47, 14–22 (2013)
8. Li, B., Zheng, C.H., Huang, D.S.: Locally linear discriminant embedding: An efficient method for face recognition. Pattern Recognition 41, 3813–3821 (2008)
9. Jafari, A., Almasganj, F.: Using Laplacian eigenmaps latent variable model and manifold learning to improve speech recognition accuracy. Speech Communication 52, 725–735 (2010)
10. Orsenigo, C., Vercellis, C.: A comparative study of nonlinear manifold learning methods for cancer microarray data classification. Expert Systems with Applications 40, 2189–2197 (2013)
11. de Silva, V., Tenenbaum, J.B.: Global versus local methods in nonlinear dimensionality reduction. In: Advances in Neural Information Processing Systems, vol. 15, pp. 705–712 (2003)
12. Shi, L., He, P.-L., Liu, E.: An Incremental Nonlinear Dimensionality Reduction Algorithm Based on ISOMAP. In: Zhang, S., Jarvis, R.A. (eds.) AI 2005. LNCS (LNAI), vol. 3809, pp. 892–895. Springer, Heidelberg (2005)
13. Iranmanesh, S.M., Mohammadi, M., Akbari, A., Nassersharif, B.: Improving Detection Rate in Intrusion Detection Systems Using FCM Clustering to Select Meaningful Landmarks in Incremental Landmark Isomap Algorithm. In: Zhou, Q. (ed.) ICTMF 2011. CCIS, vol. 164, pp. 46–53. Springer, Heidelberg (2011)
14. Gu, R.J., Xu, W.B.: An Improved Manifold Learning Algorithm for Data Visualization. In: Proc. of the 2006 International Conference on Machine Learning and Cybernetics, pp. 1170–1173 (2006)
15. Lei, Y.-K., Xu, Y., Zhang, S.-W., Wang, S.-L., Ding, Z.-G.: Fast ISOMAP Based on Minimum Set Coverage. In: Huang, D.-S., Zhang, X., Reyes García, C.A., Zhang, L. (eds.) ICIC 2010. LNCS, vol. 6216, pp. 173–179. Springer, Heidelberg (2010)

16. Lei, Y.K., You, Z.H., Dong, T., Jiang, Y.X., Yang, J.A.: Increasing reliability of protein interactome by fast manifold embedding. Pattern Recognition Letters 34, 372379 (2013)
17. de Silva, V., Tenenbaum, J.B.: Sparse multidimensional scaling using landmark points. Technical Report, Stanford University (2004)
18. Orsenigo, C., Vercellis, C.: An effective double-bounded tree-connected Isomap algorithm for microarray data classification. Pattern Recognition Letters 33, 9–16 (2012)
19. Geng, X., Zhan, D.C., Zhou, Z.H.: Supervised nonlinear dimensionality reduction for visualization and classification. IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics 35, 1098–1107 (2005)
20. Orsenigo, C., Vercellis, C.: Kernel ridge regression for out-of-sample mapping in supervised manifold learning. Expert Systems with Applications 39, 7757–7762 (2012)
21. Fisher, M.L.: The Lagrangian relaxation method for solving integer programming problems. Management Science 27, 1–18 (1981)
22. Burak Eksioglu, B., Demirerb, R., Capar, I.: Subset selection in multiple linear regression: a new mathematical programming approach. Computers & Industrial Engineering 49, 155–167 (2005)
23. Bache, K., Lichman, M.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine (2013), `http://archive.ics.uci.edu/ml`

Rough *c*-Regression Based on Optimization of Objective Function

Yasunori Endo[1], Akira Sugawara[2], and Naohiko Kinoshita[2]

[1] Faculty of Engineering, Information and Systems
University of Tsukuba
Tennodai 1-1-1, Tsukuba, Ibaraki, 305-8573, Japan
endo@risk.tsukuba.ac.jp
[2] Graduate School of Systems and Information Engineering
University of Tsukuba
Tennodai 1-1-1, Tsukuba, Ibaraki, 305-8573, Japan
{s1320626,s1220594}@u.tsukuba.ac.jp

Abstract. Clustering which is one of the pattern recognition methods is a technique automatically classifying data into some clusters. Various types of clustering are divided broadly into hierarchical and non-hierarchical clustering and crisp and fuzzy set theories have been applied to non-hierarchical clustering. Recently, clustering based on rough set theory has been attracted. Rough clustering represents a cluster by using two layers, i.e., upper and lower approximations. This paper proposes a *c*-regression method based on rough set representation which does regression analysis and clustering at the same time. Moreover, its effectiveness is shown through numerical examples.

1 Introduction

Computer system data has become large-scale and complicated in recent years due to progress in hardware technology, and the importance of data analysis techniques has been increasing accordingly. Clustering, which means a data classification method without any external criterion, has attracted many researchers as a significant data analysis technique.

Hathaway et al. proposed Hard and Fuzzy *c*-Regression (HCR and FCR) [1], which are clustering methods based on conventional regression model. With HCR and FCR, linear regression models are derived and belongingness or the membership grade of each object to each regression model is calculated. That is, those algorithms execute regression and clustering at same time.

FCR is a fuzzified HCR and fuzzy set representation plays very important role in FCR. Fuzzy set representation allows that an object belongs to two or more clusters. The belongingness is represented as a real value in a unit interval $[0, 1]$. Therefore, fuzzy set representation can be regarded as more flexible than HCR.

On the other hand, it is pointed out that "the fuzzy degree of membership may be too descriptive for interpreting clustering results." [2] In such cases, rough set representation is a more useful and powerful tool [3,4]. The basic concept of the rough set representation is based on two definitions of lower and upper approximations of a set.

V. Torra et al. (Eds.): MDAI 2013, LNAI 8234, pp. 272–283, 2013.

The lower approximation means that "an object surely belongs to the set" and the upper one means that "an object possibly belongs to the set". Clustering based on rough set representation could provide a solution that is less restrictive than conventional clustering and more descriptive than fuzzy clustering [5,2], and therefore the rough set based clustering has attracted increasing interest of researchers [6,7,8,9,10,11,2].

This paper proposes new clustering algorithms based on regression analysis and rough set representation and evaluate the algorithms through numerical examples.

2 Rough Sets

Let U be the universe and $R \subseteq U \times U$ be an equivalence relation on U. R is also called indiscernibility relation. The pair $X = (U, R)$ is called approximation space. If $x,\ y \in U$ and $(x, y) \in R$, we say that x and y are indistinguishable in X.

Equivalence class of the relation R is called elementary set in X. The family of all elementary sets is denoted by U/R. The empty set is also elementary in every X.

Each finite union of elementary sets in X is called composed set in X. The family of all composed sets is denoted by $\mathrm{Com}(X)$.

Since it is impossible to distinguish each element in an equivalence class, we may not be able to get a precise representation for an arbitrary subset $A \subset U$. Instead, any A can be represented by its lower and upper bounds. The upper bound $\overline{A}$ is the least composed set in X containing A, called the best upper approximation or, in short, upper approximation. The lower bound $\underline{A}$ is the greatest composed set in X containing A, called the best lower approximation or, briefly, lower approximation. The set $\mathrm{Bnd}(A) = \overline{A} - \underline{A}$ is called the boundary of A in X.

The pair $(\underline{A}, \overline{A})$ is the representation of an ordinary set A in the approximation space X, or simply a rough set of A. The elements in the lower approximation of A definitely belong to A, while elements in the upper bound of A may or may not belong to A.

From the above description of rough sets, we can define the following conditions for clustering:

(C1) An object x can be part of at most one lower approximation.
(C2) If $x \in \underline{A}$, $x \in \overline{A}$.
(C3) An object x is not part of any lower approximation if and only if x belongs to two or more boundaries.

3 *c*-Regression

In this section, we explain regression analysis and its error evaluation. Next, we show some representative methods of c-regression.

3.1 Regression

Regression is a way to obtain a regression model which presents the best relation between given variables x and y.

$x = (x^1, \ldots, x^p) \in R^p$, $y \in R$, and $\beta \in R^{p+1}$ mean independent variable, dependent variable, and regression coefficient, and it is assumed that objects $(x_1, y_1), \ldots, (x_n, y_n)$ are given. By using a regression model $f(x;\beta)$, the object (x_k, y_k) is denoted by

$$y_k = f(x_k;\beta) + \varepsilon_k.$$

ε_k means an error between the regression model and each dependent variable y_k. In regression analysis, a regression model which minimizes the error ε_k is derived. We can consider various functions as $f(x;\beta)$. In this paper, we consider the linear regression model.

In c-regression, i regression models $f(x,\beta_i)$ $(i = 1, \ldots, c)$ are considered and each cluster C_i is represented by the i-th regression model. The regression model is defined as follows:

$$f(x;\beta) = \sum_{j=1}^{p} \beta_i^j x^j + \beta_i^{p+1}.$$

By putting $z = (x^1, \ldots, x^p, 1)^T$, we can rewrite the above equation as follows:

$$f(x;\beta_i) = z^T \beta_i.$$

$(\bullet)^T$ means transposition.

The clustering problem is which object x_k belongs to which cluster C_i.

3.2 Error Evaluation

There are some approaches to evaluate errors between each pair (x_k, y_k) and regression models. Two of most popular approaches are least square deviation (LS) and least absolute deviation (LAD). LS minimizes $\sum_{k=1}^{n} (d_{ki})^2$ and LAD minimizes $\sum_{k=1}^{n} |d_{ki}|$. Here,

$$d_{ki} = y_k - f(x_k;\beta_i).$$

From here, LS-(name of the method) and LAD-(name of the method) mean the method with LS and LAD as error evaluation, respectively.

3.3 RKR

Peters proposed RKR (Rough k-Regression) [10] which is not based on optimization of objective function. RKR is inspired by RKM (Rough k-Means) by Lingras [2].

Algorithm 1 (RKR)

RKR1 *Give initial lower and upper approximations.*
RKR2 *Calculate the optimal regression coefficients.*
RKR3 *Update lower and upper approximations.*
RKR4 *If the stop criterion satisfies, finish. Otherwise back to* **RKR2**.

LS-RKR

β_i is calculated as follows:

$$\beta_i = \left(\sum_{k=1}^{n} u_{ki} z_k z_k^T \right)^{-1} \sum_{k=1}^{n} u_{ki} y_k z_k,$$

where

$$u_{ki} = \begin{cases} \underline{w}, & (x_i \in \underline{C}_i) \\ \overline{w}. & (x_i \in \mathrm{Bnd}(C_i)) \end{cases}$$

$\underline{w}$ and $\overline{w}$ are given constants and $\underline{w} + \overline{w} = 1$.

Here, we introduce the notations ν_{ki} and u_{ki}. ν_{ki} and u_{ki} means the belongingness of x_k to $\underline{C}_i$ and $\mathrm{Bnd}(C_i)$, respectively. Those are calculated as follows:

$$\nu_{ki} = \begin{cases} 1, & ((T_k = \phi) \wedge (i = p)) \\ 0, & (\text{otherwise}) \end{cases}$$

$$u_{ki} = \begin{cases} 1, & ((T_k \neq \phi) \vee ((i = p) \wedge (i \in T_k))) \\ 0. & (\text{otherwise}) \end{cases}$$

Here,

$$p = \arg\min_i |d_{ki}|,$$

$$T_k = \left\{ i \mid \frac{|d_{ki}|}{|d_{kp}|} \leq \text{threshold}.\ i \neq p,\ i = 1, \ldots, c \right\}$$

threshold is an arbitrary constant.

LAD-RKR

The optimal solution to β_i can be calculated from solving the following linear programming problem for each i:

$$\sum_{k=1}^{n} u_{ki} r_{ki} \to \min, \tag{1}$$

$$\text{s.t.} \quad y_k - f(x_k; \beta_i) \leq r_{ki},$$

$$y_k - f(x_k; \beta_i) \geq -r_{ki},$$

$$r_{ki} \geq 0, \quad (k = 1, \ldots, n)$$

where

$$u_{ki} = \begin{cases} \underline{w}, & (x_i \in \underline{C}_i) \\ \overline{w}. & (x_i \in \mathrm{Bnd}(C_i)) \end{cases}$$

$\underline{w}$ and $\overline{w}$ are given constants and $\underline{w} + \overline{w} = 1$.

How to calculate v_{ki} and u_{ki} is same as LS-RKR.

The two parameters $\underline{w}$ and threshold give The both methods LS-RKR and LAD-RKR flexibility. That is an advantage of the both methods that we can obtain various outputs.

On the other hand, we do not know how to determine the parameters because there is no standard of those parameters. Moreover, there is no evaluation criteria for their outputs in RKR. Outputs of many clustering algorithms based on optimization of objective function strongly depend on initial values, therefore we have to evaluate the outputs by certain criteria. Those matters are disadvantages of RKR. So, we will propose new clustering algorithms based on rough set representation and regression model, which overcome the above disadvantages of RKR.

4 Proposed Method: RCR

In this section, we propose RCR ((Rough c-Regression) based on optimization of objective function. RCR is based on optimization of objective function. In our algorithm, the number of parameters is one and the objective function can be used as evaluation criteria for the outputs.

The flow of algorithms of LS-RCR (LS-Rough c-Regression) and LAD-RCR (LAD-Rough c-Regression) are same, so we call those algorithms as RCR in a lump. Here, $N = \{v_{ki} \mid k = 1, \ldots, n,\ i = 1, \ldots, n\}$ and $U = \{u_{ki} \mid k = 1, \ldots, n,\ i = 1, \ldots, n\}$.

Algorithm 2 (RCR)

RCR1. *Give initial lower and upper approximations and calculate the optimal regression coefficients.*
RCR2. *Calculate N and U which minimize the objective function with fixing β.*
RCR3. *Calculate β which minimizes the objective function with fixing N and U.*
RCR4. *If the stop criterion satisfies, finish. Otherwise back to* **RCR2**.

LS-RCR

The objective function of LS-RCR is defined as follows:

$$J(N, U, \beta) = \sum_{i=1}^{c} \sum_{k=1}^{n} (v_{ki}\underline{w} + u_{ki}\overline{w}) d_{ki}^2$$

The constraints are as follows:

$$\begin{aligned} & v_{ki} \in \{0, 1\}, \quad u_{ki} \in \{0, 1\}, \\ & \underline{w} + \overline{w} = 1, \\ & \sum_{i=1}^{c} v_{ki} = 0 \Rightarrow \sum_{i=1}^{c} u_{ki} \geq 2, \\ & \sum_{i=1}^{c} u_{ki} = 0 \Rightarrow \sum_{i=1}^{c} v_{ki} = 1. \end{aligned} \tag{2}$$

Those constraints obviously satisfy the above conditions **C1**, **C2** and **C3**. Actually, (2) is rewritten as

$$\sum_{i=1}^{c} \nu_{ki} = 0 \Rightarrow \sum_{i=1}^{c} u_{ki} = 2.$$

When an object belongs to some boundaries, its belonging to two boundaries makes the objective function minimize in comparison with three or more boundaries.

Now, we describe how to calculate the optimal solutions to ν_{ki} and u_{ki}. For each object x_k, we first assume that x_k belongs to the lower approximation of a cluster which corresponds to the closest regression model $f(x, \beta_p)$, that is,

$$p = \arg\min_i d_{ki}^2.$$

In this case, the objective function is calculated as follows:

$$J_\nu = \nu_{kp} \underline{w} d_{kp}^2.$$

Next, We assume that x_k belongs to two boundaries. In this case, we can find the closest regression model $f(x, \beta_p)$ and the second closest one $f(x, \beta_q)$, that is,

$$q = \arg\min_{i, i \neq p} d_{ki}^2.$$

The objective function is calculated as follows:

$$J_u = u_{kp} \overline{w} d_{kp}^2 + u_{kq} \overline{w} d_{kq}^2.$$

Finally, ν_{ki} and u_{ki} is calculated as follows:

$$\nu_{ki} = \begin{cases} 1, & ((J_\nu < J_u) \wedge (i = p)) \\ 0, & (\text{otherwise}) \end{cases}$$

$$u_{ki} = \begin{cases} 1, & ((J_\nu > J_u) \wedge ((i = p) \vee (i = q))) \\ 0. & (\text{otherwise}) \end{cases}$$

Next, we consider the optimal solution to β_i. From $\frac{\partial J}{\partial \beta_i^j} = 0$, we obtain

$$\beta_i = \left(\sum_{k=1}^{n} (\nu_{ki} \underline{w} + u_{ki} \overline{w}) z_k z_k^T \right)^{-1} \cdot \sum_{k=1}^{n} (\nu_{ki} \underline{w} + u_{ki} \overline{w}) y_k z_k.$$

LAD-RCR

The objective function of LAD-RCR is defined as follows:

$$J(N, U, \beta) = \sum_{i=1}^{c} \sum_{k=1}^{n} (\nu_{ki} \underline{w} + u_{ki} \overline{w}) |d_{ki}|.$$

The constraints are the same as LS-FCR.

The optimal solutions to v_{ki} and u_{ki} can be obtained by replacing d_{ki}^2 in LS-FCR to $|d_{ki}|$.

The optimal solution to β_i is calculated from solving the following linear programming problem:

$$\begin{aligned}
&\sum_{k=1}^{n}(v_{ki}\underline{w} + u_{ki}\overline{w})r_{ki} \to \min,\\
&\text{s.t.} \quad y_k - f(x_k;\beta_i) \le r_{ki},\\
&\qquad y_k - f(x_k;\beta_i) \ge -r_{ki},\\
&\qquad\qquad r_{ki} \ge 0. \quad (k = 1,\ldots,n)
\end{aligned}$$

5 Numerical Examples

5.1 Preparation

Evaluation of Outputs of RKR

RKR has no evaluation criteria, so we can not evaluate the outputs of RKR. Therefore, we consider the objective function of RCR and the following function based on the objective function of HCR as the evaluation criterion as follows:

$$J = \sum_{i=1}^{c}\sum_{k=1}^{n}\left(v_{ki}d_{ki}^2 + \frac{u_{ki}}{\sum_{j=1}^{c}u_{kj}}d_{ki}^2\right) \tag{3}$$

(Name of the method)-R and (name of the method)-H in the results means that the objective function of RCR and (3) are used as the evaluation criterion, respectively.

Datasets

We prepare two artificial datasets and one real dataset to compare the proposed methods with the conventional ones. All datasets are in two-dimensional Euclidean space. Artificial dataset 1 consists of two line shaped clusters and 20 messed noise objects, and each cluster has 100 objects (Fig. 1). Artificial dataset 2 consists of two line shaped clusters which are very close to each other and 15 noise objects at random, and each cluster has 15 objects (Fig. 2). The real dataset represents GDP and energy consumption of Asian countries from 1973 to 1992 (Fig. 3). Horizontal and vertical axes mean real GDP (10^9\$) and primary energy consumption (10^6ton), respectively.

Evaluation Method

As quantitative evaluation, we show the total sum of errors L, i.e.,

$$L = \sum_{k=1}^{n}\sum_{i=1}^{c}u'_{ki}\,(d_{ki})^2\,.$$

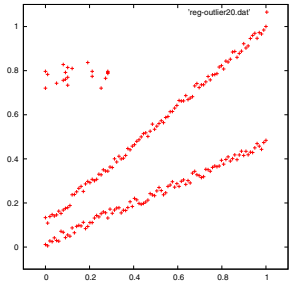


Fig. 1. Artificial dataset 1

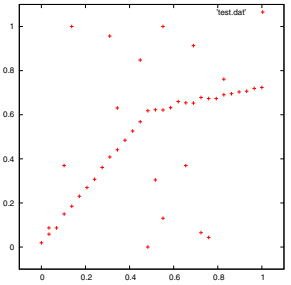


Fig. 2. Artificial dataset 2

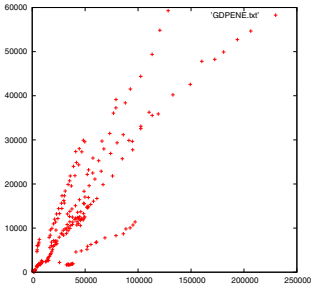


Fig. 3. GDP dataset

Here u'_{ki} means belongingness of x_k to C_i after assigning all x_k to their final clusters. Therefore, $u'_{ki} \in \{0, 1\}$. The noise objects are not included into the calculation. In case of RKR and RCR, we calculate L by two ways. One is that L is calculated without no noise objects which belong to certain boundaries, and another is that L is calculated with the object.

Parameters

In each algorithms, we prepare 1000 initial values and set parameters as $m = 3$, $c = 2$ for the artificial datasets, $c = 3$ for GDP dataset, $\underline{w} = 0.7$ and threshold $= 1.1$ in RKR, and $\underline{w} = 0.7$ or $\underline{w} = 0.8$ in RCR.

5.2 The First Artificial Dataset

Table 1 shows the total sum of errors by each methods. The values in parentheses in the table means total sum of errors between no noise objects which belong to certain boundaries and the closest regression model.

From the results, it is obvious that least absolute deviation is more robust than least square deviation. In particular, the proposed algorithms output better results. The reason is that the belongingness of noise objects to each clusters is obviously larger than $\overline{w}$ in case of RKR and RCR. For instance, the belongingness of noise objects to clusters in

LAD-FCR is about 0.5 and the belongingness in LAD-HCR is 1, while $\overline{w}$ in LAD-RCR is 0.3 or 0.2. In other words, the parameter $\overline{w}$ ($\underline{w}$) is very important role in RCR.

Table 1. Total sum of errors for Artificial dataset 1

Method	Total sum of errors
LS-HCR	1.934552
LS-FCR	2.120955
LS-RKR-H	1.747893 (1.955494)
LS-RKR-R	1.706390 (1.964700)
LS-RCR ($\underline{w} = 0.7$)	**1.651619 (1.956077)**
LS-RCR ($\underline{w} = 0.8$)	**1.139004 (2.013713)**
LAD-HCR	0.040824
LAD-FCR	0.043938
LAD-RKR-H	0.040824 (0.040824)
LAD-RKR-R	0.040824 (0.040824)
LAD-RCR ($\underline{w} = 0.7$)	**0.037886 (0.037886)**
LAD-RCR ($\underline{w} = 0.8$)	**0.033636 (0,033636)**

5.3 The Second Artificial Dataset

Table 2 shows the total sum of errors by each methods. The values in parentheses in the table means total sum of errors between no noise objects which belong to certain boundaries and the closest regression model. The results for this dataset also show that least absolute deviation is better than least square deviation.

Moreover, the results show the outputs by LS-RCR are similar to LS-HCR or S-FCR. The reason is that RCR has essentially three kinds of belongingness, ν_{ki}, u_{ki}, and $\underline{w}$. The larger $\underline{w}$ is, the more easily objects belong to boundaries, because RCR is based on optimization of objective function. Objects easily belong to lower approximations when $\underline{w} = 0.7$, and consequently, the output by RCR is similar to HCR. On the other hand, objects easily belong to boundaries when $\underline{w} = 0.8$, and the output by RCR is similar to FCR.

For the dataset, the best output is by LAD-RCR with $\underline{w} = 0.8$. The reason is that there is a few objects which must be considered when solving linear programming problem for each cluster and consequently the optimal solutions can be easily calculated. In addition, it is important that $\underline{w}$ has a big influence on the outputs by RCR.

5.4 GDP Dataset

We normalize original data into $[0, 1] \times [0, 1]$ and apply each algorithms. Table 3 shows the total sum of errors by each methods. The values in parentheses in the table means total sum of errors between no noise objects which belong to certain boundaries and the closest regression model. From the results, we can not see large difference between least square deviation and least absolute one. It can be considered to be better to apply least square deviation with considering calculation cost.

Table 2. Total sum of errors for Artificial dataset 2

Method	Total sum of errors
LS-HCR	0.373295
LS-FCR	0.156386
LS-RKR-H	0.373295 (0.373295)
LS-RKR-R	0.327269 (0.392041)
LS-RCR ($\underline{w} = 0.7$)	**0.327269 (0.392041)**
LS-RCR ($\underline{w} = 0.8$)	**0.125412 (0.157550)**
LAD-HCR	0.183568
LAD-FCR	0.049798
LAD-RKR-H	0.183568 (0.183568)
LAD-RKR-R	0.183568 (0.183568)
LAD-RCR ($\underline{w} = 0.7$)	**0.143662 (0.183568)**
LAD-RCR ($\underline{w} = 0.8$)	**0.001095 (0.001095)**

Table 3. Total sum of errors for GDP dataset

Method	Total sum of errors
LS-HCR	0.229599
LS-FCR	0.232759
LS-RKR-H	0.219475 (0.229734)
LS-RKR-R	0.219475 (0.219475)
LS-RCR ($\underline{w} = 0.7$)	**0.217145 (0.229859)**
LS-RCR ($\underline{w} = 0.8$)	**0.163177 (0.231176)**
LAD-HCR	0.238877
LAD-FCR	2.083101
LAD-RKR-H	0.238877 (0.238877)
LAD-RKR-R	0.230992 (0.239120)
LAD-RCR ($\underline{w} = 0.7$)	**0.223766 (0.239134)**
LAD-RCR ($\underline{w} = 0.8$)	**0.034578 (0.938140)**

Moreover, it is obvious that the outputs by LAD-FCR and LAD-RCR with $\underline{w} = 0.8$ are incorrect. We can consider three reasons. The first is that each object belongs to all clusters by LAD-FCR and consequently, the regression models gather. The second is that objects easily belong to boundaries by LAD-RCR and then, the outputs by LAD-RCR with $\underline{w} = 0.8$ is similar to LAD-FCR. The third is that objects are dense nearby origin, so it is difficult to classify those object clearly.

The cluster which is represented by a line with the largest slope can be regarded as a group of energy-consuming advanced countries, and the cluster which is represented by a line with the smallest slope can be regarded as a group of energy-conservation countries. The belongingness of each country depends on error evaluation. For instance, Nepal belongs to a group of middle countries with least square deviation, while it belongs to a group of energy-conservation countries with least absolute deviation. However, there is little difference between total sum of errors by least square deviation an least absolute one. Therefore, we can interpret it as the possibility that Nepal could be either.

6 Conclusion

This paper proposed new c-regression based on optimization of objective function and rough set representation, and evaluate the proposed algorithms through numerical examples.

We believe that LAD-RCR is useful in comparison with other algorithms. LAD-RCR reduces the influence of noise objects by making those objects belong to boundaries. Consequently, LAD-RCR derives better regression model. If boundaries is considered as noise clusters, RCR can be regarded as a kind of noise clustering.

When least absolute deviation is used instead of least square deviation, the RCR algorithm has an advantage of robustness against noise, while calculation cost increases because linear programming problems must be solved to calculate the optimal solutions. Moreover, there is another disadvantage that it is harder to obtain good results than least square deviation. We think that this causes the linear programming problems.

The parameter $\underline{w}$ has a big influence on the results in the proposed methods. Therefore, when the proposed methods are used, it is necessary to choose an adequate value of $\underline{w}$.

In future papers, we have to consider two matters. The first is about boundaries of RCR. In the proposed algorithms, each object belongs to just two boundaries when it does not belong to any lower approximations. Then, we have to evaluate the validity. The second is about way to choose error evaluation. There are no indication for it. The indication is necessary when we use RCR, so we have to consider it.

Acknowledgment. We would like to gratefully thank Professor Sadaaki Miyamoto of the University of Tsukuba, and Associate Professor Yuchi Kanzawa of Shibaura Institute of Technology for their advice. We would also like to sincerely thank Professor Yoji Uchiyama of the University of Tsukuba for his providing GDP Dataset.

References

1. Hathaway, R.J., Bezdek, J.C.: Switching regression models and fuzzy clustering. IEEE Trans. on Fuzzy Systems 1(3), 195–204 (1993)
2. Lingras, P., Peters, G.: Rough clustering. WIREs Data Mining and Knowledge Discovery 1(1), 64–72 (2011)
3. Pawlak, Z.: Rough Sets. International Journal of Computer and Information Sciences 11(5), 341–356 (1982)
4. Inuiguchi, M.: Generalizations of Rough Sets: From Crisp to Fuzzy Cases. In: Proceedings of Rough Sets and Current Trends in Computing, pp. 26–37 (2004)
5. Pawlak, Z.: Rough Classification. International Journal of Man-Machine Studies 20, 469–483 (1984)
6. Hirano, S., Tsumoto, S.: An Indiscernibility-Based Clustering Method with Iterative Refinement of Equivalence Relations. Journal of Advanced Computational Intelligence and Intelligent Informatics 7(2), 169–177 (2003)
7. Lingras, P., West, C.: Interval Set Clustering of Web Users with Rough K-Means. Journal of Intelligent Information Systems 23(1), 5–16 (2004)
8. Mitra, S., Banka, H., Pedrycz, W.: Rough-Fuzzy Collaborative Clustering. IEEE Transactions on Systems Man, and Cybernetics, Part B, Cybernetics 36(5), 795–805 (2006)

9. Maji, P., Pal, S.K.: Rough Set Based Generalized Fuzzy *C*-Means Algorithm and Quantitative Indices. IEEE Transactions on System, Man and Cybernetics, Part B, Cybernetics 37(6), 1529–1540 (2007)
10. Peters, G.: Rough clustering and regression analysis. In: Yao, J., Lingras, P., Wu, W.-Z., Szczuka, M.S., Cercone, N.J., Ślęzak, D. (eds.) RSKT 2007. LNCS (LNAI), vol. 4481, pp. 292–299. Springer, Heidelberg (2007)
11. Mitra, S., Barman, B.: Rough-fuzzy clustering: An application to medical imagery. In: Wang, G., Li, T., Grzymala-Busse, J.W., Miao, D., Skowron, A., Yao, Y. (eds.) RSKT 2008. LNCS (LNAI), vol. 5009, pp. 300–307. Springer, Heidelberg (2008)
12. Nakayama, Y., Miyamoto, S.: Least Square Deviations and Least Absolute Deviations in *c*-Regression Models. In: Proc. of Fuzzy System Symposium, vol. 17, pp. 583–586 (2001) (in Japanese)
13. Miyamoto, S., Umayahara, K., Nemoto, T., Takata, O.: Algorithm of Fuzzy *c*-Regression Based on Least Absolute Deviations. Journal of Japan Society for Fuzzy Theory and Systems 12(4), 578–587 (2000) (in Japanese)

Improving Automatic Edge Selection for Relational Classification

Cristina Pérez-Solà[1] and Jordi Herrera-Joancomartí[1,2]

[1] Dept. d'Enginyeria de la Informació i les Comunicacions
Universitat Autònoma de Barcelona
08193 Bellaterra, Catalonia, Spain
{cperez,jherrera}@deic.uab.cat
[2] Internet Interdisciplinary Institute (IN3) - UOC

Abstract. In this paper, we address the problem of edge selection for networked data, that is, given a set of interlinked entities for which many different kinds of links can be defined, how do we select those links that lead to a better classification of the dataset. We evaluate the current approaches to the edge selection problem for relational classification. These approaches are based on defining a metric over the graph that quantifies the goodness of a specific link type. We propose a new metric to achieve this very same goal. Experimental results show that our proposed metric outperforms the existing ones.

1 Introduction

Classification is the problem of assigning new instances (or samples) to a set of given categories or classes. It is a supervised machine learning task, where a set of already classified samples is used to infer a function that maps samples to classes. This inferred function is then used to try to classify new previously unseen samples, for which the class is unknown.

As with many other supervised machine learning tasks, samples are made of a vector of features or attributes that describes them. For instance, a sample describing a candidate for a job may contain the following features: candidate's GPA, the number of years of experience in a similar job, and the number of languages the candidate is able to fluently speak. For the samples in the training set, that is, the samples that will be used to train the classifier, the associated class label is also known. Following with the previous example, candidates that applied in the past for a similar position in the company can be classified depending on whether they got the job or they did not. For a new candidate, for which it is not known if he will be hired, we can use an inferred classifier over the training samples to make a prediction for the probability of him being hired, that is, we can try to predict its class label.

Many real world problems deal with samples that have a large number of descriptive features. Usually these many features include both redundant features and irrelevant ones. Trying to learn from this kind of datasets may be problematic. On one hand, the time needed to train the classifier increases with the

V. Torra et al. (Eds.): MDAI 2013, LNAI 8234, pp. 284–295, 2013.

size of the feature space. On the other hand, learning with irrelevant features may lead to overfitting, and thus decreased performance when evaluating unseen samples. It is thus interesting to try to reduce the number of features describing the samples before attempting to train the classifier. The problem of selecting a subset of features to work with from all the available ones is know as feature subset selection.

Networked data contains information about entities and the relationships between those entities. Networked data can be found almost everywhere: from authorship networks, that link authors sharing a common paper, to the now very popular Online Social Networks, where users are mainly linked by friendship. When working with networked data, a problem similar to the feature subset selection appears: the edge selection problem. Given a set of entities which can be linked by many kinds of edges describing different relationships between those entities, which of those relationships will lead to a better classification accuracy? Edge selection is specially critical if we study classifiers for homogeneous networks, that is, networks modeling just one type of edges.

In this paper, we focus on the edge selection problem for relational classification. When doing classification, the usual goal that we want to achieve is to maximize the accuracy of the built classifier. For this reason, in this paper we focus on selecting those edges that will maximize classification accuracy on unseen samples. We review the current proposals that tackle the problem and propose a new metric for edge selection. Then, we evaluate and compare the results of our proposal with the existing ones using a series of datasets already known by the relational learning community, and show that our proposal is able to better identify the edges leading to the best classification performance.

The rest of the paper is organized as follows. Section 2 reviews the State of the Art. Section 3 defines the problem that we want to deal with and specifies the notation that is then used through the rest of the paper. After that, Section 4 presents the proposed metric and Section 5 shows the experimental results supporting the usage of the proposed metric. Finally, Section 6 presents the conclusions and points out some lines for future work.

2 State of the Art

The traditional feature subset selection problem has been approached from two different perspectives. In the wrapper approach [1,2], the feature subset selection algorithm exists as a wrapper around the induction algorithm. The induction algorithm is taken into account during the feature selection process in order to evaluate the impact of choosing a specific set of features in classification accuracy. The induction algorithm is used as a black box, i.e., no knowledge on how the algorithm works is needed. Since exhaustively testing all possible subset selections may be impractical, the problem of feature selection is then translated into a search problem in the feature space. On the contrary, filter approaches [3,4,5] do not take into account the induction algorithm being used in the classification process. Instead, filter approaches try to evaluate the importance of the features from the data itself alone.

Some initial ideas about the problem of automatic edge selection appear in [6], where the authors identify the problem. They propose different methods to tackle the problem and try to compare their success on being able to identify the best edges for a series of datasets. However, this comparison is just preliminary work on the problem and lacks a systematic approach and a broad experimentation supporting the results.

First, the authors propose to compute the (edge) assortativity as defined previously in [7] for all the candidate edge sets E_l and select the edges that lead to the highest assortativity value. Assortativity is the tendency of the entities in a network to be connected to other entities that are like them in some way. When dealing with social networks, assortativity is usually known as homophily. Assortativity mixing can be computed according to an enumerative characteristic or a scalar characteristic. In the latter case, degree assortativity is of special interest because of its consequences on the structure of the network. The authors in [6] make use of the first alternative, assortativity according to an enumerative characteristic, where assortativity will be related to the class label of the nodes for which the classification will take place.

Because the assortativity metric as defined by [7] measures assortativity across edges and not across nodes, a node assortativity metric is also defined in [6]. This node assortativity is computed in a similar way, now using a matrix based on the node assortativity following previous works [8].

3 Problem Definition and Notation

Networked datasets are usually represented by graphs, where entities are mapped to nodes and edges describe relationships among them. Let us denote by $G = (V, E)$ the graph representing a given networked dataset. The set $V = \{v_i, \text{ for } i = 1, \cdots, n\}$ contains the nodes of the graph. On the other hand, E is the set of edges, pairs of different elements of V, representing the relationships between those nodes. Given a set of nodes V, many different sets of edges E_l can be defined based on different relationships arising in the studied dataset. Since we are dealing with weighted graphs, edges are pairs of vertexes with an associated weight, $e = (v_i, v_j, w_{ij})$ s.t. $(v_i, v_j) \in V \times V$ and $w_{ij} \in \mathbb{R}$. Because we are dealing with undirected graphs, symmetry is assumed, $e = (v_i, v_j, w_{ij}) = (v_j, v_i, w_{ji})$.

Let us denote by $\Gamma(v_i)$ the set of adjacent nodes of v_i, that is, $\Gamma(v_i) = \{v_j \in V \text{ s.t. } \exists e = (v_i, v_j) \in E\}$. We will use the words entities, nodes, or vertexes interchangeably through the rest of this paper, as we will do with edges, relationships, and links.

Classification problems consist on assigning samples of an input set into a given number of categories. We denote our category set as $\mathfrak{C} = \{\mathfrak{c}_k, \text{ for } k = 1, \cdots, m\}$. Then, the classification process will assign a value in $\mathfrak{C}$ to each node v_i from the input dataset. Let us define the ground of truth mapping, $cat : V \to \mathfrak{C}$, that assigns each node in the graph to its class.

Given a graph $G = (V, E)$, let us denote by $V_{train} \subset V$ the set of labeled samples used as the training set. We then define the test set V_{test} as the rest of the nodes, so $V_{test} = V \setminus V_{train}$.

Classification accuracy is defined as the percentage of samples the classifier is able to correctly predict:

$$accuracy(cat, cat_{pred}, V_{test}) = \frac{|v_i \in V_{test} \text{ s.t } cat_{pred}(v_i) = cat(v_i)|}{|V_{test}|}$$

where V_{test} is a given set of samples that have not been seen before by the classifier, cat the ground of truth mapping, and cat_{pred} the mapping that the classifier has learned by analyzing the samples on the training set V_{train}.

4 A Silhouette Based Metric for Edge Selection

As we already mention, we focus on the edge selection problem for classification scenarios. Given a set of vertexes V and a series of candidate edge sets E_l, for $l = 0, \cdots, L$, the edge selection problem that we want to tackle consists on selecting one of the E_l which leads to the best classification accuracy. Note that we are not interested in selecting the edges that better represent the data: we focus our goal in selecting those edges that will allow the classifier to achieve the best performance.

Figure 1 presents a toy example of the IMDB dataset (used in the experimental part of this paper). Nodes represent movies and edges describe relationships between these movies. A movie can be linked to another using three different kinds of edges, which indicate whether they share a director, a producer, or an actor. For instance, movies 1 and 2 have at least one director and one actor in common, whereas movies 3 and 4 have the same producer. Note that the homogeneous graph obtained when selecting only the *actor* edges is very different from the graph obtained when selecting just the *producer* edges (or even when selecting all the available edges, regardless of their type). Since the obtained graphs present important dissimilarities, it is thus interesting to study which of the alternatives will enable to better classify the nodes and how to identify it.

The idea of tackling the feature selection problem with a wrapper approach, by taking into consideration the induction algorithm used in the classification process, or with a filter approach, by focusing in the data alone to make the decision, can also be applied to the edge selection problem. In this paper, we evaluate different metrics to be used within the filter paradigm to create homogeneous networks with the straight forward methodology of selecting the edges presenting the highest value for the studied metric. That is, given a set of vertexes V and a series of candidate edge sets E_l, we define a metric $\overline{s}$ over the graph $G_l = (V, E_l)$, compute it for all the available edge sets $l = 0, \cdots, L$, and select the edge set that maximizes the metric value. The selected edge set $E_{l_{max}}$ is thus:

$$\{E_{l_{max}} \text{ s.t. } \overline{s}((V, E_{l_{max}})) \geq \overline{s}((V, E_l)) \, \forall l = 0, \ldots, L\}$$

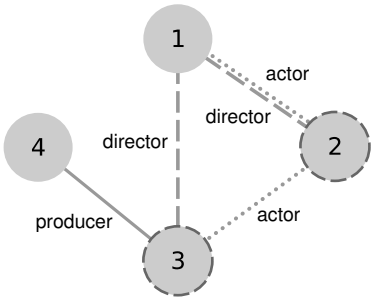


Fig. 1. An example of a graph with nodes from two different classes linked by relationships of three different kinds

4.1 General Overview

The proposed metric derives from two different ideas that have already been used in the past for similar purposes: aggregation operators and silhouette plots.

Aggregation operators have been proposed to model relational data [8,9]. Relational data contains information about entities and their relationships. These relationships include a huge amount of information that can not be discarded when analyzing the data. However, when using traditional machine learning techniques, dealing with these relationships supposes a challenge because it usually implies having to work with high-dimensional categorical attributes representing these relationships. Aggregation operators can be then used to create features representing this data.

Aggregation usually leads to information loss. For this reason, one of the characteristics that we have to take into account when selecting aggregation operators is the amount of information that is lost. Moreover, when creating aggregation operators for relational classification problems, we want the results of the aggregation to be that instances from the same class are similar while instances from different classes are distant.

Silhouettes [10] were created in the context of cluster analysis. A silhouette plot is a graphical display that represents how well samples in a cluster fit in that cluster by taking into account the distance between the sample and other samples in the same cluster, and the distance between the sample and samples in other clusters. Intuitively, the closer a sample is to others in the same cluster and the further it is from samples in other clusters, the better fit it is in that cluster.

Silhouette is mainly used for cluster validation, i.e., given a specific partition of the data, it is a useful tool to determine if the partition is good for that data or, on the contrary, a different number of clusters will lead to a better partitioning. However, silhouette plots can also be used in classification, where the number of classes is fixed. In this case, silhouette values are useful to assess how difficult a certain classification process will be. We take advantage of this characteristic, and try to evaluate how difficult the classification process will be depending on the selected edge sets.

4.2 Metric Detailed Description

Given a graph $G_l = (V, E_l)$ and a mapping cat between nodes in V and their categories in $\mathfrak{C}$, we compute the silhouette based metric as follows.

First, let us map each sample in V to its corresponding node class vector. The node's v_i class vector $CV(v_i)$ is defined as the vector of summed linkage weights to each of the classes in $\mathfrak{C}$:

$$CV(v_i)_k = \sum_{v_j \in \Gamma(v_i) \text{ s.t } cat(v_j)=\mathfrak{c}_k} w_{ij}$$

CV is thus a vector with m components (recall that $m = |\mathfrak{C}|$). Given the nodes' class vectors and a specific distance function $dist$, we can then compute the mean distance between a node and all samples in a given class $\mathfrak{c}_k$.

$$\overline{dist}(v_i, \mathfrak{c}_k) = \frac{\sum_{v_j \in V \text{ s.t } cat(v_j)=\mathfrak{c}_k, v_j \neq v_i} dist(v_i, v_j)}{|\{v_j \in V \text{ s.t } cat(v_j) = \mathfrak{c}_k, v_j \neq v_i\}|} \quad (1)$$

When $cat(v_i) \neq \mathfrak{c}_k$, the formula gives the mean distance between the sample and all samples in another class. When $cat(v_i) = \mathfrak{c}_k$, it provides the mean distance between the sample and other samples in the same class. Intuitively, the higher the first value and the lower the second one, the easier the sample will be to classify. Let us quantify this idea by defining the silhouette value for a given sample, v_i:

$$s(v_i) = \frac{\min_{\mathfrak{c}_k \in \mathfrak{C}, \mathfrak{c}_k \neq cat(v_i)} \{\overline{dist}(v_i, \mathfrak{c}_k)\} - \overline{dist}(v_i, cat(v_i))}{\max\{\min_{\mathfrak{c}_k \in \mathfrak{C}, \mathfrak{c}_k \neq cat(v_i)} \{\overline{dist}(v_i, \mathfrak{c}_k)\}, \overline{dist}(v_i, cat(v_i))\}}$$

This takes into account the mean distances from a sample to all the other classes, and consider the worst case by selecting the nearest class. It is also useful to define the silhouette value for a given class $\mathfrak{c}_k$, which is just the mean silhouette values of the samples in that class:

$$\overline{s}(\mathfrak{c}_k) = \frac{1}{|\{v_j \in V \text{ s.t } cat(v_j) = \mathfrak{c}_k\}|} \sum_{v_j \in V \text{ s.t } cat(v_j)=\mathfrak{c}_k} s(v_j)$$

Finally, the silhouette value for a whole graph is defined as the mean silhouette values of all its nodes:

$$\overline{s}(G) = \frac{1}{|V|} \sum_{v_i \in V} s(v_i)$$

Notice that Equation 1 is based on a distance metric between nodes. In Section 5, we experiment with three different distance functions: cosine distance $\overline{s}_{cos}$, Euclidean distance $\overline{s}_{Eucl}$, and Manhattan distance $\overline{s}_{Manh}$, and compare the results obtained for the different configurations.

5 Experimental Results

In this section, we evaluate the ability of the proposed metric to select the edges leading to the best classification accuracy. We also compare the results of using our silhouette based metric with those achieved when using other metrics in the literature. In order to make the results as general as possible, we make this evaluation using multiple datasets already known by the community and different relational classifiers that have been proposed in the past.

5.1 Datasets

The original datasets used in the experiments described in this paper can be found in [11] together with a more detailed description of their content. Table 1 presents a short summary of the characteristics of each dataset. A total of 14 different graphs can be created with this datasets.

For each original graph, we create additional test data by modifying the weights of the edges. The procedure to obtain different new edge sets for each graph can be described as follows:

1. We select a given graph $G = (V, E_0)$, i.e., a set of nodes V and the existing set of edges of the original graph, E_0.
2. We select a scoring function $score_{func}$ and apply it to every pair of nodes $(v_i, v_j) \in V \times V$ in the graph.
3. We compute the new weights w' by using the chosen scoring function over the edges of the graph $w'_{ij} = score_{func}(v_i, v_j) * w_{ij}$.
4. We obtain a new graph $G_l = (V, E_l)$, where E_l are the new edges with weights w'.

Table 1. Original datasets

Dataset	$\lvert\mathfrak{C}\rvert$	Edge set	$\lvert V\rvert$	$\lvert E_0\rvert$
WebKB Cornell	7	Cocitations	351	26832
WebKB Cornell	7	Links	351	1393
WebKB Texas	7	Cocitations	338	32988
WebKB Texas	7	Links	338	1002
WebKB Washington	7	Cocitations	434	30462
WebKB Washington	7	Links	434	1941
WebKB Wisconsin	7	Cocitations	354	33250
WebKB Wisconsin	7	Links	354	1155
IMDb	2	All	1441	48419
IMDb	2	Prodco	1441	20317
Industry	12	Pr	2189	13062
Industry	12	Yh	1798	14165
Cora	7	All	4240	71824
Cora	7	Cite	4240	22516

Table 2. Scoring functions

Label	Scoring function	Definition
E_1	Number of common neighbors	$score_{CN}(v_i, v_j) = \lvert\Gamma(v_i) \cap \Gamma(v_j)\rvert$
E_2	Jaccard Index	$score_{JI}(v_i, v_j) = \frac{\lvert\Gamma(v_i) \cap \Gamma(v_j)\rvert}{\lvert\Gamma(v_i) \cup \Gamma(v_j)\rvert}$
E_3	Adamic-Adar	$score_{AA}(v_i, v_j) = \sum_{v_k \in \Gamma(v_i) \cap \Gamma(v_j)} \frac{1}{\log(\lvert\Gamma(v_k)\rvert)}$
E_4	Preferential Attachment	$score_{PA}(v_i, v_j) = \lvert\Gamma(v_i)\rvert\lvert\Gamma(v_j)\rvert$
E_5	Clustering Coefficient	$score_{CC}(v_i, v_j) = \frac{\lvert\{e=(v_k,v_l) \in E \text{ s.t. } v_k, v_l \in \Gamma(v_i) \cap \Gamma(v_j)\}\rvert}{\lvert\Gamma(v_i) \cap \Gamma(v_j)\rvert(\lvert\Gamma(v_i) \cap \Gamma(v_j)\rvert - 1)}$

We use 5 scoring functions which are shown in Table 2. There are a few considerations to notice about these functions. First, all the scoring functions are defined for every pair of nodes of the graph, independently of whether an edge exists or not between the pair of nodes. Second, when evaluating the function for a pair of nodes, they take into account the existing relationships in their neighborhood but they are indifferent about the weights of these relationships. Third, the scoring functions are symmetric, in the sense that $score_{func}(v_i, v_j) = score_{func}(v_j, v_i)$. Finally, the chosen scoring functions have a common goal: they quantify, somehow, the strength of the relationship between the evaluated nodes. They do so by describing their neighborhood and measuring key aspects of its structure.

There are also a few considerations to take into account regarding the methodology for computing the new weights. On one hand, by directly multiplying the original weight by the result of the scoring function we ensure that no new edges are created. Recall that the scoring function is defined for every pair of nodes of the graph, whether they share a link or not. On the other hand, we allow all scoring functions to eliminate not relevant edges by assigning them a score of 0.

Following this procedure, we are able to obtain 14 vertex sets, each of one with 6 different sets of vertexes $E_0, E_1, \cdots, E_5$, a total of $14 \times 6 = 84$ different graphs. Through the rest of the paper, we will make use of all these 84 graphs in the experiments, and compare the results obtained when dealing with each of the 6 possible edge configurations for each dataset.

5.2 Experimental Setup

Since we want to evaluate the ability of different metrics to select the edge set that will lead to the highest classification accuracy, we need to assess the accuracy obtained when using the different edge sets. However, for a given dataset, classification accuracy is not a constant value since it is affected by the specific relational classifier used in the process. For this reason, we do the experiments with different relational classifiers.

We make use of the Netkit toolkit, which contains implementations for the most known classifiers. By using Netkit, we are able to systematically test different classifiers and compare the results. Classifiers in Netkit are comprised by a local classifier (LC), a relational classifier (RL), and a collective inference

procedure (CI). Each of the different modules[1] can be instantiated with many components. In our experiments, we allow the LC to be instantiated with either classpriors (`cp`) or uniform (`unif`); the RL component can be instantiated with Weighted-Vote Relational Neighbor Classifier (`wvrn`), its Probabilistic version (`prn`), the Class-distribution Relational Neighbor Classifier (`cdrn-norm-cos`), and Network-Only Bayes Classifier (`no-bayes`); the IC module can be specified with Relaxation Labeling (`relaxLabel`), Iterative Classification (`it`), or without any inference method (`null`). This give us $2 \times 4 \times 3 = 24$ different full classifiers. For the rest of the paper, we will use the term *full classifier* (*fc*) to refer to a specific instantiation of the three modules (LC-RC-CI).

In order to evaluate classification accuracy, we try to classify each of 86 graphs with all 24 *full classifiers*. For each experiment, that is, for a given graph and a given full classifier, we repeat the process of selecting new train and test sets 100 times and define the accuracy of the full classifier with respect to a given graph and a labeled ratio r as the mean of the accuracy over the test set of these 100 different runs. We repeated the process for different labeled ratios (train set sizes): 20%, 35%, 50%, and 65%.

5.3 Metric Comparison

In this section, we compare the performance of the different metrics with respect to selecting the best edge set, that is, the edge set that leads to the best classification accuracy. We evaluate the two assortativity variants as defined in [6], edge assortativity (A_E) and node assortativity (A_N), and compare them with the proposed silhouette based metric using as distance functions the cosine distance ($\overline{s}_{cos}$), euclidean distance ($\overline{s}_{Eucl}$), and Manhattan distance ($\overline{s}_{Manh}$).

We are interested in analyzing which metric is better correlated with classification accuracy. Given a set of nodes V and many different sets of edges E_l for $l = 0, \cdots, L$, the ideal metric should have a perfect positive correlation with classification accuracy, that is, the metric should return higher values when classification accuracy is high and lower values when classification accuracy is also low. With this kind of metric, we could simply select as the best edge set the one showing the highest value of the specific metric.

The Kendall τ rank correlation coefficient [12] is a measure of rank correlation, i.e., a measure of the similarity of the orderings of two measured quantities. Kendall's τ ranges from -1 to 1, with -1 expressing a negative correlation between the two variables (one increases with the decrease of the second), 0 expressing that the two variables are independent, and 1 expressing a perfect positive correlation between the two variables (one increases with the increase of the second). Since we are interested in deciding whether the function that describes the relationship between accuracy and the analyzed metric is monotonically increasing (without being concerned about finding the exact function that describes this relationship), we can use Kendall's τ to compare the different metrics.

[1] Readers can refer to the original Netkit paper [6] for a full explanation of these modules.

Table 3. Kendall's τ rank correlation coefficient between accuracy and each of the metrics (r=0.35)

Full classifier	A_N	A_E	$\overline{s}_{cos}$	$\overline{s}_{Eucl}$	$\overline{s}_{Manh}$
cprior-wvrn-it	0.3945	0.3764	**0.5829**	0.3649	0.3844
cprior-prn-it	**0.2878**	0.2238	0.2203	0.2020	0.2146
cprior-nobayes-it	0.3021	0.2536	0.2513	**0.3133**	0.3121
cprior-cdrn-norm-it	0.4616	0.4142	**0.5347**	0.4314	0.4280
cprior-wvrn-relaxLabel	0.2958	0.2616	**0.4624**	0.3810	0.3890
cprior-prn-relaxLabel	0.2734	0.2553	**0.3620**	0.2840	0.3104
cprior-nobayes-relaxLabel	0.2837	0.2536	0.2765	**0.3649**	0.3626
cprior-cdrn-norm-relaxLabl	0.3549	0.3133	**0.4739**	0.4360	0.4257
cprior-wvrn-null	0.3073	0.2742	**0.4796**	0.3787	0.3878
cprior-prn-null	0.2906	0.2691	**0.3574**	0.2817	0.3092
cprior-nobayes-null	0.2941	0.2570	0.2800	**0.3546**	0.3488
cprior-cdrn-norm-null	0.3612	0.3115	**0.4687**	0.4343	0.4251
unif-wvrn-it	0.3742	0.3555	**0.5374**	0.3647	0.3761
unif-prn-it	**0.2786**	0.2169	0.2042	0.1985	0.2065
unif-nobayes-it	0.2924	0.2450	0.2186	**0.2978**	0.2932
unif-cdrn-norm-it	0.4232	0.3735	**0.4504**	0.3769	0.3701
unif-wvrn-relaxLabel	0.3440	0.3247	**0.5221**	0.3775	0.3890
unif-prn-relaxLabel	0.3050	0.2823	**0.3419**	0.2628	0.2869
unif-nobayes-relaxLabel	0.3090	0.2651	0.2685	**0.3328**	0.3316
unif-cdrn-norm-relaxLabl	0.3377	0.3041	**0.4131**	0.3890	0.3775
unif-wvrn-null	0.3325	0.3155	**0.5106**	0.3890	0.3993
unif-prn-null	0.3101	0.2886	**0.3471**	0.2714	0.2920
unif-nobayes-null	0.3124	0.2616	0.2708	**0.3236**	0.3190
unif-cdrn-norm-null	0.3406	0.3012	**0.4033**	0.4045	0.3919

Table 3 shows the Kendall's τ rank correlation coefficient between classification accuracy and each of the proposed metrics for every full classifier[2], when setting the training set size to 35%.[3] The presented results take into account all the possible edge sets for each of the datasets, that is, the 6 different E_l for the 14 datasets. Each of the presented values represents the correlation between the metrics over these graphs and the 100−run mean accuracy (over the test sets) obtained when classifying those graphs. Even though datasets from very different nature are compared together, obtained τ coefficients are quite high. For all the possible configurations and analyzed metrics, τ coefficients are positive values, which denotes that there exists a positive correlation between the analyzed metrics and classification performance. Regarding the strength of this correlation, bold numbers denote the highest correlation achieved for the listed *fc*. Note that edge assortativity, A_E, is beaten for all configurations. On the other hand, node assortativity, A_N, presents better correlation with accuracy for two *fc* configurations using `prn` as the relational classifier module and iterative as the collective inference method. For the rest of the *fc* configurations, silhouette based metrics show better correlation with accuracy. Regarding the used distance function, the cosine distance exhibits the highest correlation for almost all *fc*. The exceptions are two *fc* for which A_N stands out and all the *fc* using network only Bayes as the relational module, for which using $\overline{s}_{Manh}$ leads to the best correlation.

[2] Since classification performance differs from one full classifier to another, it is thus interesting to analyse it w.r.t each full classifier.

[3] Due to space constraints, we omit the results for the other training set sizes. These results are similar to the presented ones.

Table 4. A detailed example of the usage of Kendall's τ coefficient using Cora-cite dataset classified with `cprior-wrn-it` with $r = 0.35\%$

Edge set	Accuracy	Accuracy rank	$\overline{s}_{cos}$	$\overline{s}_{cos}$ rank
E_0	0.715	1	0.525	1
E_1	0.521	4	0.373	5
E_2	0.348	6	0.206	6
E_3	0.524	3	0.383	4
E_4	0.515	5	0.391	3
E_5	0.673	2	0.453	2
τ	0.733			

Although using the proposed metric with euclidean distance, $\overline{s}_{Eucl}$, does not yield to the best correlation for any *fc*, it is important to notice that the results are very similar to those showed when using Manhattan distance, $\overline{s}_{Manh}$. The mean difference between the correlations showed for $\overline{s}_{Manh}$ and $\overline{s}_{Eucl}$ is just 0.0109, so there is no significant difference between using euclidean or Manhattan distances when evaluating the different edge sets.

Table 4 presents an example of how the τ coefficients are used for evaluating the different metrics. In the example, the displayed accuracy values are computed using `cprior-wrn-it` full classifier over the Cora-cite dataset (for a training set size of 0.35) for each of the available edge sets. We can also find the corresponding $\overline{s}_{cos}$ values. Taking into account the $\overline{s}_{cos}$ results, we will predict that the best edge set will be the original edge set, E_0, followed by E_5, E_4, E_3, E_1, and finally E_2. We can observe that the predictions are quite accurate, with E_0 being the best choice followed by E_5, and E_2 being the worst choice. However, there are three of the edge sets, E_1, E_3 and E_4 for which the predicted order is not exactly the same. Note that the three edge sets lead to very similar accuracy results, with less that 1% difference, and this is also reflected by the $\overline{s}_{cos}$ values, which are also very close. The obtained τ correlation coefficient for the set of accuracy and $\overline{s}_{cos}$ values is 0.733, denoting that there is a strong positive correlation between the two variables, although not a perfect one.

6 Conclusions and Further Research

We have presented a metric that is able to identify which edge set will lead to the best classification accuracy for a given classification problem over a relational dataset. Experimental results show that the proposed metric outperforms the ones being currently used for the same purpose.

Experimental results also indicate that classification accuracy, and thus correlation between accuracy and the studied metrics, strongly depends on the specific full classifier being used. At the same time, the full classifier obtaining

the best accuracy also depends on the specific dataset that is being classified, i.e., there is no configuration for the three modules of the classifier that works better than all the other for all datasets. All these facts suggest that using wrapper approaches, which take into consideration the classification algorithm, are more adequate than filter approaches for tackling the edge selection problem for relational classification.

This paper opens new lines for future work. It would be interesting to adapt and evaluate the wrapper approaches currently used on traditional non-relational domains to the now almost omnipresent networked datasets. Expanding this work to cover not only edge selection to create homogeneous networks but also to create heterogeneous graphs, where different kinds of entities and relationships are distinguished, is also a natural continuation of this work.

Acknowledgments. This work was partially supported by the Spanish MCYT and the FEDER funds under grants TIN2011-27076-C03 "CO- PRIVACY", TSI2007-65406-C03-03 "E-AEGIS", CONSOLIDER CSD2007-00004 "ARES", TIN2010-15764 "N-KHRONOUS", and grant FPU-AP2010-0078.

References

1. John, G.H., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In: Proceedings of the 11th Int. Machine Learning, pp. 121–129 (1994)
2. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence 97(1-2), 273–324 (1997)
3. Almuallim, H., Dietterich, T.G.: Learning with many irrelevant features. In: Proceedings of the 9th National Conf. on Artificial Intelligence, pp. 547–552 (1991)
4. Kira, K., Rendell, L.A.: The feature selection problem: traditional methods and a new algorithm. In: Proc. of the 10th Conf. on Artificial intelligence, pp. 129–134 (1992)
5. Cardie, C.: Using decision trees to improve case-based learning. In: Proceedings of the 10th Int. Conf. on Machine Learning, pp. 25–32. Morgan Kaufmann (1993)
6. Macskassy, S.A., Provost, F.: Classification in networked data: A toolkit and a univariate case study. J. Mach. Learn. Res. 8, 935–983 (2007)
7. Newman, M.E.J.: Mixing patterns in networks. Phys. Rev. E 67, 026126 (2003)
8. Perlich, C., Provost, F.: Distribution-based aggregation for relational learning with identifier attributes. Machine Learning 62(1-2), 65–105 (2006)
9. Perlich, C., Provost, F.: Aggregation-based feature invention and relational concept classes. In: Proc. of the 9th Int. Conf. on Knowledge Discovery and Data Mining, pp. 167–176 (2003)
10. Rousseeuw, P.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. J. of Computational & Applied Mathematics 20, 53–65 (1987)
11. Macskassy, S., Provost, F.: NetKit-SRL - network learning toolkit for statistical relational learning
12. Kendall, M., Gibbons, J.D.: Rank Correlation Methods, 5th edn. A Charles Griffin Title (September 1990)

Analyzing the Impact of Edge Modifications on Networks

Jordi Casas-Roma[1], Jordi Herrera-Joancomartí[2], and Vicenç Torra[3]

[1] Universitat Oberta de Catalunya (UOC), Barcelona, Spain
jcasasr@uoc.edu
[2] Universitat Autònoma de Barcelona (UAB), Bellaterra, Spain
jherrera@deic.uab.cat
[3] Artificial Intelligence Research Institute (IIIA), Spanish National Research Council (CSIC), Bellaterra, Spain
vtorra@iiia.csic.es

Abstract. Most of recent anonymization algorithms for networks are based on edge modification, i.e., adding and/or deleting edges on a network. But, no one considers the edge's relevance in order to decide which edges may be removed and which ones must be preserved. Considering edge's relevance can help us to improve data utility and reduce information loss. In this paper we analyse different measures for quantifying edge's relevance. Also, we present a new simple metric for edge's relevance on medium or large networks.

Keywords: Anonymization, Edge Relevance, Edge Modification, Networks, Graphs.

1 Introduction

In recent years, as more and more network data has been made publicly available, anonymization on network data has become an important concern. Backstrom et. al. [1] point out that the simple technique of anonymizing networks by removing the identities of the nodes before publishing the actual network does not guarantee privacy. To deal with this problem, some methods have been developed for network anonymization.

Most of these methods are based on edge modification. That is, methods that anonymize by modifying (adding and/or deleting) edges on a network. There are two basic approaches to anonymize a network via edge modification. First, randomization is the simplest way to anonymize a network by edge modification. Randomization methods are based on adding random noise on original data to hinder re-identification processes. Hay et al. [5] proposed a method to anonymize unlabelled networks which is based on removing and then adding false edges at random. Ying et al. [3] proposed a method which divides the network into blocks according to the degree sequence and implements modifications (by adding and removing edges) on the nodes at high risk of re-identification, not at random over the entire set of nodes. Both methods do not change the set of vertices and preserve the number of edges on anonymized networks.

V. Torra et al. (Eds.): MDAI 2013, LNAI 8234, pp. 296–307, 2013.

The second way to anonymize a network by edge modification is based on edge addition and deletion to meet desired objective functions. Among others, one widely adopted strategy is based on the concept of k-anonymity [2]. Several works use edge modifications to meet k-anonymity model. Among others [14] [8], Liu and Terzi [7] modify the network structure (by adding and removing edges) to ensure that all nodes satisfy the k-anonymity for the degrees of the nodes. Pei and Zhou [15] modify the network structure to meet k-anonymity for 1-neighbourhood sub-network of the objective nodes.

Edge modification techniques are widely used in network's anonymization. Nevertheless, none of these works consider the edge's relevance. Edge's relevance can help us to decide which edges can be removed or modified and which ones must be preserved. If we want to preserve the network properties, such as average distance, diameter, node centrality and more, we have to find the most relevant edges and preserve them from removing or modifying processes. Thus can lead anonymization methods to a better data utility and less information loss.

In this paper we use different metrics for edge's relevance in order to analyse the effect of edge deletion on network structure. We work with simple, undirected and unlabelled networks. We want to define a metric for edge's relevance which can help us to preserve the most important edges on network. This metric has to lead an edge modification process to remove the less important edges, keeping the basic network structural and spectral properties.

This paper is organized as follows. In Section 2, we review different metrics for edge's relevance. Section 3 presents our experimental framework, including structural and spectral metrics for network assessment and data sets used in our experiments. In Section 4, we show the experiments and discuss the results. Finally, in Section 5, we discuss conclusions and future work.

2 Metrics for Edge's Relevance

Let $G(V, E)$ be a simple network, where V is the set of nodes and E the set of edges in G. We use $v_i \in V$ to refer to node i and $e = (i, j)$ to refer to an undirected edge between nodes v_i and v_j. We define $n = |V|$ to denote the number of nodes and $m = |E|$ to denote the number of edges.

We consider the following metrics for quantifying edge's relevance:

Edge betweenness (EB) is defined as the number of the shortest paths that go through an edge in a network [4]. An edge with a high edge betweenness score represents a bridge-like connector between two parts of a network, and the removal of which may affect the communication between many pairs of nodes through the shortest paths between them. The edge betweenness of edge (i, j) is defined by:

$$EB_{(i,j)} = \frac{1}{n^2} \sum_{st} \frac{g_{st}^{(i,j)}}{g_{st}} \tag{1}$$

where $g_{st}^{(i,j)}$ is the number of shortest paths from node s to t that pass through edge (i, j), and g_{st} is the total number of shortest paths from node s to t.

Link salience (LS) [16] is defined as a linear superposition of all shortest path trees (SPTs). It quantifies the fraction of SPTs each link participates in. The salience of an edge (i, j) is computed as follows:

$$LS_{(i,j)} = \frac{1}{n} \sum_{k=1}^{n} T_{ij}(v_k) \tag{2}$$

where $T(v_k)$ is the shortest path tree (SPT) rooted at node v_k. $T(v_k)$ can be represented as a matrix with elements:

$$T_{ij}(v_k) = \begin{cases} 1 & \text{if } \sum_{l=1}^{n} \sigma_{ij}(v_l, v_k) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $\sigma_{ij}(v_l, v_k)$ is equal to 1 if edge (i, j) is on the shortest path from v_k to v_l, and otherwise is equal to 0. Despite the apparent similarity between edge betweenness and link salience, both quantities capture very different qualities of edges. Salience is insensitive to a position of an edge, acting as a uniform filter.

Calculating both above metrics of all the vertices on a network involves calculating the shortest paths between all pairs of vertices. This takes $\Theta(n^3)$ time with the Floyd-Warshall algorithm. On a sparse network, Johnson's algorithm may be more efficient, taking $\Theta(n^2 log(n) + nm)$ time. So, both metrics are not useful for large networks, since every time we remove an edge we must re-calculate all edges' values. Therefore, we present a new simple metric for quantifying edge relevance, called edge neighbourhood centrality.

Edge neighbourhood centrality (NC) of an edge (i, j) is defined as the fraction of nodes which are neighbours of v_i or v_j, but not of v_i and v_j simultaneously. The edge neighbourhood centrality is computed as follows:

$$NC_{ij} = \frac{(\Gamma(v_i) \cup \Gamma(v_j)) - (\Gamma(v_i) \cap \Gamma(v_j))}{n} \tag{4}$$

where $\Gamma(v_i)$ is the 1-neighbourhood of node v_i. Note that this metric can be computed on $\Theta(m)$ using the adjacency matrix representation of the network. An edge with high score is a bridge-like for neighbourhood nodes. All measures presented above are in range [0,1].

3 Experimental Set Up

Our experiments use edge betweenness (EB), link salience (LS) and edge neighbourhood centrality (NC) to evaluate all edges on each network. For each edge metric, we evaluate all edges of the network. One by one, we remove each edge from the network and then we compute the error introduced in the network by comparing some network characteristic metrics on original and on modified networks.

3.1 Datasets

Four different data sets are used in our experiments. Table 1 shows a summary of the network's main features. The networks considered are the following ones:

- Zachary's Karate Club [10] is a network widely used in the literature. The network shows the relationships among 34 members of a karate club.
- American College Football [4] is a network of American football games between Division IA colleges during regular season Fall 2000.
- Jazz Musicians [11] is a network of jazz musicians and their relationships.
- C.Elegans [12] is a list of edges of the metabolic network of C.elegans.

Table 1. Basic properties for selected networks

Network	Nodes	Edges	Av. degree	Av. distance	Diameter
Zachary's Karate Club	34	78	4.588	2.408	5
American College Football	115	613	10.661	2.508	4
Jazz Musicians	198	2,742	27.697	2.235	6
CElegans	453	2,025	8.940	2.663	7

3.2 Network Characteristic Metrics

We analyse the following structural and spectral metrics in order to quantify the noise introduced by edge deletion. Structural metrics are related to topological characteristics whereas spectral metrics are based on spectral characteristics.

Structural Metrics

Average distance (AD) is defined as the average of the distances between each pair of nodes in the network. It measures the minimum average number of edges between any pair of nodes. Formally, it is defined as:

$$AD(G) = \frac{\sum_{i,j=1}^{n} d(v_i, v_j)}{\binom{n}{2}} \tag{5}$$

where $d(v_i, v_j)$ is the length of the shortest path from v_i to v_j, meaning the number of edges along the path.

Diameter (D) is defined as the largest minimum distance between any two nodes in the network. Formally:

$$D(G) = max(d(v_i, v_j)), \forall i \neq j \tag{6}$$

Average distance and diameter evaluate the entire network as a unique score. We compute the error on these network metrics as follows:

$$\epsilon_{ij}(m) = |m(G) - m(G^{-})| \tag{7}$$

where m is one of the network characteristic metrics, G is the original network and G^- is the network without edge (i, j).

However, the following ones are node structural metrics, i.e., they evaluate specific structural properties for each node of the network.

Node Betweenness centrality (C_B) measures the fraction of shortest paths that go through each vertex. This measure indicates the centrality of a node based on the flow between other nodes in the network. A node with a high value indicates that this node is part of many shortest paths in the network, which will be a key node in the network structure. Formally, we define the betweenness centrality of a node v_i as:

$$C_B(v_i) = \frac{1}{n^2} \sum_{st} \frac{g_{st}^{v_i}}{g_{st}} \tag{8}$$

where $g_{st}^{v_i}$ is the number of shortest paths from s to t that pass through node v_i, and g_{st} is the total number of shortest paths from s to t.

Closeness centrality (C_C) is defined as the inverse of the average distance to all accessible nodes. Formally, we define the closeness centrality of a node v_i as:

$$C_C(v_i) = \frac{n}{\sum_{j=1}^{n} d(v_i, v_j)} \tag{9}$$

Transitivity or **Clustering coefficient** (T) is a measure widely used in the literature. The clustering of each node is the fraction of possible triangles that exist. For each node the clustering coefficient is defined by:

$$T(v_i) = \frac{2tri(v_i)}{deg(v_i)(deg(v_i) - 1)} \tag{10}$$

where $tri(v_i)$ is the number of triangles through node v_i and $deg(v_i)$ is the degree of node v_i.

And we compute the error on node metrics by:

$$\epsilon_{ij}(m) = \sqrt{\frac{1}{n}((g_1 - g_1^-)^2 + \ldots + (g_n - g_n^-)^2)} \tag{11}$$

where g_i is the value of the metric m for the node v_i of G and g_i^- is the value of the metric m for the node v_i of G^-.

Spectral Metrics

We also focus on two important eigenvalues of the network spectrum.

The largest eigenvalue of the adjacency matrix A is the value of λ_1, where λ_i are the eigenvalues of A and $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. The eigenvalues of A encode information about the cycles of a network as well as its diameter.

The second eigenvalue of the Laplacian matrix L is the value of μ_2, where μ_i are the eigenvalues of L and $0 = \mu_1 \leq \mu_2 \leq \ldots \leq \mu_m \leq m$.

The eigenvalues of L encode information about the tree structure of the network, where μ_2 is an important eigenvalue and it can be used to show how good the communities separate, with smaller values corresponding to better community structures.

4 Experimental Results

In this section we analyse the correlation based on Pearson correlation coefficient between each edge metrics and all network characteristic metrics. A high correlation points that removing edges with high score based on edge metric produces larger noise on modified network, while removing edges with low score produces fewer noise. On the other hand, a low correlation value suggests that removing edges with high or low score does not imply introducing more or less noise on modified data. We discuss in the next three sections the three edge metrics: edge betweenness, link salience, and edge neighbourhood centrality. Results on the four networks described in Section 3 are reported, and Figure 1 displays the results for the network Karate, which is the one that can be better visualized because it is the smallest dataset.

4.1 Edge Betweenness

Here we use the edge betweenness (EB) as a measure to quantify edge's relevance. As we can see on Table 2, average distance shows very high correlation for Karate and Football networks and high correlation for Jazz and CElegans. Correlation values between EB and diameter are low because diameter is much stabler than EB. Figure 1 (a) shows EB score for each edge (solid line) and the error on diameter (dashed line) introduced by this each removal on Karate network. Only twice the diameter has been modified, nevertheless it was when quiet high-scored edges were deleted from network.

Node betweenness presents very high correlation values, obtaining an average value of 0.97. On Figure 1 (b) we can see EB and node betweenness values. Clearly, EB is closely related to node betweenness centrality. On the other hand, closeness centrality presents lower values than node betweenness, except on Football network.

Results for transitivity, λ_1 and μ_2 are quite different. Pearson correlation values are low or very low on all datasets, except for transitivity on Football and λ_1 on Karate network. For example, correlation value between EB and transitivity is -0.10 on Karate (Figure 1 (c)), which means that there is not relation between EB and transitivity. I.e., removing edges with low EB score may introduce higher noise on network than removing edges with high EB score. This is because important EB-based edges are local bridges, so removing them we do not destroy any triangle, while low EB-based score edges are part of a triangle, so removing them we decrease the transitivity measure of the all involved nodes on this triangle.

Table 2. Pearson correlation values between EB and network characteristic metrics

Metric	**Karate**	**Football**	**Jazz**	**CElegans**	Average
Average Distance	0.83	0.93	0.79	0.73	0.82
Diameter	0.44	0.25	0.32	0.20	0.30
Node Betweenness	0.94	0.99	0.98	0.96	0.97
Closeness	0.44	0.95	0.37	0.25	0.50
Transitivity	-0.10	0.57	0.28	0.22	0.29
λ_1	0.26	-0.04	-0.11	0.42	0.21
μ_2	0.61	0.09	0.07	0.08	0.21
Average	0.52	0.55	0.42	0.41	0.47

4.2 Link Salience

The second measure for edge's relevance is link salience (LS). A small set of nodes are scored with high salience value, $LS \approx 1$. These nodes form the skeleton of the network whereas the others have low salience values, configuring a bi-modal distribution [16].

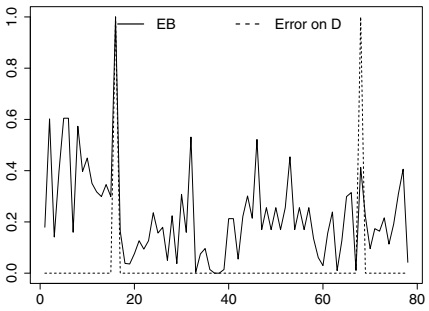
Table 3 presents the correlation values between LS and network characteristic metrics. Average distance shows good correlation values, but lower than the ones showed above. Once again, correlation values between LS and diameter presents low score. Node betweenness achieves high score, but lower than EB correlation on all networks. However, closeness gets a good score, even better than correlation with EB.

Like above metric, the correlation between LS and transitivity, λ_1 and μ_2 are not clear. They score very low values, suggesting that using salience as a edge's relevance does not guarantee better data utility and lower information loss. Figure 1 (d) exemplifies it.

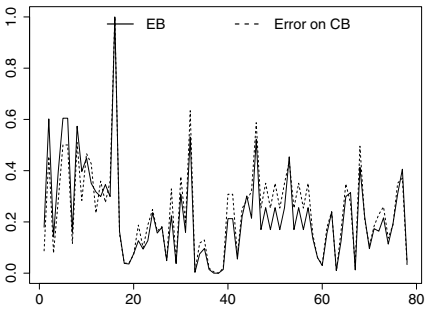
Table 3. Pearson correlation values between LS and network characteristic metrics

Metric	**Karate**	**Football**	**Jazz**	**CElegans**	Average
Average Distance	0.61	0.80	0.82	0.79	0.75
Diameter	0.20	0.10	0.13	0.22	0.16
Node betweenness	0.67	0.86	0.84	0.56	0.73
Closeness	0.45	0.86	0.53	0.33	0.54
Transitivity	0.17	0.53	0.36	0.32	0.34
λ_1	-0.10	-0.08	-0.17	0.01	0.09
μ_2	0.48	0.04	0.09	0.04	0.16
Average	0.38	0.47	0.42	0.32	0.40

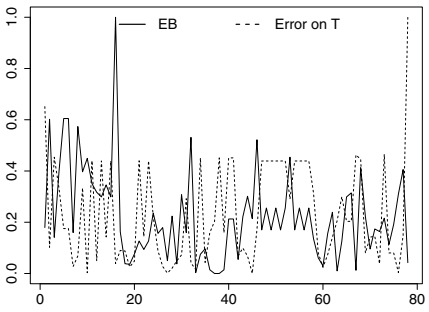
The correlation values between salience and the structural and spectral analysed metrics is lower, in general, than the ones obtained between EB and the same metrics.



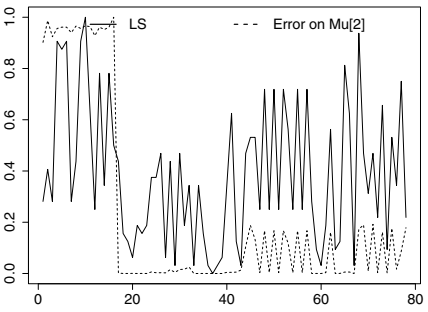
(a) EB and diameter values.



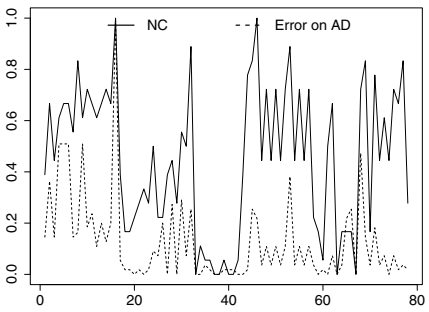
(b) EB and node betweenness centrality values.



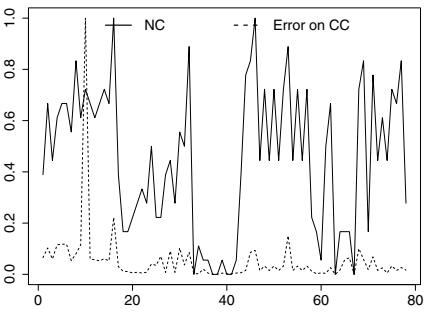
(c) EB and transitivity values.



(d) LS and μ_2 values.



(e) NC and average distance values.



(f) NC and closeness centrality values.

Fig. 1. Pairs of edge relevance metrics and network characteristics metrics on Karate network. EB stands for edge betweenness, LC stands for link salience and NC stands for edge neighbourhood centrality.

4.3 Edge Neighbourhood Centrality

Finally, we test our new simple metric for quantifying edge's relevance. As we can see on Table 4, average distance presents a moderate correlation on all networks, except Football where the correlation is not clear. Figure 1 (e) illustrates values for NC and average diameter on Karate network. Neither diameter shows correlation with NC.

Once again, node betweenness keep on moderate scores on all networks, expect Football. Closeness and transitivity do not achieve good results on Karate, Jazz nor CElegans, but they achieve quite better scores on Football network. Figure 1 (f) depicts values of NC and closeness. Finally, λ_1 and μ_2 present irregular values, showing weak correlation on some networks.

Table 4. Pearson correlation values between NC and network characteristic metrics

Metric	**Karate**	**Football**	**Jazz**	**CElegans**	Average
Average Distance	0.55	0.63	0.18	0.52	0.47
Diameter	0.24	0.07	0.03	-0.03	0.09
Node betweenness	0.75	0.78	0.36	0.46	0.59
Closeness	0.33	0.69	0.01	0.06	0.27
Transitivity	-0.06	0.51	-0.09	0.09	0.19
λ_1	0.41	-0.05	0.43	0.52	0.35
μ_2	0.48	0.21	0.35	0.19	0.31
Average	0.40	0.42	0.21	0.27	0.32

4.4 Summary

Average distance is a metric related to path lengths. Edge betweenness, as we have seen, is the edge metric which captures average distance the best, scoring the highest correlation values for this metric. Hence, removing important edge-betweenness-based edges affects it in a significant way.

Also we have seen that diameter is a stable metric with respect to network perturbation. So, it is difficult to correlate any of our edge's relevance metrics with diameter, although we have seen on Figure 1 (a) that diameter perturbation occurs when an important edge is removed from the network. It suggests that both EB and NC identify an important local bridge, and removing it produces and increment on several shortest paths along the network.

Node betweenness and closeness are widely used for clustering and community detection algorithms, so edge-modification-based anonymization algorithms must consider which edges remove or modify in order to reduce information loss and preserve data utility for clustering purposes. All metrics get a good correlation values, but the best one is EB. Clearly, EB and node betweenness are closely correlated. Some edges with high EB score are local bridges. So, removing them we introduce a high noise on node betweenness and closeness metrics. On the other hand, transitivity is not affected by removing local bridges and probably

this measure is affected by removing edges which participates on many triangles (that is, edges with low value of EB, LS and NC).

Finally, both spectral measures (λ_1 and μ_2) do not have high correlation with any edge's relevance metric. The best one is NC, which achieves moderate scores on all networks. But, using EB and LS the results are not clear. The noise introduced are not correlated with the score of these edge metrics. The eigenvalues of the adjacency matrix encode information about the cycles of a network as well as its diameter [6]. The maximum degree, chromatic number, clique number, and extend of branching in a connected network are all related to λ_1. The eigenvalues of L encode information about the tree-structure of the network [13].

Edge betweenness has been proved as a good edge's relevance metric. The values of correlation are larger than others on several cases, showing that removing or modifying edges with high values of edge betweenness introduce large noise on important structural and spectral metric.

Link salience, although is a good metric for visualizing and understanding network structure (skeleton), does not achieve the same results as edge betweenness. Despite it gets good results for structural metrics, it fails to achieve good results on spectral metrics.

Finally, neighbourhood centrality shows good results on all structural and spectral metrics. The results are not as good as the ones achieved with edge betweenness, but the complexity is very low for this metric. Therefore, it is a good metric to estimate edge's relevance on medium or large networks.

5 Conclusions

In this paper we have shown that anonymization processes should consider which edges might be removed or modified and which ones must be preserved, because the noise introduced on networks may be too large.

As we have seen in our experiments, edge betweenness is the best metric for quantifying edge's relevance. Edge betweenness identifies the most important edges, which may not be removed or modified, and the least important edges, which can be removed or modified. Edge's relevance can help edge-modification-based anonymization processes to achieve better results, raising data utility and reducing information loss. So, incorporating edge's relevance on edge modification processes can lead us to produce a more useful anonymized networks.

Although edge betweenness has proved to be a good metric to quantify edge's relevance, it is based on shortest path between all pair of nodes and it implies a high computational cost. Therefore, it is not a good metric for medium or large networks. To deal with this, we have proposed a new simple metric for edge's relevance, called edge neighbourhood centrality. It is very simple and can be applied to medium or large networks. This metric has showed good results on all structural and spectral metrics.

Many interesting directions for future research have been uncovered by this work. It may be interesting to extend this work to other network types, for

example, considering weighted or directed networks. Also, it may be interesting to create anonymization algorithms based on edge's relevance features. Then, these new algorithms can be compared with existing ones in terms of data utility, information loss and privacy.

Acknowledgements. This work was partially supported by the Spanish MCYT and the FEDER funds under grants TSI2007-65406-C03 "E-AEGIS", TIN2010-15764 "N-KHRONOUS", CONSOLIDER CSD2007-00004 "ARES", and TIN2011-27076-C03 "CO-PRIVACY".

References

1. Backstrom, L., Dwork, C., Kleinberg, J.: Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In: Proceedings of the 16th International Conference on World Wide Web, pp. 181–190. ACM, New York (2007)
2. Sweeney, L.: k-anonymity: a model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10(5), 557–570 (2002)
3. Ying, X., Pan, K., Wu, X., Guo, L.: Comparisons of randomization and k-degree anonymization schemes for privacy preserving social network publishing. In: Proceedings of the 3rd Workshop on Social Network Mining and Analysis, pp. 10:1-10:10. ACM, New York (2009)
4. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proceedings of the National Academy of Sciences of the United States of America 99(12), 7821–7826 (2002)
5. Hay, M., Miklau, G., Jensen, D., Weis, P., Srivastava, S.: Anonymizing Social Networks. Technical Report, University of Massachusetts Amherst, pp. 1–17 (2007)
6. Ying, X., Wu, X.: Randomizing Social Networks: A Spectrum Preserving Approach. In: SIAM Conference on Data Mining (SDM), pp. 739–750. SIAM, Atlanta (2008)
7. Liu, K., Terzi, E.: Towards identity anonymization on graphs. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pp. 93–106. ACM, New York (2008)
8. Hay, M., Miklau, G., Jensen, D., Towsley, D., Weis, P.: Resisting structural re-identification in anonymized social networks. Proceedings of the VLDB Endowment 1(1), 102–114 (2008)
9. Chakrabarti, D., Faloutsos, C.: Graph mining: Laws, generators, and algorithms. ACM Computing Surveys (CSUR) 38(1), 2:1-2:69 (2006)
10. Zachary, W.W.: An information flow model for conflict and fission in small groups. Journal of Anthropological Research 33(4), 452–473 (1977)
11. Gleiser, P., Danon, L.: Community structure in jazz. Advances in Complex Systems 6(04), 565–573 (2003)
12. Duch, J., Arenas, A.: Community identification using Extremal Optimization. Physical review. E, Statistical, Nonlinear, and Soft Matter Physics 72(2), 027104 (2005)
13. Seary, A.J., Richards, W.D.: Spectral methods for analyzing and visualizing networks: an introduction. In: National Research Council, Dynamic Social Network Modelling and Analysis: Workshop Summary and Papers, pp. 209–228 (2003)

14. Zou, L., Chen, L., Özsu, M.T.: K-Automorphism: A General Framework For Privacy Preserving Network Publication. In: Proceedings of the Thirtieth International Conference on Very Large Data Bases (VLDB 2009), vol. 2(1), pp. 946–957 (2009)
15. Zhou, B., Pei, J.: Preserving privacy in social networks against neighborhood attacks. In: Proceedings of the 2008 IEEE 24th International Conference on Data Engineering, pp. 506–515. IEEE Computer Society, Washington, DC (2008)
16. Grady, D., Thiemann, C., Brockmann, D.: Robust classification of salient links in complex networks. Nature Communications 3, 864:1–864:10 (2012)

Author Index