Linda C. van der Gaag (Ed.)

LNAI 7958

Symbolic and Quantitative Approaches to Reasoning with Uncertainty

12th European Conference, ECSQARU 2013
Utrecht, The Netherlands, July 2013
Proceedings

Springer

Lecture Notes in Artificial Intelligence 7958

Subseries of Lecture Notes in Computer Science

Linda C. van der Gaag (Ed.)

Symbolic and Quantitative Approaches to Reasoning with Uncertainty

12th European Conference, ECSQARU 2013
Utrecht, The Netherlands, July 8-10, 2013
Proceedings

Springer

Volume Editor

Linda C. van der Gaag
Utrecht University, Department of Information and Computing Sciences
Princetonplein 5, 3584 CC Utrecht, The Netherlands
E-mail: l.c.vandergaag@uu.nl

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-39090-6 e-ISBN 978-3-642-39091-3
DOI 10.1007/978-3-642-39091-3
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2013940605

CR Subject Classification (1998): I.2, F.4.1, F.3-4, I.2.3, I.2.4, H.3-4, J.3

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

The biennial ECSQARU conferences constitute a forum for advances in the theory and practice of reasoning under uncertainty. Contributions typically come from researchers who are interested in advancing technology and from practitioners using uncertainty techniques in real-world applications. The scope of the conference series encompasses fundamental issues, representation, inference, learning, and decision making in qualitative and numeric uncertainty paradigms.

Previous ECSQARU events were held in Marseille (1991), Granada (1993), Fribourg (1995), Bonn (1997), London (1999), Toulouse (2001), Aalborg (2003), Barcelona (2005), Hammamet (2007), Verona (2009), and Belfast (2011). The 12th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty was held in Utrecht, The Netherlands, from July 8 to July 10, 2013. The 44 papers presented at the conference were selected from 89 submitted manuscripts. Each submission underwent rigorous reviewing by at least three members of the ECSQARU Program Committee. From the 44 accepted papers, eight papers were selected for plenary presentation to honor their high quality.

In addition to the main program of paper presentations, ECSQARU 2013 featured a tutorial program on July 7; we are thankful to Hans L. Bodlaender, Fabio G. Cozman, Sébastien Destercke, and Jonathan Lawry for their most insightful tutorials. ECSQARU 2013 further included keynote talks by three outstanding researchers in the field:

Gert de Cooman (Ghent University, Ghent, Belgium):
Inference Under Exchangeability Using Sets of Desirable Gambles

A. Philip Dawid (University of Cambridge, Cambridge, UK):
Conditional Independence for Causal Reasoning

Simon Parsons (Brooklyn College, New York, USA):
The State of the Argument

We are most grateful for their highly inspiring presentations.

To conclude, we would like to thank the members of the Program Committee and the additional reviewers for their efforts; their reviews were most instrumental in selecting the best submissions for presentation during the conference. Thanks are also due to the student volunteers for letting the conference run smoothly. And, last but not least, we are most indebted to our sponsors for their financial support.

May 2013 Linda C. van der Gaag

Organization

ECSQARU 2013 was hosted by Utrecht University, Utrecht, The Netherlands, and organized by members of the Department of Information and Computing Sciences, Faculty of Science.

Executive Committee

Conference Chair	Linda C. van der Gaag
Program Chairs	Jan Broersen Linda C. van der Gaag Silja Renooij

Organizing Committee

Janneke H. Bolt
Jan Broersen
Silja Renooij (Chair)
Merel T. Rietbergen
Steven P.D. Woudenberg (Chair)

Program Committee

Stefano Aguzzoli (Italy)
Leila Amgoud (France)
Alessandro Antonucci (Switzerland)
Nahla Ben Amor (Tunisia)
Boutheina Ben Yaghlane (Tunisia)
Salem Benferhat (France)
Philippe Besnard (France)
Concha Bielza Lozoya (Spain)
Martin Caminada (UK)
Giulianella Coletti (Italy)
Fabio Gagliardi Cozman (Brazil)
Fabio Cuzzolin (UK)
Luís M. de Campos (Spain)
Cassio de Campos (Switzerland)
Gert de Cooman (Belgium)
Sébastien Destercke (France)
Antonio Di Nola (Italy)
Didier Dubois (France)
Zied Elouedi (Tunisia)
Francesc Esteva (Spain)
Hélène Fargier (France)
Angelo Gilio (Italy)
Lluís Godo (Spain)
Andreas Herzig (France)
Anthony Hunter (UK)
Gabriele Kern-Isberner (Germany)
Sébastien Konieczny (France)
Rudolf Kruse (Germany)
Christophe Labreuche (France)
Jérôme Lang (France)
Pedro Larrañaga (Spain)
Jonathan Lawry (UK)
Jan Lemeire (Belgium)
Philippe Leray (France)
Churn-Jung Liau (Taiwan)
Weiru Liu (UK)
Peter Lucas (The Netherlands)
Thomas Lukasiewicz (UK)
Jianbing Ma (UK)
Pierre Marquis (France)
Vincenzo Marra (Italy)
Andres R. Masegosa (Spain)

Thomas Meyer (South Africa)
Enrique Miranda (Spain)
Serafín Moral (Spain)
Kedian Mu (P.R. China)
Kristian G. Olesen (Denmark)
Ewa Orlowska (Poland)
Odile Papini (France)
Simon Parsons (USA)
Jose M. Peña (Sweden)
Henri Prade (France)
Erik Quaeghebeur (Belgium)
Antonino Rotolo (Italy)
Giovanni Sartor (Italy)
Torsten Schaub (Germany)
Steven Schockaert (UK)
Claudio Sossai (Italy)
Leon van der Torre (Luxembourg)
Matthias Troffaes (UK)
Barbara Vantaggi (Italy)
Bart Verheij (The Netherlands)
Jirka Vomlel (Czech Republic)

Additional Reviewers

Marco Baioletti
Libor Behounek
Christian Braune
Jasper De Bock
Ad Feelders
Marcelo Finger
Tommaso Flaminio
Brunella Gerla
Ken Halland
Pascal Held
Enrico Marchioni
Christian Moewes
Kody Moodley
Farid Nouioua
Davide Petturiti
Gavin Rens
Agnieszka Rusinowska
Fabian Schmidt
Marius Schneider
Nicolas Schwind
Piotr Wasilewski
Eric Würbel

Additional Support

Rita Jansen

Sponsors

Artificial Intelligence Journal
Benelux Association for Artificial Intelligence (BNVKI)
Department of Information and Computing Sciences, Utrecht University
HUGIN Expert
Netherlands Organisation for Scientific Research (NWO)
Netherlands Research School for Information and Knowledge Systems (SIKS)
Royal Netherlands Academy of Arts and Sciences (KNAW)
Utrecht City Council

Table of Contents

A Formal Concept View of Abstract Argumentation

Leila Amgoud and Henri Prade

IRIT University of Toulouse, 118 rte de Narbonne, Toulouse, France
{amgoud,prade}@irit.fr

Abstract. The paper presents a parallel between two important theories for the treatment of information which address questions that are apparently unrelated and that are studied by different research communities: an enriched view of formal concept analysis and abstract argumentation. Both theories exploit a binary relation (expressing object-property links, attacks between arguments). We show that when an argumentation framework rather considers the complementary relation does not attack, then its stable extensions can be seen as the exact counterparts of formal concepts. This leads to a cube of oppositions, a generalization of the well-known square of oppositions, between eight remarkable sets of arguments. This provides a richer view for argumentation in cases of bi-valued attack relations and fuzzy ones.

Keywords: argumentation, formal concept analysis, possibility theory, square of oppositions.

1 Introduction

Formal concept analysis [34, 29] exploits a binary relation that links objects and properties. This relation, called 'formal context', is usually a classical 2-valued one (i.e., an object has, or not, a property), but may be also a fuzzy relation [5–7] when properties may be a matter of degree. From this relation, the notion of 'formal concept' is defined as maximal sets of pairs made of a subset of objects and a subset of properties, such that each object in a subset has all the properties in the associated subsets, and the objects in a subset are the only ones to have all these properties, in the considered context. Formal concepts are characterized by a fixed-point equation through a Galois connection. A recent parallel [18] with possibility theory [19] has shown the interest of introducing operators in this setting other than the one underlying the notion of formal concept, which leads to consider other connexions as well [15, 22].

In a fully independent way, an abstract theory of argumentation [24] has been developed on the basis of a binary attack relation between arguments. This relation, generally a classical one, may also become fuzzy when one tries to model the strength of arguments [25]. The objective is then to determine noticeable subsets of arguments that in particular constitute stable extensions

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 1–12, 2013.

in the sense they are without internal conflict, and where each argument outside the extension is attacked by an argument of the extension.

The exploitation in each setting of a classical binary relation, which may be more generally fuzzy, may lead to wonder about a possible parallel between the two theories, and about their possible mutual enrichment. In the next section we restate the formal elements of the abstract theory of argumentation and emphasize the different existing relational equations. Then in Section 3 in the same spirit, we recall the basis of formal concept analysis enriched by the operators induced by the parallel with possibility theory. In Section 4, we make a parallel between the abstract theory of argumentation and formal concept analysis, which especially sheds light on the parallel between stable extension and formal concept. Section 5 provides an analysis in terms of opposition structures that help to get an organized view of different subsets of remarkable arguments. The concluding remarks briefly considers the case of fuzzy relations, and in particular suggests lines of research for extending abstract theory of argumentation to situations where attacks are weighted.

2 Argumentation

P. M. Dung [24], in a famous article which has raised considerable interest, has proposed to define an *argumentation system* as a pair (A, R) where A is a set of arguments, and R ($\neq \emptyset$) a binary relation over A, i.e., $R \subseteq A \times A$. Given two arguments $a \in A$ and $b \in A$, $(a, b) \in R$, or equivalently aRb, then means that a *attacks* b. An *argumentation system* (A, R) can then be seen as an oriented graph, where arguments are its nodes, and where the elements of R are the vertices. As can be seen the notion of argument, which intuitively corresponds in the logical view (see, e.g., [31]) to a minimal consistent set of formulas that in a given logical setting enable us to deduce a formula of interest, is here "abstractized", as well as the notion of attack (which amounts in practice to challenge a deduced formula, either directly, or by challenging one of the formulas appearing in the argument for establishing its conclusion). Dung's framework has been often used as a reference setting and as a starting point in many artificial intelligence works in argumentation until now.

A subset $S \subseteq A$ of arguments attacks an argument a

$$\text{if } \exists s \in S \text{ and } sRa.$$

A subset $S \subseteq A$ of arguments attacks a subset $S' \subseteq A$

$$\text{if } \exists s \in S \text{ and } \exists s' \in S' \text{ and } sRs'.$$

A subset S of arguments is *conflict free*

$$\text{if } \nexists (a, b) \in S \times S \text{ such as } aRb.$$

Given an argumentation system (A, R), a key question that naturally arises is the definition of *acceptable* subsets of arguments; an acceptable subset of arguments is called *extension*. Different forms of acceptability exist. A well-known one is the notion of *stable extension*.

A subset S of arguments without conflict is a *stable* extension if and only if

$$\forall a \notin S, \exists s \in S \text{ and } sRa.$$

In other words, a stable extension attacks all the arguments outside. Other forms of acceptability use the notion of *defense*. An argument $a \in A$ is defended by a subset of arguments S if and only if for each argument $b \in A$ that attacks a, $\exists s \in S$ such that sRb. A conflict-free subset S of arguments is an *admissible extension* if and only if each argument of S is defended by S. A stable extension is admissible.

One can then introduce remarkable sets associated with an argument a, or with a subset of arguments S in terms of attack or defense, which help to make the definitions more precise and to establish some properties :

- the set of arguments attacking a
  $Ra = \{s \in A | sRa\}$;
- the set of arguments attacked by a
  $aR = \{s \in A | aRs\}$;
- the set of arguments attacked by S
  $R^+(S) = \{a \in A | S \text{ attacks } a\}$
  $= \{a \in A | \exists s \in S, sRa\}$
  $= \{a \in A | S \cap Ra \neq \emptyset\}$;
- the set of arguments attacking S
  $R^-(S) = \{a \in A | a \text{ attacks} S\}$
  $= \{a \in A | \exists s \in S, aRs\}$
  $= \{a \in A | S \cap aR \neq \emptyset\}$;
- the set of arguments defended by S
  $Def(S) = \{a \in A | S \text{ defends } a\}$
  $= \{a \in A | \forall b \in A \text{ t.q. } bRa, \exists s \in S \text{ s.t. } sRb\}$
  $= \{a \in A | Ra \subseteq R^+(S)\}$.
  The set of arguments defended by S is indeed made of the arguments whose attackers are attacked by S.

It can be checked that

- S is *conflict-free* if and only if [1]
  $S \subseteq \overline{R^+(S)}$,
  where $\overline{T} = A \setminus T$. Indeed, the arguments that S attacks are then in $\overline{S}$ $(R^+(S) \subseteq \overline{S})$.
- S is a *stable extension* if and only if [24]

$$S = \overline{R^+(S)}. \tag{1}$$

This follows from above, and from the definition of stability that requires $\overline{S} \subseteq R^+(S)$. Note also that the set of arguments non attacked by S is equal to

$$\overline{R^+(S)} = \{a \in A | \forall s \in S, s\overline{R}a\}, \text{ i.e., we have}$$

$$\overline{R^+(S)} = \{a \in A | S \subseteq \overline{R}a\} \quad (2)$$

where $s\overline{R}a$ means that s does not attack a. One can then establish that:

- $Def(S) = \overline{R^+(\overline{R^+(S)})}$ [1].

  Indeed, applying Equation 2 one gets $\overline{R^+(\overline{R^+(S)})} = \{a \in A | \overline{R^+(S)} \subseteq \overline{R}a\}$,

  which provides the proof since $\overline{R}a = \overline{Ra}$, taking into account the definition of $Def(S)$.

Thus if S is a stable extension, Equation 1 holds, and then

$$Def(S) = S$$

In a stable extension, the arguments are thus defending themselves.

One can still establish that [8]

- S is an *admissible extension* if and only if

  $$S \subseteq Def(S) \cap \overline{R^+(S)}.$$

  Indeed, this is equivalent to $S \subseteq Def(S) \wedge S \subseteq \overline{R^+(S)}$, which indeed means that the arguments in S are both defended by S and non attacked by S (S is thus conflict-free). This condition can be still written

  $$\begin{aligned} & S \subseteq \{a \in A | \overline{R^+(S)} \subseteq \overline{R}a \wedge S \subseteq \overline{R}a\} \\ \Leftrightarrow\ & S \subseteq \{a \in A | (\overline{R^+(S)} \cup S) \subseteq \overline{R}a\} \\ \Leftrightarrow\ & S \subseteq \{a \in A | Ra \subseteq (R^+(S) \cap \overline{S})\}. \end{aligned}$$

- S is an *admissible extension* if and only if

  $$S \subseteq Def(S \cap \overline{R^-(S)}).$$

  Indeed, this condition guarantees that S is conflict-free, since it expresses that each argument in S is defended by an argument in S that does not attack S (we have $\overline{R^-(S)} = \{a \in A | S \subseteq a\overline{R}\}$). Indeed, if aRb with $(a, b) \in S^2$, b cannot be defended by c (i.e. cRa and thus $c \in R^-(S)$) with $c \in \overline{R^-(S)}$.
- An admissible extension S is said *complete* if and only if each argument which is defended by S is in S [24]. Thus S is complete if and only if S is admissible and $Def(S) \subseteq S$. Thus, we have

S is a *complete extension* if and only if

$$S = Def(S) \cap \overline{R^+(S)}.$$

Moreover, if S is a complete extension, then

$$S = Def(S).$$

3 Formal Concept Analysis

Formal concept analysis (FCA) [4, 34] provides a theoretical setting for the learning of hierarchies of concepts (from which association rules can be extracted). It starts with a *formal context* $\mathcal{K} = (\mathcal{O}, \mathcal{P}, \mathcal{R})$ where $\mathcal{R}$ is a binary relation completely defined between a set of objects $\mathcal{O}$ and a set of Boolean properties $\mathcal{P}$. Namely, $\mathcal{R} \subseteq \mathcal{O} \times \mathcal{P}$. A formal context is often visualized under the form of a table such that the presence of a cross ($\times$) (resp. its absence) in a cell indicates if an objet satisfies (resp. does not satisfy) the corresponding property.

Given an object x and a property y, let $R(x) = \{y \in \mathcal{P} \mid x\mathcal{R}y\}$ be the set of properties satisfied by object x ($x\mathcal{R}y$ means that x has property y) and let $R(y) = \{x \in \mathcal{O} \mid x\mathcal{R}y\}$ be the set of objects having property y. In FCA, one defines correspondences between the sets $2^{\mathcal{O}}$ and $2^{\mathcal{P}}$. These correspondences are called Galois derivation operators. The Galois operator, which is at the basis of FCA, here denoted $(.)^{\Delta}$ (for reasons made clear later), enables us to express the set of properties satisfied by *all* the objects in $X \subseteq \mathcal{O}$ as :

$$\begin{aligned} X^{\Delta} &= \{y \in \mathcal{P} \mid \forall x \in \mathcal{O}\ (x \in X \Rightarrow x\mathcal{R}y)\} \\ &= \{y \in \mathcal{P} \mid X \subseteq R(y)\} = \textstyle\bigcap_{x \in X} R(x) \end{aligned}$$

We can also express, in a dual manner, the set of objects satisfying all the properties in Y as :

$$\begin{aligned} Y^{\Delta} &= \{x \in \mathcal{O} \mid \forall y \in \mathcal{P}\ (y \in Y \Rightarrow x\mathcal{R}y)\} \\ &= \{x \in \mathcal{O} \mid Y \subseteq R(x)\} = \textstyle\bigcap_{y \in Y} R(y) \end{aligned}$$

The dual pair of operators $((.)^{\Delta}, (.)^{\Delta})$ applied respectively to $2^{\mathcal{O}}$ and to $2^{\mathcal{P}}$ constitutes a Galois connexion that enables the definition of formal concepts. A*formal concept* is a pair (X, Y) such as

$$X^{\Delta} = Y \text{ and } Y^{\Delta} = X.$$

In other words, X is the maximal set of objects satisfying all the properties already satisfied by all the objects in X. The set X (resp. Y) is called *extension* (resp. *intension*) of the concept. It can be shown that in an equivalent way, (X, Y) is a formal concept if and only if it is a maximal pair in the sense of set inclusion such as

$$X \times Y \subseteq \mathcal{R}.$$

The set of all the formal concepts is naturally equipped with an order relation (denoted $\preccurlyeq$) and defined as : $(X_1, Y_1) \preceq (X_2, Y_2)$ iff $X_1 \subseteq X_2$ (or $Y_2 \subseteq Y_1$). This set equipped with the order relation $\preccurlyeq$ forms a complete lattice $\mathfrak{B}(\mathcal{K})$. The operators *meet* and *join* in the lattice are described by the fundamental result due to Ganter and Wille [29] :

$$\bigwedge_{j\in J}(X_j, Y_j) = \left(\bigcap_{j\in J} X_j, \left(\left(\bigcup_{j\in J} Y_j\right)^{\Delta}\right)^{\Delta}\right)$$

$$\bigvee_{j\in J}(X_j, Y_j) = \left(\left(\left(\bigcup_{j\in J} X_j\right)^{\Delta}\right)^{\Delta}, \bigcap_{j\in J} Y_j\right)$$

In [18], on the basis of a parallel with *possibility theory* (indeed $X^{\Delta} = \bigcap_{x\in X} R(x)$ may be seen as the counterpart of the definition of a guaranteed possibility measure $\Delta(F) = \min_{x\in F} \pi(x)$ where π is a possibility distribution), other operators have been introduced: namely the possibility operator (denoted $(.)^{\Pi}$) and its dual, the necessity operator (denoted $(.)^{N}$), as well as the operator $(.)^{\nabla}$, dual of the operator $(.)^{\triangle}$ at the basis of FCA, defined as follows:

- X^{Π} is the set of properties satisfied by at least one object in X:

$$\begin{aligned} X^{\Pi} &= \{y \in \mathcal{P} \mid \exists x \in X,\ x\mathcal{R}y\} \\ &= \{y \in \mathcal{P} \mid X \cap R(y) \neq \emptyset\} \\ &= \textstyle\bigcup_{x\in X} R(x) \end{aligned}$$

- X^{N} is the set of properties that only the objects in X have:

$$\begin{aligned} X^{N} &= \{y \in \mathcal{P} \mid \forall x \in \mathcal{O}\,(x\mathcal{R}y \Rightarrow x \in X)\} \\ &= \{y \in \mathcal{P} \mid R(y) \subseteq X\} \\ &= \textstyle\bigcap_{x\notin X} \overline{R}(x) \end{aligned}$$

  (where $\overline{R}(x)$ is the set of properties that x does not have)

- X^{∇} is the set of properties that are not satisfied by at least one object outside X (X^{∇} should not be confused with the notion of weak opposition in FCA, often denoted in a similar way):

$$\begin{aligned} X^{\nabla} &= \{y \in \mathcal{P} \mid \exists x \in \overline{X},\ x\overline{\mathcal{R}}y\} \\ &= \{y \in \mathcal{P} | R(y) \cup X \neq \mathcal{O}\} \\ &= \textstyle\bigcup_{x\notin X} \overline{R}(x) \end{aligned}$$

The operators Y^{Π}, Y^{N}, Y^{∇} are obtained in a dual manner. As established in [15, 22], the pairs (X, Y) such as $X^{N} = Y$ and $Y^{N} = X$ (or in an equivalent way $X^{\Pi} = Y$ and $Y^{\Pi} = X$) characterize independent sub-contexts (i.e. which have not any objects or properties in common) inside the initial context. The pairs (X, Y) such as $X^{N} = Y$ and $Y^{N} = X$ are such that:

$$(X \times Y) \cup (\overline{X} \times \overline{Y}) \supseteq \mathcal{R}.$$

Regarding $X^{\nabla} = Y$ and $Y^{\nabla} = X$, it constitutes another characterization of formal concepts.

It has been shown [18, 22] that the four sets X^{Π}, X^{N}, X^{Δ}, X^{∇} represent complementary pieces of information, which are all necessary for a complete analysis of the situation of a set X in the formal context $\mathcal{K} = (\mathcal{O}, \mathcal{P}, \mathcal{R})$.

4 Stable Extensions in Argumentation and Formal Concepts

There is a striking parallel between Equation 2 in Section 2

$$\overline{R^{+}(S)} = \{a \in A | S \subseteq \overline{R}a\}$$

and the expression

$$X^{\Delta} = \{y \in \mathcal{P} \mid X \subseteq R(y)\} = \bigcap_{x \in X} R(x)$$

as well as between the definition 1 of a stable extension S

$$S = \overline{R^{+}(S)}$$

and the one of a formal concept (X, Y)

$$X^{\Delta} = Y \text{ and } Y^{\Delta} = X,$$

taking into account the similarity of the definitions of $\overline{R^{+}(S)}$ and X^{Δ}.

However, there is an obvious difference: in argumentation one is in the particular case $\mathcal{O} = \mathcal{P} = A$. What plays the role of the formal context is thus the relation $\overline{R}$ ("does not attack") defined on $A \times A = \mathcal{O} \times \mathcal{P}$.

It is well-known that stable extensions do not always exist. For instance, $(A = \{a, b, c, d\}, R = \{(a, b), (b, c), (c, a)\})$ has no stable extension. While formal concepts always exist when $R \neq \emptyset$, here it is no longer the case, when we work on $A \times A$, rather than with $\mathcal{O} \times \mathcal{P}$ where $\mathcal{O} \neq \mathcal{P}$. Since here the only acceptable formal concepts (X, Y) should be such that $X = Y$ $(= S$ in the above notation).

Then, one can look at the argumentative counterparts of X^{Π}, X^{N}, or X^{∇}. They are respectively:

- $\overline{R}^{+}(S) = \{a \in A | S \cap \overline{R}a \neq \emptyset\}$

the set of arguments not attacked by *all* the arguments in S. It means that for each argument in $\overline{R}^{+}(S)$ there exists at least one argument in S that does not attack it. It should not be confused with the set of arguments not attacked by *some* arguments in S: $\overline{R^{+}(S)} = \{a \in A | S \cap Ra = \emptyset\}$; Thus we have $\overline{R^{+}(S)} \subseteq \overline{R}^{+}(S)$, just as $\Delta \leq \Pi$ in possibility theory.

- $\overline{\overline{R}^{+}(\overline{S})} = \{a \in A | \overline{R}a \subseteq S\} = \{a \in A | \overline{S} \subseteq Ra\}$

the set of arguments that are attacked by all the arguments outside S ;

- $R^+(\overline{S}) = \{a \in A | S \cup \overline{R}a \neq A\}$
  $= \{a \in A | \overline{S} \cap Ra \neq \emptyset\}$

the set of arguments that are attacked by arguments outside S. We have $\overline{\overline{R}^+(\overline{S})} \subseteq R^+(\overline{S})$, as well as $N \leq \nabla$ holds in possibility theory. Moreover, if $R \neq \emptyset$ and $\overline{R} \neq \emptyset$, we have $\overline{\overline{R}^+(\overline{S})} \subseteq \overline{R}^+(S)$ and $\overline{R^+(S)} \subseteq R^+(\overline{S})$, counterparts of $N \leq \Pi$ and $\Delta \leq \nabla$ respectively. Thus, finally it holds that

$$\overline{R^+(S)} \cup \overline{\overline{R}^+(\overline{S})} \subseteq R^+(\overline{S}) \cap \overline{R}^+(S).$$

If one leaves aside complementations, it can thus be seen that given S, there are four basic sets of arguments:

$$R^+(S), \overline{R}^+(S), R^+(\overline{S}), \overline{R}^+(\overline{S}).$$

They are i) the arguments attacked by S, ii) the arguments not attacked by S, iii) the arguments attacked by non S, iv)the arguments not attacked by non S. Considering these four sets is necessary for a complete characterization of the relative position of the set of attackers of an argument a with respect to a set S of arguments (see [22] for the detailed possibilistic counterpart of this fact). It is clear that in a dual manner, there are four other noticeable sets in terms of R^- rather than of R^+.

We are thus led to consider the counterparts of the four conditions $X^\Delta = Y$ and $Y^\Delta = X$, $X^\nabla = Y$ and $Y^\nabla = X$, $X^\Pi = Y$ and $Y^\Pi = X$, and $X^N = Y$ and $Y^N = X$. They are respectively $S = \overline{R^+(S)}$, $S = R^+(\overline{S})$, which equivalently characterizes a stable extension on the one hand, and the equivalent constraints $S = \overline{\overline{R}^+(S)}$ and $S = \overline{R}^+(\overline{S})$ on the other hand, which correspond to extensions S and $\overline{S}$ that present a form of independence. Indeed $S = \overline{R}^+(\overline{S}) \Leftrightarrow \overline{S} = \overline{\overline{R}^+(\overline{S})}$ expresses that the set of arguments that are attacked by all the arguments outside S are precisely the arguments outside S.

5 Structures of Opposition and Abstract Argumentation

Structures of opposition have been studied in logic for a long time. In particular, the square of oppositions invented by Aristotle and its modern generalization to an hexagon of oppositions after the works of Robert Blanché [11] and Béziau [10] are encountered each time an internal negation and an external negation are at work on formal expressions.

Taking advantage of results presented in [23] regarding the structures of oppositions in formal concept analysis and in possibility theory, one may study in a similar manner the structures of oppositions at work in the theory of abstract argumentation, and in particular obtain the cube of oppositions pictured in Figure 1, where the four sets of arguments and their complements appear (a set

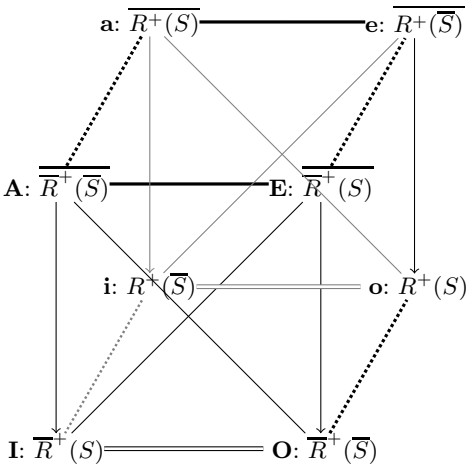


Fig. 1. Cube of oppositions between 8 remarkable sets of arguments

and its complement are at the two extremities of diagonals). The vertical arrows express inclusions. For example $\overline{R^+(S)} \subseteq R^+(\overline{S})$ (provided that $R \neq \emptyset$).

It is worth noticing that the different meaningful sets of arguments can be organized in such a structure that has played an important role through the whole history of logic. Moreover, the hexagonal structure of oppositions obtained in the logic of argumentation proposed in [3] should also be compared to the one obtained here.

6 Concluding Remarks: The Gradual Case

The idea to extend FCA to a fuzzy formal context, which enables us to express that an object satisfies a property to an intermediary degree, has been initially proposed by Burusco and Fuentes-Gonzalez [12] before being considerably developed by Belohlavek [5–7], and by a number of other authors, as in particular [30, 26, 32, 33]. For a discussion of different meaningful gradual extensions of FCA, the reader is referred to [14, 16].

We only give here the basic operator of fuzzy FCA [6]:

$$X^{\Delta}(y) = \bigwedge_{x \in O} (X(x) \to R(x, y))$$

where now R is a fuzzy relation, $R(x, y)$ is the degree to which x is in relation R with y, and X and X^{Δ} are fuzzy sets of objects and properties respectively, and $\bigwedge$ is the conjunction operator min and $\to$ an implication operator. An appropriate choice of this connective (such as Gödel residuated implication: $a \to b = 1$ if

$a \leq b$, and $a \to b = b$ if $a > b$) enables us to see a fuzzy formal concept in terms of its level cuts X_α, Y_α in such a way that

$$(X_\alpha \times Y_\alpha) \subseteq R_\alpha$$

where $X_\alpha \times Y_\alpha$ is maximal, with $R_\alpha = \{(x,y)|R(x,y) \geq \alpha\}$, $X_\alpha = \{x \in \mathcal{O}|X(x) \geq \alpha\}$, $Y_\alpha = \{y \in \mathcal{P}|Y(y) \geq \alpha\}$.

The idea of an abstract argumentation theory allowing for a *graded attack relation* has been recently advocated by some authors, in particularly in [25]. Following the parallel presented here, we are thus led to characterize a *fuzzy stable extension* by the equation

$$S(s) = \bigwedge_{a \in A} (S(s) \to \overline{R}(s,a))$$

where $S(s)$ is the degree to which the argument s belongs to the fuzzy stable extension S, $\overline{R}(s,a) = 1 - R(s,a)$, $R(s,a)$ being the degree with which s attacks a, which generalizes $S = \overline{R^+(S)} = \{a \in A|S \subseteq \overline{R}a\}$.

By exploiting the counterpart of$(X_\alpha \times Y_\alpha) \subseteq R_\alpha$, in the argumentative setting, one sees that we are back to the study of the level cuts of the relation of "non-attack" $\overline{R}$.

In the same spirit, one could define fuzzy admissible extensions, or define fuzzy extensions of $\overline{R}^+(S)$, $R^+(\overline{S})$, and $\overline{R}^+(\overline{S})$.

The association of degrees to arguments may have different meanings: They may in particular reflect the strength of the argument, or the uncertainty associated to its components. The nature of the degrees is as much important in FCA, since uncertainty and satisfaction level of a gradual property should not be handled in the same way [16]. Different treatments should as well be considered in argumentation according to the meaning of the degrees. What is suggested above rather applies to the strength of the arguments rather than to their uncertainty.

The computation of extensions in argumentation can be expressed in the setting of propositional logic in terms of algebraic equations as shown in [9] (see also [2]). This idea has been recently reused by Gabbay [28], thus putting abstract argumentation in the framework of the equational semantics of propositional logic, first developed one century ago by Louis Couturat [13]. The exploitation of this idea can be extended to fuzzy logic [28]. One can thus also reconsider what is proposed above in this paper in that perspective.

This paper is a preliminary attempt at bridging four noticeable areas in the formal treatment of information, namely abstract argumentation, formal concept analysis, but also possibility theory and squares of opposition, which have remained completely related until recently. Such parallels should contribute to enrich each domain: for instance, in argumentation by considering new sets of arguments and understanding better how they are related. It may also provide useful guidelines for introducing grades in argumentation.

References

1. Amgoud, L., Cayrol, C.: On the acceptability of arguments in preference-based argumentation. In: Cooper, G.F., Moral, S. (eds.) Proc. 14th Conf. on Uncertainty in Artif. Intellig. (UAI 1998), Madison, W.I., July 24-26, pp. 1–7. Morgan Kaufmann (1998)
2. Amgoud, L., Devred, C.: Argumentation frameworks as constraint satisfaction problems. In: Benferhat, S., Grant, J. (eds.) SUM 2011. LNCS, vol. 6929, pp. 110–122. Springer, Heidelberg (2011)
3. Amgoud, L., Prade, H.: Towards a logic of argumentation. In: Hüllermeier, E., Link, S., Fober, T., Seeger, B. (eds.) SUM 2012. LNCS (LNAI), vol. 7520, pp. 558–565. Springer, Heidelberg (2012)
4. Barbut, M., Monjardet, B.: Ordre et Classification. Algèbre et Combinatoire. Tome 2, Hachette, Paris (1970)
5. Belohlavek, R.: Fuzzy Galois connections. Math. Logic Quart. 45, 497–504 (1999)
6. Belohlavek, R.: Fuzzy Relational Systems. Foundations and principles. Kluwer (2002)
7. Belohlavek, R., Vychodil, V.: What is a fuzzy concept lattice. In: Proc. CLA 2005, Olomounc, Czech Republic, pp. 34–45 (2005)
8. Besnard, P., Doutre, S.: Characterization of semantics for argument systems. In: Dubois, D., Welty, C.A., Williams, M.A. (eds.) Proc. of the 9th Inter. Conf. on Principles of Knowledge Representation and Reasoning (KR 2004), Whistler, Canada, June 2-5, pp. 183–193. AAAI Press (2004)
9. Besnard, P., Doutre, S.: Checking the acceptability of a set of arguments. In: Proc. 10th Inter. Workshop on Non-Monotonic Reasoning (NMR 2004), Whistler, Canada, June 6-8, pp. 59–64 (2004)
10. Béziau, J.-Y.: The power of the hexagon. Logica Universalis 6 (2012)
11. Blanché, R.: Structures Intellectuelles. Essai sur l'Organisation Systématique des Concepts, Vrin, Paris (1966)
12. Burusco, A., Fuentes-Gonzalez, R.: The study of the L-fuzzy concept lattice. Mathware & Soft Comput. 3, 209–218 (1994)
13. Couturat, L.: L'Algèbre de la Logique. Gauthier-Villars, Paris (1905)
14. Djouadi, Y., Dubois, D., Prade, H.: Différentes extensions floues de l'analyse formelle de concepts. In: Actes Rencontres Francophones sur la Logique Floue et ses Applications (LFA 2009), Annecy, Cépaduès, November 5-6, pp. 141–148 (2009)
15. Djouadi, Y., Dubois, D., Prade, H.: Possibility theory and formal concept analysis: Context decomposition and uncertainty handling. In: Hüllermeier, E., Kruse, R., Hoffmann, F. (eds.) IPMU 2010. LNCS(LNAI), vol. 6178, pp. 260–269. Springer, Heidelberg (2010)
16. Djouadi, Y., Dubois, D., Prade, H.: Graduality, uncertainty and typicality in formal concept analysis. In: Cornelis, C., Deschrijver, G., Nachtegael, M., Schockaert, S., Shi, Y. (eds.) 35 Years of Fuzzy Set Theory. STUDFUZZ, vol. 261, pp. 127–147. Springer, Heidelberg (2010)
17. Djouadi, Y., Prade, H.: Possibility-theoretic extension of derivation operators in formal concept analysis over fuzzy lattices. Fuzzy Optimization and Decision Making 10(4), 287–309 (2011)
18. Dubois, D., Dupin de Saint Cyr, F., Prade, H.: A possibilty-theoretic view of formal concept analysis. Fundamenta Informaticae 75(1-4), 195–213 (2007)
19. Dubois, D., Prade, H.: Possibility Theory. Plenum Press (1988)

20. Dubois, D., Prade, H.: Possibilistic logic: a retrospective and prospective view. Fuzzy Sets and Systems 144, 3–23 (2004)
21. Dubois, D., Prade, H.: Bridging gaps between several frameworks for the idea of granulation. In: Proc. Symp. on Foundations of Computational Intelligence (in IEEE Symposium Series on Computational Intelligence - SSCI 2011) (FOCI 2011), Paris, April 11-15, pp. 59–65 (2011)
22. Dubois, D., Prade, H.: Possibility theory and formal concept analysis: Characterizing independent sub-contexts. Fuzzy Sets and Systems 196, 4–16 (2012)
23. Dubois, D., Prade, H.: From Blanché's hexagonal organization of concepts to formal concept analysis and possibility theory. Logica Universalis 6 (2012)
24. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence 77, 321–358 (1995)
25. Dunne, P.E., Hunter, A., McBurney, P., Parsons, S., Wooldridge, M.: Weighted argument systems: Basic definitions, algorithms, and complexity results. Artificial Intelligence 175, 457–486 (2011)
26. Fan, S.-Q., Zhang, W.-X., Xu, W.: Fuzzy inference based on fuzzy concept lattice. Fuzzy Sets & Syst. 157, 3177–3187 (2006)
27. Ferré, S., Ridoux, O.: Introduction to logical information systems. Information Processing and Mgmt. 40, 383–419 (2004)
28. Gabbay, D.M.: Introducing equational semantics for argumentation networks. In: Liu, W. (ed.) ECSQARU 2011. LNCS, vol. 6717, pp. 19–35. Springer, Heidelberg (2011)
29. Ganter, B., Wille, R.: Formal Concept Analysis. Springer (1999)
30. Georgescu, G., Popescu, A.: Non-dual fuzzy connections. Archive for Mathematical Logic 43, 1009–1039 (2004)
31. Gorogiannis, N., Hunter, A.: Instantiating abstract argumentation with classical logic arguments: Postulates and properties. Artificial Intellig. 175, 1479–1497 (2011)
32. Lai, H., Zhang, D.: Concept lattices of fuzzy contexts: formal concept analysis vs. rough set theory. Inter. J. Approx. Reason. (2009)
33. Medina, J., Ojeda-Aciego, M., Ruiz-Calvino, J.: Formal concept analysis via multiadjoint concept lattices. Fuzzy Sets and Systems 160(2), 130–144 (2009)
34. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival, I. (ed.) Ordered Sets, pp. 445–470. Reidel, Dordrecht (1982)

Approximating Credal Network Inferences by Linear Programming*

Alessandro Antonucci, Cassio P. de Campos, David Huber, and Marco Zaffalon

Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA)
Galleria 2, Manno-Lugano, Switzerland
{alessandro,cassio,david,zaffalon}@idsia.ch

Abstract. An algorithm for approximate credal network updating is presented. The problem in its general formulation is a multilinear optimization task, which can be linearized by an appropriate rule for fixing all the local models apart from those of a single variable. This simple idea can be iterated and quickly leads to very accurate inferences. The approach can also be specialized to classification with credal networks based on the maximality criterion. A complexity analysis for both the problem and the algorithm is reported together with numerical experiments, which confirm the good performance of the method. While the inner approximation produced by the algorithm gives rise to a classifier which might return a subset of the optimal class set, preliminary empirical results suggest that the accuracy of the optimal class set is seldom affected by the approximate probabilities.

1 Introduction

Credal networks [5] are a generalization of Bayesian networks (e.g., [11]) based on the notion of credal sets. A credal set is a set of probability mass functions, thus representing a quite general and expressive model of uncertainty. Other uncertainty models like belief functions [14] or possibility measures can be regarded as (special classes of) credal sets. A Bayesian network can be turned into a credal network by simply replacing the local models, which are conditional probability mass functions, with conditional credal sets over the same variables. Exactly as a Bayesian network defines a joint probability mass function over its whole set of variables, a credal network defines a joint credal set, which is (the convex closure of) the set of all joint mass functions obtained from the Bayesian networks consistent with the local credal sets.

Compared to the case of Bayesian networks, inference in credal networks is considerably harder. For instance, a marginalization task corresponds to a multilinear optimization problem (updating is a fractional multilinear task) [7]. This is known to be NP-hard even for singly connected networks [8], while the analogous inference in Bayesian networks can be performed in polynomial time [11].

* This work was supported by the Swiss NSF grants nos. 200020_134759 / 1, 200020_137680 / 1, and by the Hasler foundation grant n. 10030.

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 13–24, 2013.

Despite the hardness of the problem, some algorithms are known to perform reasonably well under certain conditions. Exact approaches have been proposed that implement some branch-and-bound method with local searches [4,6,8,9]. Unfortunately they all suffer from serious efficiency issues unless the credal network is very simple. For instance, none of these methods can deal well with a binary node having four ternary parents, because this setting is already equivalent to $3^4 = 81$ free optimization variables to be chosen, meaning a space of 2^{81} possible solutions just locally to this node! On the other hand, approximate methods either are fast and provide no accuracy guarantee [3,4,6] or provide theoretical guarantees but are as slow as exact methods [13]. Moreover, all these approximate methods are only capable of treating credal networks under a vertex-based representation, while a constraint-based specification of credal networks still lacks any practical algorithm.

In this paper we present a fast approximate algorithm for inferences in credal networks based on solving a sequence of linear programming problems. It uses a constraint-based specification, which allows us to deal with domains where the local credal sets are given by their linear constraints. It does not suffer from many parents and large credal sets because the optimization is done by compact linear problems. To the best of our knowledge, this is the first method for general credal networks to truly run the inference with a constraint-based specification. We describe the method and some heuristic ideas to improve its accuracy. Unlike similar ideas already proposed in the literature [6], our approach does not require an explicit enumeration of the extreme points of the credal sets and should be therefore used when the number of extreme points in the local credal sets is exponentially large (e.g., variables with many states and/or parents, credal sets defined by probability intervals, etc). We also discuss how the method can be used for decision making under the maximality criterion [15].

Sections 2 and 3 review the basic notation and definitions of Bayesian and credal networks. The proposed procedure is presented in Sections 4 and 5. Numerical experiments show that the proposed method compares favorably against other available methods in the literature (Section 7). Results are particularly positive when the algorithm is specialized to the case of classification in credal networks based on the maximality criterion. Although this problem is shown to be even harder than the marginalization inferences (discussed in Section 6), classifications based on our approximate algorithm are empirically shown to coincide with those based on exact methods.

2 Bayesian Networks

Consider a set of variables $\boldsymbol{X} := (X_0, X_1, \ldots, X_n)$ in one-to-one correspondence with the nodes of an acyclic directed graph $\mathcal{G}$. For each $i = 0, \ldots, n$, the joint variable $\Pi_i \subseteq \boldsymbol{X}$ denotes the parents of X_i according to $\mathcal{G}$. All these variables are categorical: X_i takes its values on the finite set Ω_{X_i} and so does Π_i in $\Omega_{\Pi_i} := \times_{X_j \in \Pi_i} \Omega_{X_j}$, for each $i = 0, \ldots, n$.[1] The graph $\mathcal{G}$ represents stochastic

[1] Symbol $\times$ denotes Cartesian set product.

independence relations by means of a Markov condition: any variable is conditionally independent of its non-descendant non-parents given its parents (see e.g., [11]). The specification of a conditional probability mass function $P(X_i|\pi_i)$ for each $\pi_i \in \Omega_{\Pi_i}$ and $i = 0, \ldots, n$, induces, for each $\boldsymbol{x} \in \times_{i=0}^{n} \Omega_{X_i}$, the factorization:

$$P(\boldsymbol{x}) := \prod_{i=0}^{n} P(x_i|\pi_i), \tag{1}$$

where the values of x_i and π_i are those consistent with $\boldsymbol{x}$.

We call *Bayesian network* a specification of the conditional probability mass functions $\{P(X_i|\pi_i)\}_{i=0,\ldots,n}^{\pi_i \in \Omega_{\Pi_i}}$. In particular, the mass functions associated to X_i, i.e., $\{P(X_i|\pi_i)\}_{\pi_i \in \Omega_{\Pi_i}}$ are called the *local models* of X_i, for each $i = 0, \ldots, n$. Inference in Bayesian networks is based on the joint probability mass function in Eq. (1). Marginals, for instance, are obtained by summing out other variables from the joint, i.e., the marginalization of X_0 corresponds to the computation, for each $x_0 \in \Omega_{X_0}$, of

$$P(x_0) = \sum_{x_1,\ldots,x_n} \prod_{i=0}^{n} P(x_i|\pi_i), \tag{2}$$

where $\sum_x$ is a shortcut for $\sum_{x \in \Omega_X}$. With straightforward calculations, the marginal in Eq. (2) can be expressed as a linear combination of the local probabilities associated to an arbitrary $X_j \in \boldsymbol{X}$, i.e.,

$$P(x_0) = \sum_{x_j,\pi_j} [P(x_0|x_j,\pi_j) \cdot P(\pi_j)] \cdot P(x_j|\pi_j), \tag{3}$$

where probabilities $P(\pi_j)$ and $P(x_0|x_j,\pi_j)$ can be computed from the joint as in Eq. (1),[2] while probabilities $P(x_j|\pi_j)$ are already available in the Bayesian network specification. As special case, note that for $j = 0$, Eq. (3) rewrites as $P(x_0) = \sum_{\pi_0} P(\pi_0) \cdot P(x_0|\pi_0)$; while if $X_0 \in \Pi_j$, and $\Pi_j' := \Pi_j \setminus \{X_0\}$, $P(x_0) = \sum_{x_j,\pi_j'} P(x_0,\pi_j')P(x_j|x_0,\pi_j')$. Remarkably, values of both $P(\pi_j)$ and $P(x_0|x_j,\pi_j)$ are not affected by those of the local models of X_j in the Bayesian network specification. To see that, note that when computing a marginal, the descendants and hence their local models can be removed without affecting the probability. As X_j is a child of all the variables in Π_j, the computation of $P(\pi_j)$ is not affected by the local models $\{P(X_j|\pi_j)\}_{\pi_j \in \Omega_{\Pi_j}}$. Similarly, when computing a conditional probability, arcs leaving the variables after the conditioning bar can be removed: thus, in the case of $P(x_0|x_j,\pi_j)$, we can disconnect X_j from the rest of the network, thus making its local model irrelevant for the particular calculation. This remark, together with Eq. (3) will be exploited by the approximate algorithm presented later.

[2] Given a joint probability mass function, conditionals are obtained from Bayes' rule. For instance, $P(x_0|x_j,\pi_j) = P(x_0,x_j,\pi_j)/P(x_j,\pi_j)$, provided that $P(x_j,\pi_j) > 0$.

3 Credal Networks

The Bayesian theory of subjective probability has been extended by more general uncertainty theories in order to model situations of highly incomplete or conflicting information. Among others, the theory of imprecise probability in [15] adopts credal sets, which are (convex) sets of probability mass functions, as a more general model of uncertainty about the state of a categorical variable. In particular, here we focus on finitely generated credal sets, which are specified by a finite number of linear constraints on the probabilities (e.g., see Fig. 1).

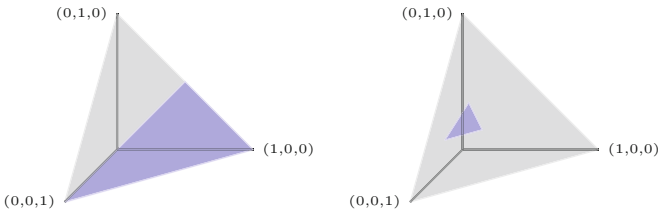


Fig. 1. Credal sets over a ternary variable X (i.e., $\Omega_X = \{x', x'', x'''\}$). The representation is in a three-dimensional space with coordinates $[P(x'), P(x''), P(x''')]$. The polytopes represent respectively: (a) the credal set defined by constraint $P(x') > P(x'')$; (b) a credal set whose extreme points are $\{[.1, .3, .6]^T, [.3, .3, .4]^T, [.1, .5, .4]^T\}$.

Credal sets can be used to extend Bayesian networks to imprecise probabilities. In order to do that, in the definition of Bayesian network, every conditional probability mass function $P(X_i|\pi_i)$ is replaced by a (conditional) *credal set* $K(X_i|\pi_i)$ for each $\pi_i \in \Omega_{\Pi_i}$ and $i = 0, \ldots, n$. As we focus on credal sets defined by a finite number of linear constraints, the set of extreme points of $K(X_i|\pi_i)$, to be denoted by $\text{ext}[K(X_i|\pi_i)]$, has finite cardinality. We call *credal network* a specification of conditional credal sets $\{K(X_i|\pi_i)\}_{i=0,\ldots,n}^{\pi_i \in \Omega_{\Pi_i}}$. Under this generalized setting, Eq. (1) can be used to obtain different joint probability mass functions. Let us consider all the possible extreme specifications, and then take the convex hull (denoted as CH), i.e., build the following joint credal set:

$$K(\boldsymbol{X}) := \text{CH}\left\{ P(\boldsymbol{X}) \,\middle|\, P(\boldsymbol{x}) := \prod_{i=0}^{n} P(x_i|\pi_i), \begin{array}{l} \forall \boldsymbol{x} \in \times_{i=0}^{n} \Omega_{X_i}, \\ \forall P(X_i|\pi_i) \in \text{ext}[K(X_i|\pi_i)] \\ \forall i = 0, 1, \ldots, n, \forall \pi_i \in \Omega_{\Pi_i} \end{array} \right\}. \tag{4}$$

The credal set in Eq. (4) is called the *strong extension* of the credal network. Here inference in credal networks is intended as based on the strong extension.

For instance, the lower bound with respect to $K(\boldsymbol{X})$ of the marginal probability in Eq. (2) is:

$$\underline{P}(x_0) := \min_{P(\boldsymbol{X})\in K(\boldsymbol{X})} P(x_0) = \min_{\substack{P(X_i|\pi_i)\in K(X_i|\pi_i)\\ \pi_i\in\Omega_{\Pi_i}, i=0,\ldots,n}} \sum_{x_1,x_2,\ldots,x_n} \prod_{i=0}^{n} P(x_i|\pi_i), \qquad (5)$$

and similarly for the upper $\overline{P}(x_0)$. Eq. (5) corresponds to the optimization of a non-linear (namely multilinear [7]) function over a feasible region defined by linear constraints on the optimization variables. In the next section we present an approximate algorithm for this task.

4 The Algorithm

The algorithm we present is based on Lukatskii and Shapot's approach [12] to approximate the solution of multilinear problems. In essence, a multilinear problem can be converted into a linear one if we fix all but one optimization variable in each of its multilinear terms. In Lukatskii and Shapot's terminology, there is a partition $S_1 \cup S_2 \cup \ldots \cup S_w$ of the optimization variables such that fixing the optimization variables in every set of the partition apart from S_j, the multilinear problem becomes linear. By iterating over j, which defines the set S_j to remain free, one can approximate the solution of the multilinear problem with a sequence of linear ones. Da Rocha et al. [10] have already used similar ideas to perform approximate inference in credal networks, but their approach had to enumerate all the extreme points of credal sets and used a less sophisticated search.

Our algorithm finds an inner approximation of the interval $[\underline{P}(x_0), \overline{P}(x_0)]$, i.e., an upper approximation of the lower probability as in Eq. (5) and a lower approximation of the upper probability. The idea is to reduce the multilinear task in Eq. (5) to a linear program by fixing all the local credal sets to singletons apart from those associated to an arbitrarily chosen variable $X_j \in \boldsymbol{X}$, which we call the *free* variable. Given a free $X_j \in \boldsymbol{X}$, we pick an extreme point $\tilde{P}(X_i|\pi_i) \in \text{ext}[K(X_i|\pi_i)]$, for each $\pi_i \in \Omega_{X_i}$ and $i = 0, \ldots, n$, $i \neq j$. These are additional constraints to the optimization problem in Eq. (5), which becomes:

$$\begin{aligned}\underline{P}'(x_0) &:= \min_{\substack{P(X_j|\pi_i)\in K(X_j|\pi_j)\\ \pi_j\in\Omega_{\Pi_j}}} \sum_{x_1,x_2,\ldots,x_n} \left[\prod_{i=0, i\neq j}^{n} \tilde{P}(x_i|\pi_i)\right] \cdot P(x_j|\pi_j) = \\ &= \min_{P(X_j|\pi_j)\in K(X_j|\pi_j)} \sum_{x_j,\pi_j} \left[\tilde{P}(x_0|x_j,\pi_j)\cdot\tilde{P}(\pi_j)\right] \cdot P(x_j|\pi_j),\end{aligned} \qquad (6)$$

where the last derivation is based on Eq. (3) and probabilities $\tilde{P}(x_0|x_j,\pi_j)$ and $\tilde{P}(\pi_j)$ are denoted by a tilde as they are computed from the joint of a Bayesian network with local models $\{\tilde{P}(X_i|\pi_i)\}$. The discussion of the special cases $j = 0$ and $X_0 \in \Pi_j$ is exactly as in the Bayesian case (see the end of Sect. 2).

We focus on marginal probabilities just for the sake of clarity. Yet, the computation with conditional probabilities is straightforward as the linear programs become linear-fractional programs.

Let us comment on two important facts about Eq. (6). First, being the solution of an optimization with additional constraints with respect to Eq. (5) (see the second term in the equation), we clearly have $\underline{P}(x_0) \leq \underline{P}'(x_0)$. Secondly, it is clear from the third term of Eq. (6) that the computation of $\underline{P}'(x_0)$ is a linear program whose optimization variables are the local probabilities of X_j, i.e., $\{P(x_j|\pi_j)\}_{x_j \in \Omega_{X_j}, \pi_j \in \Omega_{\Pi_j}}$. Moreover, as the solution of a linear program lies on an extreme point of the feasible region (i.e., an extreme point of the credal set), there is a specification $P^*(X_j|\pi_j) \in \text{ext}[K(X_j|\pi_j)]$, for each $\pi_j \in \Omega_{\Pi_j}$ such that:

$$\underline{P}'(x_0) = \sum_{x_1, x_2, \ldots, x_n} P^*(x_j|\pi_j) \left[\prod_{i=0, i \neq j}^{n} \tilde{P}(x_i|\pi_i) \right]. \tag{7}$$

Coping with Zero Probabilities. In order to obtain the coefficients of the objective function in the linear task in Eq. (6), the conditionals $\tilde{P}(x_0|x_j, \pi_j)$ (and the marginals $\tilde{P}(\pi_j)$) should be computed for each $x_j \in \Omega_{X_j}$, $\pi_j \in \Omega_{\Pi_j}$. For zero-probability conditioning events, i.e., $\tilde{P}(x_j, \pi_j) = 0$, the conditionals cannot be computed. In this case, the term of the sum in Eq. (6) associated to (x_j, π_j) rewrites as: $\tilde{P}(x_0|x_j, \pi_j) \cdot \tilde{P}(\pi_j) \cdot \tilde{P}(x_j|\pi_j) = \tilde{P}(x_0|x_j, \pi_j) \cdot \tilde{P}(x_j, \pi_j)$, being therefore zero. Thus, the corresponding term does not appear in the objective function and its coefficient can be safely set to zero.

5 Searching for the Optimum

In the previous section, we defined a procedure which, given a free variable X_j and the specification of an extreme point for all conditional credal sets of non-free variables, returns an upper approximation of the lower probability $\underline{P}(x_0)$, together with the specification of the extreme points of the local credal sets associated to the free variable which produced that optimum.

If we call *almost-Bayesian network* a credal network whose local credal sets are singletons apart from those associated to a single variable, the optimization procedure we proposed consists in taking an almost-Bayesian network consistent with the original credal network (i.e., its strong extension is included in that of the original credal network) and exploiting the fact that marginalization of almost-Bayesian networks is a linear problem. By solving the linear problem, we obtain: (i) an upper (lower) approximation of the lower (upper) probability; (ii) a specification of the extreme points of the credal sets associated to the only "non-Bayesian" variable in the almost-Bayesian network. These extreme points can be used as an assignment for the extreme points of those local credal sets, and another variable can be "freed", leading to a new linear program. In the rest of this section we suggest a possible initialization and two iteration strategies.

Initialization. The optimization in Eq. (6) requires an initialization, i.e., the specification of an almost-Bayesian network consistent with the credal network. This can be done by randomly picking an extreme point (or a simple point) from each local credal set apart for those associated to X_j. A deterministic alternative to the random choice is the center of mass of the credal set:[3]

$$P_{\mathrm{CM}}(x_i|\pi_i) := \sum_{P(X_i|\pi_i)\in \mathrm{ext}[K(X_i|\pi_i)]} \frac{P(x_i|\pi_i)}{|\mathrm{ext}[K(X_i|\pi_i)]|} \tag{8}$$

for each $x_i \in \Omega_{X_i}$, $\pi_i \in \Omega_{\Pi_i}$, $i = 0, 1, \ldots, n$, with $i \neq j$. Note that the center of mass belongs to its credal set, but it is not an extreme point of it (unless the credal set includes a single point). As we know that the exact solution of Eq. (5) corresponds to a Bayesian network whose local models are extreme points of the local credal sets, this means that if a solution includes a center of mass it cannot be exact. Yet, this can be easily overcome by iterating the procedure at least once for each variable, as all those linear problems will certainly pick extreme points.

Greedy Search. The solution in Eq. (7) of the linear program in Eq. (6) provides an approximate solution for the computation of the marginal of a credal network. This procedure can be iterated by changing the "free" variable X_j and using the optimal solution $\{P^*(X_j|\pi_j)\}_{\pi_j\in\Omega_{\Pi_j}}$ of the previous problem as a different initialization. This improves the solution as shown here.

Proposition 1. *Let* $\{\tilde{P}(X_j|\pi_j)\}_{j=0,1,\ldots,n}^{\pi_j\in\Omega_{\Pi_j}}$ *be a Bayesian network specification consistent with a credal network specification* $\{K(X_j|\pi_j)\}_{j=0,1,\ldots,n}^{\pi_j\in\Omega_{\Pi_j}}$. *As in Eq. (2), let* $\tilde{P}(x_0) := \sum_{x_1,\ldots,x_n}\prod_{i=0}^{n}\tilde{P}(x_i|\pi_i)$ *and, as in Eq. (6):*

$$\underline{\tilde{P}}'(x_0) := \min_{P(X_j|\pi_j)\in K(X_j|\pi_j)} \sum_{x_1,\ldots,x_n} \left[\prod_{\substack{i=0\\ i\neq j}}^{n} \tilde{P}(x_i|\pi_i)\right] P(x_j|\pi_j). \tag{9}$$

Then $\underline{\tilde{P}}'(x_0) \leq \tilde{P}(x_0)$.

Proof. It suffices to put in evidence the terms $\{\tilde{P}(x_j|\pi_j)\}^{\pi_j\in\Omega_{\Pi_i}}$ *in the definition of* $\tilde{P}(x_0)$ *and note that, by definition of consistency between Bayesian and credal networks,* $\tilde{P}(X_j|\pi_j) \in K(X_j|\pi_j)$ *for each* $\pi_j \in \Omega_{\Pi_j}$. □

As a corollary of Prop. 1, it follows that iterating the algorithm can only improve the quality of the approximation. A *greedy* iteration strategy is therefore the following: given a candidate solution $\underline{P}(x_0)$, we evaluate the improved solution obtained by keeping the same specification of the extreme mass functions as those used to obtain $\underline{P}(x_0)$ and we free one of the variables a time. Let $\underline{P}'_j(x_0)$

[3] In the language of evidence theory [14], this corresponds to the so-called *pignistic* transformation which associates a probability mass function to a belief functions.

denote the candidate solution obtained by freeing X_j, for each $j = 0, 1, \ldots, n$. During the first iteration, we pick as free variable X_{j^*} such that:

$$j^* := \text{argmin}_{j=0,1,\ldots,n} \underline{P}'_j(x_0). \tag{10}$$

This naturally provides us with a partition of the optimization variables as defined by Lukatskii and Shapot [12]. Hence, if all estimated solutions in Eq. (10) coincide with the previously obtained solution, a stationarity area has been reached and the algorithm stops. Often this will be a local optimum of the multilinear problem. Yet, this is not always the case because there might be a neighborhood of candidates with no improving solution, but whose neighbors might have an improving solution. The only way to ensure local optimality is to keep track of all the candidates with equal solution until such set is completely explored or an improving solution is found [12]. In practice, this is not an issue, and can be overcome by the use of multiple starts, perturbations of solutions in case of achieving a stationary area, and/or a queue of candidate solutions, as we describe in the following.

Improving the Greedy Approach. The greedy approach based on Eq. (10) and described in the previous paragraph can be improved by defining a *priority queue* of size k, which includes not only the best candidate, but the k-best ones (each candidate is tracked together with its relative Bayesian network specification). The solutions $\{\underline{P}_j(x_0)\}_{j=0}^n$ are evaluated for the candidate in the peak of the priority queue, and are themselves included back in the queue (as long as they are improving solutions). In this variant, the algorithm stops when the queue is empty, which guarantees that all candidates have been explored (this will certainly include the previously explained greedy approach). The queue can be seen as many greedy searches in distinct "directions".

Computational Complexity (Algorithm). Let m and l denote, respectively, the maximum number of states and incoming parents (i.e., the indegree) of the network variables: $m := \max_{i=0,\ldots,n} |\Omega_{X_i}|$ and $l := \max_{i=0,\ldots,n} |\Pi_i|$. Let q be the maximum number of linear constraints required to define a local credal set. A linear program as in Eq. (5) has at most m^{l+1} variables and $m^l \cdot q$ constraints. Because the input size should already be proportional to $m^l \cdot q$, the algorithm spends time equivalent to run a linear programming solver on the (local) input specification times the total number of iterations.

6 Maximality-Based classification

Credal networks have been used to implement both knowledge-based systems (e.g., [1]) and classifiers (e.g., [16]). Given a credal network over $\boldsymbol{X}$, let X_0 be the class variable and $\tilde{\boldsymbol{X}} \subseteq \boldsymbol{X} \setminus \{X_0\}$ the variables (features) for which evidential information is available. Given an instance $\tilde{\boldsymbol{x}}$ of the features, the identification of the optimal class(es) of X_0 should be therefore based on the conditional credal set $K(X_0|\tilde{\boldsymbol{x}})$ obtained by conditioning the strong extension in Eq. (4). Such an

identification depends on the adopted decision criterion. E.g., the so called Γ-maximin approach returns $x_0^* := \text{argmax}_{x_0 \in \Omega_{X_0}} \underline{P}(x_0|\tilde{\boldsymbol{x}})$. Another criterion is maximality [15], which returns the following set of classes:

$$\Omega^*_{X_0} := \left\{ x'_0 \in \Omega_{X_0} \;\middle|\; \nexists x''_0 \in \Omega_{X_0} : \begin{array}{l} P(x''_0|\tilde{\boldsymbol{x}}) > P(x'_0|\tilde{\boldsymbol{x}}) \\ \forall P(X_0|\tilde{\boldsymbol{x}}) \in K(X_0|\tilde{\boldsymbol{x}}) \end{array} \right\}. \tag{11}$$

In practice, $\Omega^*_{X_0}$ should be initialized to Ω_{X_0}. Then, for each $x'_0, x''_0 \in \Omega_{X_0}$, the following dominance should be checked:

$$\min_{P(X_0|\tilde{\boldsymbol{x}}) \in K(X_0|\tilde{\boldsymbol{x}})} [P(x''_0|\tilde{\boldsymbol{x}}) - P(x'_0|\tilde{\boldsymbol{x}})] > 0, \tag{12}$$

and, if satisfied, x'_0 removed from $\Omega^*_{X_0}$. The test in Eq. (12) cannot be directly checked by algorithms for credal networks. Nevertheless, in a recent paper [2], the test has been mapped to a standard updating task in a credal network. This is obtained by augmenting the original credal network with an auxiliary node associated to a Boolean variable Y and such that Y is a leaf child of X_0. The quantification of the conditional credal sets for Y given X_0 is precise:

$$P(Y = \text{true}|x_0) = \begin{cases} 0 \text{ if } & x_0 = x'_0 \\ 1 \text{ if } & x_0 = x''_0 \\ \frac{1}{2} \text{ otherwise.} & \end{cases} \tag{13}$$

After this quantification, the dominance test in Eq. (12) is equivalent to check whether $\underline{P}(Y = \text{true}|\tilde{\boldsymbol{x}}) > \frac{1}{2}$. The algorithm proposed in [2] can be used to evaluate the dominance for each pair of classes and determine the undominated ones according to Eq. (11). The upper approximation $\underline{P}'(Y = \text{true}|\tilde{\boldsymbol{x}}) \geq \underline{P}(Y = \text{true}|\tilde{\boldsymbol{x}})$ implies that some dominances detected by the algorithm might not really take place. Hence, the set of optimal classes evaluated by the approximate algorithm is a subset of the exact one.

Computational Complexity (classification). We characterize the computational complexity of maximality-based classification. The evaluation in Eq. (12) is called *dominance test*. Given a credal network, evidence $\tilde{\mathbf{x}}$, $q \in \Omega_Q$, and a rational r, the *inference query* decides whether exists $P \in K(\mathbf{X})$ such that $P(q|\tilde{\mathbf{x}}) \geq r$ [8]. The treewidth of a network measures the extent to which it resembles a tree (see [11] for a more formal definition).

Theorem 1. *The dominance test is coNP-complete in bounded treewidth networks and coNP^{PP}-complete in networks of general topology.*

Proof. We show hardness by demonstrating that the complementary decision, that is, whether the minimization of Eq. (12) *is less than or equal to zero, is NP^{PP}-hard in general, and NP-hard for bounded treewidth networks. For that, we reduce the marginal inference problem in a credal network to it. Marginal inference in credal networks is shown to be NP-hard in polytrees with at most two parents per node and NP^{PP}-hard in general networks [8].*

Take a credal network with inference query $\exists P : P(q|\tilde{\mathbf{x}}) \geq r$*, for a given rational* r*, query* q *and evidence* $\tilde{\mathbf{x}}$*. Build a new network by adding a binary node* X_0*, which has* Q *as sole parent and precise probability mass functions defined as* $P(x_0''|q) = \frac{r}{2}$ *and* $P(x_0''|\neg q) = \frac{1+r}{2}$*. Note that the new network has the same topology (and treewidth) of the original one. Now, the complement of the dominance test asks whether*

$$\min_P [P(x_0''|\tilde{\mathbf{x}}) - P(x_0'|\tilde{\mathbf{x}})] \leq 0 \iff \min_P [2P(x_0''|\tilde{\mathbf{x}}) - 1] \leq 0$$

$$\iff \min_P [rP(q|\tilde{\mathbf{x}}) + (1+r)P(\neg q|\tilde{\mathbf{x}})) - 1] \leq 0 \iff$$

$$\min_P [r - P(q|\tilde{\mathbf{x}})] \leq 0 \iff \max_P P(q|\tilde{\mathbf{x}}) \geq r \iff \exists P : \ P(q|\tilde{\mathbf{x}}) \geq r,$$

which is exactly the credal network marginal query. As the treewidth of the network has not been modified, the hardness results follow. Pertinence of this complementary decision in NP for the case of bounded treewidth (respectively in NP^{PP} *for the general case) is immediate, since given* $P \in K(\mathbf{X})$*, we can use a Bayesian network inference to certify that* $P(x_0''|\tilde{\mathbf{x}}) \leq P(x_0'|\tilde{\mathbf{x}})$ *(in polynomial time for bounded treewidth nets and by using the PP oracle for the general case).* □

Hence, deciding whether a class is in the maximal set is a very demanding query, because it is tested against all other classes, and each of such tests can be itself hard. For example, if the network has bounded treewidth, the problem of deciding whether a class is maximal falls in the class of decision problems that can be solved by polynomial time machines with access to non-adaptive queries to an NP oracle, namely $\mathrm{P}^{||\mathrm{NP}}$. Even if a hard task to do exactly, we shall see that our algorithm is able to recover the set of maximal classes successfully in practice (but without guarantee of exactness).

7 Experiments

To validate the performance of our algorithm, we use a benchmark made of different credal nets with random topology, either multiply or singly connected, and two classical (multiply connected) models, namely the *Alarm* and the *Insurance* networks. The maximum indegree for the networks with random topology is limited to 5. The number of states for the Alarm and the Insurance networks is the same as in their original specifications, while for the other networks the number of states is randomly chosen between 2 and 8. All the models are quantified by randomly generated conditional credal sets with a fixed number of extreme points, whose number is ranging from 2 to 8 for each network. Inferences are computed by a Java implementation of the algorithm linked to the COIN-OR linear program solver. The code is available as a free software tool.[4] In these experiments, the greedy approach described in Sect. 4 is considered and the algorithm is therefore called G-LP. Centers of mass are used for the first iteration.

[4] See `http://ipg.idsia.ch/software` and `http://www.coin-or.org`.

Table 1. Benchmark results (mean square absolute errors). Upper marginal probabilities have been computed for each state of each network in the benchmark such that the exact solver took less than three minutes to find the optimum.

Networks	# of tests	G-LP	G-LP'	GL2U	ILS
Alarm	973	.0474	.0076	.1218	.2709
Insurance	650	.0767	.0795	.1818	.2700
Random (single)	6162	.0816	.0109	.1724	.1528
Random (multi)	2963	.0855	.0140	.1594	.1269

Exact inferences are computed by mapping the problem to an integer linear program [9], which is solved by CPLEX. Comparisons are with other approximate algorithms: the iterated local search (ILS) [6] and the GL2U algorithm [3].

Before commenting on the results in Tab. 1, note that our approach assumes the local credal sets to be specified by linear constraints. This is often the case in real scenarios (e.g., credal classifiers or knowledge-based expert systems quantified by probability intervals). Conversely, credal networks used for benchmarking represent their local credal sets by explicit enumeration of the extreme points. The reason is that most of the algorithms for credal networks require the local credal sets to be described by their extreme points. In the first experiment, we evaluate the lower and upper bounds of the probabilities w.r.t. the extreme points. E.g., for the credal set in Fig. 1(b): $P(x') \in [.1, .3]$, $P(x'') \in [.3, .5]$, $P(x''') \in [.4, .6]$. These constraints define larger credal sets compared to the original ones. The third column of Tab. 1 reports the results, i.e., the mean square difference between the inner approximation obtained by G-LP and the exact inferences. The accuracy is fairly good on the whole benchmark. In the fourth column of Tab. 1, the same inferences computed by G-LP in the third column are regarded here as approximations for credal networks with local credal sets defined by the original extremes (and not by the induced linear constraints). We denote this heuristic variant as G-LP'. The inner approximation of G-LP is now balanced by the outer approximation introduced by considering the linear constraints and this produces smaller errors (MSE$<$ .02) for the Alarm and the random networks. The performance is less accurate for the Insurance network, probably because of the relatively high number of states for the variables of this network, which makes the outer approximation too coarse. Regarding the proposed improvement of the greedy approach, the results (with queue size $k = 40$ and still one minute as maximum running time) are just marginally better than those based on G-LP and for this reason are not reported. Finally, we evaluate the performance of G-LP for maximality-based classification. We consider ten benchmark networks with the Alarm topology. As classes we choose the variables with four states with no evidence (i.e., $\tilde{\boldsymbol{X}} = \emptyset$). On *all* these classification tasks the two sets of optimal classes coincide. Thus, the small approximation in the inferences based on G-LP seems to have no effect when finding the set of maximal solutions, which were recovered exactly. While this is somehow expected (because the dependency on the exact probability value is less important), this empirical result is promising for the use of credal networks in classification.

8 Conclusions

A new algorithm based on a sequence of linear optimizations is proposed for approximate credal network updating. The algorithm can deal with a constraint-based specification of credal networks, and provides inner approximation solutions. It is also extended to find the maximal classes in a classification problem. The complexities of these problems and of the algorithm are presented. In a practical perspective, preliminary results are promising: the algorithm is fast and accurate. As future work, we intend to test the algorithm on larger networks and with other search heuristics, and support other decision criteria.

References

1. Antonucci, A., Brühlmann, R., Piatti, A., Zaffalon, M.: Credal networks for military identification problems. Int. J. Approx. Reasoning 50(2), 666–679 (2009)
2. Antonucci, A., de Campos, C.P.: Decision making by credal nets. In: Proceedings of the International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC 2011), vol. 1, pp. 201–204. IEEE (2011)
3. Antonucci, A., Yi, S., de Campos, C.P., Zaffalon, M.: Generalized loopy 2U: a new algorithm for approximate inference in credal networks. Int. J. Approx. Reasoning 51(5), 474–484 (2010)
4. Cano, A., Gomez, M., Moral, S., Abellan, J.: Hill-climbing and branch-and-bound algorithms for exact and approximate inference in credal networks. Int. J. Approx. Reasoning 44(3), 261–280 (2007)
5. Cozman, F.G.: Credal networks. Artificial Intelligence 120, 199–233 (2000)
6. da Rocha, J.C.F., Cozman, F.G., de Campos, C.P.: Inference in polytrees with sets of probabilities. In: UAI 2003, pp. 217–224 (2003)
7. de Campos, C.P., Cozman, F.G.: Inference in credal networks using multilinear programming. In: Proceedings of the Second Starting AI Researcher Symposium, pp. 50–61. IOS Press, Amsterdam (2004)
8. de Campos, C.P., Cozman, F.G.: The inferential complexity of Bayesian and credal networks. In: Proceedings of the International Joint Conference on Artificial Intelligence, Edinburgh, pp. 1313–1318 (2005)
9. de Campos, C.P., Cozman, F.G.: Inference in credal networks through integer programming. In: International Symposium on Imprecise Probability: Theories and Applications (ISIPTA), Prague, pp. 145–154 (2007)
10. Ferreira da Rocha, J.C., Cozman, F.G.: Inference in credal networks: branch-and-bound methods and the A/R+ algorithm. Int. J. Approx. Reasoning 39(2-3), 279–296 (2005)
11. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press (2009)
12. Lukatskii, A.M., Shapot, D.V.: Problems in multilinear programming. Computational Mathematics and Mathematical Physics 41(5), 638–648 (2000)
13. Mauà, D.D., de Campos, C.P., Zaffalon, M.: Updating credal networks is approximable in polynomial time. Int. J. Approx. Reasoning (2012)
14. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press (1976)
15. Walley, P.: Statistical Reasoning with Imprecise Probabilities. Chapman and Hall (1991)
16. Zaffalon, M.: The naive credal classifier. J. Stat. Plann. Inference 105(1), 5–21 (2002)

A Comparative Study of Compilation-Based Inference Methods for Min-Based Possibilistic Networks

Raouia Ayachi[1,2], Nahla Ben Amor[1], and Salem Benferhat[2]

[1] LARODEC, Institut Supérieur de Gestion Tunis, Le Bardo, Tunisie, 2000
raouia.ayachi@gmail.com, nahla.benamor@gmx.fr
[2] CRIL-CNRS, Université d'Artois, France, 62307
benferhat@cril.univ-artois.fr

Abstract. Min-based possibilistic networks, which are compact representations of possibility distributions, are powerful tools for representing and reasoning with uncertain and incomplete information in the possibility theory framework. Inference in these graphical models has been recently the focus of several researches, especially under compilation. It consists in encoding the network into a *Conjunctive Normal Form* (CNF) base and compiling this latter to efficiently compute the impact of an evidence on variables. The encoding strategy of such networks can be either locally using *local structure* or globally using *possibilistic local structure*. This paper emphasizes on a comparative study between these strategies for compilation-based inference approaches in terms of CNF parameters, compiled bases parameters and inference time.

1 Introduction

Knowledge compilation [6] is a common technique for propositional logic knowledge bases. It is a mapping from a given knowledge base into a special form of propositional bases, for which queries can be answered efficiently. Assuming that the input knowledge base does not often change, so it is turned into a compiled one during an off-line compilation phase which is then used to answer the queries on-line. Answering such queries using the compiled base should be computationally easier than answering them from the input base. One of the most prominent successful applications of knowledge compilation is in the context of graphical models by including probabilistic networks [7,9]. The objective behind these works is to ensure an efficient computation of a-posteriori probability degrees given an evidence on a set of variables.

Using the possibility theory framework, we recently explored compilation-based inference approaches in while dealing with min-based possibilistic networks [2,3]. These latters are encoded in a *Conjunctive Normal Form* (CNF) base, then compiled into the appropriate target compilation language in order to ensure an efficient computation of a-posteriori possibility degrees given an evidence on a set of variables. Min-based possibilistic networks can be encoded using either *local*

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 25–36, 2013.

structure by associating a unique propositional variable per equal parameters per conditional possibility table [3] or *possibilistic local structure* by handling equal parameters from a global point of view, i.e., per all conditional possibility tables [2]. Possibilistic local structure, which is exclusively useful for a qualitative setting, goes beyond the classical local structure in terms of CNF parameters by exploiting the idempotency property of the min operator.

In [1], we emphasized on common points and unveiled differences between compilation-based inference process in the probabilistic and the possibilistic setting from a spatial viewpoint. However, no comparison has been held between proposed compilation-based inference approaches in the possibility theory framework. The investigation in this paper serves to spotlight the behavior of different encoding strategies via a detailed experimental study in terms of CNF parameters (variables and clauses), compiled bases parameters (edges) and inference time.

The remaining paper is organized as follows: Section 2 presents a brief refresher on min-based possibilistic networks. Section 3 reviews compilation-based inference approaches of min-based possibilistic networks. Section 4 compares methods from an experimental point of view.

2 Min-Based Possibilistic Networks

This section introduces min-based possibilistic networks which can be viewed as the possibilistic counterpart of Bayesian networks [13] when we consider the qualitative interpretation of the possibilistic scale. We start at first by presenting basic concepts and notations and a reminder on possibility theory.

Let $V = \{X_1, ..., X_N\}$ be a set of variables. By v we denote instantiations of all variables $X_i \in V$. We denote by $D_{X_i} = \{x_1, .., x_n\}$ the domain associated with the variable X_i. By x_i we denote any instance of X_i. By x_{ij} we denote the j^{th} instance of X_i. When there is no confusion we use x_i to mean any instance of X_i. Ω denotes the universe of discourse, which is the Cartesian product of all variable domains in V. Each element $\omega \in \Omega$ is called a state of Ω. Possibility theory [12] is seen as a simple and natural model for handling uncertain data. The basic building block in this theory is the concept of *possibility distribution* π, which is a mapping from Ω to the unit interval [0, 1] such that $\pi(\omega) = 1$ and $\pi(\omega) = 0$ refer to a totally possible state and an impossible state, respectively. It is generally assumed that there exists at least a state ω which is totally possible. In this case, π is said to be normalized. From a normalized possibility distribution π, we can compute two dual measures $\Pi(\phi) = max_{\omega \in \phi} \pi(\omega)$ and $N(\phi) = 1 - \Pi(\neg\phi)$.

A min-based possibilistic network over a set of N variables V, denoted by ΠG_{min}, is composed of:

– A *graphical component* composed of a Directed Acyclic Graph (DAG) where nodes represent variables and edges encode links between variables. The parent set of any variable X_i is denoted by $U_i = \{U_{i1}, U_{i2}, ..., U_{im}\}$ where U_{ij} is the j^{th}

parent of U_i and m is the number of parents of X_i. In what follows, we use x_i, u_i, u_{ij} to denote, respectively, possible instances of X_i, U_i and U_{ij}.
– A *numerical component* that quantifies different links. Uncertainty of each node X_i is represented by a local normalized conditional possibility table (denoted by $C\Pi T_i$) in the context of its parents. The set of all $C\Pi T_i$ is denoted by $C\Pi T$. Conditional possibility tables should respect the normalization constraint for each variable $X_i \in V$ expressed by: $\forall u_i, \max_{x_i} \ \Pi(x_i|u_i) = 1$.

Example 1. *Let us consider the ΠG_{min}, depicted by Figure 1, containing two binary variables A and B. Each node, either A or B, is quantified by a possibility distribution in the context of its parents.*

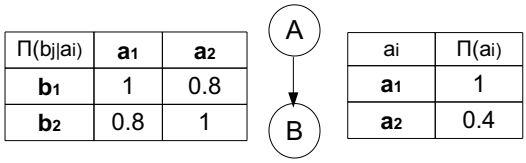


Fig. 1. Example of a min-based possibilistic network

3 Possibilistic Compilation-Based Inference Approaches

Knowledge compilation is a common technique for propositional logic knowledge bases. It is a mapping from a given knowledge base into a special form of propositional bases, for which queries can be answered efficiently. It consists mainly in splitting query answering of a particular problem into two phases [6]. In the first phase, knowledge bases are preprocessed in order to obtain the data structures the most appropriate for the given application (such a phase is called *the off-line reasoning*). In the second phase, queries are answered using the output of the first phase(such a phase is the *on-line reasoning*). A logical form is qualified to be a *target compilation language* if it supports some set of nontrivial queries, usually including *clausal entailment*, in a polynomial time. There are several compilation languages as it has been studied in the knowledge map of [11]. We are in particular interested in *Decomposable Negation Normal Form (DNNF)* [8], which is the set of all NNF sentences that satisfy *decomposability* stating that conjuncts of any conjunction share no variables. DNNF and its variants support a rich set of polynomial-time operations which can be performed efficiently, namely *clausal deduction*, *conditioning*, etc.

Inference under compilation in min-based possibilistic networks has been recently explored in [2,3]. Globally these methods are grouped into classes depending on the compilation phase input (initial min-based possibilistic network or the possibilistic knowledge base associated with it) and the encoding strategy (local structure or possibilistic local structure). This section provides a sufficient grounding relative to these approaches, as is necessary for the remainder of the article.

3.1 Π-DNNF and Its Variants

The first class of methods represents the possibilistic adaptation, denoted by Π-DNNF, of the standard approach of Darwiche [9], and its refinements using *local structure* and *possibilistic local structure* encoding strategies.

The basic idea of Π-DNNF method [3] consists in encoding the network into a CNF base using two types of propositional variables, namely: *instances indicators* $\lambda_{x_{ij}}$ associated to different instances x_{ij} of variables X_i and *parameter variables* $\theta_{x_i|u_i}$ relative to possibility degrees $\Pi(x_i|u_i)$. The CNF encoding of ΠG_{min}, denoted by C_{min}, can be defined as follows:

Definition 1. *Using the set of instance indicators and parameter variables, C_{min} contains the following clauses:*

- *Mutual exclusive clauses:*

$$\lambda_{x_{i1}} \vee \lambda_{x_{i2}} \vee \cdots \vee \lambda_{x_{in}} \tag{1}$$

$$\neg\lambda_{x_{ij}} \vee \neg\lambda_{x_{ik}}, j \neq k \tag{2}$$

- *Parameter clauses:* $\forall\ \theta_{x_i|u_{i1},u_{i2},\ldots,u_{im}}$, *we have:*

$$\lambda_{x_i} \wedge \lambda_{u_{i1}} \wedge \ldots \wedge \lambda_{u_{im}} \rightarrow \theta_{x_i|u_{i1},u_{i2},\ldots,u_{im}} \tag{3}$$

$$\theta_{x_i|u_{i1},u_{i2},\ldots,u_{im}} \rightarrow \lambda_{x_i} \tag{4}$$

$$\theta_{x_i|u_{i1},\ldots,u_{im}} \rightarrow \lambda_{u_{i1}}, \cdots, \theta_{x_i|u_{i1},\ldots,u_{im}} \rightarrow \lambda_{u_{im}} \tag{5}$$

The resulting CNF encoding C_{min} is then compiled into one of the most succinct target compilation language, namely DNNF, which is then used to efficiently compute the effect of an evidence on variables. It is clear that Π-DNNF method does not consider numerical values while encoding the network. In other terms, it associates a parameter variable for each possibility degree, regardless of its numerical value. However, parameters values can be exploited in the encoding phase locally using local structure or globally using possibilistic local structure in order to reduce CNF variables and generate more compact compiled bases. By *local structure*, we mean encoding equal parameters per conditional possibility table using the same parameter variable θ_j. This encoding strategy reduces the number of variables and clauses since equal parameters per table are encoded using only a simple implication (i.e. clause (3)) instead of a logical equivalence (i.e. clauses (3), (4) and (5)) [3]. The approach (resp. CNF encoding) that uses local structure is denoted by Π-DNNF$_{LS}$ (resp. C^{LS}_{min}).

From a global point of view, possibility degrees can be encoded using what we call *possibilistic local structure* [2]. This encoding strategy takes advantage of the idempotency property of the min operator by associating a unique parameter variable $\Pi\theta_j$ per equal parameters per all conditional possibility tables. Using possibilistic local structure, the CNF encoding C^{PLS}_{min} encodes each parameter using a simple implication (i.e. clause (3)), regardless of its occurrence number per tables. This variant is denoted by Π-DNNF$_{PLS}$.

Example 2. *Let us consider the ΠG_{min} of Figure 1. Before encoding the network, we should at first associate λ_{x_i} for each instance x_i, then encode possibility degrees by parameter variables as shown in Table 1. We can deduce that the number of parameter variables is equal to 6 when no strategy is used. It corresponds to 4 (resp. 3) in the case of local structure (resp. possibilistic local structure). We should then encode ΠG_{min} as shown in Table 2. We can point that the number of parameter clauses is decreasing from one strategy to another. In fact, it is equal to 16, 8 and 6 in the case of C_{min}, C_{min}^{LS} and C_{min}^{PLS}, respectively.*

Table 1. Parameter variables used in C_{min}, C_{min}^{LS} and C_{min}^{PLS}

Variables	**Possibility degrees**	$\mathbf{C_{min}}$	$\mathbf{C_{min}^{LS}}$	$\mathbf{C_{min}^{PLS}}$
A	$\Pi(a_1) = 1$	θ_{a_1}	θ_{a_1}	$\Pi\theta_1$
	$\Pi(a_2) = 0.4$	θ_{a_2}	θ_{a_2}	$\Pi\theta_{a_2}$
B	$\Pi(b_1\|a_1) = 1$	$\theta_{b_1\|a_1}$	θ_1	$\Pi\theta_1$
	$\Pi(b_1\|a_2) = 0.8$	$\theta_{b_1\|a_2}$	θ_2	$\Pi\theta_2$
	$\Pi(b_2\|a_1) = 0.8$	$\theta_{b_2\|a_1}$	θ_2	$\Pi\theta_2$
	$\Pi(b_2\|a_2) = 1$	$\theta_{b_2\|a_2}$	θ_1	$\Pi\theta_1$

Table 2. CNF encodings C_{min}, C_{min}^{LS} and C_{min}^{PLS}

Variables	**Mutual exclusive clauses**		
A	$(\lambda_{a_1} \vee \lambda_{a_2}) \wedge (\neg\lambda_{a_1} \vee \neg\lambda_{a_2})$		
B	$(\lambda_{b_1} \vee \lambda_{b_2}) \wedge (\neg\lambda_{b_1} \vee \neg\lambda_{b_2})$		
$\mathbf{\Pi(x_i\|u_i)}$	**Parameter clauses**		
A	$\mathbf{C_{min}}$	$\mathbf{C_{min}^{LS}}$	$\mathbf{C_{min}^{PLS}}$
$\Pi(a_1) = 1$	$(\lambda_{a_1} \rightarrow \theta_{a_1})$ $\wedge(\theta_{a_1} \rightarrow \lambda_{a_1})$	$(\lambda_{a_1} \rightarrow \theta_{a_1})$ $\wedge(\theta_{a_1} \rightarrow \lambda_{a_1})$	$(\lambda_{a_1} \rightarrow \Pi\theta_1)$
$\Pi(a_2) = 0.4$	$(\lambda_{a_2} \rightarrow \theta_{a_2})$ $\wedge(\theta_{a_2} \rightarrow \lambda_{a_2})$	$(\lambda_{a_2} \rightarrow \theta_{a_2})$ $\wedge(\theta_{a_2} \rightarrow \lambda_{a_2})$	$(\lambda_{a_2} \rightarrow \Pi\theta_{a_2})$
B	$\mathbf{C_{min}}$	$\mathbf{C_{min}^{LS}}$	$\mathbf{C_{min}^{PLS}}$
$\Pi(b_1\|a_1) = 1$	$(\lambda_{a_1} \wedge \lambda_{b_1} \rightarrow \theta_{b_1\|a_1})$ $\wedge(\theta_{b_1\|a_1} \rightarrow \lambda_{b_1})$ $\wedge(\theta_{b_1\|a_1} \rightarrow \lambda_{a_1})$	$(\lambda_{a_1} \wedge \lambda_{b_1} \rightarrow \theta_1)$	$(\lambda_{a_1} \wedge \lambda_{b_1} \rightarrow \Pi\theta_1)$
$\Pi(b_2\|a_1) = 0.8$	$(\lambda_{a_1} \wedge \lambda_{b_2} \rightarrow \theta_{b_2\|a_1})$ $\wedge(\theta_{b_2\|a_1} \rightarrow \lambda_{b_2})$ $\wedge(\theta_{b_2\|a_1} \rightarrow \lambda_{a_1})$	$(\lambda_{a_1} \wedge \lambda_{b_2} \rightarrow \theta_2)$	$(\lambda_{a_1} \wedge \lambda_{b_2} \rightarrow \Pi\theta_2)$
$\Pi(b_1\|a_2) = 0.8$	$(\lambda_{a_2} \wedge \lambda_{b_1} \rightarrow \theta_{b_1\|a_2})$ $\wedge(\theta_{b_1\|a_2} \rightarrow \lambda_{b_1})$ $\wedge(\theta_{b_1\|a_2} \rightarrow \lambda_{a_2})$	$(\lambda_{a_1} \wedge \lambda_{b_2} \rightarrow \theta_2)$	$(\lambda_{a_2} \wedge \lambda_{b_1} \rightarrow \Pi\theta_2)$
$\Pi(b_2\|a_2) = 1$	$(\lambda_{a_2} \wedge \lambda_{b_2} \rightarrow \theta_{b_2\|a_2})$ $\wedge(\theta_{b_2\|a_2} \rightarrow \lambda_{b_2})$ $\wedge(\theta_{b_2\|a_2} \rightarrow \lambda_{a_2})$	$(\lambda_{a_2} \wedge \lambda_{b_2} \rightarrow \theta_1)$	$(\lambda_{a_2} \wedge \lambda_{b_2} \rightarrow \Pi\theta_1)$

3.2 DNNF-PKB

The second approach, considered as purely possibilistic and named DNNF-PKB, is based on the transformation of min-based possibilistic networks into logic-based representations [5].

Definition 2. *Let ΠG_{min} be a min-based possibilistic network, then its possibilistic knowledge base is expressed by:*

$$\Sigma_{min} = \Sigma_{X_1} \cup \Sigma_{X_2} \cup \cdots \cup \Sigma_{X_N} \tag{6}$$

where $\Sigma_{X_i} = \{(\neg x_i \vee \neg u_i, a_i) : a_i = 1 - \Pi(x_i|u_i) \neq 0\}$, $\forall X_i \in V$.

Encoding the possibilistic base Σ_{min} associated with the possibilistic network into a CNF base is performed by affecting new propositional variables for the different necessity degrees A_i existing in the possibilistic knowledge base. This means that to each formula (α_i, a_i) corresponds the propositional formula $\alpha_i \vee A_i$. Hence, the propositional encoding of Σ_{min}, denoted by K_Σ is expressed by:

$$K_\Sigma = \{\alpha_i \vee A_i : (\alpha_i, a_i) \in \Sigma_{min}\} \tag{7}$$

Note that in this approach the notion of local structure is meaningless since equal parameters corresponding to necessity degrees are handled from a global point of view, i.e., per base.

Example 3. *Let us re-consider the ΠG_{min} of Figure 1. Then, the possibilistic knowledge base of ΠG_{min} is the following: $\Sigma_{min} = ((a_1, 0.6), (a_2 \vee b_1, 0.2), (a_1 \vee b_2, 0.2))$. We can deduce that Σ_{min} does not contain zero-weighted formulas corresponding to possibility degrees equal to 1. The CNF encoding of the possibilistic knowledge base Σ_{min} is shown in Table 3.*

Table 3. The CNF encoding K_Σ of Σ_{min}

Clauses of A	
$(a_1, 0.6)$	$(a_1 \vee A_1)$
Clauses of B	
$(a_2 \vee b_1, 0.2)$	$(a_2 \vee b_1 \vee A_2)$
$(a_1 \vee b_2, 0.2)$	$(a_1 \vee b_2 \vee A_2)$

To efficiently compute a-posteriori possibility degrees, K_Σ should be compiled into any target compilation language that supports both of conditioning and clausal entailment. This approach is qualified to be flexible since it takes advantage of existing propositional knowledge bases compilation methods.

4 Experimental Study

This section proposes an experimental study aiming to compare our possibilistic compilation-based inference algorithms in terms of CNF parameters, compiled bases parameters and the inference time w.r.t the one of the standard junction tree method [4]. To this end, we implement different CNF encodings of possibilistic networks using Matlab R2010, then compile these latters using the state of the art c2d compiler[1] [10] and finally implement inference using the resulting compiled bases. Experiments ran on a 2.27 GHz Core i3 processor with 4 GB of memory. We start with describing the experimental protocol then we compare compilation-based inference methods.

4.1 Experimental Protocol

As possibilistic networks have a graphical component and a numerical one, then it is judicious to specify which kind of networks to use during the experimental process. In fact, we randomly generate a possibilistic network by setting the number of nodes to 50, the maximum number of parents per node to 3, the number of instances per variable to 2 and the number of roots to 10. Moreover, we vary values of possibility distributions (except for the normalization value 1) using $EP_{C\Pi T}$ stating *the percent of equal parameters within conditional possibility tables (i.e., $C\Pi T$)*. We set $EP_{C\Pi T}$ to $\{0\%, 10\%, 30\%, 50\%, 70\%, 100\%\}$. When $EP_{C\Pi T}$ is equal to 50%, this means that each possibility degree appears in 50% of $C\Pi T$. The extreme case 0% states that each possibility degree, except for 1, appears in a unique conditional possibility table, i.e., $C\Pi T_i$. While the case of 100% means that there are two degrees, including the normalization one, which appear in all conditional possibility tables, i.e., $C\Pi T$. Of course when we affect equal parameters per $C\Pi T$, we should specify which tables are involved by $EP_{C\Pi T}$. In order to vary parameters positions, we propose to generate randomly indexes of tables involved by $EP_{C\Pi T}$. We perform this process 100 times, for each percentage of $EP_{C\Pi T}$.

4.2 Comparing Inference Approaches

Using the experimental protocol described above, we will compare inference approaches over 100 different randomly generated parameters locations. Interestingly enough, we establish a comparison covering the inference time averaged over 30 different randomly generated evidence sets. The experimental results are shown in Table 4. A deep analysis of these results are established for each criterion separately.

CNF Variables Let us analyze the variables behavior of each method, depicted by Figure 2 (a):

[1] Available at `http://reasoning.cs.ucla.edu/c2d/`.

Table 4. Π-DNNF vs Π-DNNF$_{LS}$ vs Π-DNNF$_{PLS}$ vs DNNF-PKB (better values are in bold)

Method	$EP_{C\Pi T}$	Variables	Clauses	Edges	Inf(sec)
Π-DNNF	0-100	358	1048	3428	0.489
Π-DNNF$_{LS}$	0	278	684	**2504**	**0.377**
	10	276	668	**2412**	**0.367**
	30	271	632	**2353**	**0.356**
	50	262	574	**2121**	**0.306**
	70	254	520	**1951**	**0.295**
	100	200	358	**1148**	**0.235**
Π-DNNF$_{PLS}$	0	179	358	76695	32.254
	10	127		251712	212.355
	30	112		30151	5.775
	50	108		7232	0.859
	70	107		3856	0.287
	100	102		608	0.152
DNNF-PKB	0	**178**	**229**	76414	31.563
	10	**126**		245368	201.809
	30	**111**		30003	4.701
	50	**107**		7019	0.748
	70	**106**		3623	0.269
	100	**101**		502	0.138

- *Π-DNNF*: Row 1 of Table 4 shows that the number of variables remains unaltered even if $EP_{C\Pi T}$ is rising. Obviously, this is an expected result since Π-DNNF does not take into consideration any numerical value by encoding each possibility degree by a parameter variable, regardless of its value.
- *Π-DNNF$_{LS}$*: Local structure exploited in Π-DNNF$_{LS}$ has a positive impact on CNF variables since equal parameters, especially the normalization degree 1, are encoded by the same parameter variable. It is also clear that the number of variables is reduced for each increase of $EP_{C\Pi T}$, as shown in Figure 2 (a).
- *Π-DNNF$_{PLS}$*: In this method, CNF variables are increasingly reduced since equal parameters per $C\Pi T$ are increased for each percentage of $EP_{C\Pi T}$. When $EP_{C\Pi T} = 100\%$, the number of variables is equal to 102 since we have 100 indicator variables and 2 parameter variables.
- *DNNF-PKB*: Row 4 of Table 4 shows that the number of variables of DNNF-PKB is reduced by one comparing to those of Π-DNNF$_{PLS}$. Indeed, the possibility degree equal to 1 in Π-DNNF$_{PLS}$ is not encoded in DNNF-PKB since it corresponds to a necessity degree equal to 0, which is not represented in possibilistic knowledge bases.

CNF Clauses. The behavior of CNF clauses of each compilation-based method of Table 4 is depicted by Figure 2 (b) and detailed below:

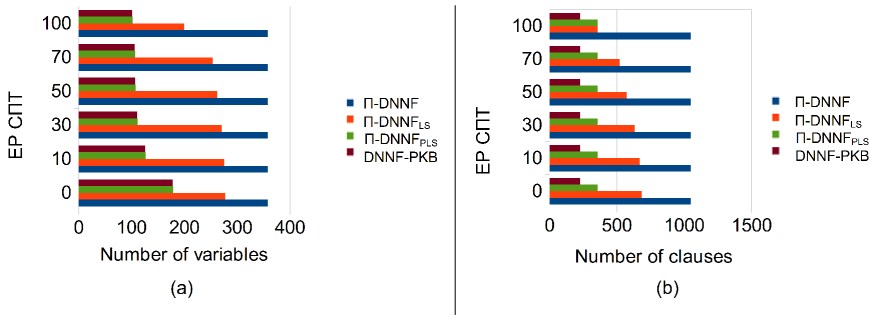


Fig. 2. (a): Number of variables and (b): Number of clauses

- *Π-DNNF*: As shown in Table 4, the number of clauses does not depend on $EP_{C\Pi T}$ since each parameter is encoded using a right-side clause and a set of left-side clauses, regardless of its numerical value.
- *Π-DNNF$_{LS}$*: The number of clauses per parameter depends on its occurrence number per $C\Pi T_i$. In fact, by increasing the number of equal parameters per $C\Pi T_i$, the number of clauses is reduced since these parameters are encoded using only right-side clauses.
- *Π-DNNF$_{PLS}$*: The number of clauses is increasingly reduced in *Π*-DNNF$_{PLS}$ comparing to those of *Π*-DNNF$_{LS}$ since the number of equal parameters per $C\Pi T$ is rising for each $EP_{C\Pi T}$ and consequently are encoded using only right-side clauses. Obviously, this number is not altered for any $EP_{C\Pi T}$ since only right-side clauses are considered.
- *DNNF-PKB*: Each clause encoding the possibility degree 1 in *Π*-DNNF$_{PLS}$ is not within DNNF-PKB's clauses since it corresponds to a zero-weighted clause. This justifies the gain of clauses.

Compiled Bases Edges. Let us now study and interpret Figure 3 (a) showing the impact of CNF encoding strategies on compiled bases edges:

- *Π-DNNF*: As shown in row 1 of Table 4, the number of edges is equal to 3428. This value remains the same for each $EP_{C\Pi T}$, as for CNF variables and clauses.
- *Π-DNNF$_{LS}$*: Encoding equal parameters per table using a unique parameter variable has a positive impact on compiled bases edges. This behavior follows the one of CNF variables and clauses.
- *Π-DNNF$_{PLS}$*: Compiled bases edges are higher when we deal with possibilistic local structure, as shown in Figure 3 (a). By paying more attention on row 3 and column 5 of Table 4, we can remark that compiled bases edges depend on $EP_{C\Pi T}$. In fact, when $EP_{C\Pi T} = 10\%$, generated compiled bases have more edges than those of $EP_{C\Pi T} = 0\%$. However, edges decrease from $EP_{C\Pi T} = 30\%$ until $EP_{C\Pi T} = 100\%$.
- *DNNF-PKB*: DNNF-PKB has a number of edges smaller than those of *Π*-DNNF$_{PLS}$. This is especially due to the reduction of CNF parameters.

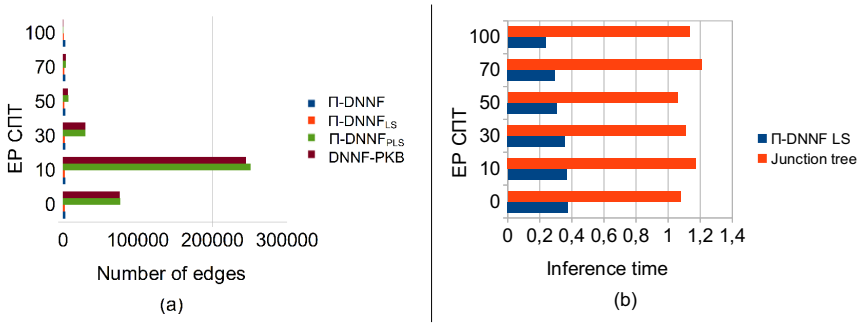


Fig. 3. (a): Number of edges and (b): Inference time

A deep analysis of these results shows that the reduction of CNF variables and clauses does not always imply more compact compiled bases. Indeed, using less CNF parameters while exploiting the encoding strategy local structure induces compiled bases with less edges. This is not the case of possibilistic local structure, which increases compiled bases parameters even with a reduced number of CNF parameters.

We can note that satisfying decomposability from CNF bases requires performing *case analysis* over the variables shared by sub-formulas. When we use local structure, equal parameters per table $C\Pi T_i$ are encoded using the same parameter variable. In this case, the compiler c2d splits common variables pertaining to the same conditional possibility table, which implies that a local interaction between clauses is performed. However, when we use possibilistic local structure the number of shared variables is increased since equal parameters are handled from a global point of view (i.e., per $C\Pi T$). Such encoding strategy introduces many interactions among clauses corresponding to different conditional possibility tables, which makes the resulting knowledge base harder to compile.

As we have mentioned above, we used c2d initially proposed to generate d-DNNFs. This compiler uses the case analysis technique that efficiently enforces the property of decomposability while enforcing determinism as well. This means that we are subjected to the determinism property as a side effect of this compiler, which is useless in the possibility theory framework.

By paying more attention on purely possibilistic approaches, we point out that these methods are very sensitive to equal parameters per tables (i.e., $C\Pi T$). Let us interpret the impact of each percentage of $EP_{C\Pi T}$:

- $EP_{C\Pi T} = 0\%$: As shown in row 3 of Table 4, the number of propositional variables in Π-DNNF$_{PLS}$ is equal to 179. Since we associate an instance indicator for each instance of $X_i \in V$ and knowing that we deal with 50 nodes, then the number of instance indicators is equal to 100. Consequently, we have 79 parameter variables where 78 encode different possibility degrees appearing once per $C\Pi T$ and only one parameter variable θ_1 encodes the possibility degree 1 pertaining to all conditional possibility tables. The re-

sulting base is more hard to compile than the one with local structure since there is an interaction between all clauses weighted by θ_1.

- $EP_{C\Pi T} = 10\%$: From row 3 of Table 4, we can deduce that the number of redundant parameter variables in Π-DNNF$_{PLS}$ is equal to 27, in which one parameter variable encoding the degree 1 appears in all tables and 26 ones appear in 10% of $C\Pi T$ (i.e., 5 tables). For DNNF-PKB method, there are 26 parameter variables encoding degrees different from 1. In such a case, the compiler c2d performs case analysis for each parameter variable θ_i holding 5 tables and pertaining to the 26 ones. Since the number of parameter variables θ_i encoding equal parameters per 5 tables is increased, interactions among clauses is rising and consequently, the base is more hard to compile.
- $EP_{C\Pi T} = 30\%, \cdots, 100\%$: We can point out from row 3 of Table 4 that the number of parameter variables appearing in 15, 25, 35 and 50 tables is equal to 11, 7, 6 and 2 when $EP_{C\Pi T} = 30\%$, $EP_{C\Pi T} = 50\%$, $EP_{C\Pi T} = 70\%$ and $EP_{C\Pi T} = 100\%$, respectively. In such cases, c2d deals with a reduced number of shared variables, which explains the reduction of compiled bases edges.

Inference Time. Inference time of compilation-based approaches follows exactly the same behavior as edges. In other terms, the smaller is the compiled base the faster inference is. Let us now compare the inference time of the junction tree method and the most compact compilation-based inference method, namely Π-DNNF$_{LS}$. We can deduce from Table 4 that the inference time in Π-DNNF$_{LS}$ decreases each time $EP_{C\Pi T}$ is rising. Yet the time for on-line inference ranges from 0.2 to 0.3 milliseconds. This illustrates the extent to which local structure can improve the inference time.

Using the junction tree algorithm, the inference time ranges from 1.059 second to 1.211 as shown in Figure 3 (b), but it does not follow a stationary behavior since $EP_{C\Pi T_i}$ and $EP_{C\Pi T}$ are not influential factors. This experimental result confirms that the junction tree is structure-based. It depends on the network topology and is invariant to parameters.

5 Conclusion

This paper proposed a study of the behavior of encoding strategies in compilation-based inference approaches of min-based possibilistic networks in terms of CNF parameters, compiled based parameters and inference time w.r.t the standard junction tree. Indeed, the impact of both of *local structure* and *possibilistic local structure* was explored.

Our experimental results point out that CNF parameters depend strongly on the used encoding strategy, which takes into consideration numerical values locally or globally. However, the reduction of CNF parameters does not involve more compact compiled bases with less edges and faster inference. Indeed, possibilistic local structure, which deals with equal parameters from a global point of view, rises compiled bases edges. This is especially due to the c2d compiler that

enforces the use of the determinism property to satisfy decomposability and in particular the use of the *case analysis* technique.

A future work is to study in depth transformations of Bayesian networks into possibilistic networks and compare our compilation-based inference approaches to those of possibilistic networks issued from Bayesian networks.

References

1. Ayachi, R., Ben Amor, N., Benferhat, S.: Experimental comparative study of compilation-based inference in bayesian and possibilitic networks. In: Petrosino, A. (ed.) WILF 2011. LNCS, vol. 6857, pp. 155–163. Springer, Heidelberg (2011)
2. Ayachi, R., Amor, N.B., Benferhat, S.: Possibilistic local structure for compiling min-based networks. In: Kruse, R., Berthold, M., Moewes, C., Gil, M.A., Grzegorzewski, P., Hryniewicz, O. (eds.) Synergies of Soft Computing and Statistics. Advances in Intelligent Systems and Computing, vol. 190, pp. 479–487. Springer, Heidelberg (2013)
3. Ayachi, R., Ben Amor, N., Benferhat, S., Haenni, R.: Compiling possibilistic networks: Alternative approaches to possibilistic inference. In: Proceedings of 26th Conference on UAI, California, pp. 40–47. AUAI Press (2010)
4. Ben Amor, N.: Qualitative possibilistic graphical models: From independence to propagation algorithms. PhD thesis, Université de Tunis, Institut supérieur de gestion (2002)
5. Benferhat, S., Dubois, D., Garcia, L., Prade, H.: On the transformation between possibilistic logic bases and possibilistic causal networks. IJAR 29(2), 135–173 (2002)
6. Cadoli, M., Donini, F.M.: A survey on knowledge compilation. AI Communications–The EJAI (10), 137–150 (1998)
7. Chavira, M., Darwiche, A.: Compiling bayesian networks with local structure. In: Proceedings of the 19th IJCAI, pp. 1306–1312 (2005)
8. Darwiche, A.: Decomposable negation normal form. Journal of the ACM 48(4), 608–647 (2001)
9. Darwiche, A.: A logical approach to factoring belief networks. In: Proceedings of the 8th International Conference on KR, Toulouse, pp. 409–420 (2002)
10. Darwiche, A.: New advances in compiling CNF to decomposable negation normal form. In: Proceedings of the 16th ECAI, pp. 328–332 (2004)
11. Darwiche, A., Marquis, P.: A knowledge compilation map. JAIR 17, 229–264 (2002)
12. Dubois, D., Prade, H.: Possibility theory. In: Encyclopedia of Complexity and Systems Science. Springer (2009)
13. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers Inc., San Francisco (1988)

Qualitative Combination of Independence Models

Marco Baioletti[1], Davide Petturiti[1], and Barbara Vantaggi[2]

[1] Dip. Matematica e Informatica, Università di Perugia, Italy
{baioletti,davide.petturiti}@dmi.unipg.it
[2] Dip. S.B.A.I., Università di Roma "La Sapienza", Italy
barbara.vantaggi@sbai.uniroma1.it

Abstract. We deal with the problem of combining sets of independence statements coming from different experts. It is known that the independence model induced by a strictly positive probability distribution has a graphoid structure, but the explicit computation and storage of the closure (w.r.t. graphoid properties) of a set of independence statements is a computational hard problem. For this, we rely on a compact symbolic representation of the closure called fast closure and study three different combination strategies of two sets of independence statements, working on fast closures. We investigate when the complete DAG representability of the given models is preserved in the combined one.

Keywords: Graphoid, Fast closure, Combination of independence models, DAG, P-map.

1 Introduction

A well-known problem concerning Bayesian Networks (BNs) is the identification of a directed acyclic graph (DAG) representing a given set of conditional independencies: this constitutes a compact and intuitive representation of probability distributions and for that it is very attractive for applications. Usually the identification task is carried out from data, by selecting one or more DAGs with a high value according to the chosen scoring criterion. This is possible when joint observations on the variables are available. However, in some applications the available data are just on subsets of variables (instead of on the whole set), so to merge the data we could use expert judgements expressing (conditional) independence statements. Moreover, even if data on all the variables are available, it is fundamental to use, without discarding, expert judgements [3, 4, 6–8, 14, 15]. In particular, we mention the area of multi-agent systems [8] where the agents need to communicate and pool their knowledge represented by BNs (or sets of independencies) and end up with a BN (or a set of independencies) which is a synthesis of the original ones. In [9] some rules are given for building a DAG encoding either all independencies implied by at least an input DAG or only those independencies implied by all input DAGs.

In this paper we address the issue of combining experts opinions about independencies, without necessarily starting from DAGs. This is done in a classical

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 37–48, 2013.

probabilistic context, so we deal with graphoid structures, induced by a strictly positive probability distribution [5]. Note that these structures are also related to other non-additive uncertainty measures, so the interest is not limited to probability theory. The basic idea is to combine graphoid structures, each obtained as the closure of a set J of independence statements. Nevertheless, the construction of the closure $\bar{J}$ is a computational hard problem [13], as its size could be exponentially larger than the size of J, thus we use a compact symbolic representation [2], called *fast closure*, which has been experimentally shown to be computationally efficient.

Concerning the fusion of experts' opinions, two strategies are easily envisaged due to their semantic. The first possibility is to consider only those independencies on which all the experts agree: this corresponds to take the intersection of the related independence models. The resulting model I has always a graphoid structure and an interesting problem is to find a compact representation of the model I in terms of fast closure or by a DAG (the so called P-map).

Another possibility is to take into account also the individual opinions, that is the union U of all the experts' models. In this case, since in general the union of two graphoids is not a graphoid, we can face the problem by finding the "closest" graphoid to U. This approximation can be performed in two different ways. A way consists in looking for the upper approximation of U, i.e., the smallest graphoid containing U: this is equivalent to compute the closure (or better, the fast closure) of U. A second alternative is to search for the lower approximation of U, i.e., the greatest graphoid contained in U. In the lower approximation all the independencies that cannot be derived by at least a single agent are not taken (i.e., an independence, obtained by applying some graphoid properties to two or more independencies coming from different agents, always belongs to the upper approximation, but not to the lower approximation if it falls outside U). As for the intersection, it is interesting to study the representability in terms of fast closure or by a DAG.

The problem of combining independence models in terms of a graphical representation has been faced by many authors, although in some restricted cases [14, 8], while the approach adopted here is rather general. In particular, in [9, 10] the combination of DAG models by means of set-theoretic (union and intersection) operations is investigated. In detail, the aim of [10] is to find a minimal I-map for the intersection of two (or more) DAG models. In [12] a further constraint is added searching for the minimal I-map minimizing the number of parameters in the corresponding BN. In [14], instead, they study how to create an undirected graph representation for the union of two or more independence models. The problem of fusing BNs is also investigated in [8], where an argumentation framework for the negotiation of a common BN is proposed.

The paper is organized as follows. In Section 2 some basic notions are recalled, while in Section 3 the intersection of two fast closures is investigated: the *fc-intersection* operator is defined and a sufficient condition for its complete DAG representability is provided. In Section 4 analogous work is carried on for the union of two fast closures, by means of the *fc-union* and *fc-subunion* operators.

2 Graphoids and Fast Closure

Let $\tilde{S} = \{Y_1, \ldots, Y_n\}$ be a finite non-empty set of variables and $S = \{1, \ldots, n\}$ the set of indices associated to $\tilde{S}$. Given a probability P on $\tilde{S}$, a conditional independence statement $Y_A \perp\!\!\!\perp Y_B | Y_C$, compatible with P, is simply denoted by the ordered triple (A, B, C), where A, B, C are disjoint subsets of S. Then, in the following we do not distinguish $\tilde{S}$ from S.

Denote with $S^{(3)}$ the set of all (ordered) triples (A, B, C) of disjoint subsets of S, such that A and B are not empty. A *conditional independence model* I is therefore a subset of $S^{(3)}$. We refer to a graphoid structure, which is a couple (S, I), where I is a ternary relation on S satisfying the following properties [5]:

G1 if $(A, B, C) \in I$, then $(B, A, C) \in I$ (Symmetry);
G2 if $(A, B \cup C, D) \in I$, then $(A, B, D) \in I$ (Decomposition);
G3 if $(A, B \cup C, D) \in I$, then $(A, B, C \cup D) \in I$ (Weak Union);
G4 if $(A, B, C \cup D) \in I$ and $(A, C, D) \in I$, then $(A, B \cup C, D) \in I$ (Contraction);
G5 if $(A, B, C \cup D) \in I$ and $(A, C, B \cup D) \in I$, then $(A, B \cup C, D) \in I$ (Intersection);

where A, B, C, D are pairwise disjoint subsets of S.

If $\theta = (A, B, C)$, we denote $\mathcal{X} = (A \cup B \cup C)$ while θ^T stands for the triple (B, A, C) obtained from θ through G1.

Given a set $J \subseteq S^{(3)}$ of conditional independence statements compatible with a strictly positive probability, a relevant problem is to find the *closure* of J with respect to graphoid rules G1–G5,

$$\bar{J} = \{\theta \in S^{(3)} : \theta \text{ is obtained from } J \text{ by G1–G5}\}.$$

In particular, in the rest of the paper we simply call *graphoid* a set $J \subseteq S^{(3)}$ such that $\bar{J} = J$. A related problem, called *deduction*, concerns to establish whether a triple $\theta \in S^{(3)}$ can be derived from J through a finite number of application of G1–G5. We stress that, as already pointed out in Section 1, the computation of $\bar{J}$ is in general infeasible both in time and space. An efficient solution to these untreatable problems is given in [2], following the line of the one proposed for *semi-graphoids* (characterized by G1–G4) in [13].

We recall some definitions and results useful in the rest of the paper. Given a pair of triples $\theta_1, \theta_2 \in S^{(3)}$, we say that θ_1 is *generalized-included* in θ_2 (briefly *g-included*), in symbol $\theta_1 \sqsubseteq \theta_2$, if θ_1 can be obtained from θ_2 by a finite number of applications of the unary rules G1, G2 and G3. The definition of g-inclusion between triples can be extended to sets of triples. Indeed, given two sets $H, J \subseteq S^{(3)}$, $H \sqsubseteq J$ if and only if for any triple $\theta \in H$ there exists a triple $\theta' \in J$ such that $\theta \sqsubseteq \theta'$.

A triple $\theta \in J$ is said to be *maximal* in J if there exists no triple $\theta' \in J$, different from θ and θ^T, such that $\theta \sqsubseteq \theta'$.

By using the relation $\sqsubseteq$, it is possible to define a set J_* which is in general much smaller than $\bar{J}$, but having the same information of $\bar{J}$. This set is called the *fast closure* of J and is defined as

$$J_* = \{\tau \in \bar{J} : \tau \text{ is maximal in } \bar{J} \text{ w.r.t } \sqsubseteq\}.$$

In general the set of maximal triples w.r.t. $\sqsubseteq$ of an arbitrary (i.e., not necessarily closed w.r.t. G1–G5) set J, denoted as $J_{/\sqsubseteq}$, carries the same information of J in a more compact form. In [2] we show that J_* can be computed by means of two inference rules G4* and G5*, which are a generalization of rules G4 and G5. In the same paper, it is provided a faster way of computing J_* using a unique inference rule which is defined in terms of the fast closure of two triples $\{\theta_1, \theta_2\}_*$ which can be computed "at once" being composed at most of 9 additional triples. Some experimental results comparing fast closure and closure are reported in [2].

2.1 DAG Representation

A set of conditional independencies can be represented in a compact way by a directed acyclic graph (DAG) [11]. A conditional independence relation (A, B, C) is encoded in a DAG G by the fact that A is *d-separated* from B by C [11]. This relation is denoted by the triple $(A, B, C)_G$. Since the d-separated triples form a graphoid, it makes sense to consider sets of triples closed with respect to properties G1–G5.

Note that it is not always possible to completely represent a probabilistic independence model by a DAG, so the following notions have been introduced in [11]:

Definition 1. *Let J be a set of conditional independence relations on S. A DAG G is a* dependence map *(briefly a D-map) for $\bar{J}$ if for each triple $(A, B, C) \in S^{(3)}$*

$$(A, B, C) \in \bar{J} \Rightarrow (A, B, C)_G.$$

Moreover, G is an independence map *(briefly an I-map) for $\bar{J}$ if for each triple $(A, B, C) \in S^{(3)}$*

$$(A, B, C)_G \Rightarrow (A, B, C) \in \bar{J}.$$

G is a minimal *I-map of $\bar{J}$ if deleting any arc, G is no more an I-map.*

G is said to be a perfect map *(briefly a P-map) for $\bar{J}$ if it is both an I-map and a D-map.*

If the DAG G is a P-map for $\bar{J}$, then there exists an ordering $\pi = \langle \pi_1, \ldots, \pi_n \rangle$ of S such that $\bar{J}$ is obtained as the closure w.r.t. the semi-graphoid properties, of the following *basic triples list* [11]

$$B^G_\pi = \{(\{\pi_i\}, S_{(\pi_i)} \setminus \mathrm{pa}^G(\pi_i), \mathrm{pa}^G(\pi_i)) \in S^{(3)} \ : \ i = 2, \ldots, n\}, \tag{1}$$

where $S_{(\pi_i)} = \{\pi_1, \ldots, \pi_{i-1}\}$, $i = 2, \ldots, n$, and $\mathrm{pa}^G(\pi_i)$ is the set of parents of the node π_i in G.

However, usually an expert is not able to assess a structured set such as a basic triples list or a closure, but rather an arbitrary set of triples. Hence, due to the difficulty of the closure computation we have provided in [1] a necessary and sufficient condition for the existence of a P-map for $\bar{J}$, exclusively relying

on the corresponding fast closure J_*. In order to recall this result we introduce the function Π defined for any $\theta = (A, B, C) \in S^{(3)}$, any $x \in S$ and $T \subseteq S$ as

$$\Pi(\theta, T, x) = \begin{cases} T \cap (A \cup C) & \text{if } C \subseteq T \subseteq A \cup B \cup C \text{ and } x \in A, \\ T \cap (B \cup C) & \text{if } C \subseteq T \subseteq A \cup B \cup C \text{ and } x \in B, \\ T & \text{otherwise.} \end{cases}$$

Theorem 1. *Let $J \subseteq S^{(3)}$ be a set of independence statements. There is a P-map for $\bar{J}$ if and only if there exists an ordering π of S such that for each $\theta = (A, B, C) \in J_*$ the following conditions hold:*

C1 for each $c \in C$ such that $S_{(c)} \cap A \neq \emptyset$ and $S_{(c)} \cap B \neq \emptyset$, there exists a triple $\theta_c \in J_*$ such that $\Pi(\theta_c, S_{(c)}, c) \cap A = \emptyset$ or $\Pi(\theta_c, S_{(c)}, c) \cap B = \emptyset$;
C2 for each $a \in A$ such that $S_{(a)} \cap B \neq \emptyset$ or $S_{(a)} \cap (S \setminus \mathcal{X}) \neq \emptyset$ there exists a triple $\theta_a \in J_*$ such that $\Pi(\theta_a, S_{(a)}, a) \cap [B \cup (S \setminus \mathcal{X})] = \emptyset$;
C3 for each $b \in B$ such that $S_{(b)} \cap A \neq \emptyset$ or $S_{(b)} \cap (S \setminus \mathcal{X}) \neq \emptyset$ there exists a triple $\theta_b \in J_*$ such that $\Pi(\theta_b, S_{(b)}, b) \cap [A \cup (S \setminus \mathcal{X})] = \emptyset$;
C4 for each $c \in C$ such that $S_{(c)} \cap (S \setminus \mathcal{X}) \neq \emptyset$, there exists a triple $\theta'_c \in J_*$ such that $\Pi(\theta'_c, S_{(c)}, c) \cap (S \setminus \mathcal{X}) = \emptyset$.

It is worth to notice that this criterion operates on J_*, instead of $\bar{J}$, thus in the following we say equivalently that a DAG G is a P-map for J_* or for $\bar{J}$.

This characterization of P-mapness can be applied whenever it is feasible to compute the fast closure, while the whole closure can be avoided to be computed (and stored) because of the aforementioned time and memory problems.

3 Intersection of Two Independence Models

The aim of this section is to study how to compute the intersection of two independence models $\bar{J}$ and $\bar{K}$ given in terms of their corresponding fast closures J_* and K_*. Notice that, from a semantic point of view, this process consists in taking only the independence statements common to the two experts.

Since it is known that the intersection of two graphoid structures is a graphoid [10, 9], the intersection of $\bar{J}$ and $\bar{K}$ is still a graphoid, thus we are interested to find the set of its maximal triples directly working with the fast closures J_* and K_*. For this we define the *fc-intersection* operator

$$J_* \sqcap_* K_* = (\bar{J} \cap \bar{K})_{/\sqsubseteq}. \tag{2}$$

We provide now a characterization of $J_* \sqcap_* K_*$ only relying on fast closures J_* and K_*. Given two triples $\theta_1 = (A_1, B_1, C_1), \theta_2 = (A_2, B_2, C_2) \in S^{(3)}$, if $A_1 \cap A_2 \neq \emptyset$, $B_1 \cap B_2 \neq \emptyset$, $C_1 \subseteq A_2 \cup B_2 \cup C_2 = \mathcal{X}_2$, and $C_2 \subseteq A_1 \cup B_1 \cup C_1 = \mathcal{X}_1$, then the triple $\tau = cm(\theta_1, \theta_2) = (A_1 \cap A_2, B_1 \cap B_2, C_1 \cup C_2)$ belongs to $S^{(3)}$ and

$$\tau \sqsubseteq \theta_1, \qquad \tau \sqsubseteq \theta_2.$$

In all the other cases, we set $cm(\theta_1, \theta_2) = \perp$.

Proposition 1. *Given two triples $\theta_1 = (A_1, B_1, C_1)$ and $\theta_2 = (A_2, B_2, C_2)$, the maximal triples of the set $R_{\theta_1,\theta_2} = \{\theta \in S^{(3)} : \theta \sqsubseteq \theta_1 \text{ and } \theta \sqsubseteq \theta_2\}$ are in the set*

$$C_{\theta_1,\theta_2} = \{cm(\theta_1^*, \theta_2^*) \in S^{(3)}\}$$

where θ_i^* $(i = 1, 2)$ stands either for θ_i or θ_i^T.

Proof. If the set R_{θ_1,θ_2} is empty, then $cm(\theta_1^*, \theta_2^*)$ are all $\bot$. Indeed, if there is an element $\theta = (A, B, C)$ in R_{θ_1,θ_2}, then θ is g-included in θ_1 and in θ_2. There are four possible cases implied by the definition of g-inclusion [2]. In the first case, suppose $C_1 \subseteq C \subseteq \mathcal{X}_1$, $A \subseteq A_1$, $B \subseteq B_1$, $C_2 \subseteq C \subseteq \mathcal{X}_2$, $A \subseteq A_2$, $B \subseteq B_2$. As a consequence, $cm(\theta_1, \theta_2) \neq \bot$ and $\theta \sqsubseteq cm(\theta_1, \theta_2)$. Indeed, it is easy to see that $\emptyset \neq A \subseteq A_1 \cap A_2$, $\emptyset \neq B \subseteq B_1 \cap B_2$, $C_1 \subseteq C \subseteq \mathcal{X}_2$, $C_2 \subseteq C \subseteq \mathcal{X}_1$. The other three cases are similar.

Hence, for each element $\theta \in R_{\theta_1,\theta_2}$ there exists an element of $\tau \in C_{\theta_1,\theta_2}$, such that $\theta \sqsubseteq \tau$. Now, let $\bar{\theta}$ be a maximal triple of R_{θ_1,θ_2}. Suppose that $\bar{\theta}$ were not an element of C_{θ_1,θ_2}. Because of the previous argument $\bar{\theta}$ would be g-included in some element of C_{θ_1,θ_2}, contradicting its maximality.

Proposition 2. *Given two sets of independence statements $J, K \subseteq S^{(3)}$, then*

$$J_* \sqcap_* K_* = \left(\bigcup\{C_{\theta_1,\theta_2} : \theta_1 \in J_*, \theta_2 \in K_*\}\right)\Big/_{\sqsubseteq}.$$

Proof. To prove the claim it is sufficient to show that for each element $\theta \in \bar{J} \cap \bar{K}$, there exists an element τ of the form $cm(\theta_1, \theta_2)$, $cm(\theta_1^T, \theta_2)$, $cm(\theta_1, \theta_2^T)$, or $cm(\theta_1^T, \theta_2^T)$ for some $\theta_1 \in J_*, \theta_2 \in K_*$, such that $\theta \sqsubseteq \tau$.

Since $\theta \in \bar{J}$, then there exists an element $\theta_1 \in J_*$, such that $\theta \sqsubseteq \theta_1$. Analogously, since $\theta \in \bar{K}$, then there exists an element $\theta_2 \in K_*$, such that $\theta \sqsubseteq \theta_2$.

Because of Proposition 1, θ is g-included in some element of C_{θ_1,θ_2}. With the same argument, all the maximal triples of $\bar{J} \cap \bar{K}$ must be elements of

$$\bigcup\{C_{\theta_1,\theta_2} : \theta_1 \in J_*, \theta_2 \in K_*\}.$$

Suppose now that J_* and K_* are completely representable by DAGs G_1 and G_2, respectively, i.e., G_1 is a P-map for J_* and G_2 is a P-map for K_*. An interesting problem is to establish (under previous hypothesis) whether also their fc-intersection is completely representable by a DAG. Next example shows this claim does not hold in general.

Example 1. Consider the set $S = \{1, 2, 3, 4\}$ and the fast closures

$$J_* = \{(\{3\}, \{1, 4\}, \{2\}), (\{4\}, \{2, 3\}, \{1\})\},$$
$$K_* = \{(\{2\}, \{3, 4\}, \{1\}), (\{3\}, \{1, 2\}, \{4\})\}.$$

Both J_* and K_* are completely representable by a DAG. Indeed, a DAG representing J_* is $4 \to 1 \to 2 \to 3$, while a DAG representing K_* is $3 \to 4 \to 1 \to 2$. Their fc-intersection is the set $J_* \sqcap_* K_* = \{(\{3\}, \{1\}, \{2, 4\}), (\{4\}, \{2\}, \{1, 4\})\}$ which is not completely representable by a DAG since there does not exist an ordering π satisfying conditions C1–C4 of Theorem 1 for all triples in $J_* \sqcap_* K_*$.

Next proposition gives a sufficient condition for the complete DAG representability of $J_* \sqcap_* K_*$ when J_* and K_* are completely DAG representable.

Proposition 3. *Given two sets of independence statements $J, K \subseteq S^{(3)}$. If there exists an ordering π on S such that conditions C1–C4 hold for every $\theta \in J_*$ and $\theta' \in K_*$, then π satisfies conditions C1–C4 also for all $\tau \in (J_* \sqcap_* K_*)$.*

Proof. Let π be the ordering of S satisfying conditions C1–C4 for J_* and K_*. This ordering π allows to define two DAGs $G_1 = (S, E_1)$ and $G_2 = (S, E_2)$ which are P-maps for $\bar{J}$ and $\bar{K}$, respectively, and their basic triples lists are

$$B_\pi^{G_1} = \{(\{\pi_i\}, S_{(\pi_i)} \setminus \mathrm{pa}^{G_1}(\pi_i), \mathrm{pa}^{G_1}(\pi_i)) \in S^{(3)} \ : \ i = 2, \ldots, n\},$$
$$B_\pi^{G_2} = \{(\{\pi_i\}, S_{(\pi_i)} \setminus \mathrm{pa}^{G_2}(\pi_i), \mathrm{pa}^{G_2}(\pi_i)) \in S^{(3)} \ : \ i = 2, \ldots, n\},$$

where it is

$$\mathrm{pa}^{G_1}(\pi_i) = \min_{\subseteq}\{\Pi(\theta, S_{(\pi_i)}, \pi_i) \ : \ \theta \in J_*\}, \ i = 2, \ldots, n,$$
$$\mathrm{pa}^{G_2}(\pi_i) = \min_{\subseteq}\{\Pi(\theta, S_{(\pi_i)}, \pi_i) \ : \ \theta \in K_*\}, \ i = 2, \ldots, n.$$

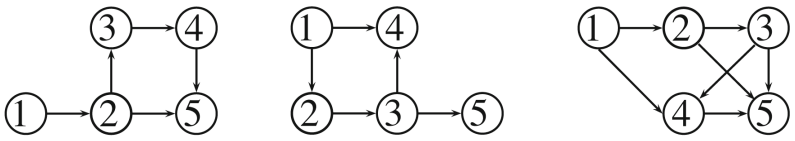
By Theorem 7 in [9] it follows that the graph $G_3 = G_1 \cup G_2 = (S, E_1 \cup E_2)$ is a minimal I-map of $\bar{J} \cap \bar{K}$, for which the corresponding basic triples list is

$$B_\pi^{G_3} = \{(\{\pi_i\}, S_{(\pi_i)} \setminus \mathrm{pa}^{G_3}(\pi_i), \mathrm{pa}^{G_3}(\pi_i)) \in S^{(3)} \ : \ i = 2, \ldots, n\},$$

with $\mathrm{pa}^{G_3}(\pi_i) = \mathrm{pa}^{G_1}(\pi_i) \cup \mathrm{pa}^{G_2}(\pi_i)$, $i = 2, \ldots, n$. Thus it remains to prove that every $\theta \in \bar{J} \cap \bar{K}$ belongs to the closure of $B_\pi^{G_3}$ w.r.t. G1–G4. We have that each triple $\theta = (A, B, C) \in \bar{J} \cap \bar{K}$ can be generated applying a finite number of times properties G1–G4 to some $\tau_{\pi_{i_1}}, \ldots, \tau_{\pi_{i_h}} \in B_\pi^{G_1}$ and some $\rho_{\pi_{j_1}}, \ldots, \rho_{\pi_{j_k}} \in B_\pi^{G_2}$. In general, there could exist several choices for the $\tau_{\pi_{i_s}}$'s and $\rho_{\pi_{j_t}}$'s, but since θ belongs to $\bar{J} \cap \bar{K}$ and for each π_i in the ordering π, $\tau_{\pi_i} = (\{\pi_i\}, B_{\pi_i}, C_{\pi_i}) \in B_{\pi_i}^{G_1}$ and $\rho_{\pi_i} = (\{\pi_i\}, B'_{\pi_i}, C'_{\pi_i}) \in B_{\pi_i}^{G_2}$ with $B_{\pi_i} \cup C_{\pi_i} = B'_{\pi_i} \cup C'_{\pi_i} = S_{(\pi_i)}$, then θ can be obtained starting from a minimal set of basic triples with the same indices in the two basic triples lists, i.e., by $\tau_{\pi_{i_1}}, \ldots, \tau_{\pi_{i_h}} \in B_\pi^{G_1}$ and $\rho_{\pi_{i_1}}, \ldots, \rho_{\pi_{i_h}} \in B_\pi^{G_2}$. In particular, up to property G1, it must hold $A = \bigcup_{s=1}^h \{\pi_{i_s}\}$, $B \subseteq \bigcap_{s=1}^h B_{\pi_{i_j}}$ and $B \subseteq \bigcap_{s=1}^h B'_{\pi_{i_j}}$ that is $B \subseteq \bigcap_{s=1}^h (B_{\pi_{i_j}} \cap B'_{\pi_{i_j}})$, $C \subseteq \bigcup_{s=1}^h C_{\pi_{i_j}} \setminus \bigcup_{s=1}^h \{\pi_{i_j}\}$ and $C \subseteq \bigcup_{s=1}^h C'_{\pi_{i_j}} \setminus \bigcup_{s=1}^h \{\pi_{i_j}\}$ that is $C \subseteq \bigcup_{s=1}^h (C_{\pi_{i_j}} \cup C'_{\pi_{i_j}}) \setminus \bigcup_{s=1}^h \{\pi_{i_j}\}$. This implies that θ can be obtained through G1–G4 also by $\kappa_{\pi_{i_1}}, \ldots, \kappa_{\pi_{i_h}} \in B_\pi^{G_3}$.

Example 2. Let $S = \{1, 2, 3, 4, 5\}$ and consider the fast closures

$$J_* = \{(\{4\}, \{1, 2\}, \{3\}), (\{5\}, \{1, 3\}, \{2, 4\}),$$
$$(\{1\}, \{3, 4, 5\}, \{2\}), (\{3\}, \{1, 5\}, \{2, 4\})\}$$
$$K_* = \{(\{4\}, \{2, 5\}, \{1, 3\}), (\{5\}, \{1, 2, 4\}, \{3\}),$$
$$(\{1\}, \{3, 5\}, \{2\}), (\{2\}, \{4, 5\}, \{1, 3\})\},$$



(a) G_1 (b) G_2 (c) $G_3 = G_1 \cup G_2$

Fig. 1. P-maps of J_*, K_* and $J_* \sqcap_* K_*$

having P-maps G_1 and G_2 (see Figure 1 (a) and (b)), respectively.

For both fast closures, the order $\pi = \langle 1, 2, 3, 4, 5\rangle$ satisfies conditions C1–C4 and is the order associated to G_1 an G_2, thus π satisfies the same conditions also for their fc-intersection

$$J_* \sqcap_* K_* = \{(\{3\}, \{1\}, \{2\}), (\{4\}, \{2\}, \{1, 3\}), (\{5\}, \{1\}, \{2, 3, 4\})\},$$

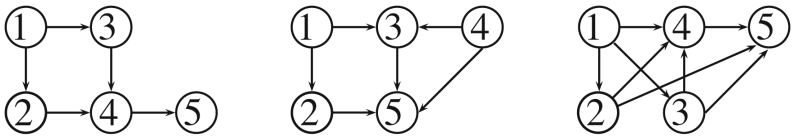
implying its complete DAG representability, in particular, a P-map for $J_* \sqcap_* K_*$ is $G_3 = G_1 \cup G_2 = (S, E_1 \cup E_2)$ in Figure 1 (c).

We stress that the condition expressed in Proposition 3 is only sufficient, i.e., $J_* \sqcap_* K_*$ can be completely DAG representable (assuming J_* and K_* are), even if there does not exist a common ordering π for J_* and K_*, as shown below.

Example 3. Take $S = \{1, 2, 3, 4, 5\}$ and consider the fast closures

$$J_* = \{(\{2\}, \{3\}, \{1\}), (\{5\}, \{1, 2, 3\}, \{4\}), (\{1\}, \{4, 5\}, \{2, 3\})\},$$
$$K_* = \{(\{2\}, \{3, 4\}, \{1\}), (\{5\}, \{1\}, \{2, 3, 4\}), (\{4\}, \{1, 2\}, \emptyset)\},$$

having P-maps G_1 and G_2 (see Figure 2 (a) and (b)), respectively.



(a) G_1 (b) G_2 (c) G_3

Fig. 2. P-maps of J_*, K_* and $J_* \sqcap_* K_*$

The orders on S satisfying C1–C4 for J_* are $\langle 1, a, b, 4, 5\rangle$ and $\langle a, 1, b, 4, 5\rangle$ with $a, b \in \{2, 3\}$, while those related to K_* are $\langle 1, 4, a, b, 5\rangle$ and $\langle 4, 1, a, b, 5\rangle$ with $a, b \in \{2, 3\}$, and $\langle 2, c, d, 3, 5\rangle$ and $\langle c, 2, d, 3, 5\rangle$ with $c, d \in \{1, 4\}$.

Thus no common ordering is present. Nevertheless, their fc-intersection is

$$J_* \sqcap_* K_* = \{(\{2\}, \{3\}, \{1\}), (\{5\}, \{1\}, \{2, 3, 4\})\},$$

for which the order $\pi = \langle 1, 2, 3, 4, 5\rangle$ satisfies conditions C1–C4, and the corresponding P-map is the DAG G_3 shown in Figure 2 (c).

4 Union of Two Independence Models

The fc-intersection conveys all the common opinions of the experts concerning independence statements. Nevertheless, a different combination strategy would take into account also the independence statements proper of each expert, and this can be realized by means of the union operation.

Contrary to the intersection operation, the union of two graphoids is generally not a graphoid, as shown by next example.

Example 4. Consider $S = \{1, 2, 3\}$ and take (we omit symmetric triples) the graphoids $\bar{J} = \{(\{1\}, \{2\}, \{3\})\}$ and $\bar{K} = \{(\{3\}, \{2\}, \{1\})\}$, the union $\bar{J} \cup \bar{K}$ is not a graphoid, indeed it is not closed with respect to G5, as it does not contain the triple $(\{1, 3\}, \{2\}, \emptyset)$. Thus, $(\bar{J} \cup \bar{K}) \subset \overline{(\bar{J} \cup \bar{K})}$.

As a consequence, in order to combine all the qualitative information provided by two experts we need to compute the closure of the union $\overline{(\bar{J} \cup \bar{K})}$. Let us stress that, even if the union $\bar{J} \cup \bar{K}$ is a graphoid, the individual complete DAG representability of $\bar{J}$ and $\bar{K}$ does not assure that also the graphoid $\bar{J} \cup \bar{K}$ is completely DAG representable. In terms of fast closures, this combination strategy corresponds to take $(J_* \cup K_*)_*$ and we briefly call it *fc-union*.

Concerning the complete DAG representability we have the following result.

Proposition 4. *Given two sets of independence statements $J, K \subseteq S^{(3)}$. If there exists an ordering π on S such that conditions C1–C4 hold for every $\theta \in J_*$ and $\theta' \in K_*$, then π satisfies conditions C1–C4 also for all $\tau \in (J_* \cup K_*)_*$.*

Proof. Consider the basic triples lists $B_\pi^{G_1}$ and $B_\pi^{G_2}$ determined by π, built as in the proof of Proposition 3. By Theorem 4 in [9] it follows that the graph $G_3 = G_1 \cap G_2 = (S, E_1 \cap E_2)$ is a minimal I-map of $\overline{(\bar{J} \cup \bar{K})}$, for which the corresponding basic triples list is

$$B_\pi^{G_3} = \{(\{\pi_i\}, S_{(\pi_i)} \setminus \mathrm{pa}^{G_3}(\pi_i), \mathrm{pa}^{G_3}(\pi_i)) \in S^{(3)} \ : \ i = 2, \ldots, n\},$$

with $\mathrm{pa}^{G_3}(\pi_i) = \mathrm{pa}^{G_1}(\pi_i) \cap \mathrm{pa}^{G_2}(\pi_i)$, $i = 2, \ldots, n$. In other terms, this means $\bar{B}_\pi^{G_3} \subseteq \overline{(\bar{J} \cup \bar{K})}$. Thus it remains to prove that every triple in $\overline{(\bar{J} \cup \bar{K})}$ belongs to the closure of $B_\pi^{G_3}$ w.r.t. G1–G4. Nevertheless, it is sufficient to show that each $\theta \in \bar{J}$ and $\theta' \in \bar{K}$ can be obtained through a finite number of applications of G1–G4 from the basic triples list $B_\pi^{G_3}$, since this implies $(\bar{J} \cup \bar{K}) \subseteq \bar{B}_\pi^{G_3}$ and, being $\bar{B}_\pi^{G_3}$ a graphoid, it must be $\overline{(\bar{J} \cup \bar{K})} \subseteq \bar{B}_\pi^{G_3}$. Last claim immediately follows since each basic triple in $B_\pi^{G_1}$ is g-included in the corresponding basic triple in $B_\pi^{G_3}$, and the same holds for each basic triple in $B_\pi^{G_2}$.

Example 5. Consider the fast closures J_* and K_* of Example 2, admitting the common order $\pi = \langle 1, 2, 3, 4, 5\rangle$. Their fc-union is

$$(J_* \cup K_*)_* = \{(\{1\}, \{3, 4, 5\}, \{2\}), (\{4\}, \{1, 2, 5\}, \{3\}), (\{5\}, \{1, 2, 3, 4\}, \emptyset), \\ (\{4, 5\}, \{1, 2\}, \{3\}), (\{3, 4\}, \{1, 5\}, \{2\})\}.$$

As J_* and K_* admit, respectively, the P-maps $G_1 = (S, E_1)$ and $G_2 = (S, E_2)$ corresponding to π, a P-map of $(J_* \cup K_*)_*$ is $G_3 = G_1 \cap G_2 = (S, E_1 \cap E_2)$.

Nevertheless, the fc-union combination can introduce some exogenous information with the only purpose of achieving a structured set, for example it could be derived from G5 on two triples $\theta_1 \in J_*, \theta_2 \in K_*$. Thus a possible alternative is to consider the maximal (with respect to set inclusion) graphoid contained in $\bar{J} \cup \bar{K}$, denoted by $M(\bar{J}, \bar{K})$. The set $M(\bar{J}, \bar{K})$ contains the graphoid $\bar{J} \cap \bar{K}$ and, in some cases, it can coincide either with $\bar{J} \cap \bar{K}$ or with $\bar{J} \cup \bar{K}$ (when it is a graphoid), or with both. However in the most general case $M(\bar{J}, \bar{K})$ can contain some triples of $(\bar{J} \cup \bar{K}) \setminus (\bar{J} \cap \bar{K})$. These additional triples can be characterized as those elements belonging to $\bar{J} \setminus \bar{K}$ (or to $\bar{K} \setminus \bar{J}$) which cannot produce new triples when they are used with the graphoid rules in connection with the elements of $\bar{K} \setminus \bar{J}$ (or $\bar{J} \setminus \bar{K}$, respectively). In other words $\theta \in M(\bar{J}, \bar{K})$ if and only if all the triples that can be obtained from θ alone (through G1–G3) or with some other triple $\tau \in \bar{J} \cup \bar{K}$ (through G4–G5) are still elements of $M(\bar{J}, \bar{K})$.

Being $M(\bar{J}, \bar{K})$ a graphoid, we are interested in finding its maximal triples, starting from the fast closure of J and K. We denote the set $M(\bar{J}, \bar{K})_{/\sqsubseteq}$ by $J_* \sqcup_* K_*$ and we call it *fc-subunion*.

It is easy to prove that $J_* \sqcup_* K_*$ is equal to

$$\Big(\{\tau \in S^{(3)} : \exists \theta \in J_*, \tau \sqsubseteq \theta \text{ and } \forall \rho \in K_*, \{\rho, \tau\}_* \sqsubseteq J_* \cup K_*\} \\ \cup \{\tau \in S^{(3)} : \exists \theta \in K_*, \tau \sqsubseteq \theta \text{ and } \forall \rho \in J_*,\ \{\rho, \tau\}_* \sqsubseteq J_* \cup K_*\}\Big)_{/\sqsubseteq}. \tag{3}$$

Indeed, τ is a maximal triple of $M(\bar{J}, \bar{K})$ if and only if τ generates only triples which are g-included in some other maximal triples of $J_* \cup K_*$.

Let us stress that for any J_* and K_* it always holds $J_* \sqcap_* K_* \sqsubseteq J_* \sqcup_* K_* \sqsubseteq (J_* \cup K_*)_*$. We also notice that, depending on J_* and K_*, it could happen $J_* \sqcap_* K_* = J_* \sqcup_* K_*$ or $J_* \sqcup_* K_* = (J_* \cup K_*)_*$, for example in the case $J_* \sqsubseteq K_*$, then trivially $J_* \sqcap_* K_* = J_*$ and $J_* \sqcup_* K_* = (J_* \cup K_*)_* = K_*$.

The set $J_* \sqcup_* K_*$ can be computed with Algorithm 1 which, in turn, relies on Algorithms 2 and 3. Notice that the procedure FindMaximal finds all the maximal triples of a set of conditional independencies [2].

It is worthwhile to notice that even when $\bar{J}$ and $\bar{K}$ are both completely representable by DAGs, $M(\bar{J}, \bar{K})$ is not necessarily completely representable by a DAG, as shown in the next example.

Example 6. Consider the fast closures J_* and K_* of Example 3. Their fc-union and fc-subunion are

$$(J_* \cup K_*)_* = \{(\{5\}, \{1, 2, 3\}, \{4\}), (\{2\}, \{3, 4, 5\}, \{1\}), (\{3\}, \{2, 5\}, \{1, 4\}),$$

Algorithm 1. Computing the fc-subunion $J_* \sqcup_* K_*$

function FCSubunion(J_*, K_*)
  return FindMaximal(SelectTriples(J_*, K_*)$\cup$ SelectTriples(K_*, J_*))

Algorithm 2. Selecting candidate maximal triples for $M(\bar{J}, \bar{K})$

function SelectTriples(U, V)
  $T \leftarrow \emptyset$
  for all $\theta = (A, B, C) \in U$ **do**
    for all $A' \subset A$ **do**
      for all $A'' \subseteq A'$ **do**
        $\tau \leftarrow (A \setminus A', B, C \cup A'')$
        if Check(τ, U, V) **then** $T \leftarrow T \cup \{\tau\}$
    for all $B' \subset B$ **do**
      for all $B'' \subseteq B'$ **do**
        $\tau \leftarrow (A, B \setminus B', C \cup B'')$
        if Check(τ, U, V) **then** $T \leftarrow T \cup \{\tau\}$
  return T

Algorithm 3. Verifying if τ can be a candidate maximal triple for $M(\bar{J}, \bar{K})$

function Check(τ, C, D)
  for all $\rho \in D$ **do**
    if not $\{\tau, \rho\}_* \sqsubseteq (C \cup D)$ **then return** FALSE
  return TRUE

$$J_* \sqcup_* K_* = \begin{array}{l}(\{1,2\},\{4,5\},\emptyset),(\{4,5\}\{1,2\},\{3\})\},\\ \{(\{2\},\{3,4\},\{1\})(\{5\},\{1\},\{2,3,4\}),(\{4\},\{1,2\},\emptyset),\\ (\{5\},\{3\},\{4\}),(\{5\},\{3\},\{1,4\}),(\{5\},\{1\},\{2,3\})\}.\end{array}$$

Even if J_* and K_* are completely DAG representable (but they do not have a common order satisfying C1–C4), neither $(J_* \cup K_*)_*$ nor $J_* \sqcup_* K_*$ is completely DAG representable.

Next example shows a case where both the sets $J_* \sqcup_* K_*$ and $(J_* \cup K_*)_*$ are completely DAG representable at the same time.

Example 7. Consider $S = \{1,2,3,4\}$, together with $J_* = \{(\{1\},\{2\},\emptyset)\}$ and $K_* = \{(\{3\},\{1,4\},\{2\})\}$. Then we have $J_* \sqcap_* K_* = \emptyset$, while $(J_* \cup K_*)_* = \{(\{1\}, \{2,3\},\emptyset),(\{3\},\{1,4\},\{2\})\}$ and $J_* \sqcup_* K_* = \{(\{1\},\{2\},\emptyset),(\{3\},\{1\},\{2,4\})\}$. It holds that the order $\pi = \langle 1,2,4,3\rangle$ satisfies conditions C1–C4 for all the fast closures, thus they all admit a complete DAG representation.

5 Conclusions

We studied the combination of graphoid structures by means of a compact symbolic representation (fast closure). We defined the intersection operator (fc-intersection) and two operators for the approximation of the union (fc-union and

fc-subunion) working on fast closures. We also provided a sufficient condition for the complete DAG representability of the fc-intersection and the fc-union. The present work has mainly a theoretical objective, we plan a computational analysis of the presented issues in a future work.

References

1. Baioletti, M., Busanello, G., Vantaggi, B.: Acyclic directed graphs representing independence models. Int. J. of App. Reas. 52(1), 2–18 (2011)
2. Baioletti, M., Busanello, G., Vantaggi, B.: Conditional independence structure and its closure: Inferential rules and algorithms. Int. J. of App. Reas. 50, 1097–1114 (2009)
3. Clemen, R.T., Fischer, G.W., Winkler, R.L.: Assessing dependence: Some experimental results. Man. Sci. 46(8), 1100–1115 (2000)
4. Cooke, R.M., Goossens, L.H.J.: TU Delft expert judgment database. Rel. Eng. & Sys. Saf. 93(5), 657–674 (2008)
5. Dawid, A.P.: Conditional independence in statistical theory. J. of the Royal Stat. Soc. B 41, 15–31 (1979)
6. Destercke, S., Chojnacki, E.: Handling dependencies between variables with imprecise probabilistic models. Saf., Rel. and Risk An.: Th., Meth. and App. 1-4, 697–702 (2009)
7. Henrion, M., Breese, J.S., Horvitz, E.J.: Decision analysis and expert systems. AI Mag. 12(4), 64–91 (1991)
8. Nielsen, S.H., Parsons, S.: An application of formal argumentation: Fusing Bayesian networks in multi-agent systems. Art. Int. 171(10-15), 754–775 (2007)
9. del Sagrado, J., Moral, S.: Qualitative Combination of Bayesian Networks. Int. J. of Int. Sys. 18, 237–249 (2003)
10. Matzkevich, I., Abramson, B.: Some complexity considerations in the combination of belief networks. In: Proc. 9th Conf. UAI, pp. 159–165. Morgan Kaufmann, San Francisco (1993)
11. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann, Los Altos (1988)
12. Peña, J.M.: Finding Consensus Bayesian Network Structures. J. of Art. Int. Res. 42, 661–687 (2011)
13. Studeny, M.: Semigraphoids and structures of probabilistic conditional independence. Ann. of Math. and Art. Int. 21, 71–98 (1997)
14. Wong, S.K.M., Butz, C.J.: Constructing the Dependency Structure of a Multiagent Probabilistic Network. IEEE Trans. on Knowl. and Data Eng. 13(3), 395–415 (2001)
15. Xiang, Y.: Verification of DAG Structures in Cooperative Belief Network-Based Multi-agent Systems. Net. 31, 183–191 (1998)

A Case Study on the Application of Probabilistic Conditional Modelling and Reasoning to Clinical Patient Data in Neurosurgery

Christoph Beierle[1], Marc Finthammer[1], Nico Potyka[1], Julian Varghese[1], and Gabriele Kern-Isberner[2]

[1] Dept. of Computer Science, FernUniversität in Hagen, 58084 Hagen, Germany
[2] Dept. of Computer Science, TU Dortmund, 44221 Dortmund, Germany

Abstract. We present a case-study of applying probabilistic logic to the analysis of clinical patient data in neurosurgery. Probabilistic conditionals are used to build a knowledge base for modelling and representing clinical brain tumor data and expert knowledge of physicians working in this area. The semantics of a knowledge base consisting of probabilistic conditionals is defined by employing the principle of maximum entropy that chooses among those probability distributions satisfying all conditionals the one that is as unbiased as possible. For computing the maximum entropy distribution we use the MEcoRe system that additionally provides a series of knowledge management operations like revising, updating and querying a knowledge base. The use of the obtained knowledge base is illustrated by using MEcoRe's knowledge management operations.

1 Introduction

In the medical domain, uncertain rules like "*If symptoms S_1, S_2, and S_3 are present, then there is a probability of 70% that the patient has disease D.*" occur frequently. An intelligent agent providing decision support for performing medical diagnosis and for choosing a therapy must be able to deal with pieces of knowledge expressed by such rules, requiring elaborate knowledge representation and reasoning facilities. For instance, in neurosurgery, such an agent should be able to answer diagnostic questions in the presence of evidential facts like "*Given the evidence that the patient has perceptual disturbances, suffers from unusual pain in the head and that there are symptoms for intracranial pressure, what is the probability that he has a cranialnerve tumor?*", and the agent should be able to perform hypothetical reasoning as in: "*There is evidence that the patient has perceptual disturbances and that there are symptoms for intracranial pressure. If we chose a surgery for therapy and if the correct diagnosis was gliobastoma, what would be the patient's chance to recover completely without any serious complications?*" Moreover, when the agent lives in an uncertain and dynamic environment, she has to adapt her epistemic state constantly to changes in the surrounding world and to react adequately to new demands (cf. [5], [10]).

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 49–60, 2013.

In this paper, we report on a case study on the application of probabilistic modelling and reasoning to clinical patient data in neurosurgery. A knowledge base BT representing and integrating both statistical frequencies of brain tumors reported in the literature as well as physicians' expert beliefs is developed and used to perform reasoning regarding the diagnosis of brain tumor types or the prognosis for patients (see [22,21] for more information on the medical background). Uncertain rules as the first one above are modelled by probabilistic conditionals, formally denoted by $(D|S_1 \wedge S_2 \wedge S_3)[0.7]$. Semantics of such conditionals are given by probability distributions over the possible worlds determined by the underlying propositional variables, and satisfaction of a conditional by a probability distribution P is defined via conditional probability, e.g., P satisfies $(D|S_1 \wedge S_2 \wedge S_3)[0.7]$ iff $P(D|S_1 \wedge S_2 \wedge S_3) = 0.7$. In order to complete any missing or unspecified knowledge, the concept of maximum entropy [16,11] is used. The required reasoning is carried out by the MEcore system [7] that implements reasoning at optimum entropy and provides knowledge management operations required for modelling an intelligent agent.

In the following section, we first recall some preliminaries of probabilistic conditional logic and features of the MEcore system as they are presented in [7]. In Sec. 3, the vocabulary of BT and a first version of this knowledge base is presented. Section 4 introduces revision and update operations for BT, and in Sec. 5 we illustrate the reasoning facilities for prognosis and hypothetical what-if-analysis, demonstrating that the results are well in accordance with a clinical physician's point of view. In Sec. 6, we conclude and point out further work.

2 Background: Probabilistic Conditionals and MEcore

2.1 Probabilistic Conditional Logic in a Nutshell

We start with a propositional language $\mathcal{L}$, generated by a finite set Σ of (binary) atoms $a, b, c, \ldots$. The formulas of $\mathcal{L}$ will be denoted by uppercase Roman letters $A, B, C, \ldots$. For conciseness of notation, we will omit the logical *and*-connector, writing AB instead of $A \wedge B$, and over-lining formulas will indicate negation, i.e. $\overline{A}$ means $\neg A$. Let Ω denote the set of possible worlds over $\mathcal{L}$; Ω will be taken here simply as the set of all propositional interpretations over $\mathcal{L}$ and can be identified with the set of all complete conjunctions over Σ. For $\omega \in \Omega$, $\omega \models A$ means that the propositional formula $A \in \mathcal{L}$ holds in the possible world ω.

By introducing a new binary operator $|$, we obtain the set $(\mathcal{L} \mid \mathcal{L}) = \{(B|A) \mid A, B \in \mathcal{L}\}$ of (unquantified) *conditionals* (or *rules*) over $\mathcal{L}$. $(B|A)$ formalizes "*if* A *then* B" and establishes a plausible, probable, possible etc connection between the *antecedent* A and the *consequent* B. We will use $Sen_{\mathcal{C}}$ to denote the set of all *probabilistic conditionals* (or *probabilistic rules*) of the form $(B|A)[x]$ where x is a probability value $x \in [0, 1]$.

To give appropriate semantics to conditionals, they are usually considered within richer structures such as *epistemic states*. Besides certain (logical) knowledge, epistemic states also allow the representation of e.g. preferences, beliefs, assumptions of an intelligent agent. Basically, an epistemic state allows

one to compare formulas or worlds with respect to plausibility, possibility, necessity, probability etc. In a quantitative framework, most appreciated representations of epistemic states are provided by *probability functions* (or *probability distributions*) $P : \Omega \to [0,1]$ with $\sum_{\omega\in\Omega} P(\omega) = 1$. Thus, in this setting, the set of *epistemic states* we will consider is $EpState = \{P \mid P : \Omega \to [0,1]$ is a probability function$\}$. The probability of a formula $A \in \mathcal{L}$ is given by $P(A) = \sum_{\omega\models A} P(\omega)$, and the probability of a conditional $(B|A) \in (\mathcal{L} \mid \mathcal{L})$ with $P(A) > 0$ is defined as $P(B|A) = P(AB)/P(A)$, the corresponding conditional probability. Conditionals are interpreted via conditional probability. So the satisfaction relation $\models_{\mathcal{C}} \subseteq EpState \times Sen_{\mathcal{C}}$ of probabilistic conditional logic is defined by $P \models_{\mathcal{C}} (B|A)\,[x]$ iff $P(B|A) = x$.

2.2 Epistemic States and Belief Management Operations

Initialization. First, a prior epistemic state has to be built up on the basis of which the agent can start her computations. If no knowledge at all is at hand, simply the uniform epistemic state is taken to initialize the system. In our probabilistic setting, this corresponds to the uniform distribution where each possible world is assigned the same probability. If, however, a set of probabilistic rules is at hand to describe the problem area under consideration, an epistemic state has to be found to appropriately represent this prior knowledge. To this end, we assume an inductive representation method to establish the desired connection between sets of sentences and epistemic states. Whereas generally, a set R of sentences allows a (possibly large) set of models (or epistemic states), in an inductive formalism we have a function $inductive : \mathcal{P}(Sen_{\mathcal{C}}) \to EpState$ (where $\mathcal{P}(S)$ denotes the power set of S) such that $inductive(R)$ selects a unique, "best" epistemic state from all those states satisfying R.

In the probabilistic framework, the *principle of maximum entropy* associates to a set R of probabilistic conditionals the unique distribution $P^* = MaxEnt(R)$ that satisfies all conditionals in R and has maximal entropy, i.e., $MaxEnt(R)$ is the *unique* solution to the maximization problem

$$\arg\max_{P'\models R} H(P') \quad \text{with } H(P') = -\sum_{\omega} P'(\omega)\log P'(\omega) \tag{1}$$

The rationale behind this is that $MaxEnt(R)$ represents the knowledge given by R most faithfully, i.e. without adding information unnecessarily (cf. [16,11]).

Example 1. Consider the three propositional variables s - *being a student*, y - *being young*, and u - *being unmarried*. *Students* and *unmarried people* are mostly *young*. This commonsense knowledge an agent may have can be expressed probabilistically e.g. by the set $R = \{(y|s)[0.8], (y|u)[0.7]\}$ of conditionals. The *MaxEnt*-representation $P^* = MaxEnt(R)$ computed by MEcoRe is:

ω	$P^*(\omega)$	ω	$P^*(\omega)$	ω	$P^*(\omega)$	ω	$P^*(\omega)$
syu	0.1950	$sy\overline{u}$	0.1758	$s\overline{y}u$	0.0408	$s\overline{y}\,\overline{u}$	0.0519
$\overline{s}yu$	0.1528	$\overline{s}y\overline{u}$	0.1378	$\overline{s}\,\overline{y}u$	0.1081	$\overline{s}\,\overline{y}\,\overline{u}$	0.1378

Querying an Epistemic State. Querying an agent about her beliefs amounts to pose a set of unquantified sentences and asking for the corresponding degrees of belief with respect to her current epistemic state.

Example 2. Suppose the current epistemic state is $currState = MaxEnt(R)$ from Ex. 1, and our question is "*What is the probability that unmarried students are young?*", i.e. the set of queries is $\{(y|su)\}$. MEcoRe returns $\{(y|su)[0.8270]\}$, that is, unmarried students are supposed to be young with probability 0.8270.

New Information and Belief Change. Belief revision, the theory of dynamics of knowledge, has been mainly concerned with propositional beliefs for a long time. The most basic approach here is the *AGM-theory* presented in the seminal paper [1] as a set of postulates outlining appropriate revision mechanisms in a propositional logical environment. This framework has been widened by Darwiche and Pearl [5] for (qualitative) epistemic states and conditional beliefs. An even more general approach, unifying revision methods for quantitative and qualitative representations of epistemic states, is described in [12]. The crucial meaning of conditionals as *revision policies* for belief revision processes is made clear by the so-called *Ramsey test*, according to which a conditional $(B|A)$ is accepted in an epistemic state Ψ, iff revising Ψ by A yields belief in B: $\Psi \models (B|A) \quad \text{iff} \quad \Psi * A \models B$ where $*$ is a belief revision operator (see e.g. [8]).

Note, that the term "belief revision" is a bit ambiguous: On the one hand, it is used to denote quite generally *any* process of changing beliefs due to incoming new information [8]. On a more sophisticated level, however, one distinguishes between different kinds of belief change. Here, *(genuine) revision* takes place when new information about a static world arrives, whereas *updating* tries to incorporate new information about a (possibly) evolving, changing world [10]. Further belief change operators are *expansion*, *focusing*, *contraction*, and *erasure* (cf. [8,6,10]). In the following, we will use the general approach to belief change developed in [12] where belief change is considered in a very general and advanced form: Epistemic states are revised by sets of conditionals – this exceeds the classical AGM-theory by far which only deals with sets of propositional beliefs.

In the probabilistic framework, a powerful operator to change probability distributions by sets of probabilistic conditionals is provided by the *principle of minimum cross-entropy* which generalizes the principle of maximum entropy in the sense of (1): Given a (prior) distribution P and a set R of probabilistic conditionals, the *MinCEnt-distribution* $P^* = MinCEnt(P, R)$ is the *unique* distribution that satisfies all constraints in R and has minimal cross-entropy H_{ce} with respect to P, i.e. P^* solves the minimization problem

$$\arg\min_{P' \models R} H_{ce}(P', P) \quad \text{with } H_{ce}(P', P) = \sum_{\omega} P'(\omega) \log \frac{P'(\omega)}{P(\omega)} \tag{2}$$

If R is basically compatible with P (i.e. *P-consistent*, cf. [12]), then P^* is guaranteed to exist (for further information and lots of examples, see [4,16,12]). The cross-entropy between two distributions can be taken as a directed (i.e. asymmetric) information distance [19] between these two distributions. Following the

principle of minimum cross-entropy means to modify the prior epistemic state P in such a way as to obtain a new distribution P^* which satisfies all conditionals in R and is as close to P as possible. So, the *MinCEnt*-principle yields a probabilistic belief change operator, associating to each probability distribution P and each P-consistent set R of probabilistic conditionals a revised distribution $P^* = \mathit{MinCEnt}(P, R)$ in which R holds. In [13] it is shown how both revision and update can be based on such a belief change operator, and the corresponding conceptual agent model MECoRe which realizes this approach is described in [2].

Example 3. Suppose that some time later, the relationships in the population from Example 1 between students and young people have changed, so that students are young with a probability of 0.9. In order to incorporate this new knowledge, the agent applies an updating operation to modify P^* appropriately. The result $P^{**} = \mathit{MinCEnt}(P^*, \{(y|s)[0.9]\})$ as determined by MECoRe is:

ω	$P^{**}(\omega)$	ω	$P^{**}(\omega)$	ω	$P^{**}(\omega)$	ω	$P^{**}(\omega)$
syu	0.2151	$sy\overline{u}$	0.1939	$s\overline{y}u$	0.0200	$s\overline{y}\,\overline{u}$	0.0255
$\overline{s}yu$	0.1554	$\overline{s}y\overline{u}$	0.1401	$\overline{s}\,\overline{y}u$	0.1099	$\overline{s}\,\overline{y}\,\overline{u}$	0.1401

It is easily checked that indeed, $P^{**}(y|s) = 0.9$ (taking rounding into account).

Diagnosis. Diagnosing a given case is one of the most common operations in knowledge based systems. Given some case-specific *evidence* E (formally, a set of quantified facts), diagnosis assigns degrees of belief to the atomic propositions D to be *diagnosed* (formally, D is a set of unquantified atomic propositions). Thus, making a diagnosis in the light of some given evidence corresponds to determine what is believed in the state obtained by focusing the current state P on the given evidence, i.e. querying the epistemic state $\mathit{MinCEnt}(P, E)$ with respect to D. Thus, here focusing corresponds to conditioning P with respect to the given evidence E.

Example 4. Let $\mathit{currState} = P^*$ from Ex. 1. If there is now certain evidence for being a student and being unmarried – i.e. $E = \{su[1]\}$ – and we ask for the degree of belief of being young – i.e. $D = \{y\}$ –, MECoRe computes $\{y[0.8270]\}$. Thus, if there is certain evidence for being an unmarried student, then the degree of belief for being young is 0.8270.

What-If-Analysis: Hypothetical Reasoning. Hypothetical reasoning asks for the degree of belief of complex relationships (goals) under some hypothetical assumptions. This is useful, e. g., to exploit in advance the benefits of some expensive or intricate medical investigations. Note that whereas in the diagnostic case both evidence E and diagnoses D are just simple propositions, in hypothetical reasoning both the *assumptions* A (formally, a set of quantified conditionals) as well as the *goals* G (formally, a set of unquantified conditionals) may be sets of full conditionals. However, since its underlying powerful *MinCEnt*-update operator can modify epistemic states by arbitrary sets of conditionals, MECoRe can handle hypothetical what-if-analysis structurally analogously to the diagnostic case, i. e. by querying the epistemic state $\mathit{focussed\_state} = \mathit{MinCEnt}(P, A)$ with

respect to G where P is the current epistemic state. Since this is hypothetical reasoning, the agent's current epistemic state remains unchanged.

Example 5. Given $currState = P^*$ from Ex. 1 as present epistemic state, a hypothetical reasoning question is given by: "What would be the probability of being young under the condition of being unmarried – i.e. $G = \{(y|u)\}$ –, provided that the probability of a student being young changed to 0.9 – i.e. $A = \{(y|s)[0.9]\}$?" MEcore's answer is $\{(y|u)[0.7404]\}$ which corresponds to the probability given by P^{**} from Ex. 3.

2.3 The MEcore System

The main objective of MEcore is to implement probabilistic reasoning at optimum entropy and to support advanced belief management operations like revision, update, diagnosis, or what-if-analysis in a most flexible and easily extendable way. MEcore is implemented in Java and uses a straight-forward, direct implementation of a well-known *MinCEnt* algorithm, computing the distribution $P^* = MinCEnt(P, R)$ in an iterative way [4], and provides a powerful and flexible interface. MEcore can be controlled by a text command interface or by scripts, i. e. text files that allow the batch processing of command sequences. These scripts and the text interface use a programming language-like syntax that allows to define, manipulate and display variables, propositions, rule sets and epistemic states. The following example shows a way to generate an epistemic state using the initialize and update operators:

```
//define a set of rules
kb := ((y|s)[0.8], (y|u)[0.7]);
// initialize an epistemic state with these rules
currState := epstate().initialze(kb);
//query and output current belief in the conditional (y|su)
currState.query((y|su));
//update the epistemic state currState by (y|s)[0.9]
currState.update((y|s)[0.9]);
```

Hence, one is able to use both previously defined rule sets and rules that are entered just when they are needed, and combinations of both. The ability to manipulate rule sets, to automate sequences of updates and revisions, and to output selected results for comparing, yields a very expressive command language. This command language is a powerful tool for experimenting and testing with different setups. All core functions of the MEcore system are also accessible through a software interface in terms of a Java API; thus, MEcore can easily be extended by a GUI or be integrated into another software application.

There are many systems performing inferences in probabilistic networks, especially in Bayesian networks. One system built upon network techniques to implement reasoning at optimum entropy is the expert system shell Spirit [18]. Graph based methods are known to feature a very efficient representation of probability distributions via junction trees and hypergraphs, while MEcore works on a model based representation of probabilities. While this is clearly inefficient, the

aim of MEcoRe is to implement subjective probabilistic reasoning, as it could be performed by agents, making various belief operations possible. In particular, it allows changing of beliefs in a very flexible way by taking new, complex information into account. This is not possible with graph based systems for probabilistic inference, as efficient methods of restructuring probabilistic networks still have to be developed.

3 BT: Modelling Clinical Brain Tumor Data

For generating an initial knowledge base for clinical brain tumor data we will use various binary and multi-valued variables considering aspects of the patient, the patient's anamnesis, the observed symptoms, the possible diagnosis, etc; a medical justification for these variables and their values along with references to the relevant medical literature is given in [21,22]. Since the prevalence of different tumor types varies with the age of patients, the variable `age` distinguishes patients with respect to the three values `le20` (less or equal 20 years old), `20to80` (between 20 and 80 years), and `ge80` (greater or equal 80 years). The binary variable `warningSymptoms` is true iff warning symptoms like perceptual disturbances or unusual pain in the head are present. Given results of a magnetic resonance tomography (MRT), the variable `malignancy` corresponds to the assumed malignancy of the tumor with respect to the WHO grading system [14]; a higher index corresponds to a higher malignancy. The binary variable `icpSymptoms` indicates whether MRT results provide symptoms for intracranial pressure (ICP). The preoperative physical fitness of patients is evaluated by the ASA (American Society of Anesthesiologists) classification system represented by the variable `ASA`. It is associated with perioperative risks, and a higher value indicates a higher risk. Only the first four states are considered here, as treatment of a brain tumor is of low priority for a higher value. Thus, so far we have:

```
                  age : le20, 20to80, ge80
     warningSymptoms : true, false
           malignancy : 1, 2, 3, 4, other
         icpSymptoms : true, false
                  ASA : 1, 2, 3, 4
```

In `BT`, the ten most common brain tumor types like gliomas and meningiomas [17] are taken into account. Together with the value `other` for any other tumor types, these brain tumor types constitute the values of the variable `diagnosis`:

```
diagnosis : pilocytic-astrocytoma, diffuse-astrocytoma,
            anaplastic-astrocytoma, glioblastoma,
            oligodendroglioma, ependymoma, meningeoma,
            medulloblastoma, cranialnerve-tumor,
            metastatic-tumor, other
```

Finally, there are three variables denoting the therapy, possible complications, and the expected health of the patient. The variable

```
therapy : conservative, surgery, none
```

diagnosis	Adults	Children
glioma		
- glioblastoma	15%	*unspecified*
- pilocytic-astrocytoma	*unspecified*	35%
- diffuse-astrocytoma	10%	*unspecified*
- anaplastic-astrocytoma	10%	*unspecified*
- oligodendroglioma	10%	*unspecified*
- ependymoma	4%	8%
meningeoma	20%	*unspecified*
medulloblastoma	7%	25%
cranialnerve-tumor	7%	*unspecified*
metastatic-tumor	10%	*unspecified*
other	*unspecified*	*unspecified*

Fig. 1. Empirical frequencies of brain tumor types, where *unspecified* stands for rare or unknown (collected from [3,9,15,20])

refers to the therapy to be chosen. We distinguish a conservative therapy without surgery, surgery, or no therapy at all. Possible complications during an inpatient stay are expressed by the variable

$$\texttt{complication}: \ 1, 2, 3$$

which distinguishes the three stages 1 (no complications or minor, completely reversible complications like temporary pain after surgery), 2 (medium or heavy complications with uncertain reversibility like neurological or other functional disorders), and 3 (life-threatening complications like serious internal bleeding or neurological deficits at the risk of brain death). Thus, higher values correspond to more serious complications. The expected health of the patient after inpatient stay is denoted by:

$$\texttt{prognosis}: \ \texttt{very\_good}, \texttt{good}, \texttt{intermediate}, \texttt{poor}, \texttt{very\_poor}$$

The knowledge base BT uses these nine propositional variables as its vocabulary to represent clinical brain tumor data and corresponding expert knowledge. Note that although we have only 9 variables, due to the multiple values they induce $2^2 \times 3^3 \times 4 \times 5^2 \times 11 = 118.800$ possible worlds.

There are various publications containing empirical frequencies of certain brain tumor types. For our initial version of our knowledge base BT, we encode the frequencies given in Fig. 1 that are collected from [3,9,15,20] and that are given relative to the patient being an adult (age=20to80 or age=ge80) or being a child (age=le20). The representation of these frequencies is given by conditionals of the following type

```
(diagnosis=meningeoma | !(age=le20))[0.20]          (3)
(diagnosis=medulloblastoma | !(age=le20))[0.07]     (4)
(diagnosis=cranialnerve-tumor | !(age=le20))[0.07]  (5)
(diagnosis=metastatic-tumor | !(age=le20))[0.10]    (6)
```

where, using the input syntax of MEcore, ! denotes negation. Additionally, BT contains the probabilistic facts (age=le20)[0.15] and (age=20to80)[0.62] reflecting the age distribution in Germany in the year 2009.

Note that there are some missing frequencies in Fig. 1, and thus, there are no conditionals in `BT` for these missing frequencies. In order to obtain a full probability distribution over all variables and their values, the missing knowledge is completed in an information-theoretically optimal way by employing the ME principle, thus by being as unbiased as possible with respect to each diagnosis with unspecified probability. In MEcoRe, the computation of an epistemic state incorporating the knowledge given by `BT` is started by

```
cmd-1:  currState := epstate.initialize(BT);
```

so that `currState` denotes the ME distribution over `BT`.

In order to be able to ask a set of queries instead of just a single query at the same time, MEcoRe allows the introduction of an identifier to denote a set of queries. Here, we will illustrate this feature with a singleton set containing an unquantified conditional for the diagnosis under the premise that the patient is older than 80 and that he suffers from warning symptoms

```
cmd-2:  queriesBT := (diagnosis|(age=ge80) ∧ warningSymptoms);
cmd-3:  currState.query(queriesBT);
```

which yields the following probabilities:

diagnosis	*probability*	*diagnosis*	*probability*
`glioblastoma`	0.150	`meningeoma`	0.200
`pilocytic-astrocytoma`	0.035	`medulloblastoma`	0.070
`diffuse-astrocytoma`	0.100	`cranialnerve-tumor`	0.070
`anaplastic-astrocytoma`	0.100	`metastatic-tumor`	0.100
`oligodendroglioma`	0.100	`other`	0.035
`ependymoma`	0.040		

Note that up to now, `BT` does not contain any information about the influence of warning symptoms or the observation that the patient is more than 80 years old. Therefore, in the ME distribution given by `currState`, the corresponding premise given in the queries in `queriesBT` (cf. command line `cmd-2`) does not cause a deviation from the probabilities given in the original conditionals in `BT` and taken from Fig. 1. Note also that the prababilities for the two possible diagnosis values `pilocytic-astrocytoma` and `other` missing for adults in Fig. 1 have also been computed as expected.

4 Revising and Updating `BT`

Besides available statistical data, another important knowledge source is the clinical expert knowledge of a physician. For example, for adults, Fig. 1 tells us that the most frequently appearing glioma tumor type is `glioblastoma`, but no information is provided about its probability given specific symptoms. An experienced physician working with brain tumor patients might state the following conditionals expressing his expert beliefs about the probability of a `glioblastoma` given various observations:

```
(diagnosis=glioblastoma | !(age=le20) ∧ warningSymptoms)[0.20]   (7)
```

```
(diagnosis=glioblastoma|!(age=le20) ∧ icpSymptoms)[0.20]        (8)
(diagnosis=glioblastoma|!(age=le20) ∧ (malignancy=4))[0.40]     (9)
(diagnosis=glioblastoma|!(age=le20) ∧ (malignancy=3))[0.10]    (10)
(diagnosis=glioblastoma|!(age=le20) ∧ (malignancy=2))[0.05]    (11)
(diagnosis=glioblastoma|!(age=le20) ∧ (malignancy=1)[0.01]     (12)
```

Taking into account only Fig. 1, the probability for `glioblastoma` is 15%. Therefore, given the respective preconditions, rules (7) - (9) would increase the probability, whereas rules (10) - (12) would decrease it.

In [21], about 90 conditionals expressing such expert knowledge from a physician's point of view are formulated. With `expertBT` denoting the set of these conditionals, we will incorporate this new knowledge into the current epistemic state. We can achieve this in such a way as if it had been available already in the original knowledge base `BT` by a kind of belief change called *genuine revision* (cf. Sec. 2 and [13,2]). In MEcoRe, this is easily expressed by

```
cmd-4:  currState.revise(expertBT);
```

Now, asking the `queriesBT` (cf. command line `cmd-2`) again, the probabilities have changed considerably in the new epistemic state:

diagnosis	*probability*	*diagnosis*	*probability*
glioblastoma	0.223	meningeoma	0.156
pilocytic-astrocytoma	0.050	medulloblastoma	0.065
diffuse-astrocytoma	0.098	cranialnerve-tumor	0.057
anaplastic-astrocytoma	0.106	metastatic-tumor	0.106
oligodendroglioma	0.086	other	0.011
ependymoma	0.039		

E.g., the probability for `glioblastoma` increased from 15% to 22.3%, while the probability for `meningeoma` decreased from 20% to 15.6%. This is well in accordance with the observations made by physicians working in this area [21].

Now suppose that later on, experts think that the probabilities of conditionals (7) - (9) should be changed to 0.15%, 0.25%, and 0.45%, respectively, and let `gliobNew` denote these three modified conditionals. Genuine revision of the current epistemic state with `gliobNew` would lead to an inconsistency since (7) - (9) and `gliobNew` cannot be satisfied simultaneously. However, MEcoRe's update operation of `currState` by `gliobNew` can incorporate the new knowledge in the current epistemic state by choosing the distribution satisfying `gliobNew` and having minimum cross entropy with respect to `currState` (cf. [13,2]). Note that update is the more appropriate operation here, since the shift of the probabilities reflects a changed environment.

5 Prognosis and What-If-Analysis

For the real documented case of a patient being older than 80 years, with `warningSymptoms`, `icpSymptoms`, and `malignancy=4`, asking MEcoRe results in a probability of 55.6% for the diagnosis glioblastoma, being very plausible from

a physician's point of view. Assuming that glioblastoma were indeed the correct diagnosis and assuming further that a surgery would be chosen, the prognosis for complications that might occur are determined by:

```
cmd-5:  whatIfQ := (complication|(diagnosis|(age=ge80) ∧
                warningSymptoms ∧ icpSymptoms ∧ malignancy=4));
cmd-6:  hypothesis := ((diagnosis=glioblastoma)[1.0],
                (therapy=surgery)[1.0]);
cmd-7:  currState.whatif(hypothesis,whatIfQ);
```

Note that what-if is similar to an update except that it does not change the current belief state. The resulting probabilities for complications of grade 1, 2, and 3 are 0.4%, 45.4%, and 54.2%, respectively. While complications of grade 2 or 3 are rare in general, the provided evidence and the given assumptions caused MEcoRe to rise the probabilities for these types of complications considerably. After surgical treatment of the given patient, there was indeed a complication of grade 2. From a clinical perspective, the probabilities for `complication` computed by MEcoRe is an adequate warning; however, the probability for grade 3 is a bit too pessimistic, since compared to similar patient-risk constellations, life-threatening complications are frequent, but less than 50%. Here, a corresponding adaptation of the conditionals constraining the probabilities for grade 3 complications might lead to a more realistic probability value for this query. Further types of queries for `BT` asking MEcoRe for the expected health of patients after inpatient stay, returned a very realistic prognosis from a medical point of view [21]. An example for what-if-analysis where the assumptions are not just facts with probability 1.0 (as in `cmd-7`) is given by `currState.whatif(gliobNew,whatIfQ)`, asking for the probability of `whatIfQ` in the current epistemic state under the assumption that the conditionals in `gliobNew` (cf. end of Sec. 4) hold.

6 Conclusions and Further Work

We reported on a case study using probabilistic logic and the principle of maximum entropy to model clinical brain tumor data and medical expert knowledge in neurosurgery. The knowledge base `BT` contains approximately 110 probabilistic conditionals over 9 multi-valued variables that medical experts identified to be at the core of clinical brain tumor data analysis. Using MEcoRe for working with `BT` produced realistic probabilities for diagnosis and prognosis from a clinical physician's point of view. We are currently working on extending `BT`, taking into account additional variables and further refining the medical modelling.

References

1. Alchourrón, C.E., Gärdenfors, P., Makinson, P.: On the logic of theory change: Partial meet contraction and revision functions. Journal of Symbolic Logic 50(2), 510–530 (1985)

2. Beierle, C., Kern-Isberner, G.: A conceptual agent model based on a uniform approach to various belief operations. In: Mertsching, B., Hund, M., Aziz, Z. (eds.) KI 2009. LNCS (LNAI), vol. 5803, pp. 273–280. Springer, Heidelberg (2009)
3. Bruch, H.P., Trentz, O.: Berchthold Chirurgie, 6. Auflage. Elsevier GmbH (2008)
4. Csiszár, I.: I-divergence geometry of probability distributions and minimization problems. Ann. Prob. 3, 146–158 (1975)
5. Darwiche, A., Pearl, J.: On the logic of iterated belief revision. Artificial Intelligence 89, 1–29 (1997)
6. Dubois, D., Prade, H.: Focusing vs. belief revision: A fundamental distinction when dealing with generic knowledge. In: Nonnengart, A., Kruse, R., Ohlbach, H.J., Gabbay, D.M. (eds.) FAPR 1997 and ECSQARU 1997. LNCS, vol. 1244. Springer, Heidelberg (1997)
7. Finthammer, M., Beierle, C., Berger, B., Kern-Isberner, G.: Probabilistic reasoning at optimum entropy with the MEcore system. In: Lane, H.C., Guesgen, H.W. (eds.) Proc. FLAIRS 2009. AAAI Press, Menlo Park (2009)
8. Gärdenfors, P.: Knowledge in Flux: Modeling the Dynamics of Epistemic States. MIT Press, Cambridge (1988)
9. Hosten, N., Liebig, T.: Computertomografie von Kopf und Wirbelsäule. Georg Thieme Verlag (2007)
10. Katsuno, H., Mendelzon, A.O.: On the difference between updating a knowledge base and revising it. In: Proceedings Second International Conference on Principles of Knowledge Representation and Reasoning, KR 1991, pp. 387–394. Morgan Kaufmann, San Mateo (1991)
11. Kern-Isberner, G.: Characterizing the principle of minimum cross-entropy within a conditional-logical framework. Artificial Intelligence 98, 169–208 (1998)
12. Kern-Isberner, G.: Conditionals in nonmonotonic reasoning and belief revision. LNCS (LNAI), vol. 2087. Springer, Heidelberg (2001)
13. Kern-Isberner, G.: Linking iterated belief change operations to nonmonotonic reasoning. In: Brewka, G., Lang, J. (eds.) Proceedings 11th International Conference on Knowledge Representation and Reasoning, KR 2008, pp. 166–176. AAAI Press, Menlo Park (2008)
14. Louis, D.N., Ohgaki, H., Wiestler, O.D., Cavenee, W.K., Burger, P.C., Jouvet, A., Scheithauer, B.W., Kleihues, P.: The 2007 WHO Classification of Tumours of the Central Nervous System. Acta Neuropathologica 114(2), 97–109 (2007)
15. Müller, M.: Chirurgie für Studium und Praxis, 9. Auflage. Medizinische Verlags- und Informationsdienste (2007)
16. Paris, J.B., Vencovska, A.: In defence of the maximum entropy inference process. International Journal of Approximate Reasoning 17(1), 77–103 (1997)
17. Park, B.J., Kim, H.K., Sade, B., Lee, J.H.: Epidemiology. In: Lee, J.H. (ed.) Meningiomas: Diagnosis, Treatment, and Outcome, p. 11. Springer (2009)
18. Rödder, W., Reucher, E., Kulmann, F.: Features of the expert-system-shell SPIRIT. Logic Journal of the IGPL 14(3), 483–500 (2006)
19. Shore, J.E.: Relative entropy, probabilistic inference and AI. In: Kanal, L.N., Lemmer, J.F. (eds.) Uncertainty in Artificial Intelligence, pp. 211–215. North-Holland, Amsterdam (1986)
20. Steiger, H.-J., Reulen, H.J.: Manual Neurochirurgie. Ecomed Medizin (2006)
21. Varghese, J.: Using probabilistic logic for the analyis and evaluation of clinical patient data in neurosurgery. B.Sc. Thesis, Univ. Hagen (2012) (in German)
22. Varghese, J., Beierle, C., Potyka, N., Kern-Isberner, G.: Using probabilistic logic and the principle of maximum entropy for the analysis of clinical brain tumor data. In: Proc. CBMS 2013. IEEE Press, New York (to appear 2013)

Causal Belief Networks: Handling Uncertain Interventions

Imen Boukhris[1,2], Salem Benferhat[2], and Zied Elouedi[1]

[1] LARODEC, Université de Tunis, ISG de Tunis, Tunisia
[2] CRIL, Université d'Artois, Faculté Jean Perrin, France
imen.boukhris@hotmail.com, benferhat@cril.univ-artois.fr,
zied.elouedi@gmx.fr

Abstract. Eliciting the cause of an event will be easier if an agent can directly intervene on some variables by forcing them to take a specific value. The state of the target variable is therefore totally dependent of this external action and independent of its original causes. However in real world applications, performing such perfect interventions is not always feasible. In fact, an intervention can be uncertain in the sense that it may uncertainly occur. It can also have uncertain consequences which means that it may not succeed to put its target into one specific value. In this paper, we use the belief function theory to handle uncertain interventions that could have uncertain consequences. Augmented causal belief networks are used to model uncertain interventions.

1 Introduction

Despite its importance, causality is undefinable if a general and precise definition is sought (i.e., not restrained to particular cases) [22]. However, causal relations should be distinguished from mere statistical correlations. A paradigmatic assertion in causal relations is that the exterior manipulation (intervention) of a genuine cause will result in the variation of an effect. Therefore, interventions play a crucial role for an efficient causal analysis.

Bayesian networks [9,11,14] are successful graphical models representing a compact joint probability distribution. Causal Bayesian networks [14] go beyond Bayesian networks where arcs between variables follow the causal process. Probabilistic causal graphical models are effective when a very complete statistical knowledge description of the modeled system is available. If not, alternative causal networks will be more appropriate (possibilistic causal networks [3,4], causal belief networks [6]). On these networks, we can compute the simultaneous effect of observations and interventions. Interventions are distinguished from observations with the "do" operator [14]. An intervention forcing a variable A_i to be at a specific value a_{ij} is denoted by $do(a_{ij})$. This action deems that the original causes of the target variable are no more responsible of its state.

However, considering an intervention as a perfect external action is not realistic. Indeed, it may happen that due to an inattention, to ethical issues or to a lack of knowledge, the experimenter may not know the state of his action or its possible consequences. In fact, the occurrence of an intervention may be uncertain (e.g., injecting

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 61–72, 2013.

a drug whose expiration date has been exceeded). Moreover, an intervention may fail to set the target variable into one specific state (e.g., the use of a nicotine patches). In these cases, choosing random values will lead to the mis-estimation of the effects and accordingly to bad policies decisions.

Only few works in the probabilistic setting addressed the issue of intervention imperfection [10,12,20]. Besides in these works, interventions are defined differently from what is considered in the scope of this paper. In fact, they are considered as external actions certainly occurring represented with dummy variables that change the local probability distribution of the target variable.

The belief function theory is an uncertain framework that is especially appropriate to represent cases of partial and total ignorance. Therefore, it is an ideal tool to deal with these imperfect interventions. Despite its representation power, no work has been presented to handle uncertain interventions in the belief function framework.

This paper focuses on the modeling of uncertain interventions (i.e., uncertainly taking place) under the belief function framework. Graphically, to represent such interventions, augmented causal belief networks where conditional distributions are defined for any number of parents are used. In these networks, a conditional table is provided for the target variable given the intervention aside for the ones specified in the context of the initial causes. By this way, interactions with other causal factors are taken into consideration. Discounting technique is used to weaken the impact of the uncertain intervention on the distribution of the target variable. Moreover, a certain intervention may have uncertain consequences [7]. In this paper, we investigate the case of uncertain interventions that may have either certain or uncertain consequences.

The rest of the paper is organized as follows: in Section 2, we recall the basic concepts of the belief function theory and explain how causal knowledge can be represented on belief causal networks. The effect of uncertain interventions with certain consequences is handled in Section 3, whereas the case of uncertain interventions with uncertain consequences is treated in Section 4. Section 5 concludes the paper.

2 Belief Function Theory

2.1 Basics

We briefly recall the belief function theory. For more details see [15,19].

Let Θ be a finite set of mutually exhaustive and exclusive events referred to as the frame of discernment. The basic belief assignment (bba), denoted by m^{Θ}, is a mapping from 2^{Θ} to [0,1] such that:

$$\sum_{A \subseteq \Theta} m^{\Theta}(A) = 1 \tag{1}$$

When there is no ambiguity, m^{Θ} will be shortened m. The part of belief exactly committed to the event A of Θ is represented with the basic belief mass (bbm) denoted by $m(A)$. Subsets of Θ such that $m(A) > 0$ are called focal elements. When the emptyset is not a focal element, the bba is called normalized. A bba is said to be certain if the whole mass is allocated to a unique singleton of Θ and Bayesian when all focal elements are singletons. If the bba has Θ as unique focal element, it is called vacuous and it represents the case of total ignorance.

Two bbas m_1 and m_2 induced by two distinct items of evidence can be aggregated using Dempster's rule of combination to give one resulting bba $m_1 \oplus m_2$.

$$m_1 \oplus m_2(A) = \begin{cases} K \cdot \sum\limits_{B \cap C = A} m_1(B) \cdot m_2(C), \ \ \forall B, C \subseteq \Theta & \text{if } A \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $K^{-1} = 1 - \sum\limits_{B \cap C = \emptyset} m_1(B) \cdot m_2(C)$ is the normalization factor.

The initial knowledge encoded with a mass value, $m(A)$, is revised using Dempster's rule of conditioning upon the arrival of a new certain piece of information B. All non vacuous events implying $\overline{B}$ will be transferred to the part of A compatible with the evidence namely, $A \cap B$ [17]. In the case, where $A \cap B = \emptyset$, several methods exist for transferring the remaining evidence [18]. $m(A|B)$ denotes the degree of belief of A in the context where B holds. It is defined as:

$$m(A|B) = \frac{\sum_{C, B \cap C = A} m(C)}{1 - \sum_{B \cap C = \emptyset} m(C)} \tag{3}$$

A basic belief assignment can be weakened (or discounted) before the combination to take into account the reliability of an expert by the discounting method defined as:

$$m^{\alpha}(A) = \begin{cases} (1 - \alpha) \cdot m(A), \forall A \subset \Theta \\ \alpha + (1 - \alpha) \cdot m(A), \ \text{ if } A = \Theta \end{cases} \tag{4}$$

The discounting operation is controlled by a *discount rate* α taking values between 0 and 1. If α = 0, the source is fully reliable and beliefs remain unchanged. However, if α = 1, the bba is transformed into the vacuous bba, meaning that the information provided by the expert is completely discarded.

When a decision has to be made, beliefs held by the agent and represented by a bba could be transformed to a probability measure called $BetP$, using the pignistic transformation. It is defined as follows:

$$BetP(A) = \sum_{B \subseteq \Theta} \frac{|A \cap B|}{|B|} \frac{m(B)}{1 - m(\emptyset)}, \forall A \in \Theta \tag{5}$$

2.2 Causal Belief Networks

Belief networks [1,6,21] are simple and efficient tools to compactly represent uncertainty distributions. They have shown their efficiency in several applications (e.g., system analysis [16], threat assessment [2]). One main advantage of these networks is that they limit the use of a priori. They differ from Bayesian networks in the definition of conditional distributions and in the way to compute the global joint distribution. Causal belief networks [6,8] are seen as belief networks with some particular properties concerning the interpretation of arcs. They are defined on two levels as follows:

- qualitative level: a DAG $\mathcal{G} = (V, E)$ where arcs describe causal influence. Each variable A_i is associated with a finite set namely its frame of discernment Θ_{A_i} representing

all its possible instances, i.e., $\{a_{ij,j=1,...,|\Theta_{A_i}|}\}$. A variable A_j is called a parent of a variable A_i if there is an edge pointing from A_j towards A_i. The set of all parents of A_i is denoted by $U(A_i)$. Some of the parents of A_i are denoted by $PA(A_i)$ where a single parent is denoted by $PA_j(A_i)$. An instance from $U(A_i)$, $PA(A_i)$ or $PA_j(A_i)$ is denoted respectively by $u(A_i)$, $Pa(A_i)$ and $Pa_j(A_i)$.

- quantitative level: represented by the set of bbas associated to each node in the graph. For each root node A_i (i.e., $PA(A_i) = \emptyset$) having a frame of discernment Θ_{A_i}, an a priori m^{A_i} is defined on the powerset of $2^{\Theta_{A_i}}$, such that $\sum_{sub_{ik} \subseteq \Theta_{A_i}} m^{A_i}(sub_{ik}) = 1$. It is possible to model the total ignorance of the a priori by defining a vacuous bba on A_i (i.e., setting m(Θ_{A_i}) = 1). For the rest of the nodes, conditional distributions can be defined for each subset of each variable A_i in the context of its parents (either one or more than one parent node).

In causal belief networks, local conditional mass distributions are aggregated using the Dempster rule of combination. Since this rule is looking for intersections, each local distribution should be first extended to a joint frame. Thus, each conditional distribution will be deconditionalized (denoted by $\wr$) and non-conditionalized distribution will be vacuously extended to a joint frame (denoted by $\uparrow$)[5].

$$m^{V=A_1,...,A_n} = \oplus_{A_i \in V} (\oplus_{Pa_j(A_i)} m^{A_i}(a_i|Pa_j(A_i))^{\wr A_i \times PA_j(A_i)})^{\uparrow V} \quad (6)$$

where the vacuous extension is computed as:

$$m^{A_i \uparrow A_i \times A_j}(a_i) = m^{A_i, A_j}(a_i \times \Theta_{A_j})$$

and a conditional distribution is deconditionalized as follows:

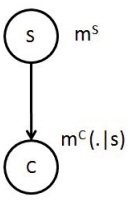
$$m^{A_i}(a_i|Pa_j(A_i))^{\wr A_i \times PA_j(A_i)} = m^{A_i, A_j}(\{a_i \times Pa_j \cup \Theta_{A_i} \times \overline{Pa_j(A_i)}\})$$

On causal belief networks, it is possible to compute the effect of observations (seeing the natural behavior of the system) and interventions (intended external acting forcing a variable to take a specific value). If a manipulation of the event B leads to a change in A, then B is considered as a cause of A. While the effects of observations are computed with conditioning rules, those of interventions are handled by means of the so-called "do" operator [14]. An intervention in this case is considered as an external that totally control the state of its target variable. Such interventions make the original causes of the manipulated variable no more responsible of its state. All the other causes than the one of the intervention will be excluded. Graphically, interventions are described in two equivalent ways, namely graph mutilation and graph augmentation. The first way consists in modifying the causal graph by cutting off the links pointing into the target variable. The second equivalent way consists in adding, for the target variable, a new parent variable denoted DO.

3 Handling Uncertain Interventions with Certain Consequences

The occurrence of interventions recalled in the last section is assumed to be certain. However, it is not realistic to always consider interventions as fully certain external actions. An intervention having the variable A_i as target may uncertainly occur by forcing A_i to take *an unknown specific value* a_{ij} ($a_{ij} \in \Theta_{A_i}$) or it may *fail to take place*.

Example 1. *This example will be used in the rest of the paper to illustrate the main results. It concerns a description of knowledge regarding the causal link between the use of sugar and the sweetness of a coffee.*



$\{s_1\}$	0.1
$\{s_2\}$	0.7
Θ_S	0.2

	$\{s_1\}$	$\{s_2\}$
$\{c_1\}$	0.8	0.1
$\{c_2\}$	0.1	0.7
Θ_C	0.1	0.2

Fig. 1. Causal belief network

Fig. 1, depicts a causal belief network where S describes the presence of sugar in the cup of coffee, Θ_S = $\{s_1, s_2\}$ where s_1 is yes and s_2 is no and C represents the sweetness of the coffee, Θ_C = $\{c_1, c_2\}$ where c_1 is sweet and c_2 is bitter.

Let us assume that you have gone to a restaurant and ordered a coffee. A friend sees on the table a container with some white powder, without tasting it, he adds some of this powder into your cup of coffee because he knows that you like sweet coffee. Unfortunately, later he realizes that it may be either sugar or something else, and since you are in a restaurant it is most likely to be salt. If afterward, you taste the coffee and you find it sweet, you do not know if it is due to the action of your friend or to the way the coffee has been prepared. This latest alternative has no relation with the intervention of your friend. Thus, links relating the sweetness of the coffee with the initial use of sugar should not be deleted.

As in handling standard interventions, to represent uncertain interventions, we will alter the belief network by adding a new fictive node (DO) as a new parent of the variable A_i concerned by a manipulation, i.e., $PA(A_i) \leftarrow PA(A_i) \cup DO$. The DO node is taking value in $do(x)$, $x \in \{\Theta_{A_i} \cup \{\text{nothing}\}\}$. do(nothing) means that there are no actions on the variable A_i, it represents the state of the system when no interventions are made or totally fail to occur. do(a_{ij}) means that the variable A_i is forced to take the value a_{ij}. This way allows to represent the effect of interventions and also observations. The augmented graph is denoted by $\mathcal{G}_{aug}$. By taking advantage of the representation of causal belief networks to define conditional distributions [8], a conditional bba in the context of the fictive node DO will be "naturally" specified.

3.1 Interventions with an Unknown Specific Value

In the following, we propose a method to handle uncertain interventions that force the target variable to take an unknown specific value. To compute the distribution of the target variable, we need to address four different issues:

1- Deciding about the nature of the external action. We propose a general method where the nature of the intervention is undefined and we have to specify it. A bba, m^I, expressing the beliefs about the genuine nature of the external action expressed on a frame of discernment $\Theta_I = \{\theta_1, \ldots, \theta_n\}$ is defined. Note that the frame Θ_I may be different from the frame of the target variable. Deciding about the actual nature of the intervention will allow us to know which states will be affected by a change. The decision operation is made using the pignistic transformation.

Example 2 *(continued). Suppose that the beliefs about the nature of the substance in the container are flexibly expressed within the belief function formalism. They are defined on Θ_I = {sugar, salt, flour} such that m^I({sugar}) = 0.2, m^I({salt}) = 0.7, m^I({flour}) = 0.01 and m^I({sugar,salt}) = 0.09. The corresponding probabilistic knowledge of this bba is computed with the pignistic probability measure as follows: $BetP^I$({sugar}) = 0.2 + 0.09 * 0.5 = 0.245, $BetP^I$({flour}) = 0.01, $BetP^I$({salt}) = 0.7 + 0.09 * 0.5 = 0.745.*

2- Defining the possible states of the intervention. The frame Θ_I is different from the frame of the target variable Θ_{A_i}. However, instances of Θ_I may affect the state of the target variable A_i by forcing it to take the value a_{ij}. Thus in the case of uncertain interventions, a matching between each θ_i and a state from Θ_{A_i} is defined as $match(\theta_i) = a_{ij}$. If θ_i has no impact on A_i, then we will say that $match(\theta_i) = nothing$. Note that more than one element of Θ_I may affect the same state a_{ij}.

Example 3 *(continued). The target variable has a frame of discernment Θ_C= {c_1=sweet, c_2=bitter} while the intervention is represented on Θ_I={sugar, salt, flour}. Table 1 presents the results of the matching between elements θ_i with instances of C.*

Table 1. Matching function: $match(\theta_i)$

θ_i	$match(\theta_i)$
sugar	c_1
salt	nothing
flour	c_2

Recall that the DO node represents the intervention. It has the same instances than its target to which the value $nothing$ is added. $do(a_{ij})$ means that the intervention attempts to set the target variable A_i into the state a_{ij}. This is achieved by performing the action θ_i. Therefore, executing θ_i amounts to $do(a_{ij})$. Accordingly, beliefs about the state of the variable DO reflecting the occurrence of the intervention will be defined from the knowledge about the decided nature of the intervention computed in the last step through BetPs. Since this latter reflects a probabilistic knowledge (i.e., computed for singletons), the bba of the DO node will be Bayesian and defined as:

$$m^{DO}(do(x)) \quad = \quad \begin{cases} \sum_{\theta_i, match(\theta_i)=a_{ij}} BetP^I(\theta_i) & \text{if } x = \{a_{ij}\} \\ \sum_{\theta_i, match(\theta_i)=nothing} BetP^I(\theta_i) & \text{if } x = \{nothing\} \end{cases} \quad (7)$$

Example 4 *(continued). According to the added substance, the coffee will be either sweet, bitter or remain as it was prepared. Therefore, forcing it to be at a specific state is not given for sure by adding the white powder. Hence, beliefs expressed about the actual occurrence of the intervention are computed using the BetP of each ingredient. In fact, the BetP takes into account all the focal elements that intersect with the substance of interest. The bba of the node DO is defined as:* $m^{DO}(\{do(c_1)\}) = BetP^I$*(sugar) = 0.245,* $m^{DO}(\{do(c_2)\}) = BetP^I$*(flour) = 0.01 and* $m^{DO}(\{do(nothing)\}) = BetP^I$*(salt) = 0.745.*

3- Defining Conditionals Given the DO Node. When occurring, an intervention $do(a_{ij})$ succeeds to force the variable A_i to take a certain value a_{ij}. Therefore, a conditional bba given an intervention is a certain bba focused on a_{ij} defined as:

$$m^{A_i}(sub_{ik}|do(a_{ij})) = \begin{cases} 1 & \text{if } sub_{ik} = \{a_{ij}\} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

One can consider that $m^{A_i}(.|do(a_{ij}))$ is provided by an information source and this latest expects that it will be a certain bba. Since the occurrence of the intervention is uncertain, the bba defined by applying Equation 8 is not appropriate. Accordingly, this source is seen as not fully reliable. In fact, even if the intervention succeeds to put its target into one specific value, its occurrence remains uncertain. A Bayesian bba expressing the actual values concerning the occurrence of the intervention has been computed with BetP as explained in the last step. It will be used to evaluate the reliability of the source.

When considering the case of an intervention forcing the variable A_i to take the value a_{ij}, the occurrence of the intervention in the form of other states does not matter. What it was predicted by the source is an intervention certainly occurring at the state a_{ij}, $m^{DO}(do(a_{ij})) = 1$, whereas the actual belief about the occurrence of the intervention succeeding to put the variable A_i into the state a_{ij} is defined as $m^{DO}(do(a_{ij})) = \alpha \in [0, 1]$. Since the degree of confidence in the reliability of a source can depend on the true value of the variable of interest, the difference between what is was predicted and the actual value is considered as its discounting factor defined as $1 - \alpha$. Consequently, the conditional distribution given the DO node is discounted by taking into account the reliability of each source, namely $\alpha_{do(a_{ij})}$. This information, will transform the conditional given the DO node from a certain bba into a weaker, less informative one. Hence, the new conditional bba of the target variable given the DO node becomes:

$$m^{A_i,\alpha_{do(a_{ij})}}(sub_{ik}|do(a_{ij})) = \begin{cases} 1-\alpha & \text{if } sub_{ik} = \{a_{ij}\} \\ \alpha & \text{if } sub_{ik} = \Theta_{A_i} \end{cases} \tag{9}$$

Proposition 1. *Standard interventions are a particular case of uncertain interventions when the source is fully reliable, i.e.,* $\alpha = 0$.

$$m^{A_i,\alpha_{do(a_{ij})}=0}(sub_{ik}|do(a_{ij})) = \begin{cases} 1 & \textit{if } sub_{ik} = \{a_{ij}\} \\ 0 & \textit{otherwise} \end{cases} \tag{10}$$

Example 5 *(continued).* *Graphically, an extra node DO representing the intervention on the variable C is added as its new parent in the augmented graph. Each conditional distribution for the target variable C given an instance of the DO node is seen as provided by a distinct source of information. These sources affirm that performing an intervention will lead to a known change in the state of the manipulated variable. The conditional distributions as presented by the sources are presented in Table 2.*

Table 2. Certain *bba*: $m^C(.|do(x))$

	$\{do(c_1)\}$	$\{do(c_2)\}$	$\{do(nothing)\}$
$\{c_1\}$	1	0	0
$\{c_2\}$	0	1	0
Θ_C	0	0	1

Since the intervention achievement is uncertain, conditional local distributions presented in Table 2 are not appropriate. In fact, even when the intervention occurs with a degree of belief and succeeds to put its target into one specific value, one should take into consideration the cases where it fails to take place. Therefore, certain conditional local distributions will be discounted according to the reliability of each source. The degree of confidence in the reliability of a source is computed according the true value of the variable of interest, i.e., the DO bba. Hence, discount rates are denoted by $1-\alpha_{do(x)}$. They are defined as 1- $\alpha_{do(c_1)}$ = 0.245, 1- $\alpha_{do(c_2)}$ = 0.01 and 1- $\alpha_{do(nothing)}$ = 0.745. The new discounted conditional bba is presented in Table 3.

Table 3. Discounted *bba*: $m^{C,\alpha_{do(x)}}(.|do(x))$

	$\{do(c_1)\}$	$\{do(c_2)\}$	$\{do(nothing)\}$
$\{c_1\}$	1*0.245=0.245	0*0.01=0	0*0.745=0
$\{c_2\}$	0*0.245=0	1*0.01=0.01	0*0.745=0
Θ_C	0*0.245+0.755 =0.755	0*0.01+0.99=0.99	1*0.745+0.255=1

4- Defining Conditionals Given an Uncertain Intervention. The impact of the uncertain intervention on the target variable will not only depend from the intervention but also from the initial causes of the variable. To get the conditional *bba* given all the parent nodes, Dempster's rule of combination is used to aggregate the conditional distribution given the initial causes with the discounted conditional given the DO parent. We use $m^{A_i}(a_j|Pa(A_i))$ to represent the conditional mass function induced on the space Θ_{A_i} given $Pa(A_i) \subseteq \Theta_{PA(A_i)}$, and $m^{A_i,\alpha_{do(x)}}(a_k|do(x))$ to represent the discounted conditional mass function induced on the space Θ_{A_i} given the intervention $do(x)$. The *bba* of the target variable $m^{A_i}(a_i|Pa(A_i), do(x))$ is computed as follows:

$$m^{A_i}(a_i|Pa(A_i), do(x)) = \sum_{a_j \cap a_k = a_i} m^{A_i}(a_j|Pa(A_i)) \cdot m^{A_i,\alpha_{do(x)}}(a_k|do(x)) \quad (11)$$

Example 6 *(continued). The conditional bbas given the initial causes and that of the DO node can be aggregated to give the conditional bba* $m^C(.|s_i, do(x))$*. For instance,* $m^C(.|s_1, do(c_1))$ *is obtained by computing* $m^C(.|s_1) \oplus m^{C,\alpha_{do(c_1)}}(.|do(c_1))$*. Results are presented in Table 4.*

Unlike the case of standard interventions, $m^C(c_1|s_1, do(c_1)) \neq 1$*. However, the action of the friend has raised the beliefs about the sweetness of the coffee. A small increase from 0.8 to 0.845 is explained by the fact that it is more likely that the used ingredient is salt. In the same way,* $m^C(c_2|s_2, do(c_1))$ *has decreased from 0.7 to 0.638.*

Table 4. Conditional *bba*: $m^C(.|s_i, do(c_1))$

	$\{(s_1, do(c_1))\}$	$\{(s_2, do(c_1))\}$
$\{c_1\}$	0.8450	0.180
$\{c_2\}$	0.0775	0.638
Θ_C	0.0775	0.182

3.2 Interventions Not Occurring

The approach we proposed for handling interventions uncertainly happening remains valid to deal with the case of non-interventions. This is represented by setting the variable DO with certainty to the value $do(nothing)$.

In this paper, we consider that the situation of non-intervention encompasses:
- not acting on the target variable and observing the spontaneous behavior of the system,
- failing to act on the target variable and therefore the intervention will not occur.

Formally, in this case:

$$\forall \theta_i, match(\theta_i) = \{nothing\} \tag{12}$$

From Equations 7 and 12, the bba of the DO node is defined by:

$$m^{DO}(do(x)) = \begin{cases} 1 & \text{if } x = \{nothing\} \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

In this case, the state of the target variable will not depend on the intervention (i.e., from the DO node). The conditional bba given the DO node is not informative. It is represented with the vacuous bba defined as:

$$m^{A_i}(sub_{ik}|do(nothing)) = \begin{cases} 1 & \text{if } sub_{ik} = \Theta_{A_i} \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

The "non-intervention" occurs certainly. Therefore, the source is fully reliable and the discounting factor is equal to zero. Hence, our approach well handles the particular case of standard interventions.

Proposition 2. *The beliefs provided about the non-occurrence of an intervention are accepted without any modification. They are defined like standard interventions.x*

$$m^{A_i,\alpha_{do(nothing)}}(.|do(nothing)) = m^{A_i}(.|do(nothing)) \tag{15}$$

The conditional bbas defined in the context of the DO node and of the initial causes are computed by combining each conditional defined per single parent as follows:

$$\begin{aligned} m^{A_i}(.|Pa(A_i), do(nothing)) &= m^{A_i}(.|do(nothing)) \oplus m^{A_i}(.|Pa(A_i)) \\ &= m^{A_i}(.|Pa(A_i)) \end{aligned} \tag{16}$$

Proposition 3. *An augmented causal belief graph where the DO node is set to the value $nothing$ encodes the same joint distribution than the initial causal belief network.*

$$m_{\mathcal{G}_{aug}}(.|do(nothing)) = m_{\mathcal{G}} \tag{17}$$

4 Handling Uncertain Intervention with Uncertain Consequences

In the last section, we dealt with interventions occurring in an uncertain way. When happening, even with a belief $m(\{do(a_{ij})\})$, they succeed to put the target variable into exactly one specific state. This situation is not always feasible. Therefore, our proposed approach in this section is to handle uncertain interventions with uncertain consequences, i.e., failing to put their target into a specific value.

4.1 Certain Interventions with Uncertain Consequences

In [7], we dealt with interventions that certainly take place but have uncertain consequences. To handle such cases, we proposed to specify a new bba on the target variable representing the consequences of the intervention. Let us denote by $\mathcal{F}_{A_i}$, the set of the focal elements representing the uncertain consequences of the intervention where a bbm β_j is allocated to each focal element. The conditional bba of the target variable given a certain intervention on the variable A_i attempting to force it to take the value a_{ij} is defined as follows:

$$m^{A_i}(sub_{ik}|do(a_{ij})) = \begin{cases} \beta_j & \text{if } sub_{ik} \in \mathcal{F}_{A_i}, \beta_j \in]0,1] \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

Example 7 *(continued).* *Let us continue with the network of Fig. 1. Imagine here that your friend puts Lactose into your cup of coffee which is a disaccharide sugar. However it is known that it is poorly soluble. Therefore, even if the substance is a kind of sugar, adding it will obviously affect the sweetness of the coffee but without certainty. The conditional bba $m^C(.|do(c_1))$ defined upon this intervention is expressed as follows: $m^C(c_1|do(c_1)) = 0.8$, $m^C(c_2|do(c_1)) = 0.05$, $m^C(\Theta_C|do(c_1)) = 0.15$.*

4.2 Uncertain Interventions with Uncertain Consequences

In this paper, we also investigate the case of uncertain interventions with uncertain consequences. In fact, an intervention even taking place with a given degree of belief may have uncertain consequences. Remember that to deal with uncertain interventions succeeding to set their target into a specific value a_{ij}, the conditional bbas given instances

of the DO node are discounted according to the actual occurrence of the intervention (see Equation 9). In the case of uncertain interventions with uncertain consequences, we take into consideration possible states that can take the target variable. Therefore, we define the resulting bba as a mixture of Equation 9 and 18 as follows:

$$m^{A_i}(sub_{ik}|do(a_{ij})) = \begin{cases} (1-\alpha)\cdot\beta_j & \text{if } sub_{ik} \in \mathcal{F}_{A_i} \\ \alpha + (1-\alpha)\cdot\beta_j & \text{if } sub_{ik} = \Theta_{A_i} \end{cases} \tag{19}$$

Proposition 4. *Uncertain interventions with a certain consequence are a particular case of uncertain ones with uncertain consequences when the parameter β_j is set to one.*

$$m^{A_i}(sub_{ik}|do(a_{ij})) = \begin{cases} 1-\alpha & \textit{if } sub_{ik} = \{a_{ij}\} \\ \alpha & \textit{if } sub_{ik} = \Theta_{A_i} \end{cases}$$

Example 8 *(continued)***.** *In context of a restaurant, it is more likely that what your friend has putted into your coffee is salt. We are focusing in the occurrence of the intervention as attempting to set its target into the value sweet, which means that the powder is sugar. However, some kinds of sugar (e.g., lactose, saccharine) are either not very soluble or may have a bitter or metallic unpleasant aftertaste. Adding them may lead to uncertain consequences. Note that the bbm that the added substance is sugar is represented with $m(\{do(c_1)\})$= 0.245. Hence, to represent this case the conditional bba given the DO node will be discounted. The resulting bba is presented in Table 5.*

Table 5. $m^{C,\alpha_{do(c_1)}}(.|do(c_1))$ upon an uncertain intervention with uncertain consequences

	$\{\text{do}(c_1)\}$
$\{c_1\}$	0.8*0.245 = 0.2
$\{c_2\}$	0.05*0.245 = 0.01
Θ_C	0.15*0.245+0.755 = 0.79

Note that as for uncertain interventions with certain consequences, the conditional distribution given the DO parent can be combined with the discounted conditional distribution given the initial causes using Dempster's rule of combination to obtain the conditional distribution given all the parent nodes.

5 Conclusion

This paper provided a causal graphical model to deal with interventions under the belief function framework. We argued that for several practical cases, interventions may be uncertain and should be consequently adequately modeled. Furthermore, we addressed the issue of uncertain interventions failing to be at one specific state so-called uncertain interventions with uncertain consequences.

We emphasized on that uncertain interventions have a natural encoding under the belief function framework and may be graphically modeled using causal belief networks. The effect of an uncertain intervention is computed on an altered structure,

namely belief augmented graphs. In these networks, conditionals can be defined for any number of parents and are can be seen as provided by distinct sources of information.

As future works, we intend to explore the relationships between interventions and the belief changes using Jeffrey-Dempster's rule [13].

References

1. Ben Yaghlane, B., Mellouli, K.: Inference in directed evidential networks based on the transferable belief model. Int. J. Approx. Reasoning 48(2), 399–418 (2008)
2. 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, 620–639 (2009)
3. Benferhat, S.: Interventions and belief change in possibilistic graphical models. Artif. Intell. 174(2), 177–189 (2010)
4. Benferhat, S., Smaoui, S.: Inferring interventions in product-based possibilistic causal networks. Fuzzy Sets and Systems 169(1), 26–50 (2011)
5. Boukhris, I., Benferhat, S., Elouedi, Z.: Representing belief function knowledge with graphical models. In: Xiong, H., Lee, W.B. (eds.) KSEM 2011. LNCS, vol. 7091, pp. 233–245. Springer, Heidelberg (2011)
6. Boukhris, I., Elouedi, Z., Benferhat, S.: Modeling interventions using belief causal networks. In: FLAIRS 2011, pp. 602–607 (2011)
7. Boukhris, I., Elouedi, Z., Benferhat, S.: Dealing with interventions with uncertain consequences in belief causal networks. In: IPMU 2012, pp. 585–595 (2012)
8. Boukhris, I., Elouedi, Z., Benferhat, S.: On the modeling of causal belief networks. In: ICMSAO 2013 (to appear, 2013)
9. Darwiche, A.: Modeling and Reasoning with Bayesian Networks. Cambridge University Press (2009)
10. Eberhardt, F., Scheines, R.: Interventions and causal inference. Philos. Sci. 74, 981–995 (2007)
11. Jensen, F., Nielsen, T.: Bayesian Networks and Decision Graphs. Springer Publishing Company (2007)
12. Korb, K.B., Hope, L.R., Nicholson, A.E., Axnick, K.: Varieties of causal intervention. In: Zhang, C., Guesgen, H.W., Yeap, W.-K. (eds.) PRICAI 2004. LNCS (LNAI), vol. 3157, pp. 322–331. Springer, Heidelberg (2004)
13. Ma, J., Liu, W., Dubois, D., Prade, H.: Bridging jeffrey's rule, agm revision and dempster conditioning in the theory of evidence. International Journal on Artificial Intelligence Tools 20(4), 691–720 (2011)
14. Pearl, J.: Causality: Models, Reasonning and Inference. Cambridge University Press (2000)
15. Shafer, G.: A Mathematical Theory of Evidence. Princeton Univ. Press, Princeton (1976)
16. Simon, C., Weber, P., Evsukoff, A.: Bayesian networks inference algorithm to implement dempster shafer theory in reliability analysis. Reliability Engineering and System Safety 93, 950–963 (2008)
17. Smets, P.: The combination of evidence in the transferable belief model. IEEE Trans. Pattern Anal. Mach. Intell. 12(5), 447–458 (1990)
18. Smets, P.: About updating. In: UAI 1991, pp. 378–385 (1991)
19. Smets, P.: The transferable belief model for quantified belief representation, vol. 1, pp. 267–301. Kluwer Academic Publisher (1998)
20. Teng, C.M.: Applications of causal inference. In: ISAIM (2012)
21. Xu, H., Smets, P.: Evidential reasoning with conditional belief functions. In: UAI 1994, pp. 598–606 (1994)
22. Zadeh, L.: Causality is undefinable. Tech. rep., Univ. of California, Berkley (2001)

On Semantics of Inference in Bayesian Networks

Cory J. Butz[1], Wen Yan[1], and Anders L. Madsen[2,3]

[1] Department of Computer Science, University of Regina, Canada
{butz,yanwe111}@cs.uregina.ca
[2] HUGIN EXPERT A/S, Aalborg, Denmark
anders@hugin.com
[3] Department of Computer Science, Aalborg University, Denmark

Abstract. An algorithm, called *Semantics in Inference* (SI) has been proposed recently for determining semantics of the intermediate factors constructed during exact inference in discrete Bayesian networks. In this paper, we establish the soundness and completeness of SI. We also suggest an alternative version of SI, one that is perhaps more intuitive as it is a simpler graphical approach to deciding semantics.

1 Introduction

Zhang and Poole [14] proposed *Variable Elimination* (VE) as a simple approach to exact inference in discrete Bayesian networks. Given a *Bayesian network* [11], which consists of a directed acyclic graph and a corresponding set of conditional probability tables, VE can compute the posterior probabilities of a set of variables given that another disjoint set of variables are observed taking certain values. Koller and Friedman [8] state that it is interesting to consider the semantics of the potentials constructed during inference, since only sometimes the probabilities are defined with respect to the joint distribution.

In [1], we gave a method for determining the semantics of the intermediate factors built by VE during inference. That method worked by checking for the existence of a particular topological ordering of the n variables in a Bayesian network, thereby having $O(n!)$ time complexity. More recently, we suggested in [2], the *Semantics in Inference* (SI) algorithm, which uses *d-separation* [11] to decide the semantics of the intermediate factors. There it was shown that SI has polynomial time complexity $O(n^3)$ and that SI is strongly complete. Due to space constraints other properties were not shown.

In this theoretical paper, we establish the soundness and completeness of SI. The work here also leads us to suggest an alternative version of SI, one that is based upon the notions of ancestors and descendants. This is a third way to decide semantics of intermediate factors in exact inference in discrete Bayesian networks. The method in [1] is based upon the notion of topological orderings, while the notion of d-separation is utilized in [2]. The approach taken here should be more intuitive, since it can be understood as a simple visual test.

The remainder of this paper is organized as follows. Background knowledge is given in Section 2. In Section 3, the soundness and completeness of SI are established. Section 4 presents an alternative version of SI. Conclusions are delivered in Section 5.

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 73–84, 2013.

2 Background Knowledge

2.1 Exact Inference in Bayesian Networks

A discrete *Bayesian network* [11] is a pair (B, C). B denotes a directed acyclic graph with a finite vertex set $U = \{v_1, v_2, \ldots, v_n\}$, where for simplified notation, we may write $\{v_1, v_2, \ldots, v_k\}$ as $v_1, v_2, \ldots, v_k$. Each vertex represents a random variable v_i, which can take a value from a finite domain, $dom(v_i)$. C is a set of *conditional probability tables* (CPTs) $\{p(v_i|P(v_i)) \mid i = 1, 2, \ldots, n\}$, where $P(v_i)$ denotes the parents (immediate predecessors) of $v_i \in B$. The product of CPTs in C is a joint probability distribution $p(U)$. For example, the directed acyclic graph in Figure 1 is called the *extended student Bayesian network* (ESBN) [8]. We give CPTs in Table 1, where only binary variables are used in examples, and probabilities not shown can be obtained by definition. By the above,

$$p(U) \;=\; p(c) \cdot p(d|c) \cdot p(i) \cdot p(g|d,i) \cdots p(h|g,j). \tag{1}$$

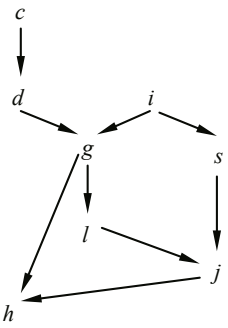


Fig. 1. The directed acyclic graph of ESBN

Table 1. CPTs for the ESBN in Figure 1

c	$p(c)$
0	0.20

i	$p(i)$
0	0.75

c	d	$p(d\|c)$
0	0	0.40
1	0	0.70

g	l	$p(l\|g)$
0	0	0.30
1	0	0.60

i	s	$p(s\|i)$
0	0	0.40
1	0	0.80

d	i	g	$p(g\|d,i)$
0	0	0	0.90
0	1	0	0.20
1	0	0	0.50
1	1	0	0.40

s	l	j	$p(j\|s,l)$
0	0	0	0.10
0	1	0	0.60
1	0	0	0.45
1	1	0	0.50

g	j	h	$p(h\|g,j)$
0	0	0	0.25
0	1	0	0.65
1	0	0	0.50
1	1	0	0.85

A *topological ordering* [8] is an ordering $\prec$ of the variables in a Bayesian network B so that for every arc (v_i, v_j) in B, v_i precedes v_j in $\prec$. For example, $c \prec d \prec i \prec g \prec s \prec l \prec j \prec h$ is a topological ordering of the variables in Figure 1, but $d \prec c \prec i \prec g \prec h \prec l \prec j \prec s$ is not. A *path* from v_1 to v_n is a sequence

$v_1, v_2, \ldots, v_n$ with arcs (v_i, v_{i+1}) in B, $i = 1, \ldots, n-1$. With respect to a variable v_i, we define three more sets: (i) the ancestors of v_i, denoted $A(v_i)$, are those variables having a path to v_i; (ii) the descendants of v_i, denoted $D(v_i)$, are those variables to which v_i has a path; and, (iii) the children of v_i are those variables v_j such that arc (v_i, v_j) is in B. The ancestors of a set $X \subseteq U$ are defined as $A(X) = (\cup_{v_i \in X} A(v_i)) - X$. The descendants $D(X)$ are defined similarly.

VE, shown as Algorithm 1, computes $p(X|E = e)$ from a discrete Bayesian network B by calling *sum-out* (SO) to eliminate variables one by one. More specifically, in Algorithm 1, Φ is the set C of CPTs for B, X is a list of query variables, E is a list of observed variables, e is the corresponding list of observed values, and σ is an elimination ordering for variables $U - XE$, where XE denotes $X \cup E$. The *evidence potential* for $E = e$, denoted $1(E = e)$, assigns probability 1 to the single value e of E and probability 0 to all other values of E. Hence, for a variable v observed taking value λ and $v \in \{v_i\} \cup P(v_i)$, the product $p(v_i|P(v_i)) \cdot 1(v = \lambda)$ keeps only those configurations agreeing with $v = \lambda$.

Algorithm 1. VE(Φ, X, E, e, σ)
Multiply evidence potentials with appropriate CPTs
While σ is not empty
  Remove the first variable v from σ
  Φ = SO(v, Φ)
$p(X, E = e)$ = the product of all potentials $\psi \in \Phi$
return $p(X, E = e) / \sum_X p(X, E = e)$

SO, shown as Algorithm 2, eliminates a single variable v from a set Φ of *potentials* [8], and returns the resulting set of potentials. The algorithm *collect-relevant* simply returns those potentials in Φ involving variable v.

Algorithm 2. SO(v,Φ)
Ψ = collect-relevant(v,Φ)
ψ = the product of all potentials in Ψ
$\tau = \sum_v \psi$
return $(\Phi - \Psi) \cup \{\tau\}$

As in [8], suppose the observed evidence for the ESBN is $i = 1$ and $h = 0$ and the query is $p(j|h = 0, i = 1)$. The weighted-min-fill algorithm [8] can yield $\sigma = (c, d, l, s, g)$. VE first incorporates the evidence: $\psi(i = 1) = p(i) \cdot 1(i = 1)$, $\psi(d, g, i = 1) = p(g|d, i) \cdot 1(i = 1)$, $\psi(i = 1, s) = p(s|i) \cdot 1(i = 1)$, $\psi(g, h = 0, j) = p(h|g, j) \cdot 1(h = 0)$. To eliminate c, the SO algorithm computes $\psi(d) = \sum_c p(c) \cdot p(d|c)$. SO computes the following to eliminate d $\psi(g, i = 1) = \sum_d \psi(d) \cdot \psi(d, g, i = 1)$. To eliminate l, $\psi(g, j, s) = \sum_l p(l|g) \cdot p(j|l, s)$. SO computes the following when eliminating s,

$$\psi(g, i = 1, j) = \sum_s \psi(i = 1, s) \cdot \psi(g, j, s). \quad (2)$$

For g, SO can compute:

$$\begin{aligned} & \sum_g \psi(g, i=1, j) \cdot \psi(g, i=1) \cdot \psi(g, h=0, j) \\ = & \sum_g \psi(g, i=1, j) \cdot \psi(g, h=0, i=1, j) \\ = & \ \psi(h=0, i=1, j). \end{aligned} \tag{3}$$

VE then multiplies all remaining potentials as $p(h=0, i=1, j) = \psi(i=1) \cdot \psi(h=0, i=1, j)$. Finally, VE answers the query by $p(j|h=0, i=1) = p(h=0, i=1, j)/\sum_j p(h=0, i=1, j)$.

2.2 Semantics in Inference

In [3], we established the CPT structure of VE's intermediate factors, namely, every multiplication in VE $\psi(X_1|Y_1) \cdot \psi(X_2|Y_2)$ yields a CPT $\psi(X_1X_2|Y_1Y_2 - X_1X_2)$ and every summation $\sum_v \psi(X|Y)$ during VE yields $\psi(X-v|Y)$.

By semantics, we mean that a CPT $\psi(X|Y)$ constructed by VE's manipulation of Bayesian network CPTs is not necessarily equal to the CPT $p(X|Y)$ obtained from the defined joint probability distribution $p(U)$. For instance, it can be verified that in the ESBN,

$$p(h|g, j) \cdot \sum_d p(g|d, i) \cdot \sum_c p(c) \cdot p(d|c) \tag{4}$$

produces the CPT $\psi(g, h|i, j)$ in Table 2 (left). In contrast, the CPT $p(g, h|i, j)$ built from the joint distribution $p(U)$ in (1) is shown in Table 2 (right).

The *evidence expanded form* [2] of ψ, denoted $F(\psi)$, is the unique expression defining how ψ was built using the multiplication and marginalization operators on the Bayesian network CPTs together with any appropriate evidence potentials. The evidence expanded form $F(\psi)$ of any potential ψ constructed by VE can always be equivalently written in *evidence normal form* [2], namely, $\gamma \cdot N$, where γ is the product of 1 and all evidence potentials in $F(\psi)$, and N is the same factorization as $F(\psi)$ except without products involving evidence potentials. The evidence expanded form $F(\psi(g, i=1, j))$ of $\psi(g, i=1, j)$ in (2) is:

$$\sum_s ((p(s|i) \cdot 1(i=1)) \cdot (\sum_l (p(l|g) \cdot p(j|l, s)))). \tag{5}$$

Its normal form $\gamma \cdot N$ is

$$1(i=1) \cdot \sum_s \sum_l p(s|i) \cdot p(l|g) \cdot p(j|l, s), \tag{6}$$

where $\gamma = 1(i=1)$ and $N = \psi(j|g, i)$.

Table 2. CPT $\psi(g,h|i,j)$ built by (4) and the CPT $p(g,h|i,j)$ built from $p(U)$ in (1)

i	j	g	h	$\psi(g,h\|i,j)$	i	j	g	h	$p(g,h\|i,j)$
0	0	0	0	0.1890	0	0	0	0	0.1960
0	0	0	1	0.5670	0	0	0	1	0.5880
0	0	1	0	0.1220	0	0	1	0	0.1080
0	1	0	0	0.4914	0	1	0	0	0.4762
0	1	0	1	0.2646	0	1	0	1	0.2564
0	1	1	0	0.2074	0	1	1	0	0.2272
1	0	0	0	0.0680	1	0	0	0	0.0846
1	0	0	1	0.2040	1	0	0	1	0.2537
1	0	1	0	0.3640	1	0	1	0	0.3309
1	1	0	0	0.1768	1	1	0	0	0.1518
1	1	0	1	0.0952	1	1	0	1	0.0817
1	1	1	0	0.6188	1	1	1	0	0.6515

The *transitive closure* [5] of B is defined as the graph $T = (U, E^*)$, where

$$E^* = \{(v_i, v_j) \mid \text{ there is a path from vertex } v_i \text{ to } v_j \text{ in } B\}.$$

The transitive closure T of the ESBN in Figure 1 is:

$$T = \begin{array}{c} \\ \mathrm{c} \\ \mathrm{d} \\ \mathrm{i} \\ \mathrm{g} \\ \mathrm{l} \\ \mathrm{s} \\ \mathrm{j} \\ \mathrm{h} \end{array} \begin{array}{c} \begin{array}{cccccccc} \mathrm{c} & \mathrm{d} & \mathrm{i} & \mathrm{g} & \mathrm{l} & \mathrm{s} & \mathrm{j} & \mathrm{h} \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}.$$

For each variable, the columns of T represent its ancestors, while the rows of T represent its descendants. For example, $A(g) = \{c, d, i\}$ and $D(l) = \{j, h\}$. More generally, for sets of variables, $A(j, l, s) = \{c, d, i, g\}$ and $D(j, l, s) = \{h\}$, while $A(\{c, d, g, h\}) = \{i, j, l, s\}$ and $D(\{c, d, g, h\}) = \{j, l\}$.

The *Semantics in Inference* (SI) algorithm [2], given as Algorithm 3, denotes the semantics of any potential ψ built by VE on a discrete Bayesian network B. Each potential ψ constructed by VE is represented in evidence normal form $\psi(X|Y)$. If the semantics of B ensure the $\psi(X|Y) = p(X|Y)$, then ψ is denoted as $p_B(X|Y)$; otherwise, it is denoted as $\phi_B(X|Y)$. S is the set of variables marginalized in $F(\psi)$. $A(XS)$ and $D(XS)$ are computed from the *transitive closure*, denoted T, of B. $I_B(M, N, O)$ means an independence statement $I(M, N, O)$ holds in B by *d-separation* [11], where $M, N, O \subseteq U$.

Algorithm 3. SI(ψ)
Compute the evidence expanded form $F(\psi)$ of ψ
Compute the normal form $\gamma \cdot N$ of $F(\psi)$
Compute the CPT structure $\psi(X|Y)$ of N
Compute $Z = A(XS) \cap D(XS)$
Compute $X_1 = X \cap P(Z)$
if $I_B(X_1, \emptyset, Y)$ holds in B by d-separation
  return $p_B(X|Y)$
else
  return $\phi_B(X|Y)$

Recall $\psi(g, i = 1, j)$ in (2). The evidence normal form is (5) and the evidence normal form is (6). The CPT structure of N is $\psi(j|i, g)$. Now $X = \{j\}$, $Y = \{i, g\}$ and $S = \{l, s\}$. By the transitive closure T of the ESBN, $A(XS) = \{c, d, i, g\}$ and $D(XS) = \{h\}$. Hence, $Z = \emptyset$, $P(Z) = \emptyset$, and $X_1 = \emptyset$. Trivially, $I_B(X_1, \emptyset, Y)$ holds. Thus, SI denotes $\psi(g, i = 1, j)$ in (2) as $p_B(j|g, i = 1)$.

Now consider $\psi(g, h = 0, i = 1, j)$ in (3). Then $F(\psi)$ is equal to

$$p(h|g, j) \cdot 1(h = 0) \cdot \sum_d p(g|d, i) \cdot 1(i = 1) \cdot \sum_c p(c) \cdot p(d|c). \tag{7}$$

In evidence normal form $\gamma \cdot N$, we have

$$1(h = 0, i = 1) \cdot \sum_d \sum_c p(h|g, j) \cdot p(g|d, i) \cdot p(c) \cdot p(d|c) \tag{8}$$

where $\gamma = 1(h = 0, i = 1)$ and $N = \psi(g, h|i, j)$. With $X = \{g, h\}$, $Y = \{i, j\}$ and $S = \{c, d\}$, from T on the ESBN we have $A(\{c, d, g, h\}) = \{i, j, l, s\}$ and $D(\{c, d, g, h\}) = \{j, l\}$. Thus, $Z = \{j, l\}$, giving $P(Z) = \{g, s\}$ and $X_1 = \{g\}$. Now, $I_B(X_1, \emptyset, Y)$ does not hold. Thereby, SI denotes $\psi(g, h = 0, i = 1, j)$ in (3) as $\phi_B(g, h = 0|i = 1, j)$.

3 Theoretical Foundation

Lemmas 1 - 4 are used to show soundness (Theorem 1), while Lemmas 4 - 7 are used to show completeness (Theorem 2).

Lemma 1. *In SI,* $I_B(X_1, \emptyset, Y) \iff X_1 = \emptyset$.

Proof. ($\Leftarrow$) Given $X_1 = \emptyset$, then $I_B(X_1, \emptyset, Y)$ holds.

($\Rightarrow$) Consider two cases. Suppose $Y = \emptyset$. As $Y = P(XS) = \emptyset$, $A(XS) = \emptyset$. Therefore, $Z = \emptyset$, $P(Z) = \emptyset$, and $X_1 = \emptyset$. Now suppose $Y \neq \emptyset$. By contradiction, suppose $v_i \in X_1$. By definition of X_1, $v_i \in P(Z)$. Thus, there exists a $v_j \in Z$ with $(v_i, v_j) \in B$. By definition of Z, $v_j \in A(XS)$. If $v_j \in P(XS)$, then $I_B(X_1, \emptyset, Y)$ does not hold. A contradiction. Otherwise, if $v_j \notin P(XS)$, there exists a $v_k \in P(XS)$ with $v_k \in D(v_j)$. As $v_k \in D(v_i)$, $I_B(X_1, \emptyset, Y)$ does not hold. A contradiction. Thus, $X_1 = \emptyset$. □

Lemma 2. *In SI,* $X_1 = \emptyset \iff A(XS) \cap D(XS) = \emptyset$.

Proof. ($\Leftarrow$) Suppose $A(XS) \cap D(XS) = \emptyset$. By definition, $Z = A(XS) \cap D(XS)$. Therefore, $P(Z) = \emptyset$. Since $X_1 = X \cap P(Z)$, we have $X_1 = \emptyset$.

($\Rightarrow$) Given $X_1 = \emptyset$. By contradiction, suppose $Z \neq \emptyset$. Then there exists a $v_k \in Z$ such that $(v_i, v_k) \in B$ and $v_i \in XS$. Suppose $v_i \in S$. By VE, all CPTs involving v_i will have been multiplied together, which includes $p(v_k|P(v_k))$ as v_k is a child of v_i in B. This would mean that v_k is in XS. A contradiction to $v_k \in Z$. Therefore, $v_i \in X$. Now, as $v_k \in Z$, we know $v_i \in P(Z)$. By definition, $v_i \in X_1$. A contradiction to $X_1 = \emptyset$. Therefore, $Z = A(XS) \cap D(XS) = \emptyset$. □

Lemma 3. *Given a discrete Bayesian network* B *on* U *and* $V \subseteq U$. *There exists a topological ordering* $\prec$ *where the variables in* V *appear consecutively if and only if* $A(V) \cap D(V) = \emptyset$.

Proof. ($\Rightarrow$) Suppose $A(V) \cap D(V) \neq \emptyset$, by contraposition. Then there is at least one variable v_j not in V that is both a descendant of a variable v_i in V and an ancestor of a variable v_k in V. This means that no topological ordering exists where the variables in V appear consecutively.

($\Leftarrow$) Suppose $A(V) \cap D(V) = \emptyset$. By definition, $V \cap D(V) = \emptyset$. This means every element in $D(V)$ is in $U - (VA(V))$. Since $V \cap A(V) = \emptyset$, a topological ordering $\prec$ can be constructed based on the directed acyclic graph of B in which the variables in $A(V)$ appear consecutively first, followed by all variables in V, followed by all variables in $U - (VA(V))$. □

A set $V \subseteq U$ in a directed acyclic graph is an *initial segment* [12] if the parents of each v_i in V are also in V. Shafer [12] showed that if V is an initial segment, then

$$p(V) \;=\; \prod_{v_i \in V} p(v_i|P(v_i)).$$

Verma and Pearl [13] showed that d-separation is sound, i.e., $I_B(X, Y, Z) \Rightarrow I_p(X, Y, Z)$, where $I_p(X, Y, Z)$ means an independence statement $I(X, Y, Z)$ holds in $p(U)$ defined by a Bayesian network (B, C), $X, Y, Z \subseteq U$. Lastly, Lemma 4 means that potentials not involving the variable being eliminated can be ignored.

Lemma 4. [12] *If* ψ_1 *is a potential on* W *and* ψ_2 *is a potential on* Z, *then the marginalization of* $\psi_1 \cdot \psi_2$ *onto* W *is the same as* ψ_1 *multiplied with the marginalization of* ψ_2 *onto* $W \cap Z$, *where* $W, Z \subseteq U$.

We now can present our first main result, namely, that SI is sound.

Theorem 1. *In a Bayesian network* (B, C) *defining a joint distribution* $p(U)$, *suppose VE computes a potential* ψ *whose evidence normal form is* $\gamma \cdot N$. *If SI denotes the semantics of* N *as* $p_B(X|Y)$, *then* $N = p(X|Y)$.

Proof. By SI, $I_B(X_1, \emptyset, Y)$ holds. By Lemmas 1, 2, and 3, there exists a topological order of B starting with $A(XS)$ and followed by XS. Then

$$p(A(W)) \;=\; \prod_{v_i \in A(W)} p(v_i|P(v_i)),$$

where W denotes XS, and

$$p(A(W)W) \;=\; \prod_{v_i \in A(W)} p(v_i|P(v_i)) \cdot \prod_{v_i \in W} p(v_i|P(v_i)).$$

By manipulation of the above two equations,

$$p(W|A(W)) \;=\; \prod_{v_i \in W} p(v_i|P(v_i)). \tag{9}$$

Consider any $v_i \in W$ and $v_j \in A(W)$. Suppose $v_j \in D(v_i)$. Then $v_j \in D(W)$, contradicting $A(W) \cap D(W) = \emptyset$. Thus, there is no path from W to $A(W)$. And, as all paths from $A(W)$ to W necessarily go through $Y = P(W)$, $I_B(W, Y, A(WY))$ holds in B by d-separation. Thus,

$$p(W|A(W)) \;=\; p(W|Y). \tag{10}$$

By (9) and (10),

$$p(W|Y) \;=\; \prod_{v_i \in W} p(v_i|P(v_i)).$$

Summing out S on both sides yields

$$p(X|Y) \;=\; \sum_{S} \prod_{v_i \in W} p(v_i|P(v_i)). \tag{11}$$

As previously mentioned, applying Lemma 4 on N in evidence normal form $\gamma \cdot N$, gives

$$N \;=\; \sum_{S} \prod_{v_i \in W} p(v_i|P(v_i)). \tag{12}$$

By (11) and (12),

$$N \;=\; p(X|Y). \qquad \square$$

Theorem 1 guarantees that if SI denotes the semantics of a VE potential ψ as $\gamma \cdot p_B(X|Y)$, then

$$\psi = \gamma \cdot p(X|Y).$$

Recall potential $\psi(g, i = 1, j)$ in (2). As illustrated in Table 3, Theorem 1 ensures that $\psi(g, i = 1, j)$ is equal to $p(j|g, i = 1)$, since SI denotes it as $p_B(j|g, i = 1)$.

With respect to inference, the question of completeness is this. Can SI determine the semantics of every VE potential defined with respect to the joint distribution? The answer is no, due to the next result.

Table 3. Potential $\psi(g, i = 1, j)$ in (2) is $p(j|g, i = 1)$.

$$\psi(g, i = 1, j) \;=\; p(j|g, i = 1) \;=$$

i	g	j	$p_B(j\|g, i = 1)$
1	0	0	0.457
1	0	1	0.543
1	1	0	0.334
1	1	1	0.666

Lemma 5. *Using B defining $p(U)$, $I_p(X_1, \emptyset, Y) \iff$ VE builds $p(X|Y)$, where X_1 is defined in SI.*

Proof. The claim follows from the discussion in [8], where, in the notation of SI, $I_B(X_1, \emptyset, Y) \iff$ VE builds $p(X|Y)$, under the assumption that $I_B(X_1, \emptyset, Y) \iff I_p(X_1, \emptyset, Y)$. □

As it is not feasible to test every $I_p(X_1, \emptyset, Y)$ in $p(U)$, we rely on d-separation to test $I_B(X_1, \emptyset, Y)$ in B. However, it is known that independencies in $p(U)$ can escape detection in B. This means that SI will make mistakes in certain situations. However, d-separation and SI satisfy a weaker notion of completeness.

Lemma 6. [10] *Suppose that d-separation indicates that $I_B(X, Y, Z)$ does not hold in a discrete Bayesian network B on U. Then there exists a set C of CPTs for B defining a joint distribution $p(U)$ such that $I_p(X, Y, Z)$ does not hold.*

$I_B(g, \emptyset, ij)$ does not hold by d-separation in the ESBN B of Figure 1. As required by Lemma 6, there must exist a set C of CPTs, such as those in Table 1, defining a $p(U)$ such that $I_p(g, \emptyset, ij)$ does not hold. Lemma 6 can be utilized to show a similar kind of completeness for SI. First, one more result is needed.

Lemma 7. *Suppose VE computes $\psi'(X - v_i|Y) = \sum_{v_i} \psi(X|Y)$. Then ψ' and ψ are both p or both ϕ.*

Proof. It is known that $p(X - v_i|Y) = \sum_{v_i} p(X|Y)$. Now consider $\sum_{v_i} \phi(X|Y)$, where $\phi(X|Y) \neq p(X|Y)$. By Lemma 5, $\phi(X|Y)$ means $I_p(X_1, \emptyset, Y)$ does not hold, where $X_1 = X \cap P(Z)$. Now marginalization gives $\psi(X'|Y)$, where $X' = X - v_i$. Note that Y did not change. Similarly, $XS = X' \cup (Sv_i)$ meaning Z did not change. Thus, $P(Z)$ did not change. Suppose $v_i \in P(Z)$. Then there exists a $v_k \in Z$ with $(v_i, v_k) \in B$. Similar to the proof of Lemma 2, this means $v_k \in XS$. A contradiction. Thus, $v_i \notin P(Z)$. Then, by definition, $v_i \notin X_1$. Hence, $X_1 = X_1'$, where $X_1' = X' \cap P(Z)$. By above, $I_p(X_1', \emptyset, Y)$ does not hold. By Lemma 5, $\psi(X'|Y) = \phi(X - v_i|Y)$. □

Theorem 2. *In a Bayesian network B on U, suppose VE computes a potential ψ whose evidence normal form is $\gamma \cdot N$. If SI denotes the semantics of N as $\phi_B(X|Y)$, there exists a set C of CPTs for B defining a joint distribution $p(U)$ such that $N \neq p(X|Y)$.*

Proof. By SI, $I_B(X_1, \emptyset, Y)$ does not hold. There exists a C for B defining $p(U)$ such that, by Lemma 6,

$$p(Y) \neq p(Y|X_1). \tag{13}$$

We define an initial segment with four pairwise disjoint subsets: $W = XS$, Y, $Z_1 = Z - Y$ and $V = A(W) - YZ_1$. Their product is $p(WYZ_1V)$, so $p(WY)$ is

$$\sum_{VZ_1} \prod_{v_i \in W} p(v_i|P(v_i)) \cdot \prod_{v_i \in VZ_1Y} p(v_i|P(v_i)).$$

By contradiction, suppose the product of W's CPTs is $p(W|Y)$. This means

$$p(WY) = \sum_{VZ_1} p(W|Y) \cdot \prod_{v_i \in VZ_1Y} p(v_i|P(v_i)).$$

Lemma 4, and rearrangement, give

$$p(Y) = \sum_{VZ_1} \prod_{v_i \in VZ_1Y} p(v_i|P(v_i)).$$

By [3],

$$p(Y) = \sum_{VZ_1} \psi(VZ_1Y|P(VZ_1Y)).$$

We now show $P(VZ_1Y) = X_1$. Here, $P(V) \subseteq Y$, $P(Z_1) \subseteq X_1VY$ and $P(Y) \subseteq X_1VZ_1$. Therefore,

$$P(V) \cup P(Z_1) \cup P(Y) \subseteq X_1VZ_1Y.$$

By definition,

$$P(VZ_1Y) \subseteq X_1.$$

To show $X_1 \subseteq P(VZ_1Y)$, let $Y_1 = Y \cap Z$ and $Y_2 = Y - Y_1$. By definition, $X_1 \subseteq P(Z)$. Now $X_1 \subseteq P(Z) - VY_2$, since $X_1 \cap VY_2 = \emptyset$. Similarly, $X_1 \subseteq P(Z) - VY_2Z$ as $P(Z) \cap Z = \emptyset$. It follows that $X_1 \subseteq P(V) \cup P(Z) \cup P(Y_2) - VZY_2$. By definition, $X_1 \subseteq P(VZY_2)$. Finally, $Z = Z_1Y_1$ means that $P(VZY_2) = P(VZ_1Y_1Y_2) = P(VZ_1Y)$. Thus,

$$X_1 \subseteq P(VZ_1Y).$$

Hence, $X_1 = P(VZ_1Y)$ giving

$$p(Y) = \sum_{VZ_1} \psi(VZ_1Y|X_1).$$

By Lemma 7,

$$p(Y) = \sum_{VZ_1} p(VZ_1Y|X_1).$$

Taking the marginalization gives

$$p(Y) = p(Y|X_1),$$

a contradiction to (13). Therefore,

$$p(W|Y) \neq \prod_{v_i \in W} p(v_i|P(v_i)).$$

By Lemma 7, marginalizing S from both sides gives

$$p(X|Y) \neq \sum_{S} \prod_{v_i \in W} p(v_i|P(v_i)).$$

But this is our desired result, $p(X|Y) \neq N$. □

Theorem 2 states that whenever SI indicates that a potential is not defined with respect to the joint distribution, then this is true for at least one set of CPTs for the given Bayesian network. Recall once again $\psi(g, h = 0, i = 1, j)$ in (3), which SI denotes as $\phi_B(g, h = 0, l|i = 1, j)$. With respect to $p(U)$ defined by the CPTs in Table 1, we have

$$\psi(g, h = 0, i = 1, j) \neq p(g, h = 0|i = 1, j).$$

4 On the Role of d-Separation in Deciding Semantics

Our work here reveals that the last six lines of SI can be replaced with:

Compute $A(XS)$ and $D(XS)$ using T
if $A(XS) \cap D(XS) = \emptyset$
  return $p_B(X|Y)$
else
  return $\phi_B(X|Y)$

Recall $\psi(g, i = 1, j)$ in (2). The evidence expanded form is (5). Its evidence normal form $\gamma \cdot N$ is $\gamma = 1(i = 1)$ and $N = \psi(j|g, i)$. Now $X = \{j\}$ and $S = \{l, s\}$. By the transitive closure T of the ESBN, $A(XS) = \{c, d, i, g\}$ and $D(XS) = \{h\}$. Hence,

$$A(XS) \cap D(XS) = \emptyset.$$

Thus, SI denotes $\psi(g, i = 1, j)$ in (2) as $p_B(j|g, i = 1)$.

Now consider $\psi(g, h = 0, i = 1, j)$ in (3). The evidence expanded form is (7). The evidence normal form $\gamma \cdot N$ is (8). Here $N = \psi(g, h|i, j)$, as seen in (4). With $X = \{g, h\}$ and $S = \{c, d\}$, from T on the ESBN we have $A(\{c, d, g, h\}) = \{i, j, l, s\}$ and $D(\{c, d, g, h\}) = \{j, l\}$. With

$$A(XS) \cap D(XS) = \{j, l\},$$

SI denotes $\psi(g, h = 0, i = 1, j)$ in (3) as $\phi_B(g, h = 0|i = 1, j)$.

5 Conclusion

In [2], we gave an algorithm for deciding semantics in Bayesian network inference that used one d-separation test, namely, $I_B(X_1, \emptyset, Y)$. Here we have shown that $I_B(X_1, \emptyset, Y) \iff A(XS) \cap D(XS) = \emptyset$. A new version of SI based upon testing $A(XS) \cap D(XS) = \emptyset$ has a visual appeal to it in the sense that one simply determines whether or not there exists a path from any variable in XS to any other variable in XS involving at least one variable not in XS. Thereby, the version of SI proposed here is perhaps more intuitive. Whether d-separation is explicitly or implicitly used, SI brings improved clarity to denoting exact inference in Bayesian network texts, including [4, 6–8, 11, 12]. Future work will include applying the results here to differential semantics in Bayesian networks [6, 9].

References

1. Butz, C.J., Yan, W.: The semantics of intermediate CPTs in variable elimination. In: Proc. of Fifth European Workshop on Probabilistic Graphical Models, pp. 41–48 (2010)
2. Butz, C.J., Yan, W., Madsen, A.L.: d-Separation: strong completeness of semantics of intermediate CPTs in variable elimination. Submitted to the Canadian Conference on Artificial Intelligence, CAI (2013)
3. Butz, C.J., Yan, W., Lingras, P., Yao, Y.Y.: The CPT structure of variable elimination in discrete Bayesian networks. In: Ras, Z.W., Tsay, L.-S. (eds.) Advances in Intelligent Information Systems. SCI, vol. 265, pp. 245–257. Springer, Heidelberg (2010)
4. Castillo, E., Gutiérrez, J., Hadi, A.: Expert Systems and Probabilistic Network Models. Springer, New York (1997)
5. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. MIT Press, Cambridge (2009)
6. Darwiche, A.: Modeling and Reasoning with Bayesian Networks. Cambridge University Press, New York (2009)
7. Kjærulff, U.B., Madsen, A.L.: Bayesian Networks and Influence Diagrams, 2nd edn. Springer, New York (2013)
8. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press, Cambridge (2009)
9. Madsen, A.L.: A differential semantics of Lazy AR Propagation. In: Proc. of Twenty-First Annual Conference on Uncertainty in Artificial Intelligence (UAI), pp. 364–371 (1995)
10. Meek, C.: Strong completeness and faithfulness in Bayesian networks. In: Proc. of Eleventh Annual Conference on Uncertainty in Artificial Intelligence (UAI), pp. 411–418 (1995)
11. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Francisco (1988)
12. Shafer, G.: Probabilistic Expert Systems. SIAM, Philadelphia (1996)
13. Verma, T., Pearl, J.: Causal networks: semantics and expressiveness. In: Proc. of Fourth Annual Conference on Uncertainty in Artificial Intelligence (UAI), pp. 352–359 (1998)
14. Zhang, N.L., Poole, D.: A simple approach to Bayesian network computations. In: Proc. of Canadian Conference on Artificial Intelligence (CAI), pp. 171–178 (1994)

Evaluating Asymmetric Decision Problems with Binary Constraint Trees

Rafael Cabañas, Manuel Gómez-Olmedo, and Andrés Cano

Dept. Computer Science and Artificial Intelligence
University of Granada, CITIC-UGR, Spain
{rcabanas,mgomez,acu}@decsai.ugr.es

Abstract. This paper proposes the use of *binary trees* in order to represent and evaluate asymmetric decision problems with Influence Diagrams (IDs). Constraint rules are used to represent the asymmetries between the variables of the ID. These rules and the potentials involved in IDs will be represented using binary trees. The application of these rules can reduce the size of the potentials of the ID. As a consequence the efficiency of the inference algorithms will be improved.

Keywords: Influence diagrams, asymmetric decision problems, binary trees, probability trees.

1 Introduction

Influence Diagrams (IDs) [11] are a tool to represent and solve decision problems under uncertainty. Their main advantage is that they can encode the independence relations between variables allowing a compact representation. However, they have weaknesses: decision problems are usually asymmetric in the sense the set of legitimate states of variables may vary depending on different states of other variables [1]. To be represented as an ID, an asymmetric decision problem must be symmetrized and a considerable amount of unnecessary computation may be involved. Several approaches have been made to solve this drawback. Call and Miller [4], Fung and Shachter [18], Smith et al. [21], Qi et al. [17], Covaliu and Oliver [7], Shenoy [20], Nielsen and Jensen [15], Demirer and Shenoy [8], Díez and Luque [9] have proposed modifications to the IDs framework in order to deal with asymmetries.

In this paper we propose representing the qualitative information about the problem (constraints, due to asymmetries) using *binary trees* (BTs). Constraints can be easily applied to potentials reducing the number of scenarios to consider. Moreover, if BTs are too large, they can be pruned and converted into smaller trees, thus leading to approximate algorithms. We compare BTs with a previous approach for representing constraints, *numerical trees* (NTs), and show that more efficient algorithms are obtained.

The paper is organized as follows: Section 2 introduces some basic concepts about IDs and trees; Section 3 describes key issues about asymmetries and how they can be represented using BTs; Section 4 describes the evaluation algorithm

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 85–96, 2013.

adapted for working with constraints; Section 5 includes the experimental work and results; finally Section 6 details our conclusions and lines for future work.

2 Preliminaries

2.1 Influence Diagrams

An ID [11] is a generalization of a Bayesian network (BN) [16] used for representing and solving decision problems under uncertainty. An ID contains three types of nodes: *chance nodes* (representing random variables), *decision nodes* (mutually exclusive actions which the decision maker can control) and *utility nodes* (representing decision maker preferences). Fig. 1 shows an example of an ID that represents the Car Buyer problem [17].

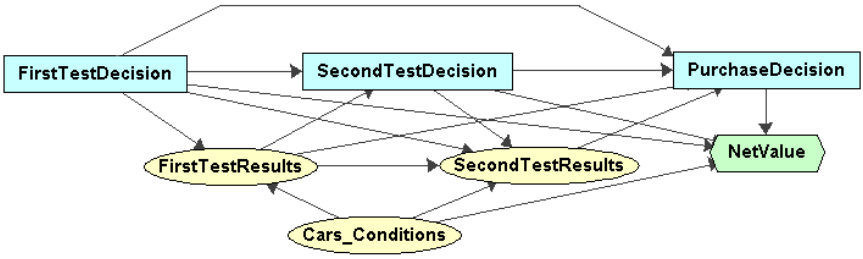


Fig. 1. Example of an ID representing the Car Buyer problem

The set of chance nodes is denoted by $\mathcal{U}_C$, the set of decision nodes is denoted by $\mathcal{U}_D$, and the set of utility nodes is denoted by $\mathcal{U}_V$. The decision nodes have a temporal order, $D_1, \ldots, D_n$, and the chance nodes are partitioned according to when they are observed: $\mathcal{I}_0$ is the set of chance nodes observed prior to any decision, and $\mathcal{I}_i$ is the set of chance nodes observed after D_i is taken and before deciding about D_{i+1}. Finally, $\mathcal{I}_n$ is the set of chance nodes never observed or observed after the last decision. That is, there is a partial temporal ordering: $\mathcal{I}_0 \prec D_1 \prec \mathcal{I}_1 \prec \cdots \prec D_n \prec \mathcal{I}_n$. For example, the temporal ordering of the ID in Fig. 1 is $FirstTestDecision \prec FirstTestResults \prec SecondTestDecision \prec SecondTestResults \prec PurchaseDecision \prec Car\_Conditions$.

Let us suppose that each variable X_i of the ID takes values on a finite set $\Omega_{X_i} = \{x_1, \ldots, x_{|\Omega_{X_i}|}\}$. If I is a set of indexes, we shall write $\mathbf{X}_I$ for the set of variables $\{X_i | i \in I\}$, defined on $\Omega_{\mathbf{X}_I} = \times_{i \in I} \Omega_{X_i}$. The elements of Ω_{X_I} are called configurations of $\mathbf{X}_I$ and will be represented as $\mathbf{x}_I$. Parents or direct predecessors of a variable X_i are denoted $pa(X_i)$.

In an ID, each chance node X_i has a conditional probability distribution $P(X_i|pa(X_i))$ attached, where $pa(X_i)$ are the parents of X_i. In the same way, each utility node V_i has a utility function $U(pa(V_i))$ attached. In general, we will talk about potentials (probability distributions are normalized potentials). Let $\mathbf{X}_I$ be the set of all variables involved in a potential, then a *probability potential*

denoted by ϕ is a mapping $\phi : \Omega_{\mathbf{X}_I} \rightarrow [0, 1]$. A *utility potential* denoted by ψ is a mapping $\psi : \Omega_{\mathbf{X}_I} \rightarrow \mathbb{R}$.

When evaluating an ID, it must be computed the best choice or *optimal policy* δ_i for each decision D_i, that is a mapping $\delta_i : \Omega_{pa(D_i)} \rightarrow \Omega_{D_i}$. The optimal policy maximizes the *expected utility* for the decision. A strategy is an ordered set of policies $\Delta = \{\delta_1, \ldots, \delta_n\}$ including a policy for each decision. An optimal strategy $\widehat{\Delta}$ returns the optimal choice the decision maker should take for each decision.

2.2 Numerical and Binary Trees

Traditionally, potentials involved in an ID have been represented using tables. An alternative representation are trees (numerical and binary)[6, 3] that will be denoted NT and BT respectively. Each internal node of the tree is labelled with a variable (random variable or decision). We use L_t to denote the *label of a node* t. Each leaf node is labelled with a number (a probability or a utility value). In a NT, each internal node has an outgoing arc for each state of the variable associated with that node. The difference between NTs and BTs is that internal nodes in BTs always have two children. As a consequence, outgoing arcs in a BT can be labelled with more than one state. We denote by $L_{lb(t)}$ and $L_{rb(t)}$ the left and right labels of a node t respectively.

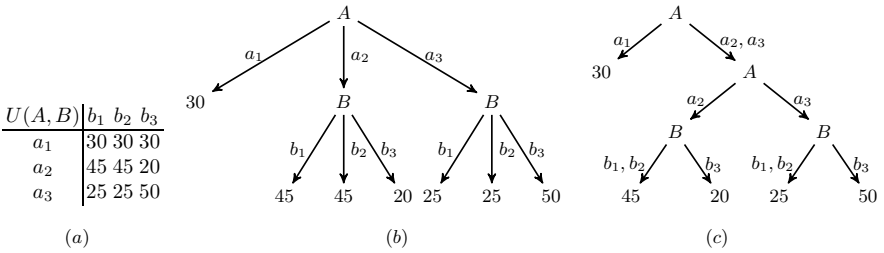


Fig. 2. Utility potential represented as (a) a table, (b) a NT and (c) a BT

Fig. 2 shows three different representations for the same utility potential: (a) a table , (b) a NT, and (c) a BT. The use of trees offer the possibility of taking advantage of *context-specific independencies* [2]. For example, when $A = a_1$, the potential will always take the value 30, regardless of the value of B. Therefore, less space is needed for representing it as a tree. Besides, BT can capture finer-grained independencies: when $A = a_2$ and $B \in \{b_1, b_2\}$, the potential will always be 45.

In a previous work [3], a comparison of the evaluation of IDs using NTs and BTs was performed using the *Variable Elimination* algorithm. The experiments showed that BTs offer better approximate solutions than NTs. The same error level will be achieved using a BT of smaller size than the corresponding NT.

2.3 Building and Approximating Trees

When a BT is built, variables are sorted in such a way that the most informative variables must be situated at the highest nodes in the tree. BTs are built from tables using a top-down approach, choosing at each step a variable and two partitions of its states that maximizes the information gain.

Definition 1 (Information Gain). *Let ϕ be the potential to be represented as a tree $\mathcal{BT}_j$ and $\mathcal{BT}_j(t, X_i, \Omega_{X_i}^{t_l}, \Omega_{X_i}^{t_r})$ the tree resulting of expanding the leaf node t with the candidate variable X_i and a partition of its available states into sets $\Omega_{X_i}^{t_l}$ and $\Omega_{X_i}^{t_r}$. Let $D(\phi, \mathcal{BT}_j)$ be the distance between a potential and a tree. The information gain can be defined as:*

$$I(t, X_i, \Omega_{X_i}^{t_l}, \Omega_{X_i}^{t_r}) = D(\phi, \mathcal{BT}_j) - D(\phi, \mathcal{BT}_j(t, X_i, \Omega_{X_i}^{t_l}, \Omega_{X_i}^{t_r})) \quad (1)$$

Kullback Leibler divergence and Euclidean distance are the distance measure used for computing the information gain with probability and utility trees respectively. If the size of a BT needs to be reduced, it can be pruned in order to get a new BT which approximates the potential. Pruning a BT consists in replacing a *terminal tree* (a node whose children are all leaves) by the average value of its leaves. If the information gain between a terminal tree and the resulting pruned tree is lower than a threshold ε, the tree is pruned. The decision of pruning a terminal tree is independent of the decision of pruning other terminal trees. Variables in trees generated during evaluation can be sorted again using the same procedure than for building: the most informative variables may not be the same than in initial trees. This process allows obtaining better approximations, but it can be a very time consuming task since it implies building again the tree. Further details about building and pruning BTs are given in [5, 3].

3 Asymmetries and Constraints

The drawback of using IDs to model asymmetric decision problem is well known. An asymmetric decision problem must be symmetrized to be represented as an ID. Therefore, a considerable amount of unnecessary computation may be involved during the evaluation. It is sometimes possible to identify the source of asymmetry and represent it with relations between variables. In our solution we try to keep qualitative (constraints due to asymmetries) and quantitative (potentials) knowledge separate, merely because qualitative knowledge may affect several distributions, with some of them not being present in the model (i.e. distributions managed during the evaluation process and derived from the initial ones). On the other hand, we attempt to store both kinds of knowledge in similar structures, making their joint application easier. In order to represent the qualitative knowledge about a decision problem, we therefore propose the use of *constraint rules*.

A *constraint rule* is an expression *antecedent* $\Rightarrow$ *consequent*. An *atomic sentence* is a pair (variable, set of values): $X_i \in \{x_i, \ldots x_j\}$. Atomic sentences can

be connected with logical operators to form *logical sentences*. Valid logical operators are $\wedge(and)$, $\vee(or)$ and $\neg(not)$. For constraint rules, both antecedents and consequents are expressed using logical sentences. For example, let us suppose that X, Y and Z take values receptively on the sets $\Omega_X = \{x_1, x_2, x_3\}$, $\Omega_Y = \{y_1, y_2\}$, $\Omega_Z = \{z_1, z_2\}$, then the constraint rule:

$$X \in \{x_1, x_3\} \wedge Y \in \{y_2\} \Rightarrow Z \in \{z_2\} \tag{2}$$

states that if X is equal to x_1 or x_3 and Y is equal to y_2 then the variable Z will always take the value z_2. Considering this constraint rule and the conditional probability $P(Z|X, Y)$, we can state that:

$$P(Z = z_1|X = x_1, Y = y_2) = P(Z = z_1|X = x_3, Y = y_2) = 0$$

The configurations $\{x_1, y_2, z_1\}$ and $\{x_3, y_2, z_1\}$ are impossible scenarios that must not be considered for computations. An atomic sentence could have an empty set of values for the consequent. For example, the constraint rule

$$X \in \{x_1, x_3\} \wedge Y \in \{y_2\} \Rightarrow Z \in \{\} \tag{3}$$

means that $\{x_1, y_2, z_1\}$, $\{x_1, y_2, z_2\}$, $\{x_3, y_2, z_1\}$ and $\{x_3, y_2, z_2\}$ are impossible scenarios.

To decide if a constraint rule for the variables $\mathbf{X}_J$ is applicable to a potential ϕ (probability or utility) for the variables $\mathbf{X}_I$, we have to check the *applicability* of the constraint rule. The applicability of the complete constraint rule depends on the logical operator involved with the atomic sentences of the rule. We use the following definitions to decide if the constraint rule is applicable. We say that an atomic sentence in a constraint rule for $\mathbf{X}_J$ is applicable to a potential $\mathbf{X}_J$ for $\mathbf{X}_I$ if the variable X_i of the atomic sentence is in $\mathbf{X}_J \cap \mathbf{X}_I$. The negation of a sentence is applicable if and only if the sentence itself is applicable. A conjunction is applicable if and only if the two conjuncts are applicable. A disjunction is applicable if and only if at least one of the disjuncts is applicable. With these definitions, the constraint rule is applicable if and only if both the antecedent and the consequent are applicable.

Sometimes the constraint rules are not applicable to any distribution of the model. For example, we could have the following situation: a constraint links the values of two decision nodes , but there is no distribution containing both variables. However, during the evaluation process the value node will depend on both of them and this will be the moment to activate the constraint.

The use of constraint rules have several advantages. First of all, they make the elicitation process easier (reducing the number of scenarios and therefore the number of parameters to assess); secondly, they help to make both qualitative and quantitative knowledge consistent; and thirdly, they clearly state invalid scenarios, making the contingent nature of the decision problem clear.

3.1 Binary Constraint Trees

In a previous work, it was proposed the use of NTs for representing constraint rules and applying them during the ID evaluation [10]. In the present paper, BTs are proposed for representing constraint rules: *binary constraint trees* (BCTs). BT can capture finer-grained independencies than those captured using NT. Potentials and constraints need less space to be represented as a BT than as NT. As a consequence, more efficient evaluation algorithms will be obtained.

Leaf nodes in a BCT contain the values 0 or 1. If $\mathcal{T}^c$ is a BCT for a constraint rule with variables $\mathbf{X}_J$, then a value of 0 in a leaf node t_n, means that the configuration of its ancestor variables corresponds to an impossible scenario in the ID. A value equal to 1 means that, taking into account only this constraint tree, the configuration is possible (it can be impossible according to another constraint).

Constraint rules and trees are useful when evaluating the ID in order to reduce the size of the potentials (probability trees and utility trees). This reduction causes that the complexity of operations (combination and marginalization) is also reduced. For applying a constraint tree $\mathcal{T}^c$ to a tree $\mathcal{T}_\phi$ from a potential ϕ, non-common variables are removed from $\mathcal{T}^c$ using max-marginalization. Then, the resulting constraint tree is combined with $\mathcal{T}_\phi$. Fig. 3 shows an example of the application of the constraint tree $\mathcal{T}^c(X, Y, X)$ obtained from the rule in Equation 3 to a utility tree $\psi(X, Y, Z)$. In the constraint tree variable Z is not present since it has previously been pruned.

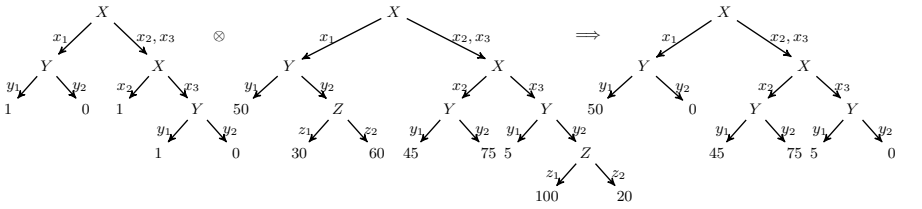


Fig. 3. Application of the constraint tree obtained from the rule in Equation 3 to a utility tree

4 Evaluating Influence Diagrams

4.1 Variable Elimination with Constraint Trees

This section shows how BCTs can be applied to evaluate IDs. In particular we have decided to work with the *Variable Elimination* algorithm (VE) and can be used for solving BNs [22] and IDs [12]. This method uses the temporal order between the decision nodes to partition the whole set of nodes according to when they are observed. Once this order has been established, the algorithm eliminates all the variables one by one, with two operations: sum-marginalization and max-marginalization. The method adapted for working with constraints is shown below.

1. Initialization phase
   (a) Build initial trees: for each potential in $\Phi \cup \Psi$ obtain a tree $\mathcal{T}_\phi$.
   (b) Apply constraints: $\forall \mathcal{T}^c$, if $\mathcal{T}^c$ is applicable to $\mathcal{T}_\phi$ then: $\mathcal{T}'_\phi = \mathcal{T}^c \otimes \mathcal{T}_\phi$, else $\mathcal{T}'_\phi = \mathcal{T}_\phi$.
   (c) Sort and prune all trees
2. While there are variables to remove
   (a) Decide next variable to remove (X) and combine all potentials (trees) containing X: Φ_X, Ψ_X. Remove X using sum-marginalization (chance nodes) or max-marginalization (decision nodes). New trees are obtained as result: $\mathcal{T}_\phi$ and $\mathcal{T}_\psi$
   (b) Apply constraints to resulting trees $\mathcal{T}_\phi$ and $\mathcal{T}_\psi$, as in 1.b.
   (c) Sort and prune the resulting trees (optional).

It can be seen in the algorithm that the global structure of VE is not changed: the only difference is that it is performed a pre-processing at the beginning and a post processing after removing each variable which modify the potentials (their size is reduced). Constraints are applied to initial potentials (1.b) and to potentials obtained from the removal of variables (2.b). This application reduces the size of potentials, but a greatest reduction can be achieved if trees are sorted and pruned (see Section 2.3). The pruning process is a time consuming task. For that reason, if the features of the problem require to evaluate the ID in a short period of time, it must only be perform during the initialization phase. However, if the reduction in the storage size is more important, it can also be performed after removing each variable.

4.2 Modified Operations

The application of constraints is performed by combining the BCTs and potential trees when needed. Combine operation was described in a previous work about using BTs for BNs inference [5]. However, the combination does not reduce the size of potentials after applying a constraint. In fact, bigger potentials can be obtained. For that reason, it should be necessary to prune the trees after applying the constraints. The inconvenient of performing the pruning process is that evaluation time can be increased.

In order to avoid pruning the trees, here we propose to modify the combine operation (see Algorithm 1). When combining two trees, it is checked if one of them is a leaf node with the value 1 or 0. In case of a 0, the algorithm will return a leaf node with the value 0 (step 1). On the other hand, if it is the value 1, it will return the other tree (steps 1 and 1). This operation requires restricting a $\mathcal{BT}$ to a set of states L of a variable X_i, denoted $\mathcal{BT}^{R(X_i,L)}$.

It must be noticed that, in order get the benefits from this new version of the operator combine, the constraint tree must be the first input argument $\mathcal{BT}_1$. The combination process is illustrated in Fig. 4. It shows the differences in the process between combining two trees with the modifications (bottom) and without them (top). The same considerations can be made for the division operation, which is used after the removal of each variable.

Input : $t1$ and $t2$ (root nodes of $\mathcal{BT}_1$ and $\mathcal{BT}_2$)
Output: The root of $\mathcal{BT} = \mathcal{BT}_1 \otimes \mathcal{BT}_2$

Build a new node t
if *($t1$ is a leaf node* ***and*** $L_{t1} == 0$*)* ***or*** *($t2$ is a leaf node* ***and*** $L_{t1} == 0$*)* **then**
  Set $L_t = 0$ the label of t
else if $t1$ *is a leaf node* ***and*** $L_{t1} == 1$ **then**
  Set $t = t2$
else if $t2$ *is a leaf node* ***and*** $L_{t2} == 1$ **then**
  Set $t = t1$
else if $t1$ *is a leaf node* **then**
  if $t2$ *is a leaf node* **then**
    $L_t = L_{t1} \cdot L_{t2}$
  else
    Set $L_t = L_{t2}$ the label of t
    Set $L_{lb(t)} = L_{lb(t2)}$ and $L_{rb(t)} = L_{rb(t2)}$ labels of the two branches of t
    Set Combine($t1$,$t2_l$) the left child of t
    Set Combine($t1$,$t2_r$) the right child of t
else
  Suppose X_j is the variable labelling node t_1
  Set $L_t = L_{t1}$ the label of t
  Set $L_{lb(t)} = L_{lb(t1)}$ and $L_{rb(t)} = L_{rb(t1)}$ labels of the two branches of t
  Set Combine($t1_l$,$\mathcal{BT}_2^{R(X_j, L_{lb(t1)})}$) the left child of t
  Set Combine($t1_r$,$\mathcal{BT}_2^{R(X_j, L_{rb(t1)})}$) the right child of t
return t

Algorithm 1. Modified combine operation

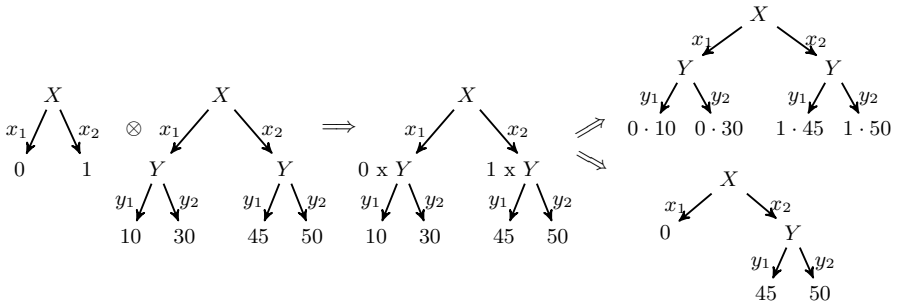


Fig. 4. Combination process: It shows the differences in the process between combining two trees with the modifications (bottom) and without them (top).

5 Experimentation

For testing purposes, two different IDs were used. First, a real world ID used for the treatment of gastric NHL disease [13] with 3 decisions, 1 utility node and 17 chance nodes. This ID contains two constraint rules between its decisions:

$$HelicobacterTreatment = \{NO\} \Longrightarrow Surgery = \{NONE\}$$
$$HelicobacterTreatment = \{NO\} \Longrightarrow CT\_RT\_Schedule = \{NONE\}$$

The second ID used represents the Car Buyer problem [17], which is shown in Fig. 1. This ID has 3 decisions, 1 utility, 3 chance nodes and the following constraint rules:

$$FirstTestDecision = \{NO\} \Longleftrightarrow FirstTestResult = \{NONE\}$$
$$FirstTestResult = \{defects2\} \Longrightarrow FirstTestDecision = \{FuelElectrical\}$$
$$SecondTestDecision = \{NO\} \Longleftrightarrow SecondTestResult = \{NONE\}$$

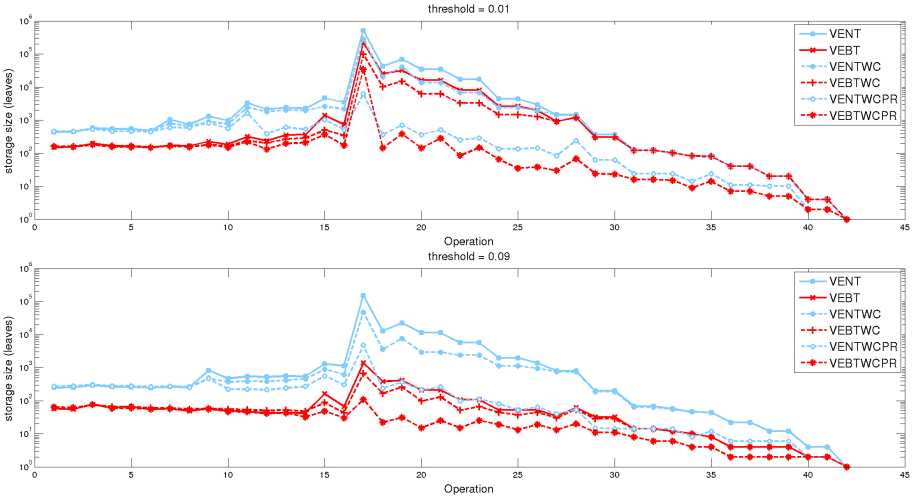


Fig. 5. Size of all potentials stored in memory during the NHL ID evaluation

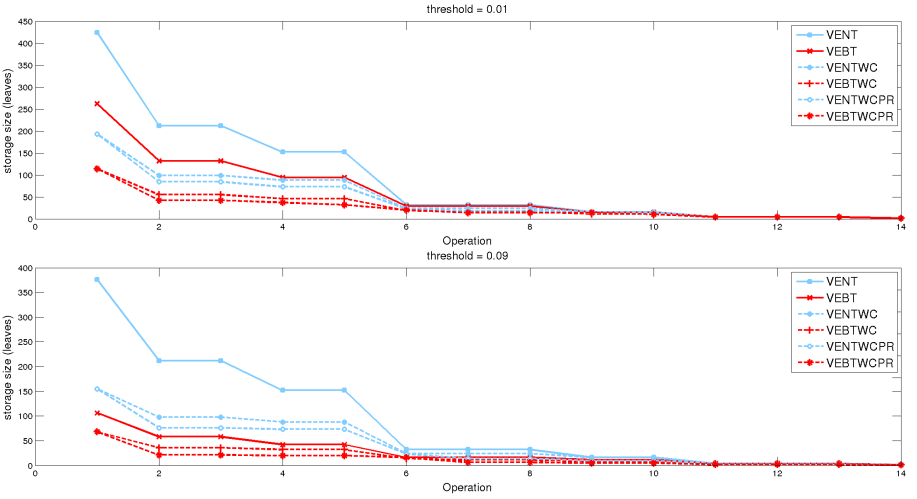


Fig. 6. Size of all potentials stored in memory during the Car Buyer ID evaluation

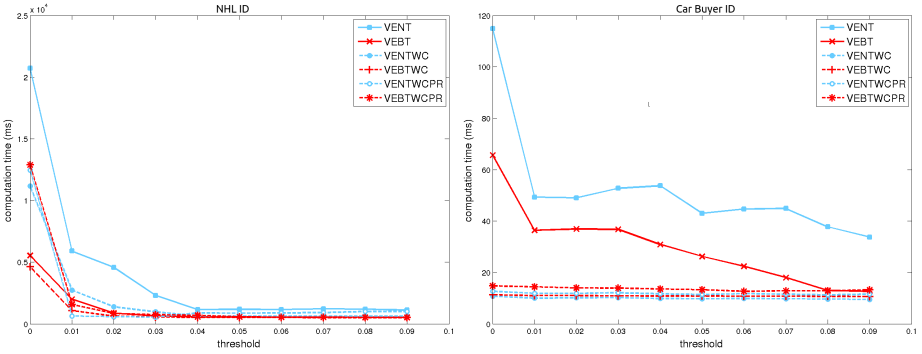


Fig. 7. Computation time for the evaluation of the diagrams NHL and Car Buyer

Both IDs were evaluated using different variations of the VE algorithm: using NTs and BTs without constraints (VENT and VEBT); using NTs and BTs with constraints (VENTWC and VEBTWC); and using NTs and BTs with constraints and using the sort and pruning operation after removing each variable (VENTWCPR and VEBTWCPR); The modifications in combination operation shown in Section 4.2 were only employed for the evaluation with constraints. The ε threshold used for pruning was ranged in the interval $[0, 0.1]$. All the algorithms were implemented in Java with the Elvira Software[1]. The tests were run on a Intel Xeon Processor E3510 (4 cores, 1.6GHz).

Graphics included in Fig. 5 and Fig. 6 show the storage requirement for both IDs using different thresholds. The vertical axis indicates the number of leaves necessary for storing all the potentials and constraints. The horizontal axis shows de number of operation. The measurement was performed after combining potentials containing a variable to be removed, and after pruning the resulting potentials of marginalization. That is, after steps 2.a and 2.c in the schema shown in Section 4.1. If evaluations with NTs (VENT, VENTWC and VENTWCPR) are compared with their equivalents using BTs (VEBT, VEBTWC and VEBTWCPR), it can be observed that, in general, less space is needed when using BTs. If constraints are applied, the size of potentials is reduced even more (VENTWC and VEBTWC), and if potentials are pruned after each operation (VENTWCPR and VEBTWCPR), the reduction is more significant. Even though constraint rules are applicable to initial potentials, an important reduction is obtained if they are applied to intermediate potentials: impossible configurations may appear in intermediate potentials during evaluation.

The reduction of the potential sizes should lead to more efficient algorithms: operations with smaller potentials should be faster. In Fig. 7 it is shown the computation time for evaluating both IDs. It can be observed that pruning after applying constraints (VENTWCPR and VEBTWCPR) is not always efficient for lower threshold values: it requires an additional computing time that is not compensated by the smaller potentials. Moreover, with higher threshold values,

[1] `http://leo.ugr.es/~elvira`

all variants of the evaluation algorithm obtain similar results. The fastest evaluation is obtained using BTs with constraints and without pruning after each operation (VEBTWC). By contrast, worst results are obtained using NTs without constraints (VENT).

6 Conclusions and Future Work

In the present paper, it is proposed a new method for representing and evaluating asymmetric decision problems with IDs: potentials and asymmetries (constraint trees) are represented using BTs. Using this kind of representation allows to reduce the number of scenarios to consider and also to approximate the potentials. The paper shows how constraints can be used to improve the efficiency of the VE algorithm. In the experimental work, it was proved that evaluating IDs with BTs and BCTs is faster and requires less storage size than using NTs. However, if BTs are pruned after removing each variable and applying constraints, the evaluation with BTs is not efficient: the overhead introduced by pruning and sorting trees is larger using BTs than with NTs.

As regards future directions of research, we shall study if applying constraints reduces the error committed when approximating potentials. It could also be interesting to study the behaviour of BTs with constraints using alternatives to the VE inference algorithm, like *Arc Reversal* [19], *Lazy propagation* [14], etc

Acknowledgments. This research was supported by the Spanish Ministry of Economy and Competitiveness under project TIN2010-20900-C04-01, the European Regional Development Fund (FEDER) and the FPI scholarship programme (BES-2011-050604). The authors have been also partially supported by "Consejería de Economía, Innovación y Ciencia de la Junta de Andalucía" under projects TIC-06016 and P08-TIC-03717.

Bibliography

1. Bielza, C., Shenoy, P.P.: A comparison of graphical techniques for asymmetric decision problems. Management Science 45(11), 1552–1569 (1999)
2. Boutilier, C., Friedman, N., Goldszmidt, M., Koller, D.: Context-specific independence in Bayesian networks. In: Proceedings of the 12th International Conference on Uncertainty in AI, pp. 115–123. Morgan Kaufmann Publishers Inc. (1996)
3. Cabañas, R., Gómez, M., Cano, A.: Approximate inference in influence diagrams using binary trees. In: Proceedings of the Sixth European Workshop on Probabilistic Graphical Models, PGM 2012 (2012)
4. Call, H.J., Miller, W.A.: A comparison of approaches and implementations for automating decision analysis. Reliability Engineering & System Safety 30(1), 115–162 (1990)
5. Cano, A., Gómez-Olmedo, M., Moral, S.: Approximate inference in Bayesian networks using binary probability trees. International Journal of Approximate Reasoning 52(1), 49–62 (2011)
6. Cano, A., Moral, S., Salmerón, A.: Penniless propagation in join trees. International Journal of Intelligent Systems 15(11), 1027–1059 (2000)

7. Covaliu, Z., Oliver, R.M.: Representation and solution of decision problems using sequential decision diagrams. Management Science 41(12), 1860–1881 (1995)
8. Demirer, R., Shenoy, P.P.: Sequential valuation networks for asymmetric decision problems. European Journal of Operational Research 169(1), 286–309 (2006)
9. Díez, F.J., Luque, M.: Representing decision problems with decision analysis networks. Technical report, UNED, Madrid, Spain (2010)
10. Gómez, M., Cano, A.: Applying numerical trees to evaluate asymmetric decision problems. In: Nielsen, T.D., Zhang, N.L. (eds.) ECSQARU 2003. LNCS (LNAI), vol. 2711, pp. 196–207. Springer, Heidelberg (2003)
11. Howard, R.A., Matheson, J.E.: Influence diagram retrospective. Decision Analysis 2(3), 144–147 (2005)
12. Jensen, F.V., Nielsen, T.D.: Bayesian networks and decision graphs. Springer (2007)
13. Lucas, P.J.F., Taal, B.: Computer-based decision support in the management of primary gastric non-hodgkin lymphoma. UU-CS, (1998-33) (1998)
14. Madsen, A.L., Jensen, F.V.: Lazy evaluation of symmetric Bayesian decision problems. In: Proceedings of the 15th Conference on Uncertainty in AI, pp. 382–390. Morgan Kaufmann Publishers Inc. (1999)
15. Nielsen, T.D., Jensen, F.V.: Representing and solving asymmetric Bayesian decision problems. In: Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence, pp. 416–425. Morgan Kaufmann Publishers Inc. (2000)
16. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausble Inference. Morgan Kaufmann Pub. (1988)
17. Qi, R., Zhang, L., Poole, D.: Solving asymmetric decision problems with influence diagrams. In: Proceedings of the Tenth International Conference on Uncertainty in Artificial Intelligence, pp. 491–497. Morgan Kaufmann Publishers Inc. (1994)
18. Shachter, R., Fung, R.: Contingent influence diagrams. Advanced Decision Systems (1990)
19. Shachter, R.D.: Evaluating influence diagrams. Operations Research, 871–882 (1986)
20. Shenoy, P.P.: Valuation network representation and solution of asymmetric decision problems. European Journal of Operational Research 121(3), 579–608 (2000)
21. Smith, J.E., Holtzman, S., Matheson, J.E.: Structuring conditional relationships in influence diagrams. Operations Research 41(2), 280–297 (1993)
22. Zhang, N.L., Poole, D.: Exploiting causal independence in Bayesian network inference. Journal of Artificial Intelligence Research 5, 301–328 (1996)

On the Equivalence between Logic Programming Semantics and Argumentation Semantics

Martin Caminada[1,2,⋆], Samy Sá[3,⋆⋆], and João Alcântara[3,⋆⋆]

[1] Université du Luxembourg
[2] University of Aberdeen
[3] Universidade Federal do Ceará

Abstract. In this paper, we re-examine the connection between formal argumentation and logic programming from the perspective of semantics. We note that one particular translation from logic programs to instantiated argumentation (the one described by Wu, Caminada and Gabbay) can serve as a basis for describing various equivalences between logic programming semantics and argumentation semantics. In particular, we are able to provide a formal connection between regular semantics for logic programming and preferred semantics for formal argumentation. We also show that there exist logic programming semantics (L-stable semantics) that cannot be captured by any abstract argumentation semantics.

1 Introduction

The link between logic programming and formal argumentation theory goes back to the seminal work of [1] in which various connections were pointed out. To some extent, the approach of abstract argumentation is a way of providing an abstraction of some aspects of logic programming. This connection is especially clear when comparing the different semantics for logic programming with the different semantics for formal argumentation. In this paper, we continue such a line of research by pointing out that the translation of [2] from logic programming to formal argumentation can account for a whole range of equivalences between logic programming semantics and formal argumentation semantics. This includes both existing results like the equivalence between stable model semantics (LP) and stable semantics (argumentation) [1], well-founded semantics (LP) and grounded semantics (argumentation) [1], and partial stable model semantics (LP) and complete semantics [2], as well as a newly proved equivalence between regular model semantics (LP) and preferred semantics (argumentation).

In this paper, besides exploiting the connection between logic programming and formal argumentation, our results shed light on some aspects of instantiated

⋆ Supported by the National Research Fund, Luxembourg (LAAMI project) and by the Engineering and Physical Sciences Research Council (EPSRC, UK), grant ref. EP/J012084/1 (SAsSy project).

⋆⋆ Supported by CNPq (Universal 2012 - Proc. n 473110/2012-1), CAPES (PROCAD 2009), CNPq/CAPES (Casadinho/PROCAD 2011).

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 97–108, 2013.

argumentation theory (e.g. [3–6]). In particular, we examine the connection between argument-labellings at the abstract level and conclusion-labellings at the instantiated level. With one notable exception, we are able to show that maximizing (or minimizing) a particular label (`in`, `out` or `undec`) at the argument level coincides with maximizing (or minimizing) the same label at the conclusion level. These results are relevant as they indicate the possibilities (and limitations) of applying argument-based abstractions to formalisms for non-monotonic reasoning.

2 Preliminaries

In the current paper, we follow the approach of Dung [1]. We will restrict ourselves to finite argumentation frameworks.

Definition 1 ([1]). *An* argumentation framework *is a pair* (Ar, Att) *where* Ar *is a finite set of arguments and* $Att \subseteq Ar \times Ar$.

Arguments are related to others by the attack relation Att: an argument A attacks B iff $(A, B) \in Att$. An argumentation framework can be seen as a directed graph where the arguments are nodes and each attack is an arrow.

Definition 2 ([1]). *(defense/conflict-free). Let* (Ar, Att) *be an argumentation framework,* $A \in Ar$ *and* $Args \subseteq Ar$. *We say* $Args$ *is* conflict-free *iff there exists no arguments* $A, B \in Args$ *such that* $(A, B) \in Att$. *We say* $Args$ defends A *iff every argument attacking* A *is attacked by some argument in* $Args$. *We define a function* $F : 2^{Ar} \rightarrow 2^{Ar}$, *such that* $F(Args) = \{A | A \; is \; defended \; by \; Args\}$, *to determine the set of all arguments defended by* $Args$. *We define* $Args^+ = \{A | A \; is \; attacked \; by \; Args\}$ *to refer to the set of arguments attacked by* $Args$.

Traditional approaches to argumentation semantics are based on extensions of arguments. Some of the mainstream approaches are summarized below:[1]

Definition 3. *(extension-based argumentation semantics). Given an argumentation framework* $AF = (Ar, Att)$, *and a conflict-free set of arguments* S:

- S *is a complete extension of* AF *iff* $S = F(S)$.
- S *is a grounded extension of* AF *iff* S *is a minimal*[2] *complete extension of* AF.
- S *is a preferred extension of* AF *iff* S *is a maximal complete extension of* AF.
- S *is a stable extension of* AF *iff* S *is a complete ext. of* AF *with* $S^+ = Ar \backslash S$.
- S *is a semi-stable extension of* AF *iff* S *is a complete ext. of* AF *with maximal* $S \cup S^+$.

As for logic programming, we will focus on propositional normal logic programs, which we will call logic programs or simply programs from now on.

[1] The characterization of the extension-based semantics in Definition 3 differs slightly from that in their original version (see [1]), but equivalence is proved in [7].

[2] When referring to minimal/maximal, we assume the underlying order is set inclusion.

Definition 4. *A* logic program *P is a set of rules of the form $c \leftarrow a_1, \ldots, a_m$, $\texttt{not}\ b_1, \ldots, \texttt{not}\ b_n$ $(m, n \in \mathbb{N})$, where c, a_i $(1 \leq i \leq m)$ and b_j $(1 \leq j \leq n)$ are atoms and* not *represents negation as failure. We say c is the* head *of the rule, and $a_1, \ldots, a_m, \texttt{not}\ b_1, \ldots, \texttt{not}\ b_n$ is its* body*. The* Herbrand Base *of P is the set HB_P of all atoms occurring in P.*

A wide range of logic programming semantics can be defined based on the 3-valued interpretations (for short, interpretation) of programs [8]:

Definition 5. *A* 3-valued interpretation *I of a program P is a pair $< T; F >$, where $T \cup F \subseteq HB_P$ and $T \cap F = \emptyset$. Atoms in T (resp. F) are intended to be true (resp. false) in I. Atoms in $U = HB_P \backslash (T \cup F)$ are considered as undefined in I.*

Let $I =< T; F >$ be a 3-valued interpretation of the program P, take P/I to be the program built by the execution of the following steps:

1. Remove any $c \leftarrow a_1, \ldots, a_m, \texttt{not}\, b_1, \ldots, \texttt{not}\, b_n \in P$ with $\{b_1, \ldots, b_n\} \cap T \neq \emptyset$;
2. Afterwards, remove any occurrence of $\texttt{not}\ b_i$ from P such that $b_i \in F$.
3. Then, replace any occurrence of $\texttt{not}\ b_i$ left by a special atom $\mathbf{u}$ ($\mathbf{u} \notin HB_P$).

We note $\mathbf{u}$ was tailored to be undefined in every interpretation of P. As shown in [8], P/I has a unique least 3-valued model: $\Psi(I) =< T_\Psi; F_\Psi >$ with minimal T_Ψ and maximal F_Ψ such that, for every $c \in HB_P$:

- $c \in T_\Psi$ if $c \leftarrow a_1, \ldots, a_m \in P/I$ and $\{a_1, \ldots, a_m\} \subseteq T_\Psi$;
- $c \in F_\Psi$ if for every $c \leftarrow a_1, \ldots, a_m \in P/I$, $\{a_1, \ldots, a_m\} \cap F_\Psi \neq \emptyset$;
- $c \in U_\Psi$ otherwise.

We now specify the logic programming semantics to be examined in this paper.

Definition 6. *Let P be a program and $I =< T, F >$ be an interpretation:*

- *I is a partial stable model (p.s.m.) of P iff $I = \Psi(I)$ [8].*
- *I is a well-founded model of P iff I is a p.s.m. of P with minimal T [8].*
- *I is a regular model of P iff I is a p.s.m. of P with maximal T [9].*
- *I is a stable model of P iff I is a p.s.m. of P where $F{=}HB_P \backslash T$, i.e., $U{=}\emptyset$ [8].*
- *I is an L-stable model of P iff I is a p.s.m. of P with maximal $T \cup F$ [9].*

3 Logic Programming as Argumentation; A 3-Step Process

The next thing to examine is how argumentation theory can be applied in the context of logic programming. Our treatment is based on [2][3]. The idea is to apply (as in [3–6]) the standard three-step process of instantiated argumentation. One starts with a knowledge base and builds the associated argumentation framework (step 1), then applies abstract argumentation semantics (step 2) and then looks at what the results of the argumentation semantics imply at the level of conclusions (step 3).

[3] One difference is that in our approach, arguments are recursive, whereas in [2], they are trees of rules. However, if one identifies the nodes of a tree with rules, one cannot apply the same rule at different positions in the argument. Our approach, which is based on [3, 4], avoids this problem.

3.1 Step 1: Argumentation Framework Construction

The approach of instantiated argumentation starts with a particular knowledge base; in our case, it will be a normal logic program. From this program, one can start to construct *arguments* recursively as follows:

Definition 7. *Let P be a logic program.*

- *If $c \leftarrow \texttt{not}\ b_1, \ldots, \texttt{not}\ b_m$ is a rule in P then it is also an argument (say A) with $\texttt{Conc}(A) = c$, $\texttt{Rules}(A) = \{c \leftarrow \texttt{not}\ b_1, \ldots, \texttt{not}\ b_m\}$, and $\texttt{Vul}(A) = \{b_1, \ldots, b_m\}$.*
- *If $c \leftarrow a_1, \ldots, a_n, \texttt{not}\ b_1, \ldots, \texttt{not}\ b_m$ is a rule in P and for each a_i ($1 \leq i \leq n$) there exists an argument A_i with $\texttt{Conc}(A_i) = a_i$ and $c \leftarrow a_1, \ldots, a_n, \texttt{not}\ b_1, \ldots, \texttt{not}\ b_m \notin \texttt{Rules}(A_i)$ then $c \leftarrow (A_1), \ldots, (A_n), \texttt{not}\ b_1, \ldots, \texttt{not}\ b_m$ is an argument (say A) with $\texttt{Conc}(A) = c$, $\texttt{Rules}(A) = \texttt{Rules}(A_1) \cup \ldots \cup \texttt{Rules}(A_n) \cup \{c \leftarrow a_1, \ldots, a_n, \texttt{not}\ b_1, \ldots, \texttt{not}\ b_m\}$, and $\texttt{Vul}(A) = \texttt{Vul}(A_1) \cup \ldots \cup \texttt{Vul}(A_n) \cup \{b_1, \ldots, b_m\}$.*

An argument A can be seen as a tree-like structure of rules (the only difference with a real tree is that a rule can occur at more than one place in A). We refer to $\texttt{Conc}(A)$ as the *conclusion* of A and $\texttt{Vul}(A)$ as the *vulnerabilities* of A.

The next step is to determine the attack relation: an argument attacks another iff its conclusion is one of the vulnerabilities of the attacked argument.

Definition 8. *Let A and B be arguments in the sense of Definition 7. We say that A* attacks *B iff $\texttt{Conc}(A) \in \texttt{Vul}(B)$.*

The notion of attack has a clear meaning: if $b \in \texttt{Vul}(A)$, then A is built using at least one rule with $\texttt{not}\ b$ in its body. Hence, A is a defeasible derivation that depends on b not being derivable. An argument B providing a (possibly defeasible) derivation of b (i.e., $\texttt{Conc}(B) = b$) can thus be seen as *attacking* A.

Now one can define the argumentation framework associated to a program:

Definition 9. *Let P be a logic program. We define its associated argumentation framework as $AF_P = (Ar_P, att_P)$ where Ar_P is the set of arguments in the sense of Definition 7 and att_P is the attack relation in the sense of Definition 8.*

3.2 Step 2: Applying Argumentation Semantics

Once the argumentation framework has been built, the next question is which arguments should be accepted and which should be rejected. As shown in Section 2, several approaches have been stated for determining this. Here we will focus on complete semantics [1], which can be defined via complete labellings [7, 10].

Definition 10. *Let $AF = (Ar, att)$ be an argumentation framework. An* argument labelling *is a function $ArgLab : Ar \rightarrow \{\texttt{in}, \texttt{out}, \texttt{undec}\}$. It is called a* complete argument labelling *iff for each $A \in Ar$ it holds that:*

- *if $ArgLab(A) = \texttt{in}$, for every $B \in Ar$ attacking A it holds $ArgLab(B) = \texttt{out}$*
- *if $ArgLab(A) = \texttt{out}$, there is a $B \in Ar$ attacking A such that $ArgLab(B) = \texttt{in}$*

– *if* $ArgLab(A) = \texttt{undec}$ *then (i) not every* $B \in Ar$ *that attacks* A *has* $ArgLab(B) = \texttt{out}$ *and (ii) no* $B \in Ar$ *that attacks* A *has* $ArgLab(B) = \texttt{in}$

With an argument labelling, one can express a position on which arguments to accept (labelled `in`), which ones to reject (labelled `out`) and which ones to abstain from having an explicit opinion about (labelled `undec`). The idea of a complete labelling is that such a position is reasonable iff one has sufficient reasons for each argument one accepts (all its attackers are rejected), for each argument one rejects (it has an attacker that is accepted) and for each argument one abstains (there are insufficient grounds to accept it and to reject it).

When *ArgLab* is an argument labelling, we write $\texttt{in}(ArgLab)$ to denote the set of $\{A \mid ArgLab(A) = \texttt{in}\}$, $\texttt{out}(ArgLab)$ for $\{A \mid ArgLab(A) = \texttt{out}\}$ and $\texttt{undec}(ArgLab)$ for $\{A \mid ArgLab(A) = \texttt{undec}\}$. As an argument labelling defines a partition among arguments, we sometimes write it as $(\mathcal{A}rgs_1, \mathcal{A}rgs_2, \mathcal{A}rgs_3)$ where $\mathcal{A}rgs_1 = \texttt{in}(ArgLab)$, $\mathcal{A}rgs_2 = \texttt{out}(ArgLab)$ and $\mathcal{A}rgs_3 = \texttt{undec}(ArgLab)$.

3.3 Step 3: Converting Argument Labellings to Conclusion Labellings

For many practical purposes, what matters are not so much the arguments themselves, but the conclusions they support. Hence, for each position on which *arguments* to accept, reject or abstain we need to specify the associated position on which *conclusions* to accept, reject or abstain. For current purposes, we follow the approach described in [11]. Here, the idea is for each conclusion to identify the "best" argument that yields it. We assume a strict total order between different individual labels such that $\texttt{in} > \texttt{undec} > \texttt{out}$. The best argument for a conclusion is the one with the highest label. If there is no argument at all for a particular conclusion, it will be labelled `out`.

Definition 11 ([11]). *Let* P *be a logic program. A conclusion labelling is a function* $ConcLab : HB_P \rightarrow \{\texttt{in}, \texttt{out}, \texttt{undec}\}$.

Let $AF_P = (Ar_P, att_P)$ *be the argumentation framework associated with* P *and ArgLab be an argument labelling of* AF_P. *We say that ConcLab is the* associated conclusion labelling *of ArgLab iff ConcLab is a conclusion labelling such that for each* $c \in HB_P$ *it holds that* $ConcLab(c) = max(\{ArgLab(A) \mid \texttt{Conc}(A) = c\} \cup \{\texttt{out}\})$ *where* $\texttt{in} > \texttt{undec} > \texttt{out}$. *We say that a conclusion labelling is* complete *iff it is associated with a complete argument labelling.*

When *ConcLab* is a conclusion labelling, we write $\texttt{in}(ConcLab)$ to denote the set of $\{c \mid ConcLab(c) = \texttt{in}\}$, $\texttt{out}(ConcLab)$ for $\{c \mid ConcLab(c) = \texttt{out}\}$ and $\texttt{undec}(ConcLab)$ for $\{c \mid ConcLab(c) = \texttt{undec}\}$. Sometimes we will write a conclusion labelling as $(\mathcal{C}oncs_1, \mathcal{C}oncs_2, \mathcal{C}oncs_3)$ where $\mathcal{C}oncs_1 = \texttt{in}(ConcLab)$, $\mathcal{C}oncs_2 = \texttt{out}(ConcLab)$ and $\mathcal{C}oncs_3 = \texttt{undec}(ConcLab)$.

4 Minimization/Maximization of Argument Labellings

In [7, 10] it was observed that for each complete argument labelling *ArgLab* of a particular argumentation framework AF, it holds that:

- `in`(*ArgLab*) is maximal among all complete argument labellings of AF iff `out`(*ArgLab*) is maximal among all complete argument labellings of AF
- `in`(*ArgLab*) is minimal among all complete argument labellings of AF iff `out`(*ArgLab*) is minimal among all complete argument labellings of AF iff `undec`(*ArgLab*) is maximal among all complete argument labellings of AF

If a complete argument labelling has maximal `in` (or equivalently, maximal `out`) we call it a *preferred argument labelling*. If it has minimal `in` (or equivalently, minimal `out` or maximal `undec`), we call it a *grounded argument labelling*. Otherwise, if it has minimal `undec`, we call it a *semi-stable argument labelling*. Lastly, if it has no argument at all that is labelled `undec`, we call it an *argstable argument labelling*.

Argument labellings and argument extensions are one-to-one related. In fact, an extension is the `in`-labelled part of the associated labelling: if *ArgLab* is a complete (resp. preferred, grounded, semi-stable or argstable) argument labelling of argumentation framework $AF = (Ar, att)$, then `in`(*ArgLab*) is a complete (resp. preferred, grounded, semi-stable or stable) extension of AF. Furthermore, if E is a complete (resp. preferred, grounded, semi-stable or stable) extension of AF then $(E, E^+, Ar \setminus (E \cup E^+))$ is a complete (resp. preferred, grounded, semi-stable or argstable) labelling of AF (see [7, 10] for details).

Note that if *ArgLab* is a complete (or respectively, preferred, grounded, semi-stable or argstable) argument labelling, then the associated conclusion labelling (Definition 11) will be called a complete (or respectively, preferred, grounded, semi-stable or argstable) conclusion labelling.

5 Minimization/Maximization of Conclusion Labellings

Preferred, grounded, semi-stable, and argstable conclusion labellings, as defined in the previous section, are based on the common idea of performing the maximization/minimization at the level of argument labellings and then identifying the associated conclusion labellings. An alternative procedure would be simply to identify *all* complete conclusion labellings and then to perform the maximization/minimization right at the level of the conclusion labellings.

It turns out that (as for argument labellings) some of the maximizations and minimisations of the conclusion labellings are equivalent to others. In [12], it is proved that for each complete conclusion labelling *ConcLab* of structured argumentation framework AF, it holds that:

- `in`(*ConcLab*) is maximal among all complete conclusion labellings of AF iff `out`(*ConcLab*) is maximal among all complete conclusion labellings of AF
- `in`(*ConcLab*) is minimal among all complete conclusion labellings of AF iff `out`(*ConcLab*) is minimal among all complete conclusion labellings of AF iff `undec`(*ConcLab*) is maximal among all complete argument labellings of AF

If a complete conclusion labelling has maximal `in` (or equivalently, maximal `out`) we call it a *regular conclusion labelling*. If it has minimal `in` (or equivalently, minimal `out` or maximal `undec`), we call it a *well-founded conclusion*

labelling. Otherwise, if it has minimal `undec`, we call it an *L-stable conclusion labelling*. Lastly, if it has no argument at all labelled `undec`, we call it a *concstable conclusion labelling*.

Conclusion labellings and logic programming models turn out to be one-to-one related. The basis of this result is [2], where the equivalence between complete conclusion labellings and partial stable models was identified. More specifically:

- if *ConcLab* is a complete conclusion labelling of structured argumentation framework AF_P (generated by a logic program P) then $< \texttt{in}(ConcLab); \texttt{out}(ConcLab) >$ is a partial stable model of P
- if $< T; F >$ is a partial stable model of P then $(T, F, HB_P \setminus (T \cup F))$ is a complete conclusion labelling of the argumentation framework AF_P

From this result, other correspondences between conclusion labellings and logic programming models follow. As a regular model is a partial stable model with maximal T, and a regular conclusion labelling is a complete conclusion labelling with maximal `in`, it follows that they correspond to each other. Similar correspondences hold between the well-founded model and the well-founded conclusion labelling, between L-stable models and L-stable conclusion labellings, and between stable models and concstable conclusion labellings. To sum up, the various types of logic programming models are actually different forms of conclusion labellings.

6 Maximizing/Minimizing Argument Labellings vs. Maximizing/Minimizing Conclusion Labellings

So far, we have selected subsets of the complete conclusion labellings as follows:

1. Perform minimization (resp. maximization) of a label at the level of complete argument labellings, then obtain the associated conclusion labellings. This procedure was described in Section 4, and is in fact similar to what is done in instantiated argumentation in general [3–6].
2. Take all complete conclusion labellings (these are the associated labellings of *all* complete argument labellings) and then perform the minimization (resp. maximization) of a particular label at the level of complete conclusion labellings. This procedure was described in Section 5 and is in fact similar to what is being done by various logic programming semantics.

An interesting question is whether the outcome of the two procedures is the same. That is, does minimizing/maximizing a label at the level of argument labellings equal to minimizing/maximizing the label at the level of conclusion labellings? We will see that the answer is "yes", with one notable exception.[4]

Theorem 1. *Let ConcLab be a conclusion labelling of logic program P and associated argumentation framework $AF_P = (Ar, att)$. It holds that ConcLab is a preferred conclusion labelling iff it is a regular conclusion labelling.*

[4] Proofs that have been omitted due to space restrictions can be found in [12].

Theorem 2. *Let ConcLab be a conclusion labelling of logic program P and associated argumentation framework $AF_P = (Ar, att)$. It holds that ConcLab is the grounded conclusion labelling iff it is the well-founded conclusion labelling.*

Theorem 3. *Let ConcLab be a conclusion labelling of logic program P and associated argumentation framework $AF_P = (Ar, att)$. It holds that ConcLab is an argstable conclusion labelling iff it is a concstable conclusion labelling.*

One can also ask whether semi-stable conclusion labellings are the same as L-stable conclusion labellings. Here, however, the answer is negative:

Example 1. Let P be the program below, whose associated argumentation framework AF_P is in Fig. 1, and let $\{A_1, A_2, A_3, A_4, A_5\}$ be arguments built from P.[5]

$$r_1 : c \leftarrow \texttt{not}\ c \qquad r_2 : a \leftarrow \texttt{not}\ b$$
$$r_3 : b \leftarrow \texttt{not}\ a \qquad r_4 : c \leftarrow \texttt{not}\ c, \texttt{not}\ a$$
$$r_5 : g \leftarrow \texttt{not}\ g, \texttt{not}\ b$$

- $A_1 = r_1$, with $\texttt{Conc}(A_1) = c$ and $\texttt{Vul}(A_1) = \{c\}$
- $A_2 = r_2$, with $\texttt{Conc}(A_2) = a$ and $\texttt{Vul}(A_2) = \{b\}$
- $A_3 = r_3$, with $\texttt{Conc}(A_3) = b$ and $\texttt{Vul}(A_3) = \{a\}$
- $A_4 = r_4$, with $\texttt{Conc}(A_4) = c$ and $\texttt{Vul}(A_4) = \{c, a\}$
- $A_5 = r_5$, with $\texttt{Conc}(A_5) = g$ and $\texttt{Vul}(A_5) = \{g, b\}$

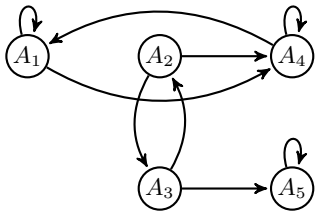


Fig. 1. The argumentation framework AF_P associated with P

The complete argument labellings of AF_P are $ArgLab_1$=($\emptyset, \emptyset$,$\{A_1, A_2, A_3, A_4, A_5\}$), $ArgLab_2$=($\{A_2\}, \{A_3, A_4\}, \{A_1, A_5\}$), and $ArgLab_3$! =($\{A_3\}, \{A_2, A_5\}, \{A_1, A_4\}$). The associated complete conclusion labellings are $ConcLab_1 = (\emptyset, \emptyset, \{a, b, c, g\})$, $ConcLab_2 = (\{a\}, \{b\}, \{c, g\})$, and $ConcLab_3 = (\{b\}, \{a, g\}, \{c\})$.

$ArgLab_2$ and $ArgLab_3$ are semi-stable argument labellings. Hence, the associated conclusion labellings $ConcLab_2$ and $ConcLab_3$ are semi-stable conclusion labellings. However, $ConcLab_2$ is not L-stable, because $\texttt{undec}(ConcLab_2)$ is not minimal. So here we have an example of a logic program where the semi-stable and L-stable conclusion labellings do not coincide.

[5] We thank Wolfgang Dvořák for this example.

7 On the Connection between Argumentation Semantics and Logic Programming Semantics

So far, we examined the general question of how argument labellings are related to conclusion labellings. We found that for complete labellings:

- maximizing `in` (or, equivalently, maximizing `out`) at the argument level yields the same result as maximizing `in` (or, equivalently, maximizing `out`) at the conclusion level. Hence, preferred conclusion labellings and regular conclusion labellings coincide.
- minimizing `in` (or, equivalently, minimizing `out` or maximizing `undec`) at the argument level yields the same result as minimizing `in` (or, equivalently, minimizing `out` or maximizing `undec`) at the conclusion level. Hence, the grounded conclusion labelling and the well-founded conclusion labelling coincide.
- minimizing `undec` at the argument level does *not* yield the same result as minimizing `undec` at the conclusion level. Hence, semi-stable conclusion labellings and L-stable conclusion labellings do *not* coincide.
- ruling out `undec` at the argument level yields the same result as ruling out `undec` at the conclusion level. Hence, argstable conclusion labellings and concstable conclusion labellings coincide.

We have now arrived at the main point of this paper: the connection between (traditional) approaches to argumentation semantics and (traditional) approaches to logic programming semantics. Let us again look at the 3-step process of Section 3. Assume that steps 1 and 3 are fixed. At step 2, it follows that

- if one applies complete semantics at step 2, the overall outcome is equivalent to calculating the partial stable models of the original logic program [2]
- if one applies preferred semantics at step 2, the overall outcome is equivalent to applying regular semantics to the original logic program
- if one applies grounded semantics at step 2, the overall outcome is equivalent to applying well-founded semantics to the original logic program
- if one applies stable semantics at step 2, the overall outcome is equivalent to applying stable model semantics to the original logic program

Thus, differences in logic programming semantics can be reduced to differences in abstract argumentation semantics (see Table 1). We are also able to explain *why* these semantics coincide, as what happens at the argument level tends to affect the conclusion level. For instance, preferred semantics coincides with regular semantics *because* maximizing `in` at either argument or conclusion level yields the same results; grounded semantics coincides with well-founded semantics *because* minimizing `in` at either argument or conclusion level yields the same results; stable semantics coincides with stable model semantics *because* ruling out `undec` at either argument or conclusion level yields the same results. Finally, semi-stable semantics does *not* coincide with L-stable model *because* minimizing `undec` at the argument level does not yield the same result as doing so at the conclusion level.

Table 1. Connections between argumentation semantics and LP semantics

Argument-Based Conclusion Labelling	Relation	Logic Programming-Based Conclusion Labelling
Preferred	$\equiv$	Regular
Grounded	$\equiv$	Well-Founded
Semi-stable	$\not\equiv$	L-stable
Argstable	$\equiv$	Concstable

8 Semi-stable and L-Stable Semantics Revisited

We will now focus on the previously observed discrepancy between semi-stable semantics and L-stable semantics. If semi-stable semantics is not able to generate L-stable conclusion labellings, then is there perhaps any other abstract argumentation semantics that can generate these? More precisely, we are interested in an abstract argumentation semantics to be applied at step 2 of the argumentation process, whose associated conclusion labellings (step 3) are precisely the L-stable labellings. Furthermore, this semantics should purely be defined on the structure of the graph (argumentation framework) and not rely on the actual contents of the arguments. That is, the semantics should satisfy the *language independence principle* [13].

Definition 12. *We say that an abstract argumentation semantics X is L-stable generating iff it is a function such that*

1. *For any logic program P, X takes as input AF_P and yields as output a set of argument labellings ArgLabs*
2. *X satisfies language independence [13, Definition 37], meaning that for any pair of argumentation frameworks AF_1, AF_2 that are isomorphic[6] by a mapping M of their arguments (the nodes in the graphs), each labelling of AF_1 can be mapped to a different labelling of AF_2 by the same mapping M.*
3. *It holds that $\{ConcLab \mid ConcLab$ is the associated conclusion labelling of some $ArgLab \in ArgLabs\}$ is precisely the set of all L-stable conclusion labellings of AF_P.*

Theorem 4. *No abstract argumentation semantics is L-stable generating.*

Proof. Consider the programs P with rules $r_1, ..., r_4$ and P' with rules $r'_1, ..., r'_4$:

$r_1 : c \leftarrow \texttt{not}\ c$	$r'_1 : d \leftarrow \texttt{not}\ c, \texttt{not}\ d$
$r_2 : a \leftarrow \texttt{not}\ b$	$r'_2 : a \leftarrow \texttt{not}\ b$
$r_3 : b \leftarrow \texttt{not}\ a$	$r'_3 : b \leftarrow \texttt{not}\ a$
$r_4 : c \leftarrow \texttt{not}\ c, \texttt{not}\ a$	$r'_4 : c \leftarrow \texttt{not}\ c, \texttt{not}\ a, \texttt{not}\ d$

The argumentation frameworks of P and P' are depicted in Fig. 2. Note that:

[6] Two argumentation frameworks AF_1, AF_2 are isomorphic (as in graph isomorphism) if there is an edge-preserving bijection from the arguments (the nodes) of AF_1 to those of AF_2, when these argumentation frameworks are perceived as graphs.

- P has three partial stable models: $S_1 =< \emptyset; \emptyset >$, $S_2 =< \{a\}; \{b\} >$ and $S_3 =< \{b\}; \{a\} >$, where S_2 and S_3 are L-stable models.
- P' has three partial stable models: $S_1 =< \emptyset; \emptyset >$, $S_2 =< \{a\}; \{b, c\} >$ and $S_3 =< \{b\}; \{a\} >$, where S_2 is the single L-stable model.

The arguments $A_1, ..., A_4$ built from P and $A'_1, ..., A'_4$ built from P' are

$A_1 : c \leftarrow \texttt{not}\ c$	$A_{1'} : d \leftarrow \texttt{not}\ c, \texttt{not}\ d$
$A_2 : a \leftarrow \texttt{not}\ b$	$A_{2'} : a \leftarrow \texttt{not}\ b$
$A_3 : b \leftarrow \texttt{not}\ a$	$A_{3'} : b \leftarrow \texttt{not}\ a$
$A_4 : c \leftarrow \texttt{not}\ c, \texttt{not}\ a$	$A_{4'} : c \leftarrow \texttt{not}\ c, \texttt{not}\ a, \texttt{not}\ d$

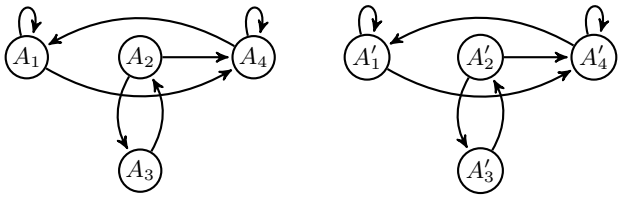


Fig. 2. The argumentation frameworks associated with P and P'

Though P has two L-stable models and P' has only one, they are indiscernible in abstract argumentation semantics. Thus, no semantics of abstract argumentation can coincide with the L-stable semantics for each and every program.

9 Discussion

In this paper, we have studied several connections between abstract argumentation semantics and logic programming semantics. We observed that various argumentation semantics are based on maximizations and minimizations (of a particular label) at the *argument level* whereas various logic programming semantics are based on maximizations and minimizations (of a particular label) at the *conclusion level*. Where performing the maximizations/minimizations at the argument level yields the same results as performing the maximizations/minimizations at the conclusion level, the associated argumentation semantics and logic programming semantics coincide. Where performing the maximizations/minimizations at the argument level does *not* yield the same results as performing the maximizations/minimizations at the conclusion level, the corresponding argumentation semantics and logic programming semantics (semistable / L-stable) do not coincide.

Although the current paper focuses mainly on instantiated argumentation based on logic programming, its main findings are in fact relevant for instantiated argumentation in general (like [3–6]) as it specifies the possibilities and impossibilities of using the argumentation approach to specify nonmonotonic entailment, or to model existing nonmonotonic formalisms. If the aim is, for instance, to model a formalism that maximizes `in` or `out` at the conclusion level

(like [14]) the argumentation approach will do fine (as evidenced by [15]). However, if the aim is to model a formalism that minimizes `undec` at the conclusion level, the argumentation approach will not be able to provide any help (Theorem 4). Hence, the current paper has shed some light on the strengths and limitations of using the argumentation approach for specifying nonmonotonic entailment.

References

1. Dung, P.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence 77, 321–357 (1995)
2. Wu, Y., Caminada, M., Gabbay, D.: Complete extensions in argumentation coincide with 3-valued stable models in logic programming. Studia Logica 93(1-2), 383–403 (2009); Special issue: new ideas in argumentation theory
3. Caminada, M., Amgoud, L.: On the evaluation of argumentation formalisms. Artificial Intelligence 171(5-6), 286–310 (2007)
4. Prakken, H.: An abstract framework for argumentation with structured arguments. Argument and Computation 1(2), 93–124 (2010)
5. Modgil, S., Prakken, H.: A general account of argumentation with preferences. Artificial Intellligence (in press, 2013)
6. Gorogiannis, N., Hunter, A.: Instantiating abstract argumentation with classical logic arguments: Postulates and properties. Artificial Intelligence 175(9-10), 1479–1497 (2011)
7. Caminada, M., Gabbay, D.: A logical account of formal argumentation. Studia Logica 93(2-3), 109–145 (2009); Special issue: new ideas in argumentation theory
8. Przymusinski, T.: The well-founded semantics coincides with the three-valued stable semantics. Fundamenta Informaticae 13(4), 445–463 (1990)
9. Eiter, T., Leone, N., Saccá, D.: On the partial semantics for disjunctive deductive databases. Ann. Math. Artif. Intell. 19(1-2), 59–96 (1997)
10. Caminada, M.: On the issue of reinstatement in argumentation. In: Fisher, M., van der Hoek, W., Konev, B., Lisitsa, A. (eds.) JELIA 2006. LNCS (LNAI), vol. 4160, pp. 111–123. Springer, Heidelberg (2006)
11. Wu, Y., Caminada, M.: A labelling-based justification status of arguments. Studies in Logic 3(4), 12–29 (2010)
12. Caminada, M., Sá, S., Alcântara, J.: On the equivalence between logic programming semantics and argumentation semantics. Technical Report ABDN–CS–13–01, University of Aberdeen (2013)
13. Baroni, P., Caminada, M., Giacomin, M.: An introduction to argumentation semantics. Knowledge Engineering Review 26(4), 365–410 (2011)
14. Pollock, J.: Cognitive Carpentry. A Blueprint for How to Build a Person. MIT Press, Cambridge (1995)
15. Jakobovits, H., Vermeir, D.: Robust semantics for argumentation frameworks. Journal of Logic and Computation 9(2), 215–261 (1999)

A Fuzzy-Rough Data Pre-processing Approach for the Dendritic Cell Classifier

Zeineb Chelly and Zied Elouedi

LARODEC, University of Tunis, High Institute of Management of Tunis, Tunisia
zeinebchelly@yahoo.fr, zied.elouedi@gmx.fr

Abstract. The Dendritic Cell Algorithm (DCA) is an immune inspired classification algorithm based on the behavior of natural dendritic cells. The DCA performance relies on its data pre-processing phase based on the Principal Component analysis (PCA) statistical method. However, using PCA presents a limitation as it destroys the underlying semantics of the features after reduction. One possible solution to overcome this limitation was the application of Rough Set Theory (RST) in the DCA data pre-processing phase; but still the developed rough DCA approach presents an information loss as data should be discretized beforehand. Thus, the aim of this paper is to develop a new DCA data pre-processing method based on Fuzzy Rough Set Theory (FRST) which allows dealing with real-valued data with no data quantization beforehand. In this new fuzzy-rough model, the DCA data pre-processing phase is based on the FRST concepts; mainly the fuzzy lower and fuzzy upper approximations. Results show that applying FRST, instead of PCA and RST, to DCA is more convenient for data pre-processing yielding much better performance in terms of accuracy.

Keywords: Dendritic cell algorithm, Fuzzy rough set theory, Feature selection, Classification.

1 Introduction

The Dendritic Cell Algorithm (DCA) [1] is a bio-inspired classification binary algorithm derived from behavioral models of natural dendritic cells (DCs) [2]. DCA has the ability to combine a series of informative signals with a sequence of repeating abstract identifiers, termed "antigens", to perform anomaly detection. To achieve this and through the pre-processing phase, DCA selects a subset of features and categorizes each selected feature into one of three signal types which are defined as "Danger Signal" (DS), "Safe Signal" (SS) and as "Pathogen-Associated Molecular Pattern"(PAMP). The resulting combination signal values are then classified to form an anomaly detection style of two-class classification.

Initially, in [3], the principal component analysis (PCA) statistical method was introduced in the DCA data pre-processing phase which is composed of two main sub-steps; namely feature reduction and signal categorization. The use of PCA aims to automatically reduce data dimension by generating new

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 109–120, 2013.

features to retain, which is achieved throughout the first sub-step, and to perform their categorization to their specific signal types (SS, DS, and PAMP), which is achieved throughout the second sub-step. However, applying PCA in the DCA data pre-processing step, destroys the underlying meaning behind the features present, initially, in the input database; seen as an undesirable property for the DCA [4].

To have a more reliable data pre-processing phase and to overcome the PCA limitation, in [4], a rough DCA version was introduced. The algorithm, named RC-DCA, is based on the application of Rough Set Theory (RST) [5] for the DCA data pre-processing task. To select features and based on the RST concepts, RC-DCA selects the most informative attributes, a subset termed *reduct*, that preserve nearly the same classification power of the original database. Furthermore, in RC-DCA, the signal categorization step is based on the RST *Reduct* and *Core* concepts. It was shown, in [4], that applying RST, instead of PCA, to DCA is more convenient for data pre-processing yielding much better performance in terms of accuracy.

However, based on rough set theory and to perform feature selection, the attribute values of the input database should be discretized beforehand. Thus, important information may be lost as a result of quantization [6]. Formally, in most databases, the attribute values may be real, and this is where RST encounters a problem. It is not possible within this theory to say whether two attribute values are similar and to what extent they are the same [6]. For instance, two close values may only differ as a result of noise, but in RST they are considered to be as different as two values of a different order of magnitude. One answer to this problem has been to discretize the dataset beforehand, producing a new database with crisp values. This is often still inadequate as it is a source of information loss; which is against the rough set objective of retaining information content [6]. This information loss may influence the RC-DCA feature selection process by generating an incorrect set of selected features; as a consequence, this will misguide the algorithm categorization phase by categorizing the features to erroneous signal categories. As a result, this will influence the algorithm classification process by generating unreliable classification results.

To overcome the RST applicability restriction, Fuzzy Rough Set Theory (FRST) [7] was introduced as a data reduction technique dealing with crisp and real-valued attributed datasets. FRST, which utilizes the extent to which values are similar, encapsulates the related but distinct concepts of vagueness (for fuzzy sets) and indiscernibility (for rough sets), both of which occur as a result of uncertainty in data; a method employing fuzzy-rough sets can handle this uncertainty. We, therefore, in this paper, propose to develop a novel fuzzy-rough DCA model based on a new feature selection and signal categorization technique. Our fuzzy-rough DCA classification model, named FBR-DCA, is based on the use of fuzzy rough set theory and more precisely on the use of the fuzzy boundary region (FBR); to guarantee a more rigorous data pre-processing phase.

The major contributions of this paper are to introduce the concept of FRST in the DCA data pre-processing phase and to show how FRST can be applied to search for the most convenient set of features to select. Additionally, we aim to show how the application of FRST can be appropriate for the categorization of each selected feature to its right type of signal. This will be achieved by avoiding the information loss already discussed, by keeping the semantics of the initial attributes and with no need for a quantization process beforehand.

2 The Dendritic Cell Algorithm

DCA is a population based system, with each agent in the system is represented as a cell. Each cell has the capacity to collect data items, termed *antigens*. Formally, the DCA initial step is the automatic data pre-processing phase where feature selection and signal categorization are achieved. More precisely, DCA selects the most important features, from the initial input database, and assigns each selected attribute to its specific signal category (SS, DS or PAMP). To do so, the PCA was used. Once data pre-processing is achieved and after calculating the values of the safe, PAMP and DS signals [8], DCA adheres these three signal categories and antigen to fix the context of each object (DC) which is the step of *Signal Processing*.

In fact, the algorithm processes its input signals (already pre-categorized) in order to get three output signals: costimulation signal (Csm), semi-mature signal (Semi) and mature signal (Mat) [8]. A migration threshold is incorporated into the DCA in order to determine the lifespan of a DC. As soon as the Csm exceeds the migration threshold; the DC ceases to sample signals and antigens. The migration state of a DC to the semi-mature state or to the mature state is determined by the comparison between cumulative $Semi$ and cumulative Mat. If the cumulative $Semi$ is greater than the cumulative Mat, then the DC goes to the semi-mature context, which implies that the antigen data was collected under normal conditions. Otherwise, the DC goes to the mature context, signifying a potentially anomalous data item. This step is known to be the *Context Assessment* phase.

The nature of the response is determined by measuring the number of DCs that are fully mature and is represented by the Mature Context Antigen Value (MCAV). $MCAV$ is applied in the DCA final step which is the *Classification* procedure and used to assess the degree of anomaly of a given antigen. The closer the $MCAV$ is to 1, the greater the probability that the antigen is anomalous. By applying thresholds at various levels, analysis can be performed to assess the anomaly detection capabilities of the algorithm. Those antigens whose $MCAV$ are greater than the anomalous threshold, which can be automatically generated from the input data, are classified as anomalous while the others are classified as normal. For a detailed description of the DCA and its implementation, please, refer to [8].

3 Rough Sets and Fuzzy-Rough Sets for Feature Selection

3.1 Fundamentals of Rough Set Theory

In Rough Set Theory (RST) [5], an *information table* is defined as a tuple $T = (U, A)$ where U and A are two finite, non-empty sets, U the *universe* of primitive objects and A the set of attributes. A may be partitioned into C and D, called *condition* and *decision* attributes, respectively. Let $P \subset A$ be a subset of attributes. The indiscernibility relation, $IND(P)$, is an equivalence relation defined as: $IND(P) = \{(x, y) \in U^2 : \forall a \in P, a(x) = a(y)\}$, where $a(x)$ denotes the value of feature a of object x. The family of all equivalence classes of $IND(P)$ is denoted by $U/IND(P)$. Equivalence classes $U/IND(C)$ and $U/IND(D)$ are respectively called *condition* and *decision* classes. For any concept $X \subseteq U$ and attribute subset $R \subseteq A$, X could be approximated using only the information contained within P by constructing the *P-lower* and the *P-upper approximations* of X defined as $\underline{P}(X) = \{x \in U | [x]_p \subseteq X\}$ and $\overline{P}(X) = \{x \in U | [x]_p \cap X \neq \emptyset\}$, respectively. The lower approximation of X is the set of objects of U that are surely in X and the upper approximation of X is the set of objects of U that are possibly in X. The tuple $< \underline{P}(X), \overline{P}(X) >$ is called a *rough set*. Let P and Q be sets of attributes inducing equivalence relations over U, then the *positive region* can be defined as: $POS_P(Q) = \bigcup_{X \in U/Q} \underline{P}(X)$. The positive region contains all objects of U that can be classified into classes of U/Q using the information in attribute P.

For feature selection, RST defines the *core* and the *reduct* concepts. The core is equivalent to the set of features which are *indispensable* attributes that cannot be removed without loss of prediction accuracy of the original database. The reduct is a combination of all these features and some features that can sometimes contribute to prediction accuracy. In RST, a subset $R \subseteq C$ is said to be a D-*reduct* of C if $POS_R(D) = POS_C(D)$ and there is no $R' \subset R$ such that $POS_{R'}(D) = POS_C(D)$. There may exist a family (F) of reducts, $RED_D^F(C)$, in T. The core is the set of attributes that are contained in all reducts, defined as: $CORE_D(C) = \bigcap RED_D^F(C)$.

3.2 Fundamentals of Fuzzy Rough Set Theory

Fuzzy Rough Set Theory (FRST) [7] comes as an extension to RST as this latter theory can only operate effectively with datasets containing discrete values. As most datasets contain real-valued attributes, it is necessary to perform a discretization step beforehand. To avoid this information loss, fuzzy rough set theory is applied.

Basic Concepts. In the same way that crisp equivalence classes are central to rough sets, *fuzzy equivalence classes* are central to the fuzzy-rough set approach. For typical applications, this means that the decision values and the conditional values may all be fuzzy. The concept of crisp equivalence classes can be extended by the inclusion of a fuzzy similarity relation S on the universe, which

determines the extent to which two elements are similar in S. The fuzzy lower and fuzzy upper approximations become $\mu_{\underline{R_P}X}(x) = inf_{y \in U} I(\mu_{R_P}(x,y), \mu_X(y))$ and $\mu_{\overline{R_P}X}(x) = sup_{y \in U} T(\mu_{R_P}(x,y), \mu_X(y))$. In the presented formulae, I is a fuzzy implicator and T is a t-norm. R_P is the fuzzy similarity relation induced by the subset of features P: $\mu_{R_P}(x,y) = \bigcup_{a \in P}\{\mu_{R_a}(x,y)\}$ where $\mu_{R_a}(x,y)$ is the degree to which objects x and y are similar for feature a. A fuzzy similarity relation can be constructed for this purpose, defined as: $\mu_{R_a}(x,y) = max(min(\frac{(a(y)-(a(x)-\sigma_a))}{(a(x)-(a(x)-\sigma_a))}, \frac{((a(x)+\sigma_a)-a(y))}{((a(x)+\sigma_a)-a(x))}), 0)$ where σ_a is the standard deviation of feature a. The tuple $< \underline{P}(X), \overline{P}(X) >$ is called a *fuzzy-rough set*. The difference between the fuzzy lower approximation, containing information regarding the extent of certainty of object membership to a given concept, and the fuzzy upper approximation, containing information regarding the degree of uncertainty of objects, generates the *fuzzy boundary region*; defined as: $\mu_{BND_{R_P}(X)}(x) = \mu_{\overline{R_P}X}(x) - \mu_{\underline{R_P}X}(x)$. This subset contains objects within the boundary region with less uncertainty.

Reduction Process. To search for the optimal subset of features, the fuzzy-rough reduct, the uncertainty for every concept has to be calculated. The uncertainty for a concept X using features in P can be calculated as follows: $U_P(X) = \frac{\sum_{x \in U} \mu_{BND_{R_P}(X)}(x)}{|U|}$. This is the average extent to which objects belong to the fuzzy boundary region for the concept X. The total uncertainty degree for all concepts, given a feature subset P, is defined as: $\gamma'_P(Q) = \frac{\sum_{X \in U/Q} U_P(X)}{|U/Q|}$.

A Fuzzy-Rough QuickReduct algorithm, defined in Fig.1, can be constructed for locating a fuzzy-rough reduct based on this measure. The task of the algorithm is to minimize the total uncertainty degree. When this reaches the minimum for the dataset, a fuzzy-rough reduct has been found. A worked example on how to compute a fuzzy-rough reduct using the Fuzzy-Rough QuickReduct algorithm, based on the fuzzy boundary region, can be found in [9].

FRQUICKREDUCT($\mathbb{C}$,$\mathbb{D}$).
$\mathbb{C}$, the set of all conditional attributes;
$\mathbb{D}$, the set of decision attributes.

(1) $R \leftarrow \{\}$; $\gamma'_{best} = 0$; $\gamma'_{prev} = 0$
(2) **do**
(3) $T \leftarrow R$
(4) $\gamma'_{prev} = \gamma'_{best}$
(5) **foreach** $x \in (\mathbb{C} - R)$
(6) **if** $\gamma'_{R \cup \{x\}}(\mathbb{D}) > \gamma'_T(\mathbb{D})$
(7) $T \leftarrow R \cup \{x\}$
(8) $\gamma'_{best} = \gamma'_T(\mathbb{D})$
(9) $R \leftarrow T$
(10) **until** $\gamma'_{best} == \gamma'_{prev}$
(11) **return** R

Fig. 1. Fuzzy-Rough QuickReduct algorithm

4 FBR-DCA: The Fuzzy-Rough Solution Approach

In this Section, we focus mainly on our FBR-DCA data pre-processing step as the rest of the fuzzy-rough FBR-DCA steps including Signal Processing, Context Assessment and the Classification procedure are performed the same as the standard DCA and as described, previously, in Section 2.

4.1 The FBR-DCA Signal Selection Process

For antigen classification, our learning problem has to select high discriminating features from the original input database which corresponds to the antigen information dataset. We may formalize this problem as an information table, where universe $U = \{x_1, x_2, \ldots, x_N\}$ is a set of antigen identifiers, the conditional attribute set $C = \{c_1, c_2, \ldots, c_A\}$ contains each feature of the information table to select and the decision attribute D of our learning problem corresponds to the class label of each sample. As FBR-DCA is based on the standard DCA concepts, except for the data pre-processing phase, and since DCA is applied to binary classification problems; then our developed FBR-DCA will be, also, applied to two-class datasets. Therefore, the decision attribute, D, of the input database of our FBR-DCA has binary values d_k: either the antigen is collected under safe circumstances reflecting a normal behavior (classified as normal) or the antigen is collected under dangerous circumstances reflecting an anomalous behavior (classified as anomalous). The condition attribute feature D is defined as follows: $D = \{normal, anomalous\}$.

For feature selection, FBR-DCA has to determine, first of all, the fuzzy boundary region for both concepts, the two-class labels, d_k. To do so, the fuzzy lower and the fuzzy upper approximations of each concept d_k for each feature c_i and for all objects x_j must be calculated. The fuzzy boundary region, the fuzzy lower and the fuzzy upper approximations are denoted by: $\mu_{BND_{R_{c_i}}(d_k)}(x_j)$, $\mu_{\underline{R_{c_i}}(\{d_k\})}(x_j)$ and $\mu_{\overline{R_{c_i}}(\{d_k\})}(x_j)$, respectively. Once the fuzzy boundary regions are measured, FBR-DCA calculates the uncertainty degrees for each attribute c_i for each concept d_k, denoted by $U_{c_i}(d_k)$, as presented in Section 3.

To find the fuzzy-rough reduct, FBR-DCA starts off with an empty set and moves to calculate the total uncertainty degrees for each feature c_i; defined as $\gamma'_{c_i}(D)$. The attribute c_m having the smallest total uncertainty degree among all the calculated total uncertainty degrees of the remaining features is added to the empty fuzzy-rough reduct set. Once the first attribute c_m is selected, FBR-DCA adds, in turn, one attribute to the selected first attribute and computes the total uncertainty degrees of each obtained attributes' couple $\gamma'_{\{c_m,c_i\}}(D)$. The algorithm chooses the couple having the smallest total uncertainty degree. The process of adding each time one attribute to the subset of the selected features continues until the total uncertainty degree of the obtained subset results in the minimal uncertainty for the dataset.

The generated subset of the selected features, constituting the fuzzy-rough reduct, shows the way of reducing the dimensionality of the original dataset by eliminating those conditional attributes that do not appear in the set. Those discarded attributes are removed in each FBR-DCA computation level since they do not add anything new to the target concept nor help the FBR-DCA to perform well its classification task. In fact, the obtained fuzzy-rough reduct includes the most informative features that preserve nearly the same classification power of the original dataset. Using the fuzzy-rough reduct concept, our method can guarantee that attributes of extracted feature patterns will be the most relevant for the FBR-DCA classification task.

4.2 The FBR-DCA Signal Categorization Process

The second step of our FBR-DCA data pre-processing phase is signal categorization. More precisely, our method has to assign for each selected attribute, produced by the previous step and which is included in the generated fuzzy-rough reduct, its definite and specific signal category. The general guidelines for signal categorization are based on the semantic of each signal type [1]:

- Safe signals: Certainly indicate that no anomalies are present.
- PAMPs: Usually mean that there is an anomalous situation.
- Danger signals: May or may not show an anomalous situation, however the probability of an anomaly is higher than under normal circumstances.

From the definitions stated above, both PAMP and SS are positive indicators of an anomalous and normal situation while the DS is measuring situations where the risk of anomalousness is high, but there is no signature of a specific cause. In other words, PAMP and SS have a certain final context (either an anomalous or a normal behavior) while the DS cannot specify exactly the final context to assign to the collected antigen. This is because the information returned by the DS is not certain as the collected antigen may or may not indicate an anomalous situation. This problem can be formulated as follows:

Based on the semantics of the mentioned signals, a ranking can be performed for these signals. More precisely, both SS and PAMP are more informative than DS which means that both of these signals can be seen as indispensable attributes; reflecting the first and the second ranking positions. To represent this level of importance, our method uses the first obtained couple of features through the fuzzy-rough reduct generation. On the other hand, DS is less informative than PAMP and SS; reflecting the last and third ranking position. Therefore, our method applies the rest of the fuzzy-rough reduct attributes, discarding the two first selected attributes that are chosen to represent the SS and PAMP signals, to represent the DS. More precisely, our method processes as follows:

As FBR-DCA has already calculated the total uncertainty degree of each attribute c_i a part, $\gamma'_{c_i}(D)$, FBR-DCA selects the first attribute c_m having the smallest total uncertainty degree to form the SS as it is considered the most informative first feature added to the fuzzy-rough reduct. With no additional

computations and since FBR-DCA has already computed the total uncertainty degree of each attributes' couple $\gamma'_{\{c_m,c_i\}}(D)$ when adding, in turn, one attribute c_i to the selected first attribute c_m that represents the SS, FBR-DCA chooses the couple having the smallest total uncertainty degree. More precisely, FBR-DCA selects that second attribute c_r having the smallest $\gamma'_{\{c_m,c_r\}}(D)$ among the calculated $\gamma'_{\{c_m,c_i\}}(D)$; to form the PAMP signal. Finally, the rest of the fuzzy-rough reduct attributes are combined and affected to the DS as it is less than certain to be anomalous.

Once the selected features are assigned to their suitable signal types, our method calculates the values of each signal category using the same process as the standard DCA [8]. The output is, thus, a new information table which reflects the signal database. In fact, the universe U of the induced signal dataset is $U = \{x'_1, x'_2, \ldots, x'_N\}$ a set of antigen identifiers and the conditional attribute set $C = \{SS, PAMP, DS\}$ contains the three signal types: SS, PAMP and DS. Once data pre-processing is achieved, FBR-DCA processes its next steps which are the Signal Processing, the Context Assessment and the Classification phase as the DCA does and as described in Section 2.

5 Experimental Setup

To test the validity of our FBR-DCA fuzzy-rough model, our experiments are performed on two-class, real-valued attributes, databases from [10]. The used databases are described in Table 1.

Table 1. Description of Databases

Database	Ref	♯ Instances	♯ Attributes
Sonar	SN	208	61
Molecular-Bio	Bio	106	59
Spambase	SP	4601	58
Cylinder Bands	CylB	540	40
Chess	Ch	3196	37
Ionosphere	IONO	351	35
Sick	Sck	3772	30
Mushroom	Mash	8124	23
Horse Colic	HC	368	23
German-Credit	GC	1000	21
Red-White-Win	RWW	6497	13

It is likely that not all of the attributes presented in the mentioned databases, are required to determine the class of each instance. Hence, feature selection, which is the first sub-step of the DCA data pre-processing phase, is needed. In [4], this is achieved by applying RST. However, as the datasets are entirely composed of real-valued attributes, discretization had to be performed. This is clearly a potential source of information loss. By applying the present work,

FBR-DCA, such loss can be reduced as attribute values are kept unchanged; no quantization is performed on the original databases. We try to show that our FBR-DCA can operate well in case of real-valued attributes avoiding the mentioned information loss while generating better classification results than when applying the crisp rough set theory. Thus, we will compare our FBR-DCA model to the crisp rough DCA approach, RC-DCA. Note that FBR-DCA and RC-DCA are based on the same concepts, except for the data pre-processing phase, as the standard DCA version, PCA-DCA. For data pre-processing, FBR-DCA applies FRST, RC-DCA applies RST and the standard DCA applies PCA.

For the DCA approaches, namely FBR-DCA, RC-DCA and PCA-DCA, each data item is mapped as an antigen, with the value of the antigen equal to the data ID of the item. For all DCA algorithms, a population of 100 cells is used. The migration threshold of an individual DC is set to 10. To perform anomaly detection, a threshold which is automatically generated from the data is applied to the MCAVs. The MCAV threshold is derived from the proportion of anomalous data instances of the whole dataset. Items below the threshold are classified as class one and above as class two. The resulting classified antigens are compared to the labels given in the original datasets. For each experiment, the results presented are based on mean MCAV values generated across a 10-fold cross validation.

We evaluate the performance of the DCA methods in terms of number of extracted features, running time, sensitivity, specificity and accuracy which are defined as: $Sensitivity = TP/(TP + FN)$; $Specificity = TN/(TN + FP)$; $Accuracy = (TP + TN)/(TP + TN + FN + FP)$; where TP, FP, TN, and FN refer respectively to: true positive, false positive, true negative and false negative. We will, also, compare the classification performance of our FBR-DCA method to well known classifiers which are the Support Vector Machine (SVM), Artificial Neural Network (ANN) and the Decision Tree (DT) and to the standard DCA version, PCA-DCA. The parameters of SVM, ANN and DT are set to the most adequate parameters to these algorithms using the Weka software. All experiments are run on a Sony Vaio G4 2.67 Ghz machine.

FRST has been experimentally evaluated with other leading feature selection techniques, such as Relif-F and entropy-based approaches in [11], and has been shown to outperform these in terms of resulting classification performance. Hence, only comparison to fuzzy rough set theory and rough set theory are given here. In addition, in [4], it was already shown that RC-DCA outperforms PCA-DCA. Thus, comparisons are made between FBR-DCA and RC-DCA.

6 Results and Analysis

Let us remind that the first step of the DCA classification algorithm is data pre-processing which is based on the use of PCA [3]. In [4], results showed that applying PCA for both feature selection and signal categorization is not convenient for the DCA as both phases are not consistent. It was, also, shown that applying rough set theory with DCA is a good alternative leading to a

better classification performance. However, the developed RC-DCA rough model suffers from a main limitation which is the performance of data discretization beforehand.

Table 2. Comparison Results of DCA Approaches

Database	Specificity(%)		Sensitivity(%)		Accuracy(%)		Time(s)		♯ Attributes	
	DCA		DCA		DCA		DCA		DCA	
	RC	FBR	RC	FBR	RC	FBR	RC	FBR	RC	FBR
SN	93.82	97.93	90.10	97.29	91.82	97.59	1705.79	14.87	20	9
Bio	79.24	92.45	77.35	86.79	78.30	89.62	1679.53	13.58	19	9
SP	98.49	99.89	98.40	99.77	98.45	99.84	3184.83	2119.95	8	8
CylB	97.75	98.39	97.00	97.00	97.46	97.85	1441.93	29.06	7	5
Ch	98.88	98.82	98.80	99.40	98.84	99.12	1779.83	714.95	11	4
IONO	97.33	99.11	96.82	98.41	97.15	98.86	668.32	41.12	19	9
Sck	97.68	99.09	96.96	96.53	97.64	98.93	1401.43	704.95	20	14
Mash	99.76	99.95	99.51	99.92	99.64	99.93	4567.34	4092.6	6	3
HC	94.73	97.36	93.05	96.29	93.75	96.73	260.08	39.84	14	5
GC	90.77	90.35	89.05	87.95	90.30	89.70	533.72	196.9	17	10
RWW	99.49	99.37	99.22	99.18	99.29	99.23	2201.98	1599.11	6	3

In this Section, we aim to show that applying FRST, instead of RST, can avoid the information loss caused by the mandatory step of data quantization. We, also, aim to show that by leaving the attribute values unchanged, our proposed FBR-DCA algorithm is able to select fewer features than the crisp rough RC-DCA approach, leading to better guide the FBR-DCA algorithm classification process. This is confirmed by the results presented in Table 2. For instance, from Table 2, we can notice that our new fuzzy-rough DCA model, FBR-DCA, has fewer features than the rough DCA model, RC-DCA. This is explained by the fact that FBR-DCA, by applying the Fuzzy-Rough QuickReduct algorithm, incorporates the information usually lost in crisp discretization by utilizing the fuzzy boundary region to provide a more informed technique. The results show that FBR-DCA selects features without much loss in information content. Our FBR-DCA new approach performs much better than traditional RST on the whole, in terms of both feature selection and classification quality. For instance, applying FBR-DCA to the Bio database, the number of selected attributes is 9. However, when applying RC-DCA to the same database, the number of selected features is set to 19. A second example can be the HC dataset where the number of selected features, by applying FBR-DCA, is reduced by more than 50% (5 features) in comparison to the number of features selected by the crisp rough DCA model, RC-DCA, which is set to 14.

Furthermore, from Table 2, we can notice that our FBR-DCA outperforms RC-DCA in terms of classification accuracy. For instance, when applying the algorithms to the SN dataset, the classification accuracy of FBR-DCA is set to 97.59%. However, when applying RC-DCA to the same database, the accuracy is set to 91.82%. Same remark is observed for the specificity and the sensitivity

criteria. When comparing the results in terms of running time, we can notice that the time taken by our FBR-DCA to process is less than the time needed by RC-DCA to function. This is explained by the fact that our FBR-DCA generates only one fuzzy-rough reduct as it is based on the Fuzzy-Rough QuickReduct algorithm. In contrast, RC-DCA generates all possible reducts that can be produced from data. Obviously, this is an expensive solution to the problem. Most of the time only one reduct is required as, typically, only one subset of features is used to reduce a dataset, so all the calculations involved in discovering the rest are pointless. Moreover, RC-DCA proposes different solutions for signal categorization; in case where the algorithm generates one reduct and when the algorithm generates a family of reducts; which is seen as a time consuming task. For example, when applying the algorithms to the Bio database, the amount of time taken by our FBR-DCA to process is 13.58(s) which is much less than the time taken by RC-DCA which is set to 1679.53(s).

We have, also, compared the performance of our FBR-DCA to other classifiers which are SVM, ANN and DT. The comparison made is in terms of the average of accuracies on the databases presented in Table 1. Fig.2 shows that the standard PCA-DCA has nearly the same classification performance as SVM and a better one than ANN and DT. It, also, shows that RC-DCA outperforms all the mentioned classifiers including the PCA-DCA in terms of overall accuracy. This is explained by the fact that RC-DCA applies rough set theory, instead of PCA, in the algorithm data pre-processing phase. Most importantly, the highest classification accuracy is noticed for our fuzzy-rough DCA new model, FBR-DCA. These promising FBR-DCA results are explained by the appropriate application of FRST to the DCA data pre-processing phase. This makes the algorithm a better classifier by generating more reliable and more pertinent results.

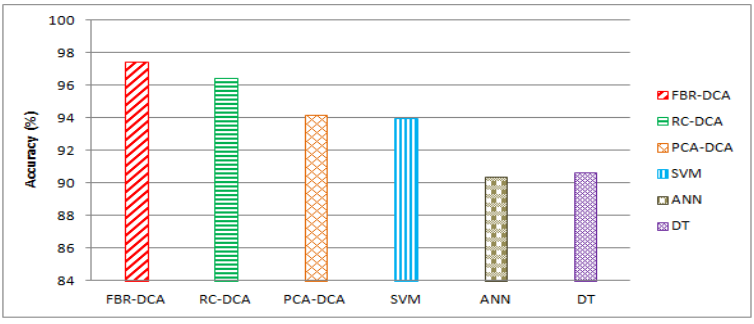


Fig. 2. Classifiers' Average Accuracies

To summarize, we have shown, in this Section, that our proposed FBR-DCA has the advantages of selecting fewer features than our proposed first work; RC-DCA. FBR-DCA is capable of avoiding the information loss caused by the use of the crisp rough set theory. The application of FBR-DCA to the unchanged attribute values led our new fuzzy-rough model to better guide its classification task yielding better performance in terms of classification accuracy. FBR-DCA

is, also, characterized by its lightweight in terms of running time in comparison to RC-DCA. Another characteristic of our FBR-DCA approach, when comparing it to the standard DCA version when applying PCA, is that it holds the semantics of the initial attributes. Adding to this, our fuzzy-rough DCA model, FBR-DCA, can effectively select features with no need for user-supplied information.

7 Conclusion and Future Works

In this paper, we have proposed a new hybrid DCA classification model based on fuzzy rough set theory. Our model aims to select the convenient set of features and to perform their signal categorization using the Fuzzy-Rough QuickReduct algorithm. Our proposed solution, FBR-DCA, ensures a more rigorous data pre-processing, for the DCA, when dealing with databases with real-valued attributes. Results show that FBR-DCA is capable of performing better its classification task than the standard DCA, the crisp rough RC-DCA model and other classifiers.

References

1. Greensmith, J., Aickelin, U., Cayzer, S.: Introducing dendritic cells as a novel immune-inspired algorithm for anomaly detection. In: Jacob, C., Pilat, M.L., Bentley, P.J., Timmis, J.I. (eds.) ICARIS 2005. LNCS, vol. 3627, pp. 153–167. Springer, Heidelberg (2005)
2. Lutz, M., Schuler, G.: Immature, semi-mature and fully mature dendritic cells: which signals induce tolerance or immunity? Trends in Immunology 23, 445–449 (2002)
3. Feng, G., Greensmith, J., Oates, R., Aickelin, U.: Pca 4 dca: The application of principal component analysis to the dendritic cell algorithm. In: Annual Workshop on Computational Intelligence (2009)
4. Chelly, Z., Elouedi, Z.: RC-DCA: A new feature selection and signal categorization technique for the dendritic cell algorithm based on rough set theory. In: Coello Coello, C.A., Greensmith, J., Krasnogor, N., Liò, P., Nicosia, G., Pavone, M. (eds.) ICARIS 2012. LNCS, vol. 7597, pp. 152–165. Springer, Heidelberg (2012)
5. Pawlak, Z.: Rough sets. International Journal of Computer and Information Science 11, 341–356 (1982)
6. Jensen, R., Shen, Q.: Fuzzy-rough sets for descriptive dimensionality reduction. In: IEEE International Conference on Fuzzy Systems, pp. 29–34 (2002)
7. Dubois, D., Prade, H.: Putting rough sets and fuzzy sets together. Kluwer Academic Publishers, Dordrecht (1992)
8. Greensmith, J., Aickelin, U., Twycross, J.: Articulation and clarification of the dendritic cell algorithm. In: Bersini, H., Carneiro, J. (eds.) ICARIS 2006. LNCS, vol. 4163, pp. 404–417. Springer, Heidelberg (2006)
9. Jensen, R., Shen, Q.: New approaches to fuzzy-rough feature selection. IEEE Transactions on Fuzzy Systems 17, 824–838 (2009)
10. Asuncion, A., Newman, D.J.: UCI machine learning repository (2007), `http://mlearn.ics.uci.edu/mlrepository.html`
11. Jensen, R., Shen, Q.: Fuzzy-rough sets assisted attribute selection. IEEE Transactions on Fuzzy Systems 15, 73–89 (2007)

Compiling Probabilistic Graphical Models Using Sentential Decision Diagrams

Arthur Choi, Doga Kisa, and Adnan Darwiche

University of California, Los Angeles, California 90095, USA
{aychoi,doga,darwiche}@cs.ucla.edu

Abstract. Knowledge compilation is a powerful approach to exact inference in probabilistic graphical models, which is able to effectively exploit determinism and context-specific independence, allowing it to scale to highly connected models that are otherwise infeasible using more traditional methods (based on treewidth alone). Previous approaches were based on performing two steps: encode a model into CNF, then compile the CNF into an equivalent but more tractable representation (d-DNNF), where exact inference reduces to weighted model counting. In this paper, we investigate a bottom-up approach, that is enabled by a recently proposed representation, the Sentential Decision Diagram (SDD). We describe a novel and efficient way to encode the factors of a given model directly to SDDs, bypassing the CNF representation. To compile a given model, it now suffices to conjoin the SDD representations of its factors, using an `apply` operator, which d-DNNFs lack. Empirically, we find that our simpler approach to knowledge compilation is as effective as those based on d-DNNFs, and at times, orders-of-magnitude faster.

1 Introduction

There are a variety of algorithms for performing exact inference in probabilistic graphical models; see, e.g., [5,12]. They typically have time complexities that are exponential in the treewidth of a given model, which make them unsuitable for models with high treewidths. Another approach to exact probabilistic inference, known as *knowledge compilation,* is capable of exploiting local structure in probabilistic graphical models, such as determinism and context-specific independence, allowing one to conduct exact inference efficiently, even in models with high treewidth. The basic idea is to encode then compile a given model into a target representation, where local structure can be exploited in more natural ways. Exact inference then reduces to weighted model counting, where the complexity of inference is now just linear in the size of the representation found [4,2]. The challenge is then to find effective encodings of a probabilistic graphical model that can be efficiently compiled to representations of manageable size.

Previous approaches based on knowledge compilation can be summarized as performing two steps [4,2]. First, a given model is encoded as a CNF, where exact inference corresponds to weighted model counting in the CNF. Second, this CNF is compiled into a more tractable representation called deterministic,

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 121–132, 2013.

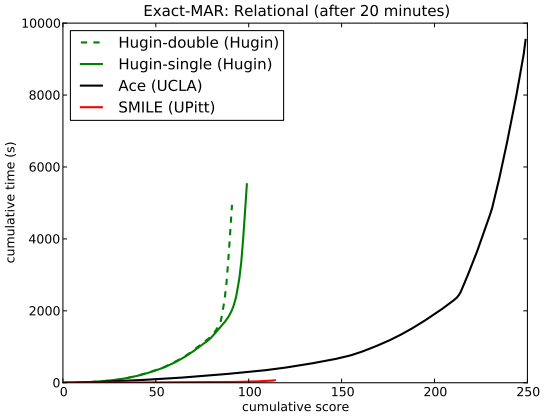


Fig. 1. The performance of ACE at the UAI'08 evaluation of probabilistic reasoning systems, the only system to exactly solve all 250 benchmarks in the `relational` suite

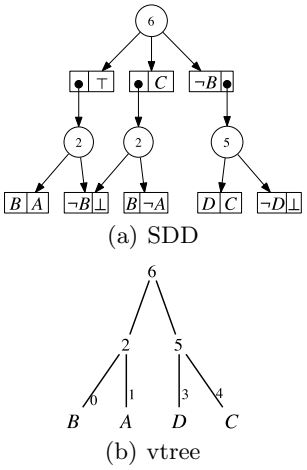


Fig. 2. An SDD and vtree for $(A \wedge B) \vee (B \wedge C) \vee (C \wedge D)$

decomposable negation normal form (d-DNNF) [8]. The effectiveness of this approach depends critically on (1) how we encode a model as a CNF, and (2) on how we compile the CNF into d-DNNF. The ACE system implements this approach, using the C2D system to compile a CNF into a d-DNNF.[1] Figure 1 illustrates the performance benefits of ACE, at the UAI'08 evaluation [7] on an example suite of high treewidth networks that are synthesized from relational models. The average cluster size for this suite is greater than 50, making them infeasible to solve without exploiting local structure.

We propose here a simpler bottom-up approach for compiling probabilistic graphical models, that is enabled by a recently proposed representation, the Sentential Decision Diagram (SDD) [6]. Unlike d-DNNFs, an efficient `apply` operation is available for SDDs, which allows one to conjoin and disjoin two SDDs efficiently. This allows us to bypass intermediate representations in CNF, and encode the factors of a model directly to SDDs. Compilation then reduces to conjoining factors together, as SDDs, also using the `apply` operator. Encoding and compilation are now expressed in common terms, enabling novel, more efficient ways to exploit local structure. Empirically, this leads to a more efficient compilation algorithm, sometimes by orders-of-magnitude. In the process, we propose further a new cardinality minimization algorithm for SDDs.

2 Probabilistic Inference as Weighted Model Counting

We first review how to reduce inference in probabilistic graphical models to weighted model counting, as in [2]. As a running example, we use a simple

[1] ACE is available at `http://reasoning.cs.ucla.edu/ace/`, and C2D is available at `http://reasoning.cs.ucla.edu/c2d/`.

Bayesian network $A \rightarrow B$, where variable A has 2 states a and $\bar{a}$, and variable B has 2 states b and $\bar{b}$. This network has two CPTs, a CPT Θ_A with 2 parameters θ_a and $\theta_{\bar{a}}$, and a CPT $\Theta_{B|A}$ with 4 parameters $\theta_{b|a}, \theta_{\bar{b}|a}, \theta_{b|\bar{a}}$ and $\theta_{\bar{b}|\bar{a}}$.

We can encode a Bayesian network as a propositional knowledge base Δ represented in CNF, whose weighted model count will correspond to the probability of evidence in a Bayesian network: $Pr(\mathbf{e}) = \sum_{\mathbf{x} \sim \mathbf{e}} \prod_X \theta_{x|\mathbf{u}}$, where $\mathbf{x}$ is a complete network instantiation, $\mathbf{e}$ is an evidence instantiation, and relation $\sim$ denotes compatibility between two instantiations (they agree on the values of common variables). For probabilistic graphical models in general, weighted model counts correspond to partition functions.

We first define the propositional variables of the CNF. First, for each BN variable X we define *indicator variables* I_x of the CNF, one variable I_x for each value x of BN variable X. Second, for each CPT $\Theta_{X|\mathbf{U}}$ of our BN, we define *parameter variables* $P_{x|\mathbf{u}}$, one variable $P_{x|\mathbf{u}}$ for each CPT parameter $\theta_{x|\mathbf{u}}$. In our running example, we have the CNF variables:

BN variables	CNF variables
A	$I_a, I_{\bar{a}}$
B	$I_b, I_{\bar{b}}$

BN CPTs	CNF variables
Θ_A	$P_a, P_{\bar{a}}$
$\Theta_{B\|A}$	$P_{b\|a}, P_{\bar{b}\|a}, P_{b\|\bar{a}}, P_{\bar{b}\|\bar{a}}$

We have two types of clauses in our CNF. First, for each BN variable, we have *indicator clauses*, which enforce a constraint that exactly one of the corresponding indicator variables is true. For each CPT, we have *parameter clauses*, which, given the indicator clauses, enforce a constraint that exactly one of the corresponding parameter variables is true (the one consistent with the indicator variables). In our example, we thus have the clauses:[2]

BN variables	CNF clauses	
A	$I_a \vee I_{\bar{a}}$	$\neg I_a \vee \neg I_{\bar{a}}$
B	$I_b \vee I_{\bar{b}}$	$\neg I_b \vee \neg I_{\bar{b}}$

BN CPTs	CNF clauses	
Θ_A	$I_a \Leftrightarrow P_a$	$I_{\bar{a}} \Leftrightarrow P_{\bar{a}}$
$\Theta_{B\|A}$	$I_a \wedge I_b \Leftrightarrow P_{b\|a}$	$I_a \wedge I_{\bar{b}} \Leftrightarrow P_{\bar{b}\|a}$
	$I_{\bar{a}} \wedge I_b \Leftrightarrow P_{b\|\bar{a}}$	$I_{\bar{a}} \wedge I_{\bar{b}} \Leftrightarrow P_{\bar{b}\|\bar{a}}$

To do weighted model counting, we need to specify weights on each CNF literal. For each indicator variable, we set both literal weights $W(I_x)$ and $W(\neg I_x)$ to one. For each parameter variable, we set the positive literal weight $W(P_{x|\mathbf{u}})$ to the value of the corresponding BN parameter $\theta_{x|\mathbf{u}}$, and the negative literal weight $W(\neg P_{x|\mathbf{u}})$ to one. The models w of the resulting knowledge base Δ are now in one-to-one correspondence with rows of the joint distribution table induced by our BN. The weight of a model w is $W(w) = \prod_{w \models \ell} W(\ell)$, and the weighted model count of Δ is $\mathsf{wmc}(\Delta) = \sum_{w \models \Delta} W(w)$. For example, we have the following model w and model weight $W(w)$:

$$w = (I_a, \neg I_{\bar{a}}, \neg I_b, I_{\bar{b}}, P_a, \neg P_{\bar{a}}, \neg P_{b|a}, P_{\bar{b}|a}, \neg P_{b|\bar{a}}, \neg P_{\bar{b}|\bar{a}})$$
$$W(w) = W(P_a) \cdot W(P_{\bar{b}|a}) = \theta_a \cdot \theta_{\bar{b}|a} = Pr(a, \bar{b}).$$

[2] $I_a \wedge I_b \Leftrightarrow P_{b|a}$ is shorthand for the clauses $(\neg I_a \vee \neg I_b \vee P_{b|a}), (I_a \vee \neg P_{b|a})$ $(I_b \vee \neg P_{b|a})$.

Further, the weighted model count is one, just as a BN's joint probability table sums to one. We incorporate evidence by setting to zero the weights of any indicator variable I_x that is not compatible with the evidence. The weighted model count then corresponds to the probability of evidence in a BN.

2.1 Exploiting Local Structure

Zero Parameters. It is straightforward to encode determinism using CNFs. Say that the parameter $\theta_{b|a}$ is zero. Any model w where parameter variable $P_{b|a}$ appears positively has a model weight that is zero, since $W(P_{b|a}) = 0$. We can thus replace the parameter clauses for the BN parameter $\theta_{x|\mathbf{u}}$ with the single clause $\neg I_a \vee \neg I_b$, which also forces to zero the weight of a model where parameter variable $P_{b|a}$ appears positively. The parameter variable $P_{b|a}$ is now superfluous, and can be removed from the domain of the knowledge base Δ.

Equal Parameters. Efficiently encoding equal parameters can be a more subtle process. Say that two parameters of a CPT, say $\theta_{b|a}$ and $\theta_{\bar{b}|\bar{a}}$, have the same value p. The parameter clauses (given the indicator clauses) guarantee that exactly one parameter variable from each CPT appears positively in any model $w \models \Delta$. This allows us to use a common parameter variable P_p, for equal parameters. If these parameters have clauses $I_a \wedge I_b \Leftrightarrow P_{b|a}$ and $I_{\bar{a}} \wedge I_{\bar{b}} \Leftrightarrow P_{\bar{a}|\bar{b}}$ we first instead assert the clauses $I_a \wedge I_b \Rightarrow P$ and $I_{\bar{a}} \wedge I_{\bar{b}} \Rightarrow P$. This by itself is not sufficient, as the resulting knowledge base Δ admits too many models (the above clauses, by themselves, do not prevent parameter variable P from being set to true when neither $I_a \wedge I_b$ nor $I_{\bar{a}} \wedge I_{\bar{b}}$ are true). We can filter out such models once we compile the resulting CNF into d-DNNF, by performing cardinality minimization [2].

3 Sentential Decision Diagrams

The Sentential Decision Diagram (SDD) is a newly introduced target representation for propositional knowledge bases [6]. It is a strict subset of deterministic, decomposable negation normal form (d-DNNF), used by the ACE system. Figure 2(a) depicts an SDD: paired-boxes $\boxed{p\,|\,s}$ are called *elements* and represent conjunctions ($p \wedge s$), where p is called a *prime* and s is called a *sub*. Circles are called *decision nodes* and represent disjunctions of the corresponding elements. SDDs satisfy stronger properties than d-DNNFs, allowing one, for example, to conjoin two SDDs in polytime. In contrast, this is not possible in general with d-DNNFs [8]. As we shall show, the ability to conjoin SDDs efficiently is critical for incremental, bottom-up compilation of probabilistic graphical models.

An SDD is constructed for a given *vtree,* which is a full binary tree whose leaves are in one-to-one correspondence with the given variables; see Figure 2(b). The SDD is canonical for a given vtree (under some conditions) and its size depends critically on the vtree used. Ordered Binary Decision Diagrams (OBDDs) [1] are a strict subset of SDDs: OBDDs correspond precisely to SDDs that are constructed using a special type of vtree, called a right-linear vtree [6]. Theoretically,

SDDs come with size upper bounds (based on treewidth) [6] that are tighter than the size upper bounds that OBDDs come with (based on pathwidth) [13,11,9]. In practice, dynamic compilation algorithms can find SDDs that are orders-of-magnitude more succinct than those found using OBDDs [3]. Compilation to d-DNNF has also compared favorably against bottom-up compilation using OBDDs in other probabilistic representations [10].

Every decision node in an SDD is *normalized* for some vtree node. In Figure 2(a), each decision node is labeled with the vtree node it is normalized for. Consider a decision node with elements $\boxed{p_1}\boxed{s_1}, \ldots, \boxed{p_n}\boxed{s_n}$, and suppose that it is normalized for a vtree node v which has variables $\mathbf{X}$ in its left subtree and variables $\mathbf{Y}$ in its right subtree. We are then guaranteed that each prime p_i will only mention variables in $\mathbf{X}$ and that each sub s_i will only mention variables in $\mathbf{Y}$ (this ensures decomposability). Moreover, the primes are guaranteed to represent propositional sentences that are consistent, mutually exclusive, and exhaustive (this ensures determinism). For example, the top decision node in Figure 2(a) has elements that represent the following sentences:

$$\{(\underbrace{A \wedge B}_{\text{prime}}, \underbrace{\mathsf{true}}_{\text{sub}}), (\underbrace{\neg A \wedge B}_{\text{prime}}, \underbrace{C}_{\text{sub}}), (\underbrace{\neg B}_{\text{prime}}, \underbrace{D \wedge C}_{\text{sub}})\}$$

One can verify that these primes and subs satisfy the properties above.

In our experiments, we use the SDD package developed by the Automated Reasoning Group at UCLA.[3] This package allows one to efficiently conjoin, disjoin and negate SDDs, in addition to computing weighted model counts in time that is linear in the size of the corresponding SDD.

4 Bottom-Up Compilation into SDDs

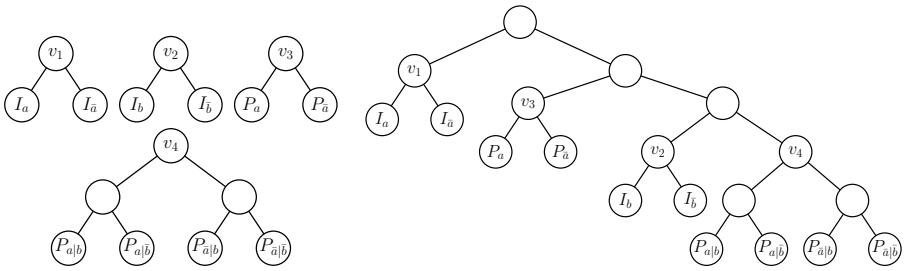
There are a number of steps we need to take in order to compile a given probabilistic graphical model into an equivalent representation as an SDD. At each step, we make certain decisions that can have a significant impact on the size of the resulting SDD, as well as on the efficiency of constructing it.

4.1 Choosing an Initial Vtree

Vtrees uniquely define SDDs (under some conditions), so the choice of an initial vtree is critical to obtaining a compact SDD. Here, we consider one approach, which we describe below, that was effective in our experiments.

We propose to obtain an initial vtree by inducing one from a variable ordering. We run the min-fill algorithm on a given model, and use the resulting variable ordering to induce a vtree, as follows. First, we construct for each model variable, a balanced vtree over indicator variables, and for each factor, a balanced vtree over parameter variables. We then simulate variable elimination: (1) when we multiply two factors, we compose their vtrees, and (2) when we forget

[3] Publicly available at `http://reasoning.cs.ucla.edu/sdd`.



(a) Initial Balanced Vtrees (b) Vtree for Network $A \rightarrow B$

Fig. 3. Choosing an initial vtree for network $A \rightarrow B$

a variable from a factor, we compose the vtrees of the variable and the factor. Here, composing two vtrees v_i and v_j means that we create a new vtree v with children v_i and v_j. In our experiments, we let the left subtree be the one with fewer variables. Figure 3(a) shows the initial vtrees over indicator and parameter variables for a Bayesian network $A \rightarrow B$. Figure 3(b) shows a vtree constructed using variable ordering $\langle B, A \rangle$. First, forget variable B from CPT $\Theta_{B|A}$ (compose vtrees v_2 and v_4), then multiply with CPT Θ_A (compose with vtree v_3), and finally forget variable A (compose with vtree v_1).

4.2 Compiling Factors into SDDs

Here, we consider how to efficiently encode the factors of a given model as an SDD. This encoding is possible as SDDs support an efficient `apply` operator, which given two SDDs α and β and a boolean operator $\circ$, will return a new SDD for $\alpha \circ \beta$, in polytime. When we encode a factor, we encode the indicator clauses for each factor variable, and the parameter clauses for each factor parameter, as in Section 2. This factor CNF can be compiled using `apply`, where we disjoin the corresponding literals of each clause, and then conjoin the resulting clauses. However, we could seek a more direct and efficient approach by relaxing our use of the CNF representation. Consider the factor $\Theta_{B|A}$ of network $A \rightarrow B$, with parameters $\theta_{b|a}, \theta_{\bar{b}|a}, \theta_{b|\bar{a}}$ and $\theta_{\bar{b}|\bar{a}}$. Our factor CNF is equivalent to the following DNF, over indicator variables I_a and I_b and parameter variables $P_{b|a}$:

$$
\begin{aligned}
&(I_a \wedge \neg I_{\bar{a}} \wedge I_b \wedge \neg I_{\bar{b}} \wedge P_{b|a} \wedge \neg P_{\bar{b}|a} \wedge \neg P_{b|\bar{a}} \wedge \neg P_{\bar{b}|\bar{a}}) \\
\vee\ &(I_a \wedge \neg I_{\bar{a}} \wedge \neg I_b \wedge I_{\bar{b}} \wedge \neg P_{b|a} \wedge P_{\bar{b}|a} \wedge \neg P_{b|\bar{a}} \wedge \neg P_{\bar{b}|\bar{a}}) \\
\vee\ &(\neg I_a \wedge I_{\bar{a}} \wedge I_b \wedge \neg I_{\bar{b}} \wedge \neg P_{b|a} \wedge \neg P_{\bar{b}|a} \wedge P_{b|\bar{a}} \wedge \neg P_{\bar{b}|\bar{a}}) \\
\vee\ &(\neg I_a \wedge I_{\bar{a}} \wedge \neg I_b \wedge I_{\bar{b}} \wedge \neg P_{b|a} \wedge \neg P_{\bar{b}|a} \wedge \neg P_{b|\bar{a}} \wedge P_{\bar{b}|\bar{a}}).
\end{aligned}
$$

The constraints implied by indicator and parameter clauses result in a DNF where each term represents a setting of indicator and parameter variables for each factor parameter $\theta_{x|\mathbf{u}}$. In particular, the term for parameter $\theta_{x|\mathbf{u}}$ has positive literals for parameter variable $P_{x|\mathbf{u}}$ and for the indicator variables consistent with instantiation $x\mathbf{u}$; all other literals are negative.

Using `apply`, it is also easy to compile a DNF into an SDD. However, there are as many terms in the DNF as there are parameters in a factor, and each term has a sub-term with the same number of literals. Naively, this entails a quadratic number of `apply` operations, just to construct the sub-terms over parameter variables, which is undesirable when we have large factors.

Instead, we observe that each sub-term over parameter variables, has all but one variable appearing negatively. We thus construct an SDD α for the sub-term composed of all negative literals, using `apply`. We can use, and re-use, this SDD to construct all parameter sub-terms, using two additional operations each. In particular, for each parameter $\theta_{x|\mathbf{u}}$ we compute $(\alpha \mid \neg P_{x|\mathbf{u}}) \wedge P_{x|\mathbf{u}}$, where $(\alpha \mid \ell)$ denotes conditioning α on a literal ℓ (which is another operation supported by SDDs). This conditioning is equivalent to replacing $\neg P_{x|\mathbf{u}}$ with the constant true, which drops the literal from the term α. The conjoin then replaces the literal with the positive one. To construct all sub-terms over parameter variables, we just need in total a linear number of `apply` operations and a linear number of conditioning operations, which is much more efficient than a quadratic number of `apply`'s. The same technique can be used to construct terms over indicator variables, which is similarly effective when a factor contains variables with many states. We can then conjoin the indicator sub-term with the parameter sub-term.

Encoding Determinism. If a factor contains a zero parameter $\theta_{x|\mathbf{u}}$, then any model w satisfying that term, in the factor DNF, will evaluate to zero, since $W(P_{x|\mathbf{u}}) = 0$. Setting variable $P_{x|\mathbf{u}}$ to false does the same, which effectively removes the term and variable from the DNF. The variable $P_{x|\mathbf{u}}$ is now vacuous, so we remove it from the domain of knowledge base Δ.

Encoding Equal Parameters. If a factor contains parameters $\theta_{x|\mathbf{u}}$ that have the same value p, then it suffices to have a single parameter variable P_p for those parameters. For example, say parameter $\theta_{b|a}$ and $\theta_{\bar{b}|\bar{a}}$ have the same value p in CPT $\Theta_{B|A}$. The corresponding DNF is:

$$\begin{aligned}
&(I_a \wedge \neg I_{\bar{a}} \wedge I_b \wedge \neg I_{\bar{b}} \wedge P_p \wedge \neg P_{\bar{b}|a} \wedge \neg P_{b|\bar{a}}) \\
\vee\ &(I_a \wedge \neg I_{\bar{a}} \wedge \neg I_b \wedge I_{\bar{b}} \wedge \neg P_p \wedge P_{\bar{b}|a} \wedge \neg P_{b|\bar{a}}) \\
\vee\ &(\neg I_a \wedge I_{\bar{a}} \wedge I_b \wedge \neg I_{\bar{b}} \wedge \neg P_p \wedge \neg P_{\bar{b}|a} \wedge P_{b|\bar{a}}) \\
\vee\ &(\neg I_a \wedge I_{\bar{a}} \wedge \neg I_b \wedge I_{\bar{b}} \wedge P_p \wedge \neg P_{\bar{b}|a} \wedge \neg P_{b|\bar{a}}).
\end{aligned}$$

Note that the weight of each term is unchanged. To compile this function using the `apply` operator, it suffices to construct the parameter sub-term for equal parameters once, and just disjoin the corresponding indicator sub-terms. This is more efficient (fewer `apply` operations) than explicitly compiling the DNF.

Note that in the CNF encoding of Section 2, we resorted to encoding a representation that contained too many models, and then filtered them by performing cardinality minimization after compiling to d-DNNF. This is more efficient than encoding equal parameters directly as a CNF, as a straightforward conversion leads to a CNF with many clauses. However, such techniques are not needed using an SDD representation, as we are not constrained to using CNFs/DNFs.

4.3 Bottom-Up Compilation

Once we have obtained SDD representation of our model's factors, we just need to conjoin these SDDs to obtain an SDD representation of our model. However, what order do we use to conjoin factor SDDs together? This decision impacts the sizes of the intermediate representations that we encounter during compilation. In the implementation we evaluate empirically, we mirror the process we used to construct our vtree, using the same min-fill variable ordering. We start with SDD representations of each factor, and simulate variable elimination: (1) when we multiply two factors, we conjoin the corresponding SDDs, and (2) when we forget a variable from a factor, we conjoin the variable's indicator clauses.[4]

4.4 CNF Encodings: Revisited

Using SDDs, it is also possible to perform bottom-up compilation using the CNF encoding [3]. Suppose we are given a probabilistic graphical model as a set of indicator and parameter clauses, as in Section 2, and a vtree over its indicator and parameter variables, as in Section 4.1. We can assign each clause c to the lowest (and unique) vtree node v which contains its variables. This labeled vtree provides a recursive partitioning of the CNF clauses, with each node v in the vtree hosting a set of clauses Δ_v. To compile a CNF, we recursively compile the clauses placed in the sub-vtrees rooted at the children of v, each child returning with its corresponding SDD. We conjoin these two SDDs using `apply`, and then iterate over the clauses at node v, compiling each into an SDD, and conjoining the result with the existing SDD, all also using `apply`. We also visit the clauses hosted by node v according to their size, visiting shorter clauses first.

4.5 Minimizing Cardinality

When exploiting local structure with a CNF as in Section 2, we appealed to cardinality minimization in a compiled d-DNNF. We need to be able to do the same when compiling CNFs to SDDs, which is not as straightforward.

Formally, the minimum cardinality of a given SDD α is defined as:[5]

$$\texttt{mcard}(\alpha) = \begin{cases} 0 & \text{if } \alpha \text{ is a negative literal or true;} \\ 1 & \text{if } \alpha \text{ is a positive literal;} \\ \infty & \text{if } \alpha \text{ is false.} \\ \min_i\{\texttt{mcard}(p_i) + \texttt{mcard}(s_i)\} & \text{if } \alpha = \{(p_1, s_1), ..., (p_n, s_n)\} \end{cases}$$

Algorithm 1 describes how to recursively obtain an SDD $\alpha_{\min}$ representing the minimum cardinality models of a given SDD α, which we call a minimized SDD.

[4] Conjoining indicator clauses is typically redundant, since we can normally assume they are encoded in the SDD of each factor mentioning that variable.

[5] More intuitively, the cardinality of a model w is the number of positive literals that appear in that model. The minimum cardinality of a knowledge base Δ is the minimum cardinality of all its models. Minimizing a knowledge base Δ produces another knowledge base representing the minimum cardinality models of Δ.

Algorithm 1. `Minimize-SDD`

Input: An SDD α, a vtree v for which α is normalized
Output: A minimized SDD $\alpha_{\min}$, normalized for v, for SDD α

```
1  if α ∈ {⊥, X, ¬X} and v is leaf with variable X then return α;
2  else if α = ⊤ and v is leaf with variable X then return ¬X;
3  else
4      if cache(α) ≠ nil then return cache(α);
5      α_min ← empty decision node;
6      foreach element (p, s) in α do
7          if mcard(p) + mcard(s) > mcard(α) then add (p, ⊥) to α_min;
8          else
9              p_min ← Minimize-SDD(p, v^l), s_min ← Minimize-SDD(s, v^r);
10             add (p_min, s_min) to α_min;
11             p_carry ← apply(p, ¬p_min, ∧);
12             if p_carry ≠ ⊥ then add (p_carry, ⊥) to α_min;
13     add α_min to cache;
14     return α_min;
```

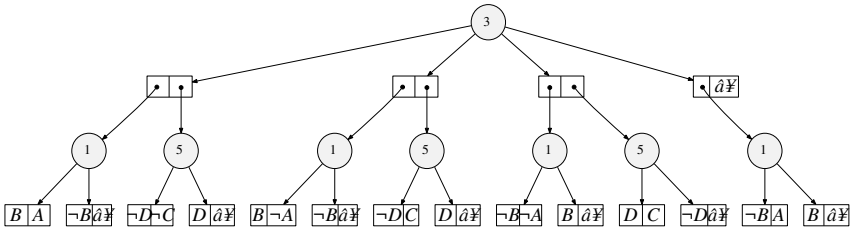


Fig. 4. Minimized SDD for $(A \wedge B) \vee (B \wedge C) \vee (C \wedge D)$

For each element $(p, s) \in \alpha$, if $\texttt{mcard}(p) + \texttt{mcard}(s) > \texttt{mcard}(\alpha)$, then $(p, \bot)$ is an element of $\alpha_{\min}$. If $\texttt{mcard}(p) + \texttt{mcard}(s) = \texttt{mcard}(\alpha)$, then $(p_{\min}, s_{\min})$ is an element of $\alpha_{\min}$, where $p_{\min}$ and $s_{\min}$ are the minimized SDD's for p and s respectively. If a prime is minimized, then to ensure the exhaustiveness of the primes, we need to add a new element to the minimized SDD (Line 12).

Figure 4 shows the minimized SDD $\alpha_{\min}$ for SDD α in Figure 2. Each element of α has a minimum cardinality of 2, so the minimum cardinality of α is 2. For each element (p, s) of α, the minimization $\alpha_{\min}$ has a minimized element $(p_{\min}, s_{\min})$. The minimization $\{(\neg B,\neg A),(B,\bot)\}$ of prime $\neg B$ is not equal to itself, so $\alpha_{\min}$ has an element with prime $\{(\neg B,A),(B,\bot)\}$ and sub $\bot$.

5 Experiments

We evaluate our approach to compiling probabilistic graphical models (PGMs) into SDDs, where we consider the impact that different encodings can have on

the succinctness of the resulting compilation, and also on the time that it takes to compile it. We compare 4 methods to compile a probabilistic graphical model:[6]

- compiling to SDD without exploiting local structure (denoted by `none`);
- compiling to SDD, exploiting local structure as in Section 4.2 (`sdd`);
- encoding the PGM as a CNF, and compiling to SDD (`cnf`);
- encoding the PGM as a CNF, and compiling to d-DNNF with C2D (`c2d`).

Note that method `c2d` is the one that underlies the ACE system. All methods here are driven by initial structures (vtrees for SDDs and dtrees for d-DNNFs), based on min-fill variable orderings. We restrict ourselves to static initial structures, although SDDs support the ability to dynamically optimize the size of an SDD [3]; top-down approaches to compilation, like method `c2d`, typically do not.

Table 1 highlights statistics for a selection of benchmarks and their SDD and d-DNNF compilations. Here, the size of an SDD is the aggregate size of an SDD's decompositions, and the number of nodes is the number of decision nodes. For methods that compile to SDDs, we observe that encoding local structure (`sdd`, `cnf`) can obtain much more compact SDDs than without (`none`). For example, in network `water`, method `sdd` produced an SDD that was 73× more compact than `none`. Such improvements are typical for knowledge compilation approaches to exact inference, when there is sufficient local structure [2]. Next, methods `cnf` and `sdd` encode the same local structure, so both approaches yield the same compiled SDDs. However, by not constraining ourselves to CNFs, as method `cnf` does, we can obtain these SDDs in much less time, often by orders-of-magnitude.

As for d-DNNFs compiled by C2D, we report the number of NNF edges as the compilation size, and the number of AND-nodes and OR-nodes in an NNF as the number of nodes. While the sizes for SDDs and d-DNNFs are not directly comparable, we note that for instances where we obtained both an SDD and a d-DNNF, the reported sizes are within an order-of-magnitude of each other (or better). This suggests that methods `sdd` and `c2d` are performing comparably, relatively speaking, across these benchmarks. In other cases, method `sdd` could compile benchmarks that method `c2d` was unable to in one hour. For example, method `sdd` was able to compile network `diabetes` in under 25 seconds (at least 144× faster), and method `sdd` was the only one to compile network `munin1`.

Finally, we note that d-DNNFs are in general more succinct than SDDs, and for any given SDD there is a corresponding d-DNNF that is at least as succinct. However, SDDs enjoy an efficient `apply` operator, which is critical for certain applications that are out of the scope of d-DNNFs, which does not support an `apply`. Our results here suggest that our simplified approach (method `sdd`) can be orders-of-magnitude more efficient than other alternatives. In some cases, in fact, the ability to encode directly to an SDD alone (`none`) appears to outweigh the ability to exploit local structure using CNFs (given enough memory).

[6] Our experiments were performed on an Intel i7-3770 3.4GHz CPU with 16GB RAM, except for method `none`, which were on an Intel Xeon E5440 2.83GHz CPU with 32GB RAM (SDD size is the relevant comparison here, and less compilation time).

Table 1. Under PGM stats, X is # of variables, θ is # of model parameters, 0 is # of zeros, p is # of distinct parameter values, C is $\log_2$ size of largest jointree cluster. We have 4 compilation methods: `none` is compilation to SDDs without encoding local structure, `sdd` is to SDDs with encoding local structure, `cnf` is to SDDs via encoding CNFs with local structure, `c2d` is to d-DNNFs via the same CNF using the C2D compiler. Time reported in seconds. — is a 1 hour timeout, ⋆ is out-of-memory.

	PGM stats					compilation stats			
benchmark	X	θ	0	p	C	method	size	nodes	time
barley	48	130180	0	36926	23.4	none	231,784,907	96,062,825	227.46
						sdd	49,442,901	17,678,076	32.48
						cnf	—	—	—
						c2d	—	—	—
diabetes	413	461069	352224	17574	17.5	none	78,641,365	32,312,892	308.74
						sdd	21,704,366	7,882,652	24.49
						cnf	—	—	—
						c2d	—	—	—
diagnose-b	329	34704	51	976	18.1	none	15,622,318	7,115,750	67.23
						sdd	227,170	102,856	0.84
						cnf	227,170	102,856	245.12
						c2d	369,426	124,393	77.56
mildew	35	547158	509234	6713	19.6	none	54,726,909	26,136,443	188.51
						sdd	2,981,951	1,156,072	5.55
						cnf	—	—	—
						c2d	167,676,317	3,120,074	1430.90
munin1	189	19466	10910	4246	26.2	none	⋆	⋆	⋆
						sdd	139,855,161	61,376,880	339.34
						cnf	—	—	—
						c2d	—	—	—
munin2	1003	83920	46606	22852	17.4	none	25,068,547	10,453,726	122.08
						sdd	8,007,175	3,430,400	19.12
						cnf	8,007,175	3,430,400	2377.67
						c2d	—	—	—
munin3	1044	85855	47581	24102	17.3	none	43,069,070	19,066,130	158.25
						sdd	9,623,616	4,431,843	21.73
						cnf	—	—	—
						c2d	66,048,268	2,297,199	132.96
water	32	13484	6970	3578	20.8	none	29,881,265	12,566,205	36.17
						sdd	405,538	195,502	3.83
						cnf	405,538	195,502	106.07
						c2d	1,342,307	141,167	3.24
mm-5-8-3	1616	11278	5531	3367	28.0	none	⋆	⋆	⋆
						sdd	870,867	400,872	255.38
						cnf	870,867	400,872	1830.93
						c2d	4,920,481	214,281	8.78
gr-90-26-1	676	5202	2360	1704	40.0	none	⋆	⋆	⋆
						sdd	600,816	288,632	190.01
						cnf	600,816	288,632	62.48
						c2d	216,935	28,241	3.38

6 Conclusion

In this paper, we outlined a knowledge compilation approach for exact inference in probabilistic graphical models, that is enabled by a recently proposed representation, the Sentential Decision Diagram (SDD). SDDs support an efficient `apply` operation, which was not available in previous approaches based on compilation to d-DNNFs. As we illustrated, an efficient `apply` operation enables a more unified approach to knowledge compilation, that allows us to encode a model, exploit its local structure, and compile it to a more compact representation, in common and simplified terms. Empirically, we found that by bypassing the auxiliary CNF representations that were previously used, we can obtain SDDs that are of comparable succinctness to d-DNNFs found by C2D, but more efficiently, by orders-of-magnitude in some cases. In the process, we further proposed a new algorithm for minimizing cardinality in SDDs.

Acknowledgments. This work has been partially supported by ONR grant #N00014-12-1-0423, NSF grant #IIS-1118122, and NSF grant #IIS-0916161.

References

1. Bryant, R.E.: Graph-based algorithms for Boolean function manipulation. IEEE Transactions on Computers C-35, 677–691 (1986)
2. Chavira, M., Darwiche, A.: On probabilistic inference by weighted model counting. Artificial Intelligence Journal 172(6-7), 772–799 (2008)
3. Choi, A., Darwiche, A.: Dynamic minimization of sentential decision diagrams. In: Proceedings of the 27th Conference on Artificial Intelligence (AAAI) (2013)
4. Darwiche, A.: A differential approach to inference in Bayesian networks. Journal of the ACM 50(3), 280–305 (2003)
5. Darwiche, A.: Modeling and Reasoning with Bayesian Networks. Cambridge University Press (2009)
6. Darwiche, A.: SDD: A new canonical representation of propositional knowledge bases. In: IJCAI, pp. 819–826 (2011)
7. Darwiche, A., Dechter, R., Choi, A., Gogate, V., Otten, L.: Results from the probablistic inference evaluation of UAI 2008 (2008), `http://graphmod.ics.uci.edu/uai08/Evaluation/Report`
8. Darwiche, A., Marquis, P.: A knowledge compilation map. Journal of Artificial Intelligence Research 17, 229–264 (2002)
9. Ferrara, A., Pan, G., Vardi, M.Y.: Treewidth in verification: Local vs. global. In: Sutcliffe, G., Voronkov, A. (eds.) LPAR 2005. LNCS (LNAI), vol. 3835, pp. 489–503. Springer, Heidelberg (2005)
10. Fierens, D., Van den Broeck, G., Thon, I., Gutmann, B., Raedt, L.D.: Inference in probabilistic logic programs using weighted CNF's. In: UAI, pp. 211–220 (2011)
11. Huang, J., Darwiche, A.: Using DPLL for efficient OBDD construction. In: Hoos, H.H., Mitchell, D.G. (eds.) SAT 2004. LNCS, vol. 3542, pp. 157–172. Springer, Heidelberg (2005)
12. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press (2009)
13. Prasad, M.R., Chong, P., Keutzer, K.: Why is ATPG easy? In: DAC, pp. 22–28 (1999)

Independence in Possibility Theory under Different Triangular Norms

Giulianella Coletti[1], Davide Petturiti[1], and Barbara Vantaggi[2]

[1] Dip. Matematica e Informatica, Università di Perugia, Italy
{coletti,davide.petturiti}@dmi.unipg.it
[2] Dip. S.B.A.I., Università di Roma "La Sapienza", Italy
barbara.vantaggi@sbai.uniroma1.it

Abstract. In this paper we consider coherent T-conditional possibility assessments, with T a continuous t-norm, and introduce for them a concept of independence already studied for the minimum and strict t-norms. As a significant particular case of T-conditional possibility we explicitly consider T_{DP}-conditional possibility (obtained through the minimum specificity principle) introduced by Dubois and Prade.

Keywords: Independence, T-conditional possibility, Coherence.

1 Introduction

The notion of conditioning is a problem of long-standing interest and it involves different uncertainty measures. In this paper we focus on (finitely maxitive) possibility measures that can be seen as specific upper probabilities, arising from a convex set of probabilities or as a result of a probabilistic inferential process [8,18,22]. Various definitions of conditional possibility have been introduced: by analogy with the Kolmogorovian probabilistic framework or by using some criterion as the minimum specificity principle (see, e.g., [12,17]). All these definitions have in common the fact that a conditional measure is obtained as a derived concept from an "unconditional" one. In [3] a general notion of T-conditional possibility has been introduced as a primitive concept: the conditional possibility is directly defined as a function on a set of conditional events which satisfies a suitable set of axioms and it is not induced just by a single unconditional possibility (as solution of an equation involving joint and marginal possibilities).

Referring to the aforementioned definition, we provide a comparison with the conditioning notion obtained through the minimum specificity principle, called here T_{DP}-conditioning, introduced in [17].

However, all these definitions are not able to deal with partial assessments, so a notion of coherence for possibility, which assures that a partial assessment is the restriction of a T-conditional possibility, has been introduced in [10].

In this paper we present a notion of independence for coherent T-conditional (and T_{DP}-conditional) possibility, which is an extension to a general continuous t-norm of that given for the minimum and strict t-norms (see [9,19]). One of the

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 133–144, 2013.

main motivations for introducing this definition of independence is to capture the following natural implication: the independence under an uncertainty measure (and so, in particular, under a possibility) must imply logical independence. In other words if an event is "logically" related to another one, the two events cannot be independent under any uncertainty measure. This implication, even though very intuitive, can fail when we adopt the classical definitions of independence. Moreover, taking into account logical constraints is interesting not only from a theoretical point of view, but also for practical situations.

A brief comparison with the classical notions of possibilistic independence [11,13,17,20,21] is shown and the main properties are studied. Among the properties we emphasize that our (conditional) independence structures do not necessarily satisfy symmetry. The lack of symmetry is not surprising in possibility theory: actually, even some of the above definitions do not satisfy this property [1]. Our definition can be reinforced in order to satisfy symmetry in a way to get symmetric independence models, that can capture associations among variables and possibly can be represented through graphs. In fact, just few separation criteria are present to represent asymmetric independence structures [15,25].

2 Conditioning in Possibility Theory

An *event* E is singled out by a Boolean proposition, that is a statement that can be either true or false. Since in general it is not known whether E is true or not, we are uncertain on E, which is said to be *possible*. Two particular events are the *certain event* Ω and the *impossible event* $\emptyset$, that coincide with, respectively, the top and the bottom of every Boolean algebra $\mathcal{B}$ of events, i.e., a set of events closed w.r.t. the familiar Boolean operations of *contrary* ^{c}, *conjunction* $\wedge$ and *disjunction* $\vee$ and equipped with the partial order $\subseteq$. A *conditional event* $E|H$ is an ordered pair (E, H), with $H \neq \emptyset$, where E and H are events of the same "nature", but with a different role (in fact H acts as a "possible hypothesis"). In particular any event E can be seen as the conditional event $E|\Omega$.

In what follows, $\mathcal{B} \times \mathcal{H}$ denotes a set of conditional events with $\mathcal{B}$ a Boolean algebra and $\mathcal{H}$ an additive set (i.e., closed with respect to finite disjunctions) such that $\mathcal{H} \subseteq \mathcal{B}^0 = \mathcal{B} \setminus \{\emptyset\}$. Moreover, given a finite set $\mathcal{G} = \{E_i|H_i\}_{i=1,\ldots,n}$, let $\mathcal{B} = \langle \{E_i, H_i\}_{i=1,\ldots,n} \rangle$ be the Boolean algebra spanned by the events $\{E_i, H_i\}_{i=1,\ldots,n}$ and if $\mathcal{B}$ is a finite Boolean algebra denote with $\mathcal{C}_\mathcal{B}$ the relevant set of atoms.

Definition 1. *Let T be any t-norm. A function $\Pi : \mathcal{B} \times \mathcal{H} \to [0,1]$ is a T-conditional possibility if it satisfies the following properties:*

(CP1) $\Pi(E|H) = \Pi(E \wedge H|H)$*, for every* $E \in \mathcal{B}$ *and* $H \in \mathcal{H}$*;*
(CP2) $\Pi(\cdot|H)$ *is a finitely maxitive possibility on* $\mathcal{B}$*, for any* $H \in \mathcal{H}$*;*
(CP3) $\Pi(E \wedge F|H) = T(\Pi(E|H), \Pi(F|E \wedge H))$*, for any* $H, E \wedge H \in \mathcal{H}$ *and* $E, F \in \mathcal{B}$*.*

Let us stress that condition *(CP2)* requires that, for every $H \in \mathcal{H}$, $\Pi(\cdot|H)$ is a normalized finitely maxitive function [24] defined on $\mathcal{B}$, i.e., it holds

- $\Pi(\emptyset|H) = 0$ and $\Pi(\Omega|H) = 1$;
- $\Pi(\bigvee_{i=1,\ldots,n} A_i|H) = \max_{i=1,\ldots,n} \Pi(A_i|H)$, for every $\{A_1, \ldots, A_n\} \subseteq \mathcal{B}$.

In this paper we consider finitely maxitive possibility measures (even when not explicitly stated), so we do not require that $\Pi(\cdot|H)$ must be supremum preserving, i.e., $\Pi(\bigvee_{i\in I} A_i|H) = \sup_{i\in I} \Pi(A_i|H)$ with $\{A_i\}_{i\in I} \subseteq \mathcal{B}$, I arbitrary. Moreover, any unconditional possibility measure $\Pi(\cdot)$ on $\mathcal{B}$ can be viewed as a T-conditional possibility $\Pi(\cdot|\Omega)$ on $\mathcal{B} \times \{\Omega\}$, where T is an arbitrary t-norm.

Mimiking the concept of full conditional probability on $\mathcal{B}$ defined in [16], given any t-norm T we say that Π is a *full T-conditional possibility on $\mathcal{B}$* if it is defined on $\mathcal{B}\times\mathcal{B}^0$. In the following T is assumed to be continuous (even when not explicitly stated), since for non-continuous t-norms a T-conditional possibility on $\mathcal{B} \times \mathcal{H}$ could be non-extendable as a full T-conditional possibility on $\mathcal{B}$ (see Example 1 in [10]), while it is always possible if T is continuous.

A full T-conditional possibility $\Pi(\cdot|\cdot)$ on $\mathcal{B}$ is not necessarily "represented" by means of a single unconditional possibility, even when $\mathcal{B}$ is finite (see [10]), but in this latter case there is a *unique class* of possibility measures $\mathcal{P} = \{\Pi_0, \ldots, \Pi_k\}$ all defined on $\mathcal{B}$, called *T-nested class agreeing with $\Pi(\cdot|\cdot)$*, whose distributions on the set of atoms $\mathcal{C}_\mathcal{B}$ satisfy the following properties for $\alpha = 1, \ldots, k$:

1. $\Pi_{\alpha-1}(C) \leq \Pi_\alpha(C)$ if $C \in \mathcal{C}_\alpha$;
2. $\Pi_\alpha(C) = 0$ for all the atoms $C \in \mathcal{C}_0 \setminus \mathcal{C}_\alpha$;
3. for any $C \in \mathcal{C}_0$ there exists a (unique) $j_C \in \{0, \ldots, k\}$ such that $\Pi_{j_C}(C) = 1$;
4. for any $C_1, C_2 \in \mathcal{C}_\alpha$, $\Pi_{\alpha-1}(C_1) < \Pi_{\alpha-1}(C_2) \Rightarrow \Pi_\alpha(C_1) < \Pi_\alpha(C_2)$;
5. for any $C \in \mathcal{C}_\alpha$, $\Pi_{\alpha-1}(C) = T(\Pi_\alpha(C), \Pi_{\alpha-1}(H_0^\alpha))$;

where $\mathcal{C}_0 = \mathcal{C}_\mathcal{B}$, $\mathcal{C}_\alpha = \{C \in \mathcal{C}_{\alpha-1} : \Pi_{\alpha-1}(C) < 1\}$, for $\alpha \geq 1$, and $H_0^\alpha = \bigvee_{C\in\mathcal{C}_\alpha} C$.

Previous structure implies that for every $E|H \in \mathcal{B}\times\mathcal{B}^0$, the value $\Pi(E|H)$ is a (non-necessarily unique) solution of equations in x

$$\Pi_\alpha(E \wedge H) = T(x, \Pi_\alpha(H)),$$

for $\alpha = 0, \ldots, j_H$, where $j_H \in \{0, \ldots, k\}$ is the minimum index such $\Pi_{j_H}(H) = 1$.

This highlights an important difference with other approaches to conditioning, where the conditional possibility $\Pi(E|H)$ is *defined*, starting from a single unconditional possibility $\Pi(\cdot)$, as a solution of the equation in x

$$\Pi(E \wedge H) = T(x, \Pi(H)). \tag{1}$$

Indeed, continuity of T assures only the solvability of (1), while to reach the uniqueness of the solution a further constraint must be imposed. At this aim, the Dubois and Prade's *minimum specificity principle* [17] consists in always selecting the greatest solution of equation (1), by means of the residuum $\rightarrow_T$ of a continuous t-norm, defined as

$$x \rightarrow_T y = \sup\{z \in [0, 1] : T(x, z) \leq y\}.$$

In [2] the issue of the existence and uniqueness of the solution computed using the possibilistic counterparts of Jeffrey's rule is addressed: under product t-norm, possibilistic version of Jeffrey's rule admits a unique solution, while under

minimum, it does not guarantee the existence of a solution (satisfying the two conditions underlying Jeffrey's rule of conditioning).

By referring to the conditioning notion based on the minimum specificity principle (see, e.g., [17]) we deal with T_{DP}-conditional possibility.

Definition 2. *Let T be a continuous t-norm. A function $\Pi : \mathcal{B} \times \mathcal{H} \to [0,1]$ is a T_{DP}-conditional possibility if it satisfies the following conditions:*

(DP1) $\Pi(E|H) = \Pi(E \wedge H|H)$ for every $E \in \mathcal{B}$ and $H \in \mathcal{H}$;
(DP2) $\Pi(\cdot|H)$ is a finitely maxitive possibility for every $H \in \mathcal{H}$;
(DP3) for every $E|H \in \mathcal{B} \times \mathcal{H}$ it holds (with $H_0^0 = \bigvee_{H \in \mathcal{H}} H \in \mathcal{H}$)

$$\Pi(E|H) = \begin{cases} \Pi(H|H_0^0) \to_T \Pi(E \wedge H|H_0^0) & \text{if } E \wedge H \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases}$$

Next proposition shows that T_{DP}-conditional possibilities are particular T-conditional possibilities.

Proposition 1. *Let T be a continuous t-norm. If Π on $\mathcal{B} \times \mathcal{H}$ is a T_{DP}-conditional possibility, then Π is a T-conditional possibility.*

Proof. It is sufficient to show that, for every $E|H \in \mathcal{B} \times \mathcal{H}$ it holds

$$\Pi(E \wedge F|H) = T(\Pi(E|H), \Pi(F|E \wedge H)). \tag{2}$$

We recall that the residuum satisfies the following properties of monotonicity: for every $x \leq x'$ and $y \leq y'$, $x \to_T y \leq x \to_T y'$ and $x \to_T y \geq x' \to_T y$. Moreover, $x \to_T y = 1$ if and only if $x \leq y$.

By Definition 2 it follows:

- $\Pi(E|H) = (\Pi(H|H_0^0) \to_T \Pi(E \wedge H|H_0^0))$,
- $\Pi(F|E \wedge H) = (\Pi(E \wedge H|H_0^0) \to_T \Pi(E \wedge F \wedge H|H_0^0))$,
- $\Pi(E \wedge F|H) = (\Pi(H|H_0^0) \to_T \Pi(E \wedge F \wedge H|H_0^0))$.

Therefore $\Pi(E \wedge F|H) \leq \Pi(E|H)$ and $\Pi(E \wedge F|H) \leq \Pi(F|E \wedge H)$.

If $\Pi(E \wedge F|H) = 1$ then equation (2) trivially holds.

If $\Pi(E \wedge F|H) < 1$, and $\Pi(E|H) = 1$ (or analogously $\Pi(F|E \wedge H) = 1$), then

- $\Pi(E \wedge H|H_0^0) = \Pi(H|H_0^0)$,
- $\Pi(E \wedge F \wedge H|H_0^0) = T(\Pi(E \wedge F|H), \Pi(H|H_0^0))$,
- $\Pi(E \wedge F \wedge H|H_0^0) = T(\Pi(F|E \wedge H), \Pi(E \wedge H|H_0^0)) = T(\Pi(F|E \wedge H), \Pi(H|H_0^0))$,

so $\Pi(F|E \wedge H)$ is the unique value satisfying the above equation and it must hold $\Pi(F|E \wedge H) = \Pi(F \wedge E|H)$, thus equation (2) is satisfied.

If all the three values are strictly less than 1, then

$$\begin{aligned} \Pi(F \wedge E \wedge H|H_0^0) &= T(\Pi(F|E \wedge H), T(\Pi(E|H), \Pi(H|H_0^0))) \\ &= T(T(\Pi(F|E \wedge H), \Pi(E|H)), \Pi(H|H_0^0)) \end{aligned}$$

and

$$\Pi(F \wedge E \wedge H|H_0^0) = T(\Pi(F \wedge E|H), \Pi(H|H_0^0)).$$

So the validity of equation (2) follows by the continuity of T.

Let us stress that not all the definitions of conditioning (defined over a Boolean algebra) present in the literature are particular cases of T-conditional possibilities: this is the case of Zadeh's conditioning rule (see [9]). Actually, T-conditional possibilities according to Definition 1 are consistent with the definition given in [12] which solves the problem of conditioning by following the classic Kolmogorovian line and by defining the concept of (Π, T)-almost everywhere equality. Nevertheless, the class of conditional measures consistent with the definition given in [12] contains also functions not consistent with our Definition 1, in the sense that, for some conditioning event H we can have $\Pi(H|H) \neq 1$.

Now we study T_{DP}-conditional possibility in the case the Boolean algebra $\mathcal{B}$ (and so the the additive set $\mathcal{H} \subseteq \mathcal{B}$) is finite. In the next Proposition 2 we show that every T_{DP}-conditional possibility on $\mathcal{B} \times \mathcal{H}$ can be extended (non-necessarily in a unique way) to a full T_{DP}-conditional possibility on $\mathcal{B}$ (i.e., a T_{DP}-conditional possibility on $\mathcal{B} \times \mathcal{B}^0$).

Proposition 2. *Let T be a continuous t-norm, and $\mathcal{B}$ a finite Boolean algebra. If $\Pi : \mathcal{B} \times \mathcal{H} \to [0,1]$ is a T_{DP}-conditional possibility, then there exists a full T_{DP}-conditional possibility $\Pi' : \mathcal{B} \times \mathcal{B}^0 \to [0,1]$ such that $\Pi'_{|\mathcal{B}\times\mathcal{H}} = \Pi$.*

Proof. The proof for $T = \min$ has been given in [4,23]. Here assume T is a continuous t-norm T. Denote $H_0^0 = \bigvee_{H\in\mathcal{H}} H$. If $H_0^0 = \Omega$, then the extension $\Pi'(\cdot|\cdot)$ is obtained through *(DP3)* by setting for any $E|H \in \mathcal{B}\times(\mathcal{B}^0\backslash\mathcal{H})$, $\Pi'(E|H)$ equal to 0 if $E \wedge H = \emptyset$, and $\Pi(H|H_0^0) \to_T \Pi(E \wedge H|H_0^0)$ otherwise.

When $H_0^0 \neq \Omega$, in order to define a possibility $\Pi'(\cdot|\Omega)$ on $\mathcal{B}$, put for each atom $C_r \in \mathcal{C}_\mathcal{B}$,

$$\Pi'(C_r|\Omega) = \begin{cases} 1 & \text{if } C_r \wedge H_0^0 = \emptyset, \\ \Pi(C_r|H_0^0) & \text{otherwise,} \end{cases}$$

that is trivially seen to be a possibility distribution and so it induces a possibility on $\mathcal{B}$. The possibility $\Pi'(\cdot|\Omega)$ defines through axiom *(DP3)* a full T-conditional possibility $\Pi'(\cdot|\cdot)$ on $\mathcal{B}$ extending $\Pi(\cdot|\cdot)$, since for any $E|H \in \mathcal{B}\times\mathcal{H}$ all equalities and strict inequalities of $\Pi(\cdot|H_0^0)$ are preserved by $\Pi'(\cdot|\Omega)$.

Since a full T_{DP}-conditional possibility is a particular full T-conditional possibility, it can be "represented" by means of a unique T-nested class $\mathcal{P} = \{\Pi_0, \ldots, \Pi_k\}$ agreeing with it [10].

Remark 1. By referring to the T-nested class of a full T_{DP}-conditional possibility, note that given $\Pi_0(\cdot) = \Pi(\cdot|\Omega)$, if Π_0 takes k distinct values $1 > \pi_1 > \pi_2 > \ldots > \pi_k \geq 0$, then for $\alpha = 1, \ldots, k$, the distribution of Π_α is obtained assigning $\Pi_\alpha(C_r) = 0$ to all those atoms $C_r \notin \mathcal{C}_\alpha$ and $\Pi_\alpha(C_r) = \pi_\alpha \to_T \Pi_{\alpha-1}(C_r)$ to all the atoms $C_r \in \mathcal{C}_\alpha$. Therefore the fact that T_{DP}-conditional possibilities are particular elements of the class of T-conditional possibilities can be captured directly through the specificity of the structure of their corresponding T-nested classes.

In the next proposition we show that every full T_{DP}-conditional possibility on $\mathcal{B}$ can be extended as a full T_{DP}-conditional possibility on every finite Boolean superalgebra $\mathcal{B}' \supseteq \mathcal{B}$.

Proposition 3. *Let T be a continuous t-norm, $\mathcal{B}$ a finite Boolean algebra and $\mathcal{B}' \supseteq \mathcal{B}$ a finite Boolean superalgebra. If $\Pi : \mathcal{B} \times \mathcal{B}^0 \to [0,1]$ is a full T_{DP}-conditional possibility, then there exists a full T_{DP}-conditional possibility $\Pi' : \mathcal{B}' \times \mathcal{B}'^0 \to [0,1]$ such that $\Pi'_{|\mathcal{B}\times\mathcal{B}^0} = \Pi$.*

Proof. Any $C_r \in \mathcal{C}_{\mathcal{B}}$ belongs to $\mathcal{B}'$, moreover, for any $C'_s \in \mathcal{C}_{\mathcal{B}'}$ there exists a unique $C_r \in \mathcal{C}_{\mathcal{B}}$ such that $C'_s \subseteq C_r$. For any $C'_s \in \mathcal{C}_{\mathcal{B}'}$ define $\Pi'(C'_s|\Omega) = \Pi(C_r|\Omega)$. $\Pi'(\cdot|\Omega)$ determines through maxitivity a possibility on $\mathcal{B}'$ which extends $\Pi(\cdot|\Omega)$ and generates a full T_{DP}-conditional possibility on $\mathcal{B}'$ (through *(DP3)*) extending $\Pi(\cdot|\cdot)$.

All definitions of conditioning given so far deeply rely on a specific algebraic structure of the domain of the function Π, thus in order to remove any restriction on the domain we go back to the concept of *coherence*, originally introduced by de Finetti for (finitely additive) probabilities [14].

Definition 3. *Let T be a continuous t-norm. Given a set $\mathcal{G} = \{E_i|H_i\}_{i=1,\ldots,n}$ of conditional events, an assessment $\Pi : \mathcal{G} \to [0,1]$ is a coherent T-conditional [T_{DP}-conditional] possibility if and only if there exits a full T-conditional [T_{DP}-conditional] possibility Π' on $\mathcal{B} = \langle \{E_i, H_i\}_{i=1,\ldots,n} \rangle$, extending Π.*

Proposition 1 implies that any coherent T_{DP}-conditional possibility is a coherent T-conditional possibility. Obviously the converse is not true, as the following example shows:

Example 1. Let us consider $\mathcal{G} = \{H, E|H\}$ with $E \wedge H \neq \emptyset$ and the relevant assessment $\Pi(H) = 0$, $\Pi(E|H) = \gamma$. It is easy to see that, for any continuous t-norm T the function Π is a coherent T-conditional possibility for every $\gamma \in [0,1]$, but is a T_{DP}-conditional possibility only for $\gamma = 1$.

In [10] the coherence of a T-conditional possibility assessment Π on a finite $\mathcal{G}$ has been characterized also in terms of a proper sequence of compatible systems $\mathcal{S}_0^{\Pi}, \ldots, \mathcal{S}_k^{\Pi}$, whose solutions are the possibility distributions related to a T-nested class of possibilities $\mathcal{P} = \{\Pi_0, \ldots, \Pi_k\}$.

We prove a characterization theorem for coherent T_{DP}-conditional possibility.

Theorem 1. *Let T be a continuous t-norm and $\mathcal{G} = \{E_i|H_i\}_{i=1,\ldots,n}$. For a function $\Pi : \mathcal{G} \to [0,1]$, the following statements are equivalent:*

(a) Π is a coherent T_{DP}-conditional possibility on $\mathcal{G}$;
(b) for any $E_i|H_i \in \mathcal{G}$ such that $E_i \wedge H_i = \emptyset$, it is $\Pi(E_i|H_i) = 0$, and the following system with unknowns $x_r \geq 0$ for $C_r \in \mathcal{C}_0 = \mathcal{C}_{\langle \{E_i,H_i\}_{i=1,\ldots,n} \rangle}$, is compatible

$$\mathcal{S}_T^{DP} = \begin{cases} \max\limits_{C_r \subseteq H_i} x_r \to_T \max\limits_{C_r \subseteq E_i \wedge H_i} x_r = \Pi(E_i|H_i) & \text{if } E_i \wedge H_i \neq \emptyset \\ \max\limits_{C_r \in \mathcal{C}_0} x_r = 1. \end{cases} \tag{3}$$

Proof. The assessment Π is a coherent T_{DP}-conditional possibility if and only if there exists a full T_{DP}-conditional possibility Π' on $\mathcal{B} = \langle \{E_i, H_i\}_{i=1,\ldots,n} \rangle$ extending it. For the function Π', it must hold $\Pi'(E_i|H_i) = 0$ whenever $E_i \wedge H_i = \emptyset$, moreover the restriction of $\Pi'(\cdot|\Omega)$ to $\mathcal{C}_{\mathcal{B}}$ (which determines the whole Π' through axiom *(DP3)*) must satisfy all the constraints in system $\mathcal{S}_T^{DP}$, and so it is a solution.

3 Possibilistic Independence under Different T-Norms

We extend to coherent T-conditional [T_{DP}-conditional] possibilities a notion of possibilistic independence (introduced in [9,19] for the minimum and strict t-norms) able to avoid pathological situations whenever logical constraints are involved. In what follows, E^* stands either for E or E^c.

We first briefly recall the concept of significant layer for a coherent T-conditional possibility assessment.

Definition 4. *Let Π be a coherent T-conditional [T_{DP}-conditional] possibility on an arbitrary finite set of conditional events $\mathcal{G}$, and $\mathcal{P}$ be a T-nested class agreeing with Π. Then, for every event $E \in \mathcal{B}^0$, the significant layer of E (denoted as $\circ(E)$) related to $\mathcal{P}$ is defined as the minimum index α such that $\Pi_\alpha(E) = 1$. Moreover, define $\circ(\emptyset) = +\infty$. For every $E|H \in \mathcal{B} \times \mathcal{B}^0$ the significant layer $\circ(E|H)$ of $E|H$ related to $\mathcal{P}$, is defined as the (non-negative) number $\circ(E|H) = \circ(E \wedge H) - \circ(H)$.*

Now we are able to introduce a definition of independence.

Definition 5. *Let T be a continuous t-norm, $\mathcal{G}$ a set of conditional events containing $\mathcal{D} = \{A^*|B^*, B^*|A^*\}$. Given a coherent T-conditional [T_{DP}-conditional] possibility Π on $\mathcal{G}$, A is independent of B under Π, in symbol $A \perp\!\!\!\perp B[\Pi]$, if both the following conditions hold:*

(i) $\Pi(A|B) = \Pi(A|B^c)$ and $\Pi(A^c|B) = \Pi(A^c|B^c)$;
(ii) there exists a T-nested class $\mathcal{P}_{\mathcal{D}} = \{\Pi_\alpha\}_{\alpha=0}^t$ agreeing with $\Pi_{|\mathcal{D}}$ such that

$$\circ(A|B) = \circ(A|B^c) \text{ and } \circ(A^c|B) = \circ(A^c|B^c). \tag{4}$$

Remark 2. The notion of independence given in [9] for strict t-norms relies on *zero layers* instead of significant layers, nevertheless it is possible to prove that the relevant characterizations remain the same by using significant layers.

The next theorem shows the connection between the logical independence and possibilistic independence (according to Definition 5), and this holds for any T-nested class.

Theorem 2. *For any continuous t-norm T and for any coherent T-conditional [T_{DP}-conditional] possibility Π on $\mathcal{G} \supseteq \mathcal{D}$, if $A \perp\!\!\!\perp B[\Pi]$, then A and B are logically independent.*

Proof. The proof is direct and is based on the fact that if two events A^* and B^* are incompatible then $\circ(A^*|B^*) = +\infty$.

Since condition *(ii)* depends on the choice of the T-nested class $\mathcal{P}_{\mathcal{D}}$ agreeing with $\Pi_{|\mathcal{D}}$ (which is generally non-unique), for coherent T-conditional possibility, the next theorem proves that the validity of condition *(ii)* is invariant with respect to the choice of the T-nested class.

Theorem 3. *Let A and B be two logically independent events, and let Π be a coherent T_{DP}-conditional possibility (with T any continuous t-norm), defined on $\mathcal{G}$ containing $\mathcal{D} = \{A^*|B^*, B^*|A^*\}$ such that condition (i) of Definition 5 holds. If there exists a T-nested class agreeing with $\Pi_{|\mathcal{D}}$ such that equation (4) is satisfied, then equation (4) holds for any T-nested class agreeing with $\Pi_{|\mathcal{D}}$.*

Proof. A sketch of the proof is done here for lack of space, a detailed version is available in [5].

Consider the atoms generated by A, B, i.e., $C_1 = A \wedge B$, $C_2 = A \wedge B^c$, $C_3 = A^c \wedge B$ and $C_4 = A^c \wedge B^c$. We refer to the characterization of coherence in terms of a sequence of systems $\mathcal{S}_0^{\Pi}, \ldots, \mathcal{S}_k^{\Pi}$ given in [10]. In particular, for $\alpha = 0$, putting $x_r^0 = \Pi_0(C_r)$, $r = 1, \ldots, 4$, it must be

$$\mathcal{S}_0^{\Pi} = \begin{cases} x_1^0 = T(\Pi(A|B), \max\{x_1^0, x_3^0\}) \\ x_1^0 = T(\Pi(B|A), \max\{x_1^0, x_2^0\}) \\ x_2^0 = T(\Pi(A|B^c), \max\{x_2^0, x_4^0\}) \\ x_2^0 = T(\Pi(B^c|A), \max\{x_1^0, x_2^0\}) \\ x_3^0 = T(\Pi(A^c|B), \max\{x_1^0, x_3^0\}) \\ x_3^0 = T(\Pi(B|A^c), \max\{x_3^0, x_4^0\}) \\ x_4^0 = T(\Pi(A^c|B^c), \max\{x_2^0, x_4^0\}) \\ x_4^0 = T(\Pi(B^c|A^c), \max\{x_3^0, x_4^0\}) \\ \max\{x_1^0, x_2^0, x_3^0, x_4^0\} = 1 \\ x_r^0 \geq 0 \qquad\qquad r = 1, \ldots, 4. \end{cases}$$

(C1). If $\Pi(A|B) = \Pi(A^c|B) = 1$, then $\circ(A^*|B^*) = 0$ under any T-nested class.

(C2). If $\Pi(A|B) = 0$ we can have the following subcases.

(C2.1). If $\Pi(B|A^c) = 0$, then $x_1^0 = x_2^0 = x_3^0 = 0$ and $x_4^0 = 1$, thus next system is

$$\mathcal{S}_1^{\Pi} = \begin{cases} x_1^1 = T(0, \max\{x_1^1, x_3^1\}) \\ x_1^1 = T(\Pi(B|A), \max\{x_1^1, x_2^1\}) \\ x_2^1 = T(\Pi(B^c|A), \max\{x_1^1, x_2^1\}) \\ x_3^1 = T(1, \max\{x_1^1, x_3^1\}) \\ \max\{x_1^1, x_2^1, x_3^1\} = 1 \\ x_r^1 \geq 0 \qquad\qquad r = 1, \ldots, 3. \end{cases}$$

(C2.1.1). If $\Pi(B|A) < 1$, then different cases can occur and all T-nested classes are such that (4) holds.

(C2.1.2). If $\Pi(B|A) = 1$, then $x_1^1 = 0$, $x_2^1 = 0$ and $x_3^1 = 1$, and the next system is

$$\mathcal{S}_2^{\Pi} = \begin{cases} x_1^2 = T(1, \max\{x_1^2, x_2^2\}) \\ x_2^2 = T(\Pi(B^c|A), \max\{x_1^2, x_2^2\}) \\ \max\{x_1^2, x_2^2\} = 1 \\ x_r^2 \geq 0 \qquad\qquad\qquad\qquad r = 1, 2, \end{cases}$$

for which the unique solution is $x_1^2 = 1$ and $x_2^2 = \Pi(B^c|A)$. Thus condition (4) never holds, since $\circ(A|B^c) = \circ(C_2) - 0 = \circ(C_2) \geq 2 > \circ(A|B) = 2 - 1 = 1$.

(C2.2) When $\Pi(B|A^c) \in (0, 1)$, the proof follows analogously to the case *(C2.1)*, so (4) holds only if $\Pi(B|A) < 1$.

(C2.3). If $\Pi(B|A^c) = \Pi(B^c|A^c) = 1$, then the unique solution of $\mathcal{S}_0^{\Pi}$ is $x_1^0 = x_2^0 = 0$, $x_3^0 = x_4^0 = 1$, thus condition (4) holds if and only if $\Pi(B|A) = \Pi(B^c|A) = 1$.

(C2.4). The case $\Pi(B^c|A^c) < 1$ is as *(C2.1), (C2.2)*, i.e., (4) holds if and only if $\Pi(B^c|A) < 1$.

(C3). If $\Pi(A|B) = \alpha \in (0, 1)$ we can have the following subcases.

(C3.1). If $\Pi(B|A^c) = 0$, then the system $\mathcal{S}_0^{\Pi}$ has solution if and only if either $\Pi(B|A) = 0$ or $\Pi(B|A) = \beta \in (0, 1)$ with β a zero divisor of α. If $\Pi(B|A) = 0$, condition (4) holds under any T-nested class. Otherwise ($0 < \Pi(B|A) < 1$), a contradiction arises in one of the following systems.

(C3.2). If $0 < \Pi(B|A^c) < \alpha$, then $0 < \Pi(B|A) < \alpha$ and condition (4) holds as $\circ(A|B) = 3 - 2 = 1 = 1 - 0 = \circ(A|B^c)$ and $\circ(A^c|B) = 2 - 2 = 0 = 0 - 0 = \circ(A^c|B^c)$.

(C3.3). If $\Pi(B|A^c) = \alpha \in (0, 1)$, then $\alpha \leq \Pi(B|A)$ and $\alpha \leq \Pi(B^c|A)$. In particular, if $\alpha \leq \Pi(B|A) < 1$, then any T-nested class verifies (4). If $\alpha \leq \Pi(B^c|A) < 1$, then no T-nested class verifies (4). If $\Pi(B|A) = \Pi(B^c|A) = 1$, then no T-nested class verifies (4).

(C3.4). If $\Pi(B|A^c) = \Pi(B^c|A^c) = 1$, then condition (4) holds if and only if $\Pi(B|A) = \Pi(B^c|A) = 1$.

(C3.5). The case $\Pi(B^c|A^c) = 0$ is symmetric to $\Pi(B|A^c) = 0$ *(C3.1)*. While the case $\Pi(B^c|A^c) = \beta \in (0, \alpha)$ [$\Pi(B^c|A^c) = \alpha \in (0, 1)$] is symmetric to $\Pi(B|A^c) = \beta \in (0, \alpha)$ *(C3.2)* [$\Pi(B|A^c) = \alpha \in (0, 1)$ *(C3.3)*].

All the remaining cases are obtained by symmetry from *(C2)* and *(C3)* by exchanging A and A^c.

Notice that, since T_{DP}-conditional possibilities are particular T-conditional possibilities, previous theorem establishes also the invariance of condition *(ii)* of Definition 5 for coherent T_{DP}-conditional possibilities.

Next Theorem 4 characterizes independence of two events in terms of the values of $\Pi(B^*|A^*)$, giving up any direct reference to significant layers, in the case of coherent T-conditional possibility with T any continuous t-norm.

Theorem 4. *Let T be any continuous t-norm, and A and B two logically independent events. If a coherent T-conditional possibility is such that $\Pi(A|B) = \Pi(A|B^c)$ and $\Pi(A^c|B) = \Pi(A^c|B^c)$, then $A \perp\!\!\!\perp B[\Pi]$ if and only if one (and only one) of the following conditions holds:*

(a) $\Pi(A|B) = \Pi(A^c|B) = 1$*;*

(b) $\min\{\Pi(A|B), \Pi(A^c|B)\} = 0$ *and the extension of* Π *on* $\{B^*|A^*\}$ *must satisfy one of the following conditions whenever the values are coherent*

- $\Pi(B|A) < 1$, $\Pi(B|A^c) < 1$*;*
- $\Pi(B^c|A) < 1$, $\Pi(B^c|A^c) < 1$*;*
- $\Pi(B^*|A^*) = 1$*;*

(c) $\Pi(A|B) = \alpha \in (0,1)$ *and the extension of* Π *on* $\{B^*|A^*\}$ *must satisfy one of the following conditions whenever the values are coherent*

- $\Pi(B|A^*) = 0$ *or* $\Pi(B^c|A^*) = 0$*;*
- $0 < \Pi(B|A^*) < \alpha$ *or* $0 < \Pi(B^c|A^*) < \alpha$*;*
- $\Pi(B|A^c) = \alpha$ *and* $\alpha \leq \Pi(B|A) < 1$ *or* $\Pi(B^c|A^c) = \alpha$ *and* $\alpha \leq \Pi(B^c|A) < 1$*;*
- $\Pi(B^*|A^*) = 1$*;*

(d) $\Pi(A^c|B) = \alpha \in (0,1)$ *and the extension of* Π *on* $\{B^*|A^*\}$ *must satisfy one of the following conditions whenever the values are coherent*

- $\Pi(B|A^*) = 0$ *or* $\Pi(B^c|A^*) = 0$*;*
- $0 < \Pi(B|A^*) < \alpha$ *or* $0 < \Pi(B^c|A^*) < \alpha$*;*
- $\Pi(B|A) = \alpha$ *and* $\alpha \leq \Pi(B|A^c) < 1$ *or* $\Pi(B^c|A) = \alpha$ *and* $\alpha \leq \Pi(B^c|A^c) < 1$*;*
- $\Pi(B^*|A^*) = 1$*.*

Proof. The proof follows directly from the proof of Theorem 3.

Remark 3. From previous theorem it is possible to derive also an analogous characterization for coherent T_{DP}-conditional possibilities. This can be done (once $\Pi(A|B^*)$ and $\Pi(A^c|B^*)$ are fixed) by taking into account only the coherent values for $\Pi(B|A^*)$ and $\Pi(B^c|A^*)$ with respect to the T_{DP}-conditioning that satisfy condition *(ii)* of Definition 5. In this case the significant layers are implied by Remark 1.

Theorem 4 implies that our definition of independence is stronger than usual ones, in fact if $A \perp\!\!\!\perp B[\Pi]$ under a T-conditional possibility (or a T_{DP}-conditional possibility), then

$$\begin{aligned}\Pi(A) &= \max\{\Pi(A \wedge B), \Pi(A \wedge B^c)\} \\ &= \max\{T(\Pi(A|B), \Pi(B)), T(\Pi(A|B^c), \Pi(B^c))\} = \Pi(A|B)\end{aligned} \tag{5}$$

and moreover

$$\Pi(A \wedge B) = T(\Pi(A|B), \Pi(B)) = T(\Pi(A), \Pi(B)). \tag{6}$$

The proposed notion of independence is not symmetric, nevertheless, as there are just few separation criteria able to represent asymmetric independence models, symmetry is often required. From Theorem 4 we can obtain the corresponding result related to the symmetric property.

Corollary 1. *Let A and B be two logically independent events. Consider a coherent T-conditional [T_{DP}-conditional] (with T any continuous t-norm) possibility Π on a set $\mathcal{G}$ containing $\mathcal{D} = \{A^*|B^*, B^*|A^*\}$, then $A \perp\!\!\!\perp B[\Pi]$ and $B \perp\!\!\!\perp A[\Pi]$ if and only if $\Pi(A|B) = \Pi(A|B^c)$ and $\Pi(A^c|B) = \Pi(A^c|B^c)$ and $\Pi(B|A) = \Pi(B|A^c)$ and $\Pi(B^c|A) = \Pi(B^c|A^c)$.*

4 Conclusions

In every framework devoted to manage uncertainty, conditioning and independence are the main concepts for updating information and for reasoning under hypotheses. For that the concept of conditioning cannot be relegated only to the role of restriction of the domain of possible events, when an event is occurred, but it is important to regard the conditioned and conditioning events as entities of the same kind, having in a certain moment a different role.

This consideration makes preferable to introduce a conditional measure as a function directly defined on a structured set of conditional events, satisfying suitable axioms. Such approach implies that, to give a Kolmogorovian-like representation, it is necessary to refer not to a single unconditional measure, but rather to a "structured" class of unconditional measures. The concept of independence introduced here for T-conditional possibility (which is inspired to the one given in [6,7] for conditional probability and generalizes the ones given in [9,19] in the particular cases $T = \min$ or is strict) deeply relies on this class representation. This notion in fact requires not only a classical condition based on T-conditional possibility values $\Pi(A^*|B^*)$, but also a reinforcement condition, regarding the significant layers of a T-nested class agreeing with Π on $\mathcal{D} = \{A^*|B^*, B^*|A^*\}$. This last condition aims to guarantee that logical independence among A and B is a necessary condition for possibilistic independence. Since to handle significant layers can be non-immediately understandable, we provided a characterization of independence only using the values of the T-conditional possibility on $\mathcal{D}$. Due to the generality of continuous t-norms, this characterization needs to take into account many different situations.

In the paper we consider also T-conditional possibilities obtained through the minimum specificity principle, introduced by Dubois and Prade, regarded as a specific class of T-conditional possibilities. For them it is possible to introduce exactly the same notion of independence, that however has a different characterization in terms of the values of the T_{DP}-conditional possibility on $\mathcal{D}$.

As a future work we plan to deal with an ensuing notion of conditional independence for variables and to study the related graphoid properties as already done for $T = \min$ or strict [9,19].

References

1. Ben Amor, N., Benferat, S.: Graphoid properties of qualitative possibilistic independence relations. Int. J. of Unc., Fuzz. and K.-B. Sys. 13(1), 59–96 (2005)
2. Benferhat, S., Tabia, K., Sedki, K.: Jeffrey's rule of conditioning in a possibilistic framework. Ann. of Math. and Art. Int. 66(3), 185–202 (2011)

3. Bouchon-Meunier, B., Coletti, G., Marsala, C.: Independence and Possibilistic Conditioning. Ann. of Math. and Art. Int. 35(1-4), 107–123 (2002)
4. Coletti, G., Petturiti, D., Vantaggi, B.: Possibilistic and probabilistic likelihood functions and their extensions: Common features and specific characteristics. Submitted to Fuzz. Sets and Sys.
5. Coletti, G., Petturiti, D., Vantaggi, B.: Independence for coherent T-conditional possibilities. Tech. Rep. n. 2, 2013 of Dip. di Matematica e Informatica, Università di Perugia (2013), http://www.dmi.unipg.it/davide.petturiti/TR_2_2013.pdf
6. Coletti, G., Scozzafava, R.: Zero probabilities in stochastic independence. In: Bouchon-Meunier, B., Yager, R.R., Zadeh, L.A. (eds.) Information, Uncertainty and Fusion, pp. 185–196. Kluwer, Dordrecht (2000)
7. Coletti, G., Scozzafava, R.: Stochastic Independence in a coherent setting. Ann. of Math. and Art. Int. 35, 151–176 (2002)
8. Coletti, G., Vantaggi, B.: Inferential processes leading to possibility and necessity. Inf. Sci. (in press), doi:10.1016/j.ins.2012.10.034
9. Coletti, G., Vantaggi, B.: Possibility theory: Conditional independence. Fuzz. Sets and Sys. 157(11), 1491–1513 (2006)
10. Coletti, G., Vantaggi, B.: T-conditional possibilities: Coherence and inference. Fuzz. Sets and Sys. 160(3), 306–324 (2009)
11. de Campos, L.M., Huete, J.F.: Independence concepts in possibility theory: part I. Fuzz. Sets and Sys. 103, 127–152 (1999)
12. de Cooman, G.: Possibility theory II: Conditional Possibility. Int. J. Gen. Sys. 25, 325–351 (1997)
13. de Cooman, G.: Possibility theory III: Possibilistic independence. Int. J. Gen. Sys. 25, 353–371 (1997)
14. de Finetti, B.: Sull'impostazione assiomatica del calcolo delle probabilità, Annali Univ. di Trieste 19, 3–55 (1949); English translation in: Probability, Induction and Statistics, ch. 5. Wiley, London (1972)
15. Didelez, V.: Asymmetric Separation for Local Independence Graphs. In: Proc. of 22nd Conf. UAI, pp. 130–137 (2006)
16. Dubins, L.E.: Finitely Additive Conditional Probabilities, Conglomerability and Disintegrations. Ann. Prob. 3(1), 89–99 (1975)
17. Dubois, D., Prade, H.: Possibility Theory: An Approach to Computerized Processing of Uncertainty. Plenum Press, New York (1988)
18. Dubois, D., Prade, H.: When upper probabilities are possibility measures. Fuzz. Sets and Sys. 49, 65–74 (1992)
19. Ferracuti, L., Vantaggi, B.: Independence and conditional possibility for strictly monotone triangular norms. Int. J. of Int. Sys. 21(3), 299–323 (2006)
20. Fonck, P.: Conditional independence in possibility theory. In: Proc. 10th Conf. UAI, pp. 221–226 (1994)
21. Hisdal, E.: Conditional possibilities independence and noninteraction. Fuzz. Sets and Sys. 1(4), 283–297 (1978)
22. Miranda, E., de Cooman, G., Couso, I.: Lower previsions induced by multi-valued mappings. J. of Stat. Plan. and Inf. 133(1), 173–197 (2005)
23. Petturiti, D.: Coherent Conditional Possibility Theory and Possibilistic Graphical Modeling in a Coherent Setting. PhD thesis, Università degli Studi di Perugia (2013)
24. Shilkret, N.: Maxitive measure and integration. Ind. Math. 74, 109–116 (1971)
25. Vantaggi, B.: The L-separation criterion for description of cs-independence models. Int. J. of App. Reas. 29(3), 291–316 (2002)

Probabilistic Satisfiability and Coherence Checking through Integer Programming

Fabio Gagliardi Cozman and Lucas Fargoni di Ianni

Universidade de Sao Paulo
Av. Prof. Mello Moraes, 2231, Sao Paulo, SP - Brazil

Abstract. This paper presents algorithms based on integer programming, both for probabilistic satisfiability and coherence checking. That is, we consider probabilistic assessments for both standard probability measures (Kolmogorovian setup) and full conditional measures (de Finettian coherence setup), and in both cases verify satisfiability/coherence using integer programming. We present empirical evaluation of our method, with evidence of phase-transitions.

1 Introduction

The analysis of arguments that combine propositions and probabilities has deserved attention for quite some time. For instance, in Boole's work [8] we find interesting examples such as:

> The probability that it thunders upon a given day is p, the probability that it both thunders and hails is q, but of the connexion of the two phenomena of thunder and hail, nothing further is supposed to be known. Required the probability that it hails on the proposed day.

Here we have propositions A and B, assessments $P(A) = p$ and $P(A \cap B) = q$. Boole asks for $P(B)$ and obtains the tight interval $[q, 1-(p-q)]$. The assessments are *coherent*: there is a probability measure that *satisfies* them.

Suppose we have propositional sentences $\{\phi_i\}_{i=1}^{M}$, each containing a subset of atomic propositions $\{A_j\}_{j=1}^{n}$. We may associate one or more of these sentences with probabilities, writing for instance $P(\phi_i) = \alpha_i$. To establish semantics for these assessments, we consider a probability measure over the set of truth assignments. The *Probabilistic Satisfiability (PSAT)* problem is to determine whether it is possible to find a probability measure over truth assignments such that all assessments are satisfied [14, 18–20, 23]. When assessments involve conditional probabilities such as $P(\phi_i'|\phi_i'') = \alpha_i$, there are two paths to follow. The Kolmogorovian setup reduces such assessments to ratios of probabilities. The other path is to use de Finetti's theory of coherent probabilities, where full conditional measures are used to interpret conditional assessments [11, 12, 32]. The *Coherent Probability Assessment (CPA)* problem is to determine whether it is possible to find a full conditional measure that satisfies all assessments [3, 4].

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 145–156, 2013.

Probabilistic satisfiability and coherence checking are central problems in reasoning under uncertainty. They serve as a foundation for logical and probabilistic inference, as a basis for probabilistic rules [30], and as an initial necessary step in the understanding of combinations of first-order logic and probabilities [21, 27, 31].

The most direct way to solve a PSAT problem is to write down the problem as a linear consistency problem [19]. The difficulty is that the resulting linear program may be too large. One may resort to column generation techniques [25], or to inference rules that capture probabilistic relationships [3, 16], or even to combinations of column generation and inference rules [24]. There is also a different approach to probabilistic satisfiability that tackles it by transformation into logical satisfiability [15].

In this paper we present another approach to Probabilistic Satisfiability, where the original problem is written as an integer linear program of size that is polynomial on the size of the original problem. The algorithm is extremely simple to state; our implementation shows that it is quite efficient compared to alternatives. Using our implementation we have studied the issue of phase transitions. We report these experiments in this paper.

Section 2 summarizes necessary background. Our basic algorithm is described in Section 3. Implementation and experiments, with a discussion of phase transitions, are presented in Section 4. Conditional probabilities are handled in Section 5, and inference problems are discussed in Section 6.

2 SAT and PSAT

Consider n atomic propositions A_j and M sentences ϕ_i in propositional logic. If a truth assignment ω is such that sentence ϕ is True, write $\omega \models \phi$. The Satisfiability (SAT) problem is to determine whether or not there exists a truth assignment to all variables such that all sentences evaluate to True [10, 17]. If every sentence ϕ_i is a conjunction of clauses, then we have a SAT problem in *Conjunctive Normal Form (CNF)*. A SAT problem in CNF is a k-SAT problem when each clause has k literals. The 2-SAT problem has a polynomial solution, while k-SAT is NP-complete for $k > 2$.

For a fixed n, m and k, one may generate a random k-SAT with n propositions and a single sentence in CNF with m clauses, as follows. For each one of the m clauses: select k variables at random, and for each variable produce a literal that may be negated or not, with probability half. There has been intense study of *phase transition* phenomena in random k-SAT; that is, study of the observed fact that for small values of m/n the probability that a random k-SAT is satisfiable tends to one as n grows (at fixed m/n), while for large values of m/n the probability that a random k-SAT is satisfiable tends to zero as n grows. Moreover, in the regions where satisfiability has probability approaching zero or one we observe that generated random k-SAT problems can be easily solved, while in the transition between the two regions we find hard problems.

Suppose that some sentences, say ϕ_1 to ϕ_q, for $q \leq M$, are associated with probabilities through *assessments* of the form $P(\phi_i) \bowtie \alpha_i$, where $\bowtie$ is one of $\geq$,

$=, \leq$. The semantics of such an assessment is as follows. Take the set of 2^n truth assignments that can be generated for the n propositions. A probability measure P over this set satisfies the assessments if, for each assessment $P(\phi_i) \bowtie \alpha_i$,

$$\sum_{\omega \models \phi_i} P(\omega) \bowtie \alpha_i. \tag{1}$$

The Probabilistic Satisfiability (PSAT) problem is to determine whether a given set of sentences and probabilistic assessments can be satisfied. That is, to determine whether there is a probability measure over those truth assignments that satisfy sentences not associated with probabilities, such that all assessments are satisfied by this probability measure. The k-PSAT problem is a PSAT problem where each sentence is in CNF and where each clause has k literals. The k-PSAT is NP-complete for *all* values of $k > 1$; note that even for $k = 2$ we obtain NP-completeness [23]. A few polynomial special cases of PSAT are known [2].

There are many algorithms for PSAT. The most obvious one is to write down M constraints of the form (1), one for each sentence; some will be actually associated with assignments $P(\phi_i) \bowtie \alpha_i$, while others will encode "pure" logical sentences as $P(\phi_i) = 1$. Each constraint can be written as

$$\sum_{j=1}^{2^n} I_{\phi_i}(\omega_j) P(\omega_j) \bowtie \alpha_i, \qquad \text{where } I_{\phi_i}(\omega_j) = \begin{cases} 1 & \text{if } \omega_j \models \phi_i \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

while truth assignments ω_j are ordered from 1 to 2^n (say by the n-bit binary number obtained by writing 0 for False and 1 for True as assigned to $A_1, \ldots, A_n$). Add to these M linear constraints the necessary constraints $\sum_\omega P(\omega) = 1$ and $P(\omega) \geq 0$ for all ω. Probabilistic Satisfiability is then obtained when the resulting set of linear constraints has a solution. The challenge is that we have 2^n truth assignments, so the size of the linear constraints is exponential in the input.

The most efficient algorithms for PSAT combine linear programming techniques and inference rules to simplify the problem [24]. These algorithms use the fact that a PSAT problem is satisfiable if and only if there is a probability measure that assigns positive probability mass to $(M + 1)$ truth assignments; all other truth assignments get zero probability mass [18]. Hence we can write down a $(M + 1) \times (M + 1)$ matrix $\mathbf{C}$ and write the PSAT problem as feasibility of $\mathbf{Cp} \bowtie \boldsymbol{\alpha}$, where $\boldsymbol{\alpha}$ denotes a vector of values α_i and $\bowtie$ refers to $\geq$, $=$ or $\leq$ as appropriate. Each column of $\mathbf{C}$ corresponds to a truth assignment; the challenge is to select $(M + 1)$ truth assignments. This is done through column generation techniques from linear programming [5]. Initially a set of $(M + 1)$ columns is selected, and then pivoting operations exchange columns until the problem is determined to be satisfiable or not. At each pivoting operation, a column is removed from $\mathbf{C}$, and the choice of the column to enter $\mathbf{C}$ happens through an auxiliary optimization problem (there are several possible formulations for this auxiliary problem) [23, 24]. Performance improvements are obtained if column

generation is preceded by application of inference rules.[1] This combination has produced the best results so far, being able to solve PSAT problems with up to 200 propositions and 800 clauses, each one of them a clause associated with a probability.

An entirely different approach to PSAT has been developed by Finger and De Bona [15]; here the selection of columns of $\mathbf{C}$ is reduced to a SAT problem. All operations with linear constraints are encoded into SAT by careful analysis of numerical precision, and a SAT solver is used to solve the PSAT problem.

The resulting methods are fairly sophisticated and require numerical care. Moreover, the extension of such methods to conditional probabilities in de Finetti's coherency framework is difficult, and existing methods require sequences of linear programs (Section 5). In this paper we propose a novel approach that addresses these concerns.

A PSAT is in *Normal Form* if a single sentence ϕ is given, and each probabilistic assessment is an equality associated with a single proposition (that is, every probabilistic assessment is of the form $P(A_i) = \alpha_i$) [15]. Even though this form may seem restrictive, every PSAT can be brought to it with polynomial effort: basically, for each assessment $P(\phi_i) \bowtie \alpha_i$, introduce if necessary fresh propositions to transform the assessment into $P(\phi_i') = \alpha_i$; then introduce a new proposition A_i' and exchange the original assessment by a sentence $A_i' \Leftrightarrow \phi_i'$ and an assessment $P(A_i') = \alpha_i$; finally, generate a single sentence ϕ that is a conjunction of all previous sentences. Every k-PSAT for $k > 2$ can be reduced to Normal Form with q assessments $P(A_i) = \alpha_i$ plus one CNF ϕ consisting of clauses with exactly 3 literals each.

3 PSAT through Integer Programming

Assume our PSAT problem is in Normal Form with assessments $\{P(A_j) = \alpha_j\}_{j=1}^{q}$ and a sentence ϕ in CNF with m clauses, each clause with k literals. So our problem is parameterized by the number of propositions n, the number of assessments q, the number of clauses m, and the number of literals per clause k. Such a parameterized Normal Form neatly separates the probabilistic and the propositional aspects of Probabilistic Satisfiability.

Our problem is: find the $(q+1)$ columns of $\mathbf{C}$, each one corresponding to a truth assignment ω such that $\omega \models \phi$, in such a way that $\mathbf{Cp} = \boldsymbol{\alpha}$.

Hence we have $(q+1)^2$ optimization variables (elements of $\mathbf{C}$ to look for); all of them are binary with values 0 and 1. As noted previously, Finger and De Bona reduce the search for these variables to a SAT problem [15]. We instead find $\mathbf{C}$ by solving an integer program.

Consider looking for the jth column of $\mathbf{C}$; denote it by $\mathbf{C}_j$. Such a column corresponds to a truth assignment that satisfies ϕ. We explore the well known connection between SAT and integer programming to find such a truth assignment [10]. Start by generating a vector $\mathbf{a}_j$ with n binary variables $\{a_{i,j}\}_{i=1}^{n}$, all

[1] An example of an inference rule [24]: if $P(A_1) \in [\underline{\alpha}_1, \overline{\alpha}_1]$ and $P(\neg A_1 \vee A_2) \in [\underline{\alpha}_1, \overline{\alpha}_2]$ for $\overline{\alpha}_1 + \overline{\alpha}_2 \geq 1$, then $P(A_2) \in [\max(0, \underline{\alpha}_1 + \underline{\alpha}_2 - 1), \min(1, \overline{\alpha}_2)]$.

1: **procedure** PSAT-IP(propositions $\{A_j\}_{j=1}^{n}$, assessments $\{P(A_i) = \alpha_i\}_{i=1}^{q}$, sentence ϕ in CNF with m clauses)
2: ▷ Variables $a_{i,j}$ are binary; variables $b_{i,j}$ and p_j are real-valued in $[0, 1]$.
3: **for** $j \in \{1, \ldots, q+1\}$ and each clause $(\vee_{l'=1}^{k'} A_{i_{l'}}) \vee (\vee_{l''=1}^{k''} \neg A_{i_{l''}})$ of ϕ **do**
4: Generate linear constraint $(\sum_{l'=1}^{k'} a_{i_{l'},j}) + (\sum_{l''=1}^{k''} (1 - a_{i_{l''},j})) \geq 1$.
5: **for** $i \in \{1, \ldots, q+1\}$ **do**
6: Generate linear constraint $\sum_{j=1}^{q+1} b_{i,j} = \alpha_i$.
7: **for** $j \in \{1, \ldots, q+1\}$ **do**
8: Generate linear constraints $0 \leq b_{i,j} \leq a_{i,j}$ and $a_{i,j} - 1 + p_j \leq b_{i,j} \leq p_j$.
9: **return** Satisfiable if linear constraints have a solution, Unsatisfiable otherwise.

Fig. 1. PSAT solution based on integer linear program

with values 0 and 1. Now take one clause of ϕ; suppose it is written as

$$(\vee_{l'=1}^{k'} A_{i_{l'}}) \vee (\vee_{l''=1}^{k''} \neg A_{i_{l''}}).$$

For this clause, generate the linear inequality:

$$\left(\sum_{l'=1}^{k'} a_{i_{l'},j}\right) + \left(\sum_{l''=1}^{k''} (1 - a_{i_{l''},j})\right) \geq 1. \tag{3}$$

Consider the m inequalities generated this way (one per clause). A vector $\mathbf{a}_j$ that satisfies these m inequalities yields a truth assignment for ϕ by assigning True to A_i when $a_{i,j}$ is one, and assigning False to A_i when $a_{i,j}$ is zero. Note that the elements of $\mathbf{C}_j$ are exactly $a_{1,j}$ to $a_{q,j}$.

We generate the whole matrix $\mathbf{C}$ by generating $(q+1)$ sets of variables $\mathbf{a}_j$ and their related inequalities. We now have inequalities for all elements of $\mathbf{C}$, and we need to solve $\mathbf{Cp} = \boldsymbol{\alpha}$. To do so, note that each row of $\mathbf{C}$ represents an equality as follows:

$$\sum_{j=1}^{q+1} a_{i,j} p_j = \alpha_i, \tag{4}$$

where p_j denotes the jth element of $\mathbf{p}$. The challenge is to reduce the bilinear term $a_{i,j}p_j$ to linear constraints. We do that by introducing a new fresh variable $b_{i,j}$ and the constraints:

$$0 \leq b_{i,j} \leq a_{i,j} \qquad \text{and} \qquad a_{i,j} - 1 + p_j \leq b_{i,j} \leq p_j. \tag{5}$$

Note that if $a_{i,j} = 0$, then $b_{i,j} = 0$; and if $a_{i,j} = 1$, then $b_{i,j} = p_j$.

The whole algorithm is presented in Figure 1; it basically collects constraints from Expressions (3), (4), and (5). The algorithm produces an integer linear program that has a solution if and only if the original PSAT problem is satisfiable.

4 Implementation, Experiments, and Phase Transition

We have coded our PSAT method using the Java language with calls to CPLEX version 12, and run experiments in iMac computers with 4GBytes of memory.

The algorithm is very compact, using only 45 lines of code (basically a direct translation of the algorithm in Figure 1 into CPLEX calls).

We focused on two values of k, namely, 2 and 3. We investigated $k = 2$ because 2-SAT is polynomial and 2-PSAT is NP-complete, a property not shared by any other k-PSAT. And we investigated $k = 3$ because any PSAT problem can be polynomially reduced to a 3-PSAT problem; in fact, Finger and De Bona pay attention to 3-PSAT for this reason [15].

Additionally, we were particularly interested in investigating phase transition phenomena. Until the work of Baioletti et al. [4], and Finger and De Bona [15], there was no evidence of phase transition in the literature. Consequently it makes more sense at this stage to examine the behavior of PSAT for various values of n, m and q, rather than to randomly try out large problems that may in the end be easy.

Figure 2 summarizes a number of experiments for $k = 2$. In all of them, PSAT problems were randomly generated from parameters n, m, q and k: m clauses with k literals each were randomly generated by selecting propositions randomly out of the n propositions; each literal was negated or not with probability $1/2$; finally, the first q propositions were associated with probabilities randomly selected in the interval $[0, 1]$. Each point in each graph conveys mean values for 50 different random PSAT problems. We set a time limit of 10 minutes per problem; some of the more difficult problems did not finish within this time limit.

The left graphs in Figure 2 show typical behavior for random 2-PSAT: the darker line indicates the percentage of satisfiable problems, and the lighter line indicates mean time spent in their solution (mean of 50 distinct random PSAT problems). The top graph deals with 2-PSAT problems with 1000 variables and up to 1500 clauses; these are rather large problems and the phase transition phenomenon is clear (note that we are using larger values of q than in the previous investigation by Finger and De Bona [15]). The lower graph conveys the same information, but now for $n = 100$. The main point to note is that the phase transition seems to occur for much smaller m/n. Indeed the presence of probabilities seems to create relationships between n, q and m in ways that are not observed in 2-SAT (where the phase transition occurs for $m/n = 1$). An interesting display of this phenomenon can be found in the right graph, where one can see that the phase transition is affected by q.

Similar results are displayed in Figure 3. In the left graph we see typical phase transition behavior, now centered around $m/n \approx 3.5$. The reason we show this particular graph (with $n = 40$, $q = 4$) is that the same experiment is reported by Finger and De Bona [15]; their reported times are about 10 times larger than ours. The right graph shows the change in the location of phase transition as q varies, similarly to what happens with 2-PSAT.

To give a better feel of the times involved in solving PSAT problems with our method, Table 1 summarizes a large variety of tests; each entry is the mean of 50 distinct PSAT problems. Note that it is not correct to expect that the larger the problem, the more time it takes; due to phase transition, some large problems may be easy, while some apparently small problems may be hard.

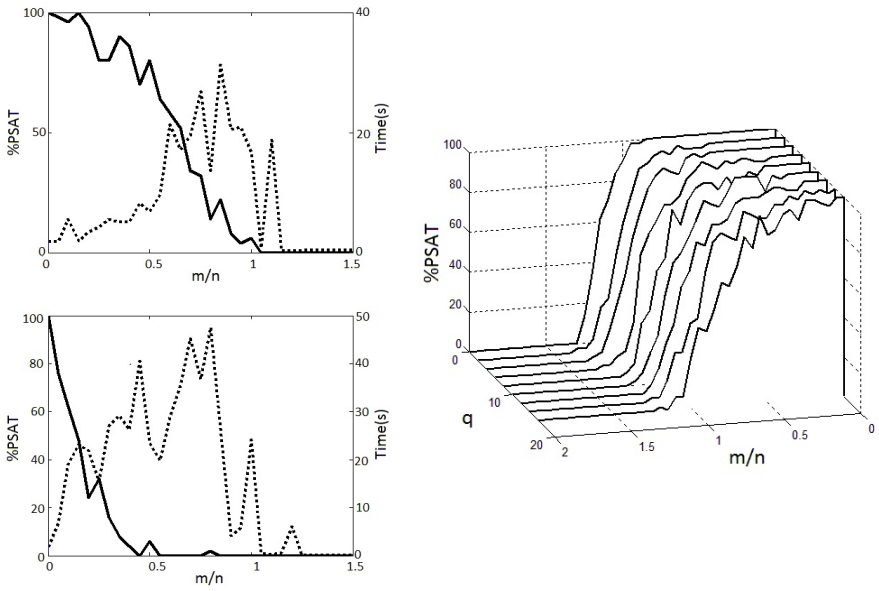


Fig. 2. Experiments with 2-PSAT

5 Checking Coherence of Conditional Assessments through Integer Programming

Suppose that conditional assessments $P(\phi'_i|\phi''_i) = \alpha_i$ must be processed. In the standard, Kolmogorovian style, probability theory, this assessment means that $P(\phi'_i \wedge \phi''_i)/P(\phi''_i) = \alpha_i$ if $P(\phi''_i) > 0$. This holds if and only if

$$P(\phi'_i \wedge \phi''_i) - \alpha_i P(\phi''_i) = 0. \tag{6}$$

The only change from "unconditional" PSAT is that each element of the matrix $\mathbf{C}$ is now a linear expression. Indeed, if we only take conditional assessments of the form $P(A'_i|A''_i) = \alpha_i$, then the element $\mathbf{C}_{i,j}$ is given by the nonlinear expression $a'_{i,j}a''_{i,j} - \alpha_i a''_{i,j}$, where $a'_{i,j}$ and $a''_{i,j}$ are binary variables corresponding to propositions $A'_{i,j}$ and $A''_{i,j}$ respectively. To handle this, the only change in our previous algorithm is that the constraints in its line 6 must be replaced by $\sum_{j=1}^{q+1} \left(b'_{i,j} - \alpha_i b''_{i,j}\right) = 0$, and constraints in line 8 must be replaced by

$$0 \leq b'_{i,j} \leq a'_{i,j}, \qquad 0 \leq b'_{i,j} \leq a''_{i,j}, \qquad a'_{i,j} + a''_{i,j} - 2 + p_j \leq b'_{i,j} \leq p_j,$$

$$0 \leq b''_{i,j} \leq a''_{i,j}, \qquad a''_{i,j} - 1 + p_j \leq b''_{i,j} \leq p_j.$$

This Kolmogorovian setup requires some care when interpreting conditional assessments. Suppose first that A''_i has probability zero in every probability

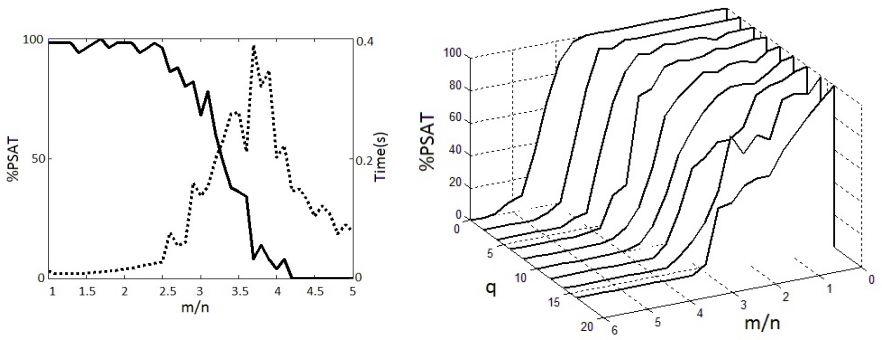


Fig. 3. Experiments with 3-PSAT

Table 1. Experiments with 2-PSAT and 3-PSAT

n	q	m	2-PSAT, mean time (sec.)	3-PSAT, mean time (sec.)
500	25	500	2.5441801	1.8612529
500	25	750	0.1588879	1.8934227
500	25	1000	0.1802416	12.0530050
500	50	500	18.271062	16.1097538
500	50	750	0.3348024	48.0028159
500	50	1000	0.9317918	177.037949
750	25	500	8.9984422	4.6699495
750	25	750	7.6501204	5.5394115
750	25	1000	0.2109049	3.6071930
750	50	500	0.3424100	19.9149997
750	50	750	0.4221426	23.5807273
750	50	1000	1.0847125	30.1025869
1000	25	500	2.4382244	2.1816687
1000	25	750	1.5742295	2.2077639
1000	25	1000	0.5740323	3.1606671
1000	25	1500	0.3041526	3.0211587
1000	25	2000	0.5616025	28.1456910
1000	50	500	3.5237384	16.2340889
1000	50	750	1.0613374	15.0277582
1000	50	1000	0.4471337	17.4252168

measure that satisfies all assessments. In this case we may take $P(A_i'|A_i'') = \alpha_i$ to be a misguided assignment to a quantity that should really be left undefined. However suppose that some satisfying probability measures assign zero probability to A_i'', while others do not. The most reasonable interpretation of this situation is that only those probability measures that assign positive probability to A_i'' should be retained; the others do not satisfy the fact that $P(A_i'|A_i'')$ has been actually assessed.

An entirely different view of conditional probability can be found in de Finetti's theory of coherence. Here conditional probability is not a derived concept, but rather the primary object of interest. Assessments can be given on arbitrary events, and coherence of assessments is equated to existence of a *full conditional measure* that satisfies the assessments. The selection of a particular conditional measure imposes considerable structure on events, while de Finetti's approach assumes little algebraic structure on the assessments [7, 11]. A full conditional

measure $P : \mathcal{B} \times (\mathcal{B}\backslash\emptyset) \to \Re$, where $\mathcal{B}$ is a Boolean algebra over a set Ω, is a two-place set-function such that for every nonempty event C [13]:

- $P(C|C) = 1$ and $P(A|C) \geq 0$ for all A;
- $P(A \cup B|C) = P(A|C) + P(B|C)$ when $A \cap B = \emptyset$;
- $P(A \cap B|C) = P(A|B \cap C)P(B|C)$ when $B \cap C \neq \emptyset$.

Note that conditioning is defined for every nonempty event; whenever the conditioning event is Ω, we suppress it and write the "unconditional" probability $P(A)$. There are other names for full conditional measures in the literature, such as *conditional probability measures* [26]; and *complete conditional probability systems* [29]. Full conditional measures have been applied in economics [22, 29], philosophy [1, 28], artificial intelligence [9].

So, suppose we have the same propositions and assessments as before, and events are interpreted as sets of truth assignments. We say the assessments are *coherent* if there is a full conditional measure that satisfies them [11, 12]. Note that a set of assessments may be coherent even if $P(B) = 0$ and $P(A|B) = \alpha > 0$; a probability measure that assigns probability zero to a conditioning event need not be discarded.

There are algorithms for coherency checking that basically work by dividing the space of truth assignments into "layers": the first layer contains the truth assignments with positive unconditional probability; the second layer contains the truth assignments with positive conditional probability given the complement of the first layer, and so on [9, 11]. For each layer an appropriately specified PSAT problem is solved, and the collection of PSAT problems yields the desired coherency check. Alternative algorithms employ local rules that mimic logical inference [3, 4]. To the extent that these methods solve linear programs in intermediate steps, and these linear programs are of size exponential in the input, the reductions to integer programming that we have explored before can be used.

To illustrate the last comment, consider the formulation of coherence checking that is due to Walley et al. [32]. They consider that assessments are of the form $P(A'_i|A''_i) \geq \alpha_i$, and show that existence of a satisfying full conditional measure is equivalent to:

$$\sup_{\omega \models S_\lambda} \left(\sum_{i=1}^{q} \lambda_i G_i(\omega) \right) \geq 0 \quad \text{whenever } \forall i : \lambda_i \geq 0 \text{ and } \exists i : \lambda_i > 0,$$

where $G_i(\omega) = I_{A''_i}(\omega)(I_{A'_i}(\omega) - \alpha_i)$, $S_\lambda = \vee_{i:\lambda_i>0} A''_i$ and $I_A(\omega)$ is the indicator function defined in Expression (2): 1 if $\omega \models A$ and 0 otherwise.

Walley et al. offer the following algorithm to check coherence, where a sequence of linear programs with more than 2^n constraints each is generated. First, set $\mathcal{I} = \{1, \ldots, q\}$. Solve the linear program in Expression (7) below. If $\tau_i = 1$ for all $i \in \mathcal{I}$, then coherence fails (problem is Unsatisfiable). Otherwise, replace $\mathcal{I}$

by $\{i \in \mathcal{I} : \tau_i = 1\}$. If $\mathcal{I}$ becomes empty, then coherence holds (problem is Satisfiable); otherwise solve linear problem in Expression (7) again, and so on.

$$\max \sum_{i \in \mathcal{I}} \tau_i \tag{7}$$
$$\text{s.t.: } \forall \omega : \sum_{i \in \mathcal{I}} \lambda_i G_i(\omega) + \sum_{i \in \mathcal{I}} \tau_i I_{A_i''}(\omega) \leq 0; \forall \lambda_i : \lambda_i \geq 0; \forall i \in \mathcal{I} : \tau_i \in [0, 1].$$

Note that at each step we have the dual of a linear program with 2^n optimization variables; one can therefore write a compact integer linear program at each step, using the techniques described in previous sections.

6 Inference

In this section we offer a few brief comments on the *inference* problem: given a satisfiable or coherent set of assessment, find all possible values for $P(\varphi)$, the probability of an additional sentence φ, such that all assessments together are still satisfiable/coherent.

In the Kolmogorovian setup that is usually adopted for PSAT, both $\min P(\varphi)$ and $\max P(\varphi)$ can be obtained by adding appropriate linear objective functions to our methods. If additionally one wants tight bounds on a conditional probability $P(\varphi'|\varphi'')$, then linear fractional programming [5] can be used in the Kolmogorovian setup to transform $\min P(\varphi' \wedge \varphi'')/P(\varphi'')$ into a linear objective function (and similarly for $\max P(\varphi' \wedge \varphi'')/P(\varphi'')$).

The inference problem is considerably more complex in de Finetti's framework. Walley et al. [32, Algorithm 5] present solutions for such a situation that rely on sequences of linear programs. The discussion in Section 5 applies to that case. Here the verification of coherence is a preliminary step, because only coherent assessments are allowed to be used in inference [6].

7 Conclusion

In this paper we have introduced an approach to probabilistic satisfiability and coherence checking that translates these problems into integer linear programming. Our algorithms have the advantage of simplicity when compared to alternative approaches. Because we can rely on existing highly optimized linear programming solvers, we do not worry about numerical stability; likewise, our algorithms can inherit any gains from parallelization and heuristics applied to integer linear programming.

Experiments indicate that our algorithms are quite effective for random PSAT problems. Moreover, we have presented an analysis of phase transition in PSAT that improves previous results in the literature. Of course more testing is necessary to fully understand the properties of probabilistic satisfiability and coherence checking.

Acknowledgements. Both authors received support by CNPq. We thank the reviewers for very useful suggestions, in particular for pointing us to Ref. [3] and [4].

References

1. Adams, E.W.: A Primer of Probability Logic. CSLI Publications, Stanford (2002)
2. Andersen, K.A., Pretolani, D.: Easy cases of probabilistic satisfiability. Annals of Mathematics and Artificial Intelligence 33(1), 69–91 (2001)
3. Baioletti, M., Capotorti, A., Tulipani, S., Vantaggi, B.: Simplification rules for the coherent probability assessment problem. Annals of Mathematics and Artificial Intelligence 35(1-4), 11–28 (2002)
4. Baioletti, M., Capotorti, A., Tulipani, S.: An empirical complexity study for a 2CPA solver. In: Bouchon-Meunier, Coletti, G., Yager, R.R. (eds.) Modern Information Processing: From Theory to Applications, pp. 73–84 (2005)
5. Bertsimas, D., Tsitsiklis, J.N.: Introduction to Linear Optimization. Athena Scientific, Belmont (1997)
6. Biazzo, V., Gilio, A.: A generalization of the fundamental theorem of de Finetti for imprecise conditional probability assessments. International Journal of Approximate Reasoning 24, 251–272 (2000)
7. Biazzo, V., Gilio, A., Lukasiewicz, T., Sanfilippo, G.: Probabilistic logic under coherence: Complexity and algorithms. Annals of Mathematics and Artificial Intelligence 45(1-2), 35–81 (2005)
8. Boole, G.: The Laws of Thought. Dover edition (1958)
9. Capotorti, A., Galli, L., Vantaggi, B.: How to use locally strong coherence in an inferential process based on upper-lower probabilities. Soft. Computing 7(5), 280–287 (2003)
10. Chandru, V., Hooker, J.: Optimization Methods for Logical Inference. John Wiley & Sons Inc. (1999)
11. Coletti, G., Scozzafava, R.: Probabilistic Logic in a Coherent Setting. Trends in logic, vol. 15. Kluwer, Dordrecht (2002)
12. de Finetti, B.: Theory of Probability, vol. 1-2. Wiley, New York (1974)
13. Dubins, L.E.: Finitely additive conditional probability, conglomerability and disintegrations. Annals of Statistics 3(1), 89–99 (1975)
14. Fagin, R., Halpern, J.Y., Megiddo, N.: A logic for reasoning about probabilities. Information and Computation 87, 78–128 (1990)
15. Finger, M., De Bona, G.: Probabilistic satisfiability: Logic-based algorithms and phase transition. In: IJCAI, pp. 528–533 (2011)
16. Frisch, A.M., Haddawy, P.: Anytime deduction for probabilistic logic. Artificial Intelligence 69, 93–122 (1994)
17. Gent, I.P., Walsh, T.: The SAT phase transition. In: European Conference on Artificial Intelligence, pp. 105–109 (1994)
18. Georgakopoulos, G., Kavvadias, D., Papadimitriou, C.H.: Probabilistic satisfiability. Journal of Complexity 4, 1–11 (1988)
19. Hailperin, T.: Best possible inequalities for the probability of a logical function of events. American Mathematical Monthly 72, 343–359 (1965)
20. Hailperin, T.: Boole's Logic and Probability: a Critical Exposition from the Standpoint of Contemporary Algebra, Logic, and Probability Theory. North-Holland, Amsterdam (1976)

21. Halpern, J.Y.: Reasoning about Uncertainty. MIT Press, Cambridge (2003)
22. Hammond, P.J.: Elementary non-Archimedean representations of probability for decision theory and games. In: Humphreys, P. (ed.) Patrick Suppes: Scientific Philosopher, vol. 1, pp. 25–59. Kluwer, Dordrecht (1994)
23. Hansen, P., Jaumard, B.: Probabilistic Satisfiability. Technical Report G-96-31, Les Cahiers du GERAD, École Polytechique de Montréal (1996)
24. Hansen, P., Perron, S.: Merging the local and global approaches to probabilistic satisfiability. International Journal of Approximate Reasoning 47(2), 125–140 (2008)
25. Jaumard, B., Hansen, P., de Aragão, M.P.: Column generation methods for probabilistic logic. ORSA Journal on Computing 3(2), 135–148 (1991)
26. Krauss, P.: Representation of conditional probability measures on Boolean algebras. Acta Mathematica Academiae Scientiarum Hungaricae 19(3-4), 229–241 (1968)
27. Lukasiewicz, T.: Expressive probabilistic description logics. Artificial Intelligence 172(6-7), 852–883 (2008)
28. McGee, V.: Learning the impossible. In: Bells, E., Skyrms, B. (eds.) Probability and Conditionals, pp. 179–199. Cambridge University Press (1994)
29. Myerson, R.: Game Theory: Analysis of Conflict. Harvard University Press, Cambridge (1991)
30. Ng, R., Subrahmanian, V.S.: Probabilistic logic programming. Information and Computation 101(2), 150–201 (1992)
31. Nilsson, N.J.: Probabilistic logic. Artificial Intelligence 28, 71–87 (1986)
32. Walley, P., Pelessoni, R., Vicig, P.: Direct algorithms for checking consistency and making inferences from conditional probability assessments. Journal of Statistical Planning and Inference 126(1), 119–151 (2004)

Extreme Lower Previsions and Minkowski Indecomposability

Jasper De Bock and Gert de Cooman

Ghent University, SYSTeMS Research Group
Technologiepark–Zwijnaarde 914, 9052 Zwijnaarde, Belgium
{jasper.debock,gert.decooman}@UGent.be

Abstract Coherent lower previsions constitute a convex set that is closed and compact under the topology of point-wise convergence, and Maaß [2] has shown that any coherent lower prevision can be written as a 'countably additive convex combination' of the extreme points of this set. We show that when the possibility space has a finite number n of elements, these extreme points are either degenerate precise probabilities, or in a one-to-one correspondence with the (Minkowski) indecomposable compact convex subsets of $\mathbb{R}^{n-1}$.

Keywords: Extreme lower previsions, extreme credal sets, fully imprecise lower previsions, fully imprecise credal sets, Minkowski decomposition.

1 Introduction

In his Ph.D. dissertation, Maaß [2] proved a general, Choquet-like representation result for what he called inequality preserving functionals. When we apply his results to coherent lower previsions, which have an important part in the theory of imprecise probabilities, we find that the set of all coherent lower previsions defined on a subset of the linear space of all bounded real-valued maps (gambles) on a possibility space $\mathscr{X}$ constitute a convex set, that is furthermore closed and compact under the topology of point-wise convergence, and that any coherent lower prevision can be written as a 'countably additive convex combination' of the extreme points of this set.

It became apparent quite soon, however, that finding these extreme coherent lower previsions was a non-trivial task. Contributions to solving this problem were made by Quaeghebeur [5], who essentially concentrated on coherent lower previsions defined on finite domains. In this paper, we look at the extreme points of the set of all coherent lower previsions defined on the space of all real-valued maps on a finite set $\mathscr{X}$, containing n elements. We begin by defining (extreme) coherent lower previsions in Section 2. In Section 3, we recall that coherent lower previsions are in a one-to-one relationship with compact convex sets of probability mass functions, which allows us, in Sections 4 and 5, to establish a link between extreme coherent lower previsions on the one hand, and (Minkowski) indecomposable compact convex subsets of $\mathbb{R}^{n-1}$ on the other.

This link allows us to reduce the problem of finding all extreme coherent lower previsions to a problem that has received quite a bit of attention in the mathematical literature, and to use existing solutions for that problem. We give a short discussion of what can and could be learned from this connection in Section 6, and go on to discuss a number of avenues for further research and possible applications.

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 157–168, 2013.

2 Coherent Lower Previsions

Consider a variable X taking values in some non-empty set $\mathscr{X}$, called *possibility space*. We will restrict ourselves to finite possibility spaces $\mathscr{X} = \{x_1, \ldots, x_n\}$, with $n \in \mathbb{N}_{>1}$.[1,2] The theory of coherent lower previsions models a subject's beliefs regarding the uncertain value of X by means of lower and upper previsions of so-called gambles. A *gamble* is a real-valued map on $\mathscr{X}$ and we use $\mathscr{G}(\mathscr{X})$ to denote the set of all of them. A lower prevision $\underline{P}$ is a real-valued functional defined on this set $\mathscr{G}(\mathscr{X})$. $\underline{P}$ is said to be *coherent* if it satisfies the following three conditions: for all $f, g \in \mathscr{G}(\mathscr{X})$ and all real $\lambda > 0$

C1. $\underline{P}(f) \geq \min f$
C2. $\underline{P}(\lambda f) = \lambda \underline{P}(f)$ [non-negative homogeneity]
C3. $\underline{P}(f+g) \geq \underline{P}(f) + \underline{P}(g)$ [super-additivity]

The set of all coherent lower previsions on $\mathscr{G}(\mathscr{X})$ is denoted by $\underline{\mathbb{P}}(\mathscr{X})$. The conjugate of a lower prevision $\underline{P} \in \underline{\mathbb{P}}(\mathscr{X})$ is called an *upper prevision*. It is denoted by $\overline{P}$ and defined by $\overline{P}(f) := -\underline{P}(-f)$ for all gambles $f \in \mathscr{G}(\mathscr{X})$. Coherent lower and upper previsions can be given a behavioural interpretation in terms of buying and selling prices, turning the three conditions above into criteria for rational behaviour; see Ref. [9] for an in-depth study, and Ref. [4] for a recent survey.

2.1 Extreme Lower Previsions

Coherence is preserved under taking convex combinations [9, Section 2.6.4]. Consider two coherent lower previsions $\underline{P}_1$ and $\underline{P}_2$ in $\underline{\mathbb{P}}(\mathscr{X})$ and any $\lambda \in [0,1]$. Then the lower prevision $\underline{P} = \lambda \underline{P}_1 + (1-\lambda)\underline{P}_2$, defined by $\underline{P}(f) := \lambda \underline{P}_1(f) + (1-\lambda)\underline{P}_2(f)$ for all $f \in \mathscr{G}(\mathscr{X})$, will also be coherent. One can now wonder whether every coherent lower prevision can be written as such a convex combination of others: given a coherent lower prevision $\underline{P} \in \underline{\mathbb{P}}(\mathscr{X})$, is it possible to find coherent lower previsions $\underline{P}_1$ and $\underline{P}_2$ in $\underline{\mathbb{P}}(\mathscr{X})$ and $\lambda \in [0,1]$ such that $\underline{P} = \lambda \underline{P}_1 + (1-\lambda)\underline{P}_2$? If we exclude the trivial decompositions, where $\lambda = 0$, $\lambda = 1$ or $\underline{P}_1 = \underline{P}_2 = \underline{P}$, then the answer can be no. We will refer to those coherent lower previsions for which no non-trivial decomposition exists as *extreme lower previsions*. The goal of this paper is to characterise, and where possible to find, the set $\operatorname{ext}\underline{\mathbb{P}}(\mathscr{X})$ of all extreme lower previsions on $\mathscr{G}(\mathscr{X})$.

2.2 Special Kinds of Coherent Lower Previsions

In order to find these extreme lower previsions, it will be useful to split the set $\underline{\mathbb{P}}(\mathscr{X})$ into three disjoint subsets: linear previsions, lower previsions that are fully imprecise and lower previsions that are partially imprecise.

[1] $\mathbb{N}$ denotes the positive integers (excluding zero) and $\mathbb{R}$ the real numbers. Subsets are denoted by using predicates as subscripts; e.g., $\mathbb{N}_{\leq n} := \{i \in \mathbb{N} : i \leq n\} = \{1, \ldots, n\}$ denotes the positive integers up to n and $\mathbb{R}_{>0} := \{r \in \mathbb{R} : r > 0\}$ the strictly positive real numbers.

[2] We do not consider $n = 1$ because this case is both trivial and of no practical use. Indeed, a variable that can only assume a single value has no uncertainty associated with it.

A coherent lower prevision $\underline{P} \in \underline{\mathbb{P}}(\mathscr{X})$ is called a *linear prevision* if it has the additional property that $\underline{P}(f+g) = \underline{P}(f) + \underline{P}(g)$ for all $f, g \in \mathscr{G}(\mathscr{X})$. It is then generically denoted by P and we use $\mathbb{P}(\mathscr{X})$ to denote the set of all of them. It can be shown that for every *mass function* p in the so-called $\mathscr{X}$-simplex

$$\Sigma_{\mathscr{X}} := \left\{ p \in \mathbb{R}^{\mathscr{X}} : \sum_{i=1}^{n} p(x_i) = 1 \text{ and } p(x_i) \geq 0 \text{ for all } i \in \mathbb{N}_{\leq n} \right\}, \tag{1}$$

the corresponding expectation operator P_p, defined by $P_p(f) := \sum_{i=1}^{n} f(x_i)p(x_i)$ for all $f \in \mathscr{G}(\mathscr{X})$, is a linear prevision in $\mathbb{P}(\mathscr{X})$. Conversely, every linear prevision $P \in \mathbb{P}(\mathscr{X})$ has a unique mass function $p \in \Sigma_{\mathscr{X}}$ for which $P = P_p$. It is defined by $p(x_i) := P(\mathbb{I}_{\{x_i\}})$, $i \in \mathbb{N}_{\leq n}$, where $\mathbb{I}_{\{x_i\}}$ denotes the *indicator* of $\{x_i\}$: for all $x \in \mathscr{X}$, $\mathbb{I}_{\{x_i\}}(x) = 1$ if $x = x_i$ and $\mathbb{I}_{\{x_i\}}(x) = 0$ otherwise.

Another special kind of coherent lower previsions are those that are *fully imprecise*. They are uniquely characterised by the property that $\underline{P}(\mathbb{I}_{\{x_i\}}) = 0$ for all $i \in \mathbb{N}_{\leq n}$. As we shall see further on, we can interpret $\underline{P}(\mathbb{I}_{\{x_i\}})$ as the lower probability of x_i, thereby making fully imprecise lower previsions those for which the lower probability of all elements in the possibility space is zero. We will use $\underline{\underline{\mathbb{P}}}(\mathscr{X})$ to denote the set of all such fully imprecise lower previsions. The reason why we call them fully imprecise is because they differ most from the precise, linear previsions. This distinction is already apparent from the following Proposition, but will become even clearer in Section 5.1, where we prove that every coherent lower prevision that is neither linear nor fully imprecise can be uniquely decomposed into a linear and a fully imprecise part.

Proposition 1. *$\mathbb{P}(\mathscr{X})$ and $\underline{\underline{\mathbb{P}}}(\mathscr{X})$ are disjoint subsets of $\underline{\mathbb{P}}(\mathscr{X})$: linear previsions are never fully imprecise.*

We refer to coherent lower previsions in $\underline{\mathbb{P}}(\mathscr{X})$ that are neither fully imprecise nor linear previsions as *partially imprecise*, and we denote by $\underline{\tilde{\mathbb{P}}}(\mathscr{X})$ the set of all partially imprecise lower previsions. The next corollary is a direct consequence of Proposition 1.

Corollary 1. *$\mathbb{P}(\mathscr{X})$, $\underline{\underline{\mathbb{P}}}(\mathscr{X})$ and $\underline{\tilde{\mathbb{P}}}(\mathscr{X})$ constitute a partition of $\underline{\mathbb{P}}(\mathscr{X})$.*

3 Credal Sets

Linear previsions are not the only coherent lower previsions that can be characterised by means of mass functions in $\Sigma_{\mathscr{X}}$. It is well known [9, Section 3.6] that every coherent lower prevision can be uniquely characterised by a so-called *credal set*, which is a closed (and therefore compact[3]) convex subset of $\Sigma_{\mathscr{X}}$. We denote a generic credal set by $\mathscr{M}$ and use $\underline{\mathbb{M}}(\mathscr{X})$ to denote the set of all of them. For any $\underline{P} \in \underline{\mathbb{P}}(\mathscr{X})$, its corresponding credal set $\mathscr{M}_{\underline{P}}$ is the set of all mass functions that define a dominating linear prevision:

$$\mathscr{M}_{\underline{P}} := \left\{ p \in \mathscr{M} : P_p(f) \geq \underline{P}(f) \text{ for all } f \in \mathscr{G}(\mathscr{X}) \right\}. \tag{2}$$

[3] Since we only consider finite possibility spaces $\mathscr{X}$, we can use the Euclidean topology instead of the weak*-topology that is usually adopted for credal sets.

The original lower prevision $\underline{P}$ and its conjugate upper prevision $\overline{P}$ can be derived from the credal set $\mathscr{M}_{\underline{P}}$: for all $f \in \mathscr{G}(\mathscr{X})$

$$\underline{P}(f) = \min\{P_p(f)\colon p \in \mathscr{M}_{\underline{P}}\} \text{ and } \overline{P}(f) = \max\{P_p(f)\colon p \in \mathscr{M}_{\underline{P}}\}. \tag{3}$$

We can use this equation to justify our earlier statement in Section 2.2 that for all $i \in \mathbb{N}_{\leq n}$, we can interpret $\underline{P}(\mathbb{I}_{\{x_i\}})$ as the lower probability of x_i. Indeed, we find that

$$\underline{P}(\mathbb{I}_{\{x_i\}}) = \min\{P_p(\mathbb{I}_{\{x_i\}})\colon p \in \mathscr{M}_{\underline{P}}\} = \min\{p(x_i)\colon p \in \mathscr{M}_{\underline{P}}\} \tag{4}$$

is the smallest probability of x_i corresponding with the mass functions in $\mathscr{M}_{\underline{P}}$.

Credal sets are therefore in a one-to-one correspondence with coherent lower previsions, allowing us to think of a coherent lower prevision as a closed and convex set of mass functions instead of as an operator on gambles. This geometric approach will be useful in our search for extreme lower previsions, since it will enable us to establish links with results already proved in fields other than coherent lower prevision theory.

3.1 Extreme Credal Sets

Similarly to what we have done in Section 2.1 for coherent lower previsions, we can also take convex combinations of credal sets. Consider two credal sets $\mathscr{M}_1$ and $\mathscr{M}_2$ in $\underline{\mathbb{M}}(\mathscr{X})$ and any $\lambda \in [0,1]$. Then the set $\mathscr{M} := \lambda\mathscr{M}_1 + (1-\lambda)\mathscr{M}_2$, given by

$$\mathscr{M} := \{\lambda p_1 + (1-\lambda)p_2\colon p_1 \in \mathscr{M}_1 \text{ and } p_2 \in \mathscr{M}_2\}, \tag{5}$$

will again be a credal set in $\underline{\mathbb{M}}(\mathscr{X})$. Due to the equivalence between credal sets and coherent lower previsions, the following proposition should not cause any surprise.

Proposition 2. *Consider coherent lower previsions $\underline{P}$, $\underline{P}_1$ and $\underline{P}_2$ in $\underline{\mathbb{P}}(\mathscr{X})$ and their corresponding credal sets $\mathscr{M}_{\underline{P}}$, $\mathscr{M}_{\underline{P}_1}$ and $\mathscr{M}_{\underline{P}_2}$ in $\underline{\mathbb{M}}(\mathscr{X})$. Then for all $\lambda \in [0,1]$:*

$$\underline{P} = \lambda\underline{P}_1 + (1-\lambda)\underline{P}_2 \Leftrightarrow \mathscr{M}_{\underline{P}} = \lambda\mathscr{M}_{\underline{P}_1} + (1-\lambda)\mathscr{M}_{\underline{P}_2}. \tag{6}$$

We now define an *extreme credal set* as a credal set $\mathscr{M} \in \underline{\mathbb{M}}(\mathscr{X})$ that cannot be written as a convex combination of two other credal sets $\mathscr{M}_1$ and $\mathscr{M}_2$ other than in a trivial way, trivial meaning that $\lambda = 0$, $\lambda = 1$ or $\mathscr{M}_1 = \mathscr{M}_2 = \mathscr{M}$. We will denote the set of all such extreme credal sets as $\operatorname{ext}\underline{\mathbb{M}}(\mathscr{X})$. The following immediate corollary of Proposition 2 shows that they are in a one-to-one correspondence with extreme lower previsions.

Corollary 2. *A coherent lower prevision is extreme iff its credal set is. For all $\underline{P} \in \underline{\mathbb{P}}(\mathscr{X})$:*

$$\underline{P} \in \operatorname{ext}\underline{\mathbb{P}}(\mathscr{X}) \Leftrightarrow \mathscr{M}_{\underline{P}} \in \operatorname{ext}\underline{\mathbb{M}}(\mathscr{X}). \tag{7}$$

3.2 Special Kinds of Credal Sets

Because of the one-to-one correspondence between coherent lower previsions and credal sets, the special subsets of $\underline{\mathbb{P}}(\mathscr{X})$ that were introduced in Section 2.2 immediately lead to corresponding subsets of $\underline{\mathbb{M}}(\mathscr{X})$. The set

$$\mathbb{M}(\mathscr{X}) := \{\mathscr{M}_P\colon P \in \mathbb{P}(\mathscr{X})\} = \{\{p\}\colon p \in \Sigma_{\mathscr{X}}\} \tag{8}$$

of credal sets that correspond to linear previsions in $\mathbb{P}(\mathscr{X})$ is the easiest one.

Another subset of $\underline{\mathbb{M}}(\mathscr{X})$, which will become very important further on, contains those credal sets that correspond to fully imprecise coherent lower previsions:

$$\underline{\underline{\mathbb{M}}}(\mathscr{X}) := \{\mathscr{M}_{\underline{P}} \colon \underline{P} \in \underline{\underline{\mathbb{P}}}(\mathscr{X})\} \tag{9}$$

$$= \{\mathscr{M} \in \underline{\mathbb{M}}(\mathscr{X}) \colon \min\{p(x_i) \colon p \in \mathscr{M}\} = 0 \text{ for all } i \in \mathbb{N}_{\leq n}\}, \tag{10}$$

where the second equality is a consequence of Eq. (4) and the definition of fully imprecise lower previsions. It should also clarify our statement in Section 2.2 that for fully imprecise lower previsions the lower probability of all elements of the possibility space is zero. We refer to elements of $\underline{\underline{\mathbb{M}}}(\mathscr{X})$ as *fully imprecise credal sets*.

The final subset of $\underline{\mathbb{M}}(\mathscr{X})$ that we need to consider contains the *partially imprecise credal sets*, corresponding to partially imprecise lower previsions in $\underline{\mathbb{P}}(\mathscr{X})$:

$$\underset{\sim}{\mathbb{M}}(\mathscr{X}) := \{\mathscr{M}_{\underline{P}} \colon \underline{P} \in \underline{\mathbb{P}}(\mathscr{X})\} = \underline{\mathbb{M}}(\mathscr{X}) \setminus \{\mathbb{M}(\mathscr{X}) \cup \underline{\underline{\mathbb{M}}}(\mathscr{X})\}. \tag{11}$$

Finally, the following result is a direct consequence of Corollary 1.

Corollary 3. *$\mathbb{M}(\mathscr{X})$, $\underline{\underline{\mathbb{M}}}(\mathscr{X})$ and $\underset{\sim}{\mathbb{M}}(\mathscr{X})$ constitute a partition of $\underline{\mathbb{M}}(\mathscr{X})$.*

3.3 Projected Credal Sets

Mass functions on the possibility space $\mathscr{X} = \{x_1, \ldots, x_n\}$ are uniquely characterised by the probability of the first $n-1$ elements because the final probability follows from the requirement that $\sum_{i=1}^{n} p(x_i) = 1$. This leads us to identify a mass function p on $\mathscr{X}$ with a point v_p in $\mathbb{R}^{n-1}$, defined by $(v_p)_i := p(x_i)$ for all $i \in \mathbb{N}_{<n}$. Similarly, a credal set $\mathscr{M}$ can be identified with a subset of $\mathbb{R}^{n-1}$ by letting

$$K_{\mathscr{M}} := \{v_p \colon p \in \mathscr{M}\}. \tag{12}$$

We call $K_{\mathscr{M}}$ the *projected credal set* of $\mathscr{M}$. We will use $K_{\underline{P}}$ as a shorthand notation for $K_{\mathscr{M}_{\underline{P}}}$ and call it the projected credal set of $\underline{P}$. For all $\mathscr{M} \in \underline{\mathbb{M}}(\mathscr{X})$, $K_{\mathscr{M}}$ is a closed and convex subset of the so-called *projected $\mathscr{X}$-simplex*

$$\mathbf{K}_{\mathscr{X}} = \left\{ v \in \mathbb{R}^{n-1} \colon \sum_{i=1}^{n-1} v_i \leq 1 \text{ and } v_i \geq 0 \text{ for all } i \in \mathbb{N}_{<n} \right\}, \tag{13}$$

which is a compact, closed and convex subset of $\mathbb{R}^{n-1}$. The set of all closed (and therefore compact) convex subsets of $\mathbf{K}_{\mathscr{X}}$ is denoted by $\underline{\mathbb{K}}(\mathscr{X})$. To show that both representations are indeed equivalent, let us define for every point $v \in \mathbf{K}_{\mathscr{X}}$ a corresponding mass function p_v on $\mathscr{X}$, defined by $p_v(x_i) := v_i$ for all $i \in \mathbb{N}_{<n}$ and $p_v(x_n) := 1 - \sum_{i=1}^{n-1} v_i$. It should be clear that $v_{p_v} = v$ and $p_{v_p} = p$, whence the equivalence. Similarly, we can define for all $K \in \mathbf{K}_{\mathscr{X}}$ a corresponding credal set

$$\mathscr{M}_K := \{p_v \colon v \in K\}. \tag{14}$$

Again, we have that $K_{\mathscr{M}_K} = K$ and $\mathscr{M}_{K_{\mathscr{M}}} = \mathscr{M}$. Finally, the following intuitive result shows that projecting credal sets on $\mathbf{K}_{\mathscr{X}}$ preserves convex combinations.

Proposition 3. *Consider credal sets $\mathscr{M}$, $\mathscr{M}_1$ and $\mathscr{M}_2$ in $\underline{\mathbb{M}}(\mathscr{X})$ and their corresponding projected credal sets $K_{\mathscr{M}}$, $K_{\mathscr{M}_1}$ and $K_{\mathscr{M}_2}$ in $\underline{\mathbb{K}}(\mathscr{X})$. Then for all $\lambda \in [0,1]$:*

$$\mathscr{M} = \lambda \mathscr{M}_1 + (1-\lambda)\mathscr{M}_2 \Leftrightarrow K_{\mathscr{M}} = \lambda K_{\mathscr{M}_1} + (1-\lambda) K_{\mathscr{M}_2}. \tag{15}$$

3.4 Special Kinds of Projected Credal Sets

Due the equivalence between credal sets and their projected versions, we can use the partition of $\underline{\mathbb{M}}(\mathscr{X})$ in Corollary 3 to construct a similar partition of $\underline{\mathbb{K}}(\mathscr{X})$. The first set in that partition corresponds to the credal sets of linear previsions and is equal to

$$\mathbb{K}(\mathscr{X}) := \{K_{\mathscr{M}} : \mathscr{M} \in \mathbb{M}(\mathscr{X})\} = \{K \in \underline{\mathbb{K}}(\mathscr{X}) : K = \{v\}, \text{ with } v \in \mathbf{K}_{\mathscr{X}}\}. \tag{16}$$

The second set consists of the projections of the credal sets in $\underline{\underline{\mathbb{M}}}(\mathscr{X})$:

$$\underline{\underline{\mathbb{K}}}(\mathscr{X}) := \{K_{\mathscr{M}} : \mathscr{M} \in \underline{\underline{\mathbb{M}}}(\mathscr{X})\} \tag{17}$$

$$= \left\{K \in \underline{\mathbb{K}}(\mathscr{X}) : \min_{v \in K} v_i = 0 \text{ for all } i \in \mathbb{N}_{<n} \text{ and } \max_{v \in K} \sum_{i=1}^{n-1} v_i = 1\right\}. \tag{18}$$

The final set contains the projected credal sets of partially imprecise lower previsions:

$$\underline{\mathbb{K}}(\mathscr{X}) := \{K_{\mathscr{M}} : \mathscr{M} \in \underline{\mathbb{M}}(\mathscr{X})\} = \underline{\mathbb{K}}(\mathscr{X}) \setminus \{\mathbb{K}(\mathscr{X}) \cup \underline{\underline{\mathbb{K}}}(\mathscr{X})\}. \tag{19}$$

4 Minkowski Decomposition

Given two compact convex subsets A_1 and A_2 of $\mathbb{R}^{n-1}$, their *Minkowski sum* or *vector sum* is given by $A_1 + A_2 := \{a_1 + a_2 : a_1 \in A_1 \text{ and } a_2 \in A_2\}$. They are called *homothetic* if $A_1 = v + \lambda A_2 := \{v + \lambda a_2 : a_2 \in A_2\}$ for some $\lambda > 0$ and $v \in \mathbb{R}^{n-1}$. If $A = A_1 + A_2$, with A, A_1 and A_2 compact convex subsets of $\mathbb{R}^{n-1}$, then A_1 and A_2 are called *summands* of A. We say that A is written as a Minkowski sum in a non-trivial way, if neither of its summands is homothetic to A or a singleton. If such a non-trivial decomposition exists, we say that A is *Minkowski decomposable*. Otherwise, A is called *Minkowski indecomposable*. Sections 6.2 and 6.3 point to relevant literature, where, incidentally, the prefix "Minkowski" is not always used. We add it in the present paper to avoid confusion with the decomposition of credal sets and lower previsions.

4.1 Connecting Both Theories

One of the main contributions of this paper will be to show how the extensive literature on Minkowski decomposition of convex sets can be related to the search for extreme lower previsions in imprecise probability theory. The results in this section take the first step towards doing so, and will turn out to be crucial for our main theorem further on.

We start by associating with any compact set $A \subseteq \mathbb{R}^{n-1}$ a point $m(A) \in \mathbb{R}^{n-1}$, defined by $m_i(A) := \min\{v_i : v \in A\}$ for all $i \in \mathbb{N}_{<n}$ and a real number $\mu(A)$, given by

$$\mu(A) := \max\left\{\sum_{i=1}^{n-1} v_i : v \in A\right\} - \sum_{i=1}^{n-1} m_i(A). \tag{20}$$

Both $m(A)$ and $\mu(A)$ are well-defined due to the compactness of A. If A is not a singleton, then it is easy to see that $\mu(A) > 0$ and we can define

$$\underline{\underline{A}} := \frac{1}{\mu(A)}\big(A - m(A)\big) = \left\{\frac{1}{\mu(A)}(v - m(A)) : v \in A\right\}. \tag{21}$$

Proposition 4. *For any compact convex subset A of $\mathbb{R}^{n-1}$ that is not a singleton, the corresponding set $\underline{\underline{A}}$ is an element of $\underline{\underline{\mathbb{K}}}(\mathscr{X})$.*

Proposition 5. *A compact convex subset A of $\mathbb{R}^{n-1}$ that is not a singleton is Minkowski decomposable iff the corresponding set $\underline{\underline{A}}$ is Minkowski decomposable.*

The following result shows how the transformation that we have just introduced can be usefully exploited to reformulate the property of Minkowski decomposability.

Theorem 1. *A compact convex subset A of $\mathbb{R}^{n-1}$ that is not a singleton is Minkowski decomposable iff its corresponding set $\underline{\underline{A}}$ can be written as a non-trivial convex combination $\lambda K_1 + (1-\lambda)K_2$, with K_1 and K_2 both elements of $\underline{\underline{\mathbb{K}}}(\mathscr{X})$, $K_1 \neq K_2$ and $0 < \lambda < 1$.*

5 Characterising Extreme Lower Previsions

We now have all the tools needed to characterise the set $\operatorname{ext}\underline{\mathbb{P}}(\mathscr{X})$ of all extreme lower previsions on $\mathscr{G}(\mathscr{X})$, or equivalently, the set $\operatorname{ext}\underline{\mathbb{M}}(\mathscr{X})$ of all extreme credal sets. We will show that partially imprecise lower previsions are never extreme as they can be split up in a linear and a fully imprecise part. The only extreme linear previsions are the degenerate ones, and the extreme fully imprecise models will turn out to be closely related to the Minkowski indecomposable convex compact sets of Section 4.

5.1 Partially Imprecise Lower Previsions

We claimed earlier on in Section 2.2 that every partially imprecise lower prevision can be uniquely decomposed in a linear and a fully imprecise part. To see why this is true, first consider the following proposition, which is the counterpart of that statement in the language of credal sets. The desired property is then a direct consequence of this result.

Proposition 6. *Any partially imprecise credal set $\mathscr{M} \in \underset{\sim}{\mathbb{M}}(\mathscr{X})$ can be uniquely written as a convex combination $\lambda\mathscr{M}_1 + (1-\lambda)\mathscr{M}_2$ of a credal set $\mathscr{M}_1 \in \mathbb{M}(\mathscr{X})$ that contains only a single mass function $p_1 \in \Sigma_{\mathscr{X}}$ and a fully imprecise credal set $\mathscr{M}_2 \in \underline{\underline{\mathbb{M}}}(\mathscr{X})$. Moreover, $0 < \lambda := \sum_{i=1}^{n} \min\{p(x_i) \colon p \in \mathscr{M}\} < 1$, the mass function p_1 that characterises $\mathscr{M}_1$ is given by $p_1(x_i) = \frac{1}{\lambda}\min\{p(x_i) : p \in \mathscr{M}\}$ for all $i \in \mathbb{N}_{\leq n}$, and*

$$\mathscr{M}_2 = \left\{ \frac{1}{1-\lambda}p - \frac{\lambda}{1-\lambda}p_1 \colon p \in \mathscr{M} \right\}. \tag{22}$$

Corollary 4. *Any partially imprecise lower prevision $\underline{P} \in \underset{\sim}{\mathbb{P}}(\mathscr{X})$ can be uniquely written as a convex combination $\lambda P_1 + (1-\lambda)\underline{P}_2$ of a linear prevision $P_1 \in \mathbb{P}(\mathscr{X})$ and a fully imprecise lower prevision $\underline{P}_2 \in \underline{\underline{\mathbb{P}}}(\mathscr{X})$. Moreover, $0 < \lambda := \sum_{i=1}^{n} \underline{P}(\mathbb{I}_{\{x_i\}}) < 1$ and*

$$P_1(f) = \frac{1}{\lambda}\sum_{i=1}^{n} f(x_i)\underline{P}(\mathbb{I}_{\{x_i\}}) \text{ and } \underline{P}_2(f) = \frac{1}{1-\lambda}\underline{P}(f) - \frac{\lambda}{1-\lambda}P_1(f) \text{ for all } f \in \mathscr{G}(\mathscr{X}). \tag{23}$$

The fact that partially imprecise models can be decomposed in this way has some immediate important consequences for extreme credal sets and lower previsions.

Corollary 5. *Extreme credal sets and lower previsions are never partially imprecise:*

$$\mathscr{M} \in \underset{\sim}{\mathbb{M}}(\mathscr{X}) \Rightarrow \mathscr{M} \notin \operatorname{ext}\underline{\mathbb{M}}(\mathscr{X}) \text{ and } \underline{P} \in \underset{\sim}{\mathbb{P}}(\mathscr{X}) \Rightarrow \underline{P} \notin \operatorname{ext}\underline{\mathbb{P}}(\mathscr{X}). \tag{24}$$

In our search for extreme lower previsions, we therefore only need to look at the subsets of the linear previsions and of the fully imprecise lower previsions.

5.2 Linear Previsions

A special class of linear previsions are those that correspond to degenerate mass functions. For every $i \in \mathbb{N}_{\leq n}$, the corresponding *degenerate mass function* $p_i^\circ \in \Sigma_{\mathscr{X}}$ has all its probability mass in x_i and is therefore defined by $p_i^\circ := \mathbb{I}_{\{x_i\}}$. They satisfy the following important property.

Proposition 7. *A credal set $\mathscr{M} \in \mathbb{M}(\mathscr{X})$ containing only a single mass function is extreme iff that single mass function is degenerate. Furthermore, any other mass function can be written as a convex combination of those degenerate ones.*

The linear previsions that correspond to such a degenerate mass function are called *degenerate linear previsions*. For every $i \in \mathbb{N}_{\leq n}$, we have a corresponding degenerate linear prevision P_i°, defined for all $f \in \mathscr{G}(\mathscr{X})$ by $P_i^\circ(f) := f(x_i)$. As a direct consequence of Proposition 7, we find that these degenerate linear previsions are the only linear previsions that are extreme.

Corollary 6. *A linear prevision $P \in \underline{\mathbb{P}}(\mathscr{X})$ is extreme iff it is degenerate. Furthermore, any other linear prevision can be written as a convex combination of degenerate ones.*

For coherent lower previsions that are defined on a finite domain $\mathscr{K} \subset \mathscr{G}(\mathscr{X})$, a result that combines Corollary 5 and 6 was already mentioned in Ref. [5, Proposition 1].

5.3 Fully Imprecise Lower Previsions

So far, we have shown that partially imprecise models are never extreme and that the extreme linear models are those that are degenerate. The only models that are thus left to investigate are those that are fully imprecise. We start with a property of decompositions of fully imprecise credal sets.

Proposition 8. *If a fully imprecise credal set $\mathscr{M} \in \underline{\underline{\mathbb{M}}}(\mathscr{X})$ can be written as a non-trivial convex combination $\lambda\mathscr{M}_1 + (1-\lambda)\mathscr{M}_2$, with $\mathscr{M}_1, \mathscr{M}_2 \in \underline{\mathbb{M}}(\mathscr{X})$, $\mathscr{M}_1 \neq \mathscr{M}_2$ and $0 < \lambda < 1$, then $\mathscr{M}_1$ and $\mathscr{M}_2$ are both fully imprecise and therefore elements of $\underline{\underline{\mathbb{M}}}(\mathscr{X})$.*

In the language of coherent lower previsions, this turns into the following corollary.

Corollary 7. *If a fully imprecise coherent lower prevision $\underline{P} \in \underline{\underline{\mathbb{P}}}(\mathscr{X})$ can be written as a non-trivial convex combination $\lambda\underline{P}_1 + (1-\lambda)\underline{P}_2$, with $\underline{P}_1, \underline{P}_2 \in \underline{\mathbb{P}}(\mathscr{X})$, $\underline{P}_1 \neq \underline{P}_2$ and $0 < \lambda < 1$, then $\underline{P}_1$ and $\underline{P}_2$ are both fully imprecise and therefore elements of $\underline{\underline{\mathbb{P}}}(\mathscr{X})$.*

Combined with Proposition 3 and Theorem 1, Proposition 8 leads to a crucial result.

Theorem 2. *A fully imprecise credal set $\mathscr{M} \in \underline{\underline{\mathbb{M}}}(\mathscr{X})$ can be written as a non-trivial convex combination $\lambda\mathscr{M}_1 + (1-\lambda)\mathscr{M}_2$, with $\mathscr{M}_1, \mathscr{M}_2 \in \underline{\underline{\mathbb{M}}}(\mathscr{X})$, $\mathscr{M}_1 \neq \mathscr{M}_2$ and $0 < \lambda < 1$ iff its projected credal set $K_{\mathscr{M}}$ is Minkowski decomposable.*

When stated in terms of coherent lower previsions, this result looks as follows.

Corollary 8. *A fully imprecise coherent lower prevision $\underline{P} \in \underline{\underline{\mathbb{P}}}(\mathscr{X})$ can be written as a non-trivial convex combination $\lambda\underline{P}_1 + (1-\lambda)\underline{P}_2$, with $\underline{P}_1, \underline{P}_2 \in \underline{\underline{\mathbb{P}}}(\mathscr{X})$, $\underline{P}_1 \neq \underline{P}_2$ and $0 < \lambda < 1$ iff its projected credal set $K_{\underline{P}}$ is Minkowski decomposable.*

The importance of these two results is that they provide us with an easy characterisation of the extreme models that are fully imprecise.

Corollary 9. *A fully imprecise credal set $\mathscr{M} \in \underline{\underline{\mathbb{M}}}(\mathscr{X})$ is extreme iff its projected credal set $K_{\mathscr{M}}$ is Minkowski indecomposable. Equivalently, a fully imprecise lower prevision $\underline{P} \in \underline{\underline{\mathbb{P}}}(\mathscr{X})$ is extreme iff its projected credal set $K_{\underline{P}}$ is Minkowski indecomposable.*

These alternative characterisations of fully imprecise extreme credal sets and lower previsions will allow us to import known results from the literature on Minkowski decomposability, using them to find the sets $\operatorname{ext}\underline{\underline{\mathbb{M}}}(\mathscr{X})$ and $\operatorname{ext}\underline{\underline{\mathbb{P}}}(\mathscr{X})$, containing all extreme credal sets and lower previsions respectively.

To conclude this section, we want to mention a very special fully imprecise credal set. It contains every single mass function in $\Sigma_{\mathscr{X}}$ and will be denoted as $\mathscr{M}_V := \Sigma_{\mathscr{X}}$. It is used to model complete ignorance and is called the *vacuous* credal set. The corresponding (fully imprecise) lower prevision $\underline{P}_V$ is referred to as the *vacuous* lower prevision and is given, for all $f \in \mathscr{G}(\mathscr{X})$, by $\underline{P}_V(f) = \min f$.

Proposition 9. *The vacuous credal set is extreme: $\mathscr{M}_V \in \operatorname{ext}\underline{\underline{\mathbb{M}}}(\mathscr{X})$.*

Corollary 10. *The vacuous lower prevision is extreme: $\underline{P}_V \in \operatorname{ext}\underline{\underline{\mathbb{P}}}(\mathscr{X})$.*

6 Finding All Extreme Lower Previsions

The size of $\operatorname{ext}\underline{\underline{\mathbb{M}}}(\mathscr{X})$ and $\operatorname{ext}\underline{\underline{\mathbb{P}}}(\mathscr{X})$ and the complexity of their elements, turns out to depend heavily on the number of elements in the possibility space $\mathscr{X} = \{x_1, \ldots, x_n\}$. We consider three distinct cases: $n = 2$, $n = 3$ and $n > 3$. We focus on constructing $\operatorname{ext}\underline{\underline{\mathbb{M}}}(\mathscr{X})$, since $\operatorname{ext}\underline{\underline{\mathbb{P}}}(\mathscr{X})$ can be derived from it by applying Corollary 2.

6.1 Possibility Spaces with Two States

For $n = 2$, constructing $\operatorname{ext}\underline{\underline{\mathbb{M}}}(\mathscr{X})$ is almost trivial. Nevertheless, it serves as a good didactic exercise to get to know the basic tools in this paper.

It follows from the results in Section 5 that in our search for the extreme credal sets, we do not need to consider the partially imprecise ones. It suffices to look at the precise and the fully imprecise credal sets. We know from Proposition 7 that of all the precise credal sets (those consisting of only a single mass function) the only extreme ones are

those that correspond to a degenerate mass function. In the current binary case, with $\mathscr{X} = \{x_1, x_2\}$, this yields the extreme credal sets $\mathscr{M}_1^\circ := \{p_1^\circ\}$ and $\mathscr{M}_2^\circ := \{p_2^\circ\}$. All other extreme credal sets will be fully imprecise. We know from Proposition 9 that $\mathscr{M}_V$ is one of those fully imprecise extreme credal sets, but finding the other ones would normally require the use of Corollary 9. However, in this simple binary case, $\mathscr{M}_V$ is the only fully imprecise credal set (we leave the simple proof of this statement as an exercise for the reader) and we can therefore conclude that for binary possibility spaces:

$$\operatorname{ext}\underline{\mathbb{M}}(\mathscr{X}) = \{\mathscr{M}_1^\circ, \mathscr{M}_2^\circ, \mathscr{M}_V\}. \tag{25}$$

By applying Corollary 2, we obtain the corresponding result for lower previsions:

$$\operatorname{ext}\underline{\mathbb{P}}(\mathscr{X}) = \{P_1^\circ, P_2^\circ, \underline{P}_V\}. \tag{26}$$

6.2 Possibility Spaces with Three States

For $n = 3$, finding $\operatorname{ext}\underline{\mathbb{M}}(\mathscr{X})$ becomes a bit more involved. As always, the partially imprecise credal sets are never extreme and the only precise extreme credal sets are the degenerate ones. Finding the fully imprecise credal sets that are extreme is however more difficult then it was in the binary case. Here, the vacuous credal set $\mathscr{M}_V$ will not be the only fully imprecise extreme credal set. In order to find the others, we rely on Corollary 9, using it to import the following result by Silverman into our framework.

Theorem 3 ([8, Theorem 4]). *A compact convex subset of $\mathbb{R}^2$ is Minkowski indecomposable if and only if it is a triangle or a line segment.*

This theorem is highly non-trivial since it holds for general compact convex subsets of $\mathbb{R}^2$ and not only for convex polygons. It allows us to derive the next result, which concludes our search for the extreme credal sets of ternary possibility spaces.

Corollary 11. *For possibility spaces $\mathscr{X} = \{x_1, x_2, x_3\}$ containing only three elements, a fully imprecise credal set $\mathscr{M} \in \underline{\mathbb{M}}(\mathscr{X})$ is extreme if and only if it is the convex closure of three probability mass functions: we can find $p_1, p_2, p_3 \in \Sigma_{\mathscr{X}}$ such that*

$$\mathscr{M} = \left\{ \sum_{i=1}^{3} \lambda_i p_i \colon (\lambda_1, \lambda_2, \lambda_3) \in \Sigma_{\mathscr{X}} \right\}. \tag{27}$$

Figure 1 should provide this result with some intuition. It presents an example of a fully imprecise credal set with four vertices and its decomposition into two extreme ones with three vertices. We depict the credal sets using the well-known simplex representation [9, Section 4.2.3].

In order to obtain the extreme lower previsions of a ternary possibility space, all we need to do now is apply Corollary 2. We find that apart from the three degenerate linear previsions P_1°, P_2° and P_3°, all other extreme lower previsions are characterised by the following translation of Corollary 11.

Corollary 12. *For possibility spaces $\mathscr{X} = \{x_1, x_2, x_3\}$ containing only three elements, a fully imprecise lower prevision $\underline{P} \in \underline{\mathbb{M}}(\mathscr{X})$ is extreme if and only if it is the lower envelope of three linear previsions: one can find $P_1, P_2, P_3 \in \mathbb{P}(\mathscr{X})$ such that*

$$\underline{P}(f) = \min_{i \in \mathbb{N}_{\leq 3}} P_i(f) \text{ for all } f \in \mathscr{G}(\mathscr{X}). \tag{28}$$

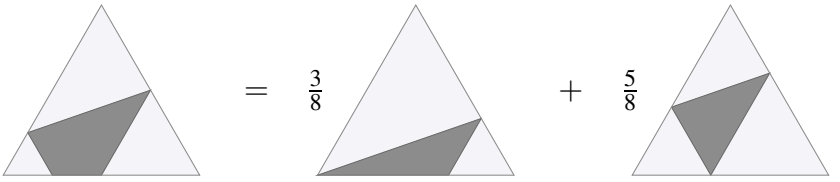


Fig. 1. Decomposition of a fully imprecise credal set into two extreme ones

6.3 General Possibility Spaces

Due to the page limit constraint, we are not able to discuss the case $n > 3$ in full detail. In contrast to the cases $n = 2$ and $n = 3$, we will not construct the set of all extreme credal sets. It should however be clear that all extreme credal sets will again be fully imprecise, except for the degenerate precise ones. We restrict ourselves to stating some relevant results from the theory of Minkowski decomposability. Their implications for extreme credal sets (and thus also extreme lower previsions) are fairly intuitive, but we defer any more formal result to future work.

We know from Corollary 9 that fully imprecise extreme credal sets correspond to Minkowski indecomposable compact and convex subsets of $\mathbb{R}^{n-1}$. For $n = 3$, we were dealing with Minkowski indecomposability in the plane, which is completely determined by Theorem 3. In higher dimensions, Minkowski indecomposability is not yet fully resolved in the literature.

Most known results deal only with polytopes. Grünbaum [1, Chapter 15] provides a good summary, explaining (amongst other interesting results) why every simplicial polytope is indecomposable and every simple polytope, with the exception of a simplex, is decomposable. Meyer [3, Theorem 3] provides two rather complicated algebraic conditions, which are both necessary and sufficient for a polytope to be indecomposable.

For non-polytopes, the most important reference seems to be Ref. [7], in which Sallee shows that a wide class of compact convex sets is decomposable, the only condition being that they have on their boundary a sufficiently smooth neighbourhood. However, unlike in the case of $\mathbb{R}^2$, in higher dimensions Minkowski indecomposable compact convex sets need not be polytopes.

7 Conclusions

We have shown that when $\mathscr{X}$ has a finite number n of elements, then the extreme coherent lower previsions on $\mathscr{G}(\mathscr{X})$ are either degenerate linear previsions or fully imprecise and in a one-to-one correspondence with (Minkowski) indecomposable compact convex subsets of $\mathbb{R}^{n-1}$. Using this connection, we have constructed the set of all extreme lower previsions for the cases $n = 2$ and $n = 3$ and suggested what these sets might look like for $n > 3$. For the case $n = 3$, we have found that a fully imprecise coherent lower prevision is extreme if and only if it is the lower envelope of three linear previsions.

A first and rather obvious avenue of future research would be to use the results mentioned in Section 6.3 to try and construct $\operatorname{ext}\underline{\mathbb{M}}(\mathscr{X})$ and $\operatorname{ext}\underline{\mathbb{P}}(\mathscr{X})$ if $n > 3$, or to at least get a better idea of what kind of elements they contain. Consider for example the case

$n = 4$. Can one find non-degenerate extreme lower previsions that are not the lower envelope of four linear ones? And are fully imprecise lower previsions that are the lower envelope of four linear previsions always extreme? We intend to answer these questions in an extended journal version of this paper.

It would also be interesting to compare our results with those in Ref. [5], which concentrated on coherent lower previsions defined on finite domains, and Ref. [6], which investigated the even more particular case of extreme lower probabilities. We conjecture that our results subsume (at least some of) those obtained in Refs. [5] and [6], but a detailed study is beyond the scope of this conference paper. Ref. [6] also looked at the extreme points of sets formed by all lower probabilities that satisfy certain properties, such as k-monotonicity and permutation invariance. We suspect that our results can be adapted to conduct a similar study for extreme coherent lower previsions as well.

Finally, we would like to see to what extent extreme lower previsions can be used to tackle practical problems. One idea would be to adapt the existing algorithms for Minkowski decomposition to decompose coherent lower previsions into convex combinations of extreme ones. Such decompositions can then be used to approximate coherent lower previsions in such a way as to satisfy certain properties or to develop a generalisation of the so-called random set product from the theory of belief functions.

Acknowledgements. Jasper De Bock is a Ph.D. Fellow of the Fund for Scientific Research – Flanders (FWO) and wishes to acknowledge its financial support. The authors also wish to thank three anonymous referees for their helpful comments.

References

1. Grünbaum, B.: Convex polytopes, 2nd edn. Springer (2003); prepared by Kaibel, V., Klee, V., Ziegler, G.M. (eds.)
2. Maaß, S.: Exact functionals, functionals preserving linear inequalities, Lévy's metric. Ph.D. thesis, Universität Bremen (2003)
3. Meyer, W.: Indecomposable polytopes. Transactions of the American Mathematical Society 190, 77–86 (1974)
4. Miranda, E.: A survey of the theory of coherent lower previsions. International Journal of Approximate Reasoning 48(2), 628–658 (2008)
5. Quaeghebeur, E.: Characterizing the set of coherent lower previsions with a finite number of constraints or vertices. In: Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence, pp. 466–473 (2010)
6. Quaeghebeur, E., De Cooman, G.: Extreme lower probabilities. Fuzzy Sets and Systems 159, 2163–2175 (2008)
7. Sallee, G.T.: Minkowski decomposition of convex sets. Israel Journal of Mathematics 12, 266–276 (1972)
8. Silverman, R.: Decomposition of plane convex sets, part I. Pacific Journal of Mathematics 47, 521–530 (1973)
9. Walley, P.: Statistical Reasoning with Imprecise Probabilities. Chapman and Hall, London (1991)

Qualitative Capacities as Imprecise Possibilities

Didier Dubois[1], Henri Prade[1], and Agnès Rico[2]

[1] IRIT, Université Paul Sabatier
118 route de Narbonne, 31062 Toulouse cedex 9, France
[2] ERIC, Université Claude Bernard Lyon 1
43 bld du 11 novembre 69100 Villeurbanne, France

Abstract. This paper studies the structure of qualitative capacities, that is, monotonic set-functions, when they range on a finite totally ordered scale equipped with an order-reversing map. These set-functions correspond to general representations of uncertainty, as well as importance levels of groups of criteria in multi-criteria decision-making. More specifically, we investigate the question whether these qualitative set-functions can be viewed as classes of simpler set-functions, typically possibility measures, paralleling the situation of quantitative capacities with respect to imprecise probability theory. We show that any capacity is characterized by a non-empty class of possibility measures having the structure of an upper semi-lattice. The lower bounds of this class are enough to reconstruct the capacity, and their number is characteristic of its complexity. We introduce a sequence of axioms generalizing the maxitivity property of possibility measures, and related to the number of possibility measures needed for this reconstruction. In the Boolean case, capacities are closely related to non-regular multi-source modal logics and their neighborhood semantics can be described in terms of qualitative Moebius transforms.

1 Introduction

A fuzzy measure (or a capacity) is a set-function that is monotonic under inclusion. If its range is a finite totally ordered scale, the capacity is said to be qualitative. Then, the connection with probability measures is lost as well, and a number of notions, meaningful in the quantitative setting, are lost, like the Möebius transform, the conjugate, nor can any qualitative capacity be viewed as encoding a family of probability distributions. Yet it seems that qualitative counterparts of many such quantitative notions can be defined if we replace probability measures by possibility measures. For instance the process of generation of belief functions, introduced by Dempster [6], was applied to possibility measures by Dubois and Prade [11,12] so as to define upper and lower possibilities and necessities. It was noticed that upper possibilities and lower necessities are still possibility and necessity measures respectively, but upper necessities and lower possibilities are not. This study was pursued by Tsiporkova and De Baets [21] in a more general setting. More recently in [18], it was shown that qualitative capacities can be viewed as counterparts of belief functions, using the possibilistic counterpart of a basic probability assignment. In [5] it was proved that the upper envelope of the possible extensions of a probability is a possibility measure.

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 169–180, 2013.

A natural question is then whether a qualitative capacity can be viewed as a family of possibility measures as in Walley's theory of imprecise probability [22]. A recent paper [7] addressed this issue, taking up a pioneering work by Banon [2]. It is shown that in the case of qualitative information, special subsets of possibility measures play a role similar to convex sets of probability measures. This should not come as a surprise. Indeed, it has been shown that possibility measures can be refined by probability measures using a lexicographic refinement of the basic axiom of possibility measures, and that capacities on a finite set can be refined by belief functions [9,10]. The aim of this paper is to show that the maxitivity and minitivity axiom of possibility theory can be generalized to define families of qualitative capacities of increasing complexity. This property enables qualitative capacities to be seen as necessity modalities in a non-regular class of modal logics, extending the links between possibility theory and modal logic.

2 Capacities as Imprecise Possibilities and Necessities

Consider a finite set S and a finite totally ordered scale L with top 1 and bottom 0. A capacity (or fuzzy measure) is a mapping $\gamma : 2^S \to L$ such that $\gamma(\emptyset) = 0$; $\gamma(S) = 1$; and if $A \subseteq B$ then $\gamma(A) \leq \gamma(B)$. A special case of capacity is a possibility measure. In possibility theory, the available information is represented by means of a possibility distribution. This is a function, usually denoted π, from the universe of discourse S to the scale L. The function π is supposed to rank-order potential values of (some aspect of) the state of the world - according to their plausibility. The value $\pi(s)$ is understood as the possibility that s be the actual state of the world. Precise information corresponds to the situation where $\exists s^*, \pi(s^*) = 1$, and $\forall s \neq s^*, \pi(s) = 0$, while complete ignorance is represented by the vacuous possibility distribution $\pi^?$ such that $\forall s \in S, \pi^?(s) = 1$. The *possibility measure* is defined by $\Pi(A) = \max_{s \in A} \pi(s)$.

A possibility distribution π is said to be more specific than another possibility distribution ρ if $\forall s \in S, \pi(s) \leq \rho(s)$. Denote by γ^c the conjugate of γ, defined as $\gamma^c(A) = \nu(\gamma(A^c)), \forall A \subseteq S$, where A^c is the complement of set A, and ν the order-reversing map on L. The conjugate of a possibility measure is called a necessity measure. The conjugate necessity measure is then of the form $N(A) = \nu(\max_{s \notin A} \pi(s)) = \min_{s \notin A} N(S \setminus \{s\})$.

It is well-known that in the numerical setting some capacities g can be equivalently represented by a convex set of probabilities of the form $\mathcal{P}(g) = \{P, P(A) \geq g(A), \forall A \subseteq S\}$. For instance, g can be a convex capacity ($g(A \cup B) \geq g(A) + g(B) - g(A \cap B)$) or a belief function. Then it holds that $g(A) = \min\{P(A) : P \in \mathcal{P}(g)\}$. This is one example of a coherent lower probability in the sense of Walley [22] (exact capacity after Schmeidler [20]). In the qualitative case this construction is impossible. The natural question is then whether a similar construction may make sense with qualitative possibility measures replacing probability measures.

2.1 Imprecise Possibility and Necessity

There is always at least one possibility measure that dominates any capacity: the vacuous possibility measure, based on the distribution $\pi^?$ expressing ignorance, since then

$\forall A \neq \emptyset \subset S, \Pi(A) = 1 \geq \gamma(A), \forall$ capacity γ, and $\Pi(\emptyset) = \gamma(\emptyset) = 0$. Let

$$\mathcal{R}(\gamma) = \{\pi : \Pi(A) \geq \gamma(A), \forall A \subseteq S\}$$

be the set of possibility distributions whose corresponding set-functions Π dominate γ. We call $\mathcal{R}(\gamma)$ the *possibilistic credal set* induced by the capacity γ. In this section we recall some results on the structure of this set of possibility distributions.

Let σ be a permutation of the $n = |S|$ elements in S. The ith element of the permutation is denoted by $s_{\sigma(i)}$. Moreover let $S^i_\sigma = \{s_{\sigma(i)}, \ldots, s_{\sigma(n)}\}$. Define the possibility distribution π^γ_σ as follows:

$$\forall i = 1 \ldots, n, \pi^\gamma_\sigma(s_{\sigma(i)}) = \gamma(S^i_\sigma) \tag{1}$$

There are at most $n!$ (number of permutations) such possibility distributions. It can be checked that the possibility measure Π^γ_σ induced by π^γ_σ lies in $\mathcal{R}(\gamma)$ and that the $n!$ such possibility distributions enable γ to be reconstructed (already in [2]):

Proposition 1. *For each permutation σ : $\forall A \subseteq S, \Pi^\gamma_\sigma(A) \geq \gamma(A)$.*
Moreover, $\forall A \subseteq S, \gamma(A) = \min_\sigma \Pi^\gamma_\sigma(A)$

As a consequence,

Proposition 2. *$\forall \pi \in \mathcal{R}(\gamma), \pi(s) \geq \pi^\gamma_\sigma(s), \forall s \in S$ for some permutation σ of S.*

Proof: Just consider a permutation σ induced by π, that is $\sigma(i) \geq \sigma(j) \iff \pi(s_i) \leq \pi(s_j)$. For this permutation, $\Pi(S^i_\sigma) = \pi(s_i) \geq \gamma(S^i_\sigma) = \pi^\gamma_\sigma(s_i), \forall i = 1, \ldots, n$. ■

This result says that the possibility distributions π^γ_σ (we call the *marginals* of γ) include the least elements of $\mathcal{R}(\gamma)$ in the sense of fuzzy set inclusion, i.e., the most specific possibility distributions dominating γ. In other terms, $\mathcal{R}(\gamma) = \{\pi, \exists \sigma, \pi \geq \pi^\gamma_\sigma\}$. Of course the maximal element of $\mathcal{R}(\gamma)$ is the vacuous possibility distribution $\pi^?$. In the qualitative case, $\mathcal{R}(\gamma)$ is closed under the qualitative counterpart of a convex combination: if $\pi_1, \pi_2 \in \mathcal{R}(\gamma)$, then $\forall \alpha, \beta \in L$, such that $\max(\alpha, \beta) = 1$, it holds that $\max(\min(\alpha, \pi_1), (\min(\beta, \pi_2)) \in \mathcal{R}(\gamma)$. In fact $\mathcal{R}(\gamma)$ is an upper semi-lattice. Not all the $n!$ possibility distributions π^γ_σ are least elements of $\mathcal{R}(\gamma)$. As a trivial example, if $\gamma = \Pi$, this least element is unique and is precisely π. But other permutations yield other less specific possibility distributions.

Conversely, for any set $\mathcal{T}$ of possibility distributions, the set-function $\gamma(A) = \min_{\pi \in \mathcal{T}} \Pi(A)$ is a capacity. It is easy to see that $\mathcal{T} \subseteq \mathcal{R}(\gamma)$ and that if $\mathcal{T}$ contains only possibility distributions that are not comparable with respect to specificity, $\mathcal{T}$ forms the most specific elements of $\mathcal{R}(\gamma)$. Note that the set-function $\gamma(A) = \max_{\pi \in \mathcal{T}} \Pi(A)$ is not only a capacity, but also a possibility measure with possibility distribution $\pi^{\max}(s) = \max_{\pi \in \mathcal{T}} \pi(s)$ [13].

We denote by $\mathcal{R}_*(\gamma)$ the set of minimal elements in $\mathcal{R}(\gamma)$. They are by construction a finite set of possibility distributions none of which is more specific that another. It is clear that the complexity of a qualitative capacity is clearly measured by the number of elements in $\mathcal{R}_*(\gamma)$. These findings also show that any capacity can be viewed as a lower possibility measure:

$$\gamma(A) = \min\{\Pi(A), \pi \in \mathcal{R}_*(\gamma)\}.$$

This is similar to the case of a convex capacity g understood as a lower probability with respect to a (probabilistic) credal set $\mathcal{P}(g)$ [22]. This probability set forms a convex polyhedron whose vertices are among probability assignments P_σ^γ of the form $p_\sigma^\gamma(s_{\sigma(i)}) = g(S_\sigma^i) - g(S_\sigma^{i+1})$, and $\mathcal{P}(g)$ is the convex hull of these probabilities.

Dually, though, we can describe capacities as upper necessities by means of a family of necessity functions that stem from the lower possibility description of their conjugates. Then we can define two sets of possibility functions from γ:

- The set $\mathcal{R}(\gamma)$ of possibility measures that dominate γ;
- The set $\mathcal{R}(\gamma^c)$ of possibility measures that dominate its conjugate γ^c.

Clearly, possibility measures that dominate γ^c are conjugates of necessity measures dominated by γ. In other words γ is also an upper necessity measure in the sense that

$$\gamma(A) = \max\{N(A), \pi \in \mathcal{R}_*(\gamma^c)\}.$$

We can denote the set of minimal possibility distributions generating maximal necessity measures dominated by γ by $\mathcal{R}^*(\gamma) = \mathcal{R}_*(\gamma^c)$. One representation of γ (by means of $\mathcal{R}_*(\gamma)$ or $\mathcal{R}_*(\gamma^c)$) may be simpler than the other. For instance, if γ is a necessity measure based on possibility distribution π, then $\mathcal{R}^*(\gamma) = \{\pi\}$ while $\mathcal{R}_*(\gamma)$ contains several possibility distributions including π. Note that $\Pi(A) \geq N(A) = \Pi^c(A)$, so that it looks more natural to reach N from below and Π from above. More generally if a capacity γ is such that $\gamma(A) \geq \gamma^c(A), \forall A \subseteq S$, ($\gamma$ is an upper capacity) then it is clear that $\mathcal{R}_*(\gamma)$ is more natural than $\mathcal{R}_*(\gamma^c)$ for representing γ by a family of possibility measures that dominate it.

2.2 Generalized Minitivity and Maxitivity Axioms

For each capacity γ, there is a least integer n along with n necessity measures such that $\gamma(A) = \max_{i=1}^n N_i(A)$. We now show that this property can be described by means of an axiom of the form:

n-**adjunction**: $\forall A_i, i = 1, \ldots n+1, \min_{i=1}^{n+1} \gamma(A_i) \leq \max_{1 \leq i < j \leq n+1} \gamma(A_i \cap A_j)$

that generalizes the minitivity axiom of necessity measures. Indeed, When $n = 1$, this is the usual adjunction property $\min(\gamma(A), \gamma(B)) \leq \gamma(A \cap B)$. It is then equivalent to the minitivity axiom of necessity measures: $N(A \cap B) = \min(N(A), N(B))$ since γ is inclusion-monotonic: 1-adjunctive capacities are necessity measures. Let us consider the next step: 2-adjunction.

Proposition 3. $\min(\gamma(A), \gamma(B), \gamma(C)) \leq \max(\gamma(A \cap B), \gamma(B \cap C), \gamma(A \cap C)), \forall A, B, C$, *if and only if there exist two necessity measures such that* $\forall A, \gamma(A) = \max(N_1(A), N_2(A))$.

Proof:

$\Leftarrow$: Suppose $\gamma(A) = \max(N_1(A), N_2(A))$. We can assume without loss of generality that $N_1(A) \geq N_2(A), N_1(B) \geq N_2(B), N_2(C) \geq N_1(C)$ with one strict inequality, for some A, B, C (otherwise γ is a necessity measure) and then

$$\min(\gamma(A), \gamma(B), \gamma(C)) = \min(N_1(A), N_1(B), N_2(C))$$

follows. Now consider $\gamma(A \cap B)$. We have

$$\gamma(A \cap B) = \max(\min(N_1(A), N_1(B)), \min(N_2(A), N_2(B)).$$

Developing: $\gamma(A \cap B) = \min(\max(N_1(A), N_2(A)), \max(N_1(A), N_2(B)),$
$\max(N_1(B), N_2(A)), \max(N_1(B), N_2(B)))$.
Now since by construction $N_1(A) \geq N_2(A), N_1(B) \geq N_2(B)$, it follows that

$$\gamma(A \cap B) = \min(N_1(A), \max(N_1(A), N_2(B)), \max(N_1(B), N_2(A)), N_1(B))$$

$$= \min(N_1(A), N_1(B)) = \min(\gamma(A), \gamma(B)) \geq \min(\gamma(A), \gamma(B), \gamma(C))$$

Hence $\max(\gamma(A \cap B), \gamma(B \cap C), \gamma(B \cap C)) \geq \min(\gamma(A), \gamma(B), \gamma(C))$.

$\Rightarrow$: To get the converse, suppose that non trivially, $\gamma(A) = \max_{i=1}^{3} N_i(A)$. Then one may find distinct sets A, B, C such that

$$\min(\gamma(A), \gamma(B), \gamma(C)) = \min(N_1(A), N_2(B), N_3(C)).$$

It is easy to find an example for which $\min(\gamma(A), \gamma(B), \gamma(C)) > \max(\gamma(A \cap B), \gamma(B \cap C), \gamma(B \cap C))$. For instance,we can choose the three distinct sets A, B, C such that $\gamma(A) = N_1(A)$ and $\gamma(A') = 0, \forall A' \subset A$, $\gamma(B) = N_2(B)$ and $\gamma(B') = 0, \forall B' \subset B$, $\gamma(C) = N_3(C)$ and $\gamma(C') = 0, \forall C' \subset C$. These are the least elements of the family : $\{D, \gamma(D) > 0\}$ that forms a union of three filters exactly (they are the cores of the possibility distributions inducing necessity functions $N_i, i = 1, 2, 3$). It is then clear that A, B, C are not included into one another, so that $\max(\gamma(A \cap B), \gamma(B \cap C), \gamma(A \cap C)) = 0$. Indeed, for instance $A \cap B \subset A$ and $A \cap B \subset B$ (strict inclusion), and $\gamma(A \cap B) = 0$ by construction. The same reasoning holds for $B \cap C, A \cap C$. ■

Note that in general, if $\gamma(A) = \max(N_1(A), N_2(A))$, there can be a strict inequality $\min(\gamma(A), \gamma(B), \gamma(C)) < \max(\gamma(A \cap B), \gamma(B \cap C), \gamma(B \cap C))$. Indeed it is enough that $\gamma(C) < \gamma(A \cap B)$. It contrasts with the case of $n = 1$ that comes down to $\gamma(A \cap B) \geq \min(\gamma(A), \gamma(B))$ and implies $\gamma(A \cap B) = \min(\gamma(A), \gamma(B))$, due to monotonicity of γ.

In the general case, it holds that

Proposition 4. $\forall A_i, i = 1, \ldots n+1, \min_{i=1}^{n+1} \gamma(A_i) \leq \max_{i \neq j} \gamma(A_i \cap A_j)$ *if and only if there exist* n *necessity measures such that* $\forall A, \gamma(A) = \max_{j=1}^{n} N_j(A)$.

Proof

$\Leftarrow$: Suppose $\forall A, \gamma(A) = \max_{j=1}^{n} N_j(A)$. As a consequence:

$$\min_{i=1}^{n+1} \gamma(A_i) = \min_{i=1}^{n+1} \max_{j=1}^{n} N_j(A_i) = \min_{i=1}^{n+1} N_{j_i}(A_i)$$

where $N_{j_i}(A_i) \geq N_k(A_i), \forall k \neq j_i, k = 1, \ldots n, i = 1, \ldots n+1$. It is clear that at least two among indices $j_i, i = 1, n+1$ are equal, since there are only n distinct values of j. Suppose they are $j_1 = 1 = j_2$ without loss of generality, that is, $\min_{i=1}^{n+1} \gamma(A_i) = \min(N_1(A_1), N_1(A_2), \min_{i=3}^{n+1} N_{j_i}(A_i))$.

Now $\gamma(A_1 \cap A_2) = \max_{i=1}^{n} N_i(A_1 \cap A_2) = \max_{i=1}^{n} \min(N_i(A_1), N_i(A_2))$. However by assumption $N_1(A_1) \geq N_k(A_1), k = 2, \ldots n$ and $N_1(A_2) \geq N_k(A_2), k = 2, \ldots n$, so $\min(N_1(A_1), N_1(A_2)) \geq \min(N_k(A_1), N_k(A_2)), k = 2, \ldots n$. As a consequence, $\gamma(A_1 \cap A_2) = \min(N_1(A_1), N_1(A_2)) = \min(\gamma(A_1), \gamma(A_2)) \geq \min_{i=1}^{n+1} \gamma(A_i)$.

$\Rightarrow$: For the converse, the proof is the same as for the case $n = 3$: suppose that non trivially, $\gamma(A) = \max_{i=1}^{n+1} N_i(A)$. Then one may find a family of $n+1$ distinct sets A_i such that $\gamma(A_i) = N_i(A_i), i = 1, \ldots, n+1$ and also choose them such that

$$\min_{i=1}^{n+1} \gamma(A_i) > \max_{1 \leq i < j \leq n+1} \gamma(A_i \cap A_j).$$

Indeed, choose the $n+1$ distinct sets A_i with $\gamma(A_i) = N_i(A_i)$ and $\gamma(A) = 0, \forall A \subset A_i, i = 1, \ldots, n+1$. These are the least elements of the family: $\{D, \gamma(D) > 0\}$ that is formed by a union of $n+1$ filters exactly (they are the cores of the possibility distributions inducing $N_i, i = 1, n+1$). It is then clear that none of the $A_i' s$ are included into one another, so that $\forall i < j, A_i \cap A_j \subset A_i$ and $A_i \cap A_j \subset A_j$ (strict inclusion) hence $\gamma(A_i \cap A_j) = 0$ by construction; so, $\max_{1 \leq i < j \leq n+1} \gamma(A_i \cap A_j) = 0$. ■

Note that if a capacity possesses n-adjunction it provides an upper bound on the number of its focal sets having a given weight. Indeed, if γ_λ denotes the Boolean capacity obtained as $\gamma_\lambda(A) = 1$ if $\gamma(A) \geq \lambda$, and 0 otherwise, then since $\gamma(A) = \max_{i=1}^{n} N_i(A)$, it follows that the set of focal sets of γ_λ is made of the n subsets E_i such that $N_i(A) \geq \lambda \iff E_i \subseteq A$.

In fact, if E is a focal set of γ, i.e. $E \in \mathcal{F}^\gamma$, define the necessity measure N_E by $\forall A \neq S, N_E(A) = \gamma_\#(E)$ if $E \subseteq A$ and 0 otherwise. It is clear that $\gamma(A) = \max_{E \in \mathcal{F}^\gamma} N_E(A)$. This is not the minimal form of course. To get the minimal form one may consider all chains of nested subsets in $\mathcal{F}^\gamma$: each such chain i defines a necessity measure N_i whose nested focal sets form the chain. If a capacity possesses n-adjunction, it means that there are exactly n chains of focal sets in $\mathcal{F}^\gamma$.

Note that in the extreme case where the focal sets in $\mathcal{F}^\gamma$ are singletons, each necessity measure N_E is also a possibility measure (it is a Dirac measure based on $E = \{s_E\}$), hence γ is a possibility measure.

Of course the above results can be adapted, replacing necessity measures by possibility measures, thus weakening the notion of maxitivity. We can consider the following axiom, dual to n-adjunction:

$$n\textbf{-max-dominance}: \max_{i=1}^{n+1} \gamma(A_i) \geq \min_{1 \leq i < j \leq n+1} \gamma(A_i \cup A_j)$$

$\forall A_i, i = 1, \ldots n+1$, and prove the counterpart to the above proposition:

Proposition 5. $\max_{i=1}^{n+1} \gamma(A_i) \geq \min_{i \neq j} \gamma(A_i \cup A_j)$ *if and only if if there exist* n *possibility measures such that* $\gamma(A) = \min_{i=1}^{n} \Pi_i(A)$.

Comment: In the numerical setting, the n-superadditivity of a capacity is implied by but does not imply its $(n+1)$-superadditivity. The above concept of n-minitivity (in fact n-adjunction) seems to play a similar role: we can generalize necessity functions by steps since n-minitivity implies, but is not implied by $(n+1)$-minitivity.

2.3 Qualitative Focal Sets, n-Adjunction and k-Maxitivity

The inner (qualitative) Moebius transform of a capacity γ is a mapping $\gamma_\# : 2^S \to L$ defined by

$$\gamma_\#(E) = \gamma(E) \text{ if } \gamma(E) > \max_{B \subsetneq E} \gamma(B) \tag{2}$$

and 0 otherwise. In the above definition, due to the monotonicity property, the condition $\gamma(E) > \max_{B \subsetneq E} \gamma(B)$ can be replaced by $\max_{x \in E} \gamma(E \setminus \{x\})$. It is easy to check that

- $\gamma_\#(\emptyset) = 0$; $\max_{A \subseteq S} \gamma_\#(A) = 1$;
- If $A \subset B$, and $\gamma_\#(A) > 0, \gamma_\#(B) > 0$, then $\gamma_\#(A) < \gamma_\#(B)$.

Let $\mathcal{F}^\gamma = \{E, \gamma_\#(E) > 0\}$ be the family of focal sets associated to γ. The last property says that the inner qualitative Moebius transform of γ is strictly monotonic with inclusion on $\mathcal{F}^\gamma$. It is clear that the inner qualitative Moebius transform of a possibility measure coincides with its possibility distribution: $\Pi_\#(A) = \pi(s)$ if $A = \{s\}$ and 0 otherwise. This property makes it clear that $\gamma_\#$ generalizes the notion of possibility distribution to the power set of S.

The inner (qualitative) Moebius transform contains the minimal information needed to reconstruct the capacity γ since, by construction [14,9]:

$$\gamma(A) = \max_{E \subseteq A} \gamma_\#(E) \tag{3}$$

The reader can check that if one of the values $\gamma_\#(E)$ is changed, the corresponding capacity will be different, namely the values $\gamma(A)$ such that $\gamma(A) = \gamma_\#(E)$. In a previous paper [7], it was shown that the qualitative Moebius transform is instrumental in finding the most specific possibility distributions dominating γ, via a selection process picking an element in each focal set.

The similarity between capacities and belief functions [19] is striking on the above equation: $\max$ replaces the sum in the expression of a belief function, and $\gamma_\#$ plays the role of the mass assignment, which is the Moebius transform of the belief function [15]. The subsets E in $\mathcal{F}^\gamma$ receive positive support and play the same role as the focal sets in Dempster-Shafer's theory: they are the primitive items of knowledge.

A capacity is said to be k-maxitive if and only if its focal sets have at most k elements. This notion was introduced by Mesiar [17] and Grabisch [14] as a class of simpler capacities. We show here a connection between the k-adjunction of capacities and the notion of k-maxitivity. The minitivity (1-adjunction) of necessity measures N go along with the fact that the focal elements of the conjugate possibility measure $\Pi(A) = \nu(N(A^c))$ are obviously the singletons $\{s\}$ such that $s \in A$ (1-maxitivity).

This construction can be generalized first to any qualitative capacity γ that ranges on $\{0, 1\}$. Let $\mathcal{F}^\gamma$ be its focal sets ($\gamma_\#(E) = 1$), and γ^c is its conjugate. Then obviously,

$$\gamma(A) = 1 \iff \exists E \in \mathcal{F}^\gamma, E \subset A \tag{4}$$

Lemma 1. *Suppose $\mathcal{F}^\gamma = \{E_1, \ldots E_k\}$ for a Boolean capacity γ. Then $\gamma^c(A) = 1$ if only if A contains a set the form $\{s_1, \ldots s_k\}, s_i \in E_i, i = 1 \ldots, k$.*

Proof: Indeed: $\gamma^c(A) = 1 \iff \gamma(A^c) = 0 \iff \forall E \in \mathcal{F}^\gamma, E \not\subseteq A^c$
hence: $\gamma^c(A) = 1 \iff \forall E \in \mathcal{F}^\gamma, E \cap A \neq \emptyset$. We can write this as follows:
$\gamma^c(A) = 1 \iff \forall E \in \mathcal{F}^\gamma, \exists s_E \in E \cap A \iff \exists F = \{s_E : E \in \mathcal{F}^\gamma\}, F \subseteq A$,
where for each focal set E of γ, s_E is picked in E. ■

Proposition 6. *The set of focal sets of γ^c is $\mathcal{F}^{\gamma^c} = \min_\subseteq\{\{s_1, \ldots s_k\}, s_i \in E_i, i = 1 \ldots, k\}$, where $\min_\subseteq$ picks the smallest subsets for inclusion.*

Proof: Note that $\mathcal{F}^{\gamma^c} = \min_\subseteq\{A, \gamma^c(A) = 1\}$. The result follows from Lemma 1. ■

Clearly, the elements s_E picked in focal sets E need not be distinct, in case the focal sets overlap. For instance, if $\mathcal{F}^\gamma = \{E_1, E_2\}$ with $E_1 = \{s_0, s_1, s_3\}, E_2 = \{s_0, s_2, s_4\}$, then the focal elements of the conjugate are the least elements among the family $\{\{s_0\}\} \cup \{\{s_0, s_i\}, i = 1, \ldots, 4\} \cup \{\{s_1, s_2\}, \{s_1, s_4\}, \{s_3, s_2\}, \{s_3, s_4\}\}$, that is $\mathcal{F}^{\gamma^c} = \{\{s_0\}\{s_1, s_2\}, \{s_1, s_4\}, \{s_3, s_2\}, \{s_3, s_4\}\}$.

Denoting by $c(\mathcal{F}^\gamma)$ the transformation from $\mathcal{F}^\gamma$ to $\mathcal{F}^{\gamma^c}$, we can prove:

Proposition 7. $c(c(\mathcal{F}^\gamma)) = \mathcal{F}^\gamma$

Proof: It is obvious because $(\gamma^c)^c = \gamma$. A direct proof is far less obvious.

For instance, if $\mathcal{F}^\gamma = \{A, B\}$. Then $\mathcal{F}^{\gamma^c} = \{\{s\} : s \in A \cap B\} \cup \{\{s_A, s_B\} : s_A \in A \setminus B, s_B \in B \setminus A\}$. To build dual focal sets from the latter family, each such focal set must contain $A \cap B$. Then suppose we pick $s_A \in \{s_A, s_B\}$. Clearly, this choice covers all focal sets $\{s_A, s\}, s \in B \setminus A$. It thus prevents us from picking the next element in $B \setminus A$. So the next elements to be picked lie in A. In fact, the focal sets left $\{s, s_B\}, s \neq s_A$ can be deprived of s_B since there is a focal set of the form $\{s_A, s_B\}$ that forbids s_B from further consideration. So this process reconstructs the focal set A.

From Prop. 6, it is clear that if a Boolean capacity is k-adjunctive (it has k focal sets), then its conjugate is k-maxitive, since the focal sets of its conjugate will have not more than k elements. In the next section, we shall see that the computation of the focal sets of a capacity from the ones of its conjugate corresponds in the modal logic setting to the swapping of modalities. In the following, we denote by $\mathcal{F}^\gamma_\beta$ the set of the focal elements A of a capacity γ such that $\gamma(A) = \beta$

Proposition 8. *For a general capacity γ, suppose $\mathcal{F}^\gamma = \{E_1, \ldots E_k\}$. Then, $\gamma^c(A) = 1$ if and only if $\forall i = 1 \cdots, k : E_i \cap A \neq \emptyset$. Moreover, $\mathcal{F}^{\gamma^c}_1 = \min_\subseteq\{\{s_1, \ldots s_k\}, s_i \in E_i, i = 1 \ldots, k\}$.*

Proof: It is like the proof of Lemma 1 and the subsequent proposition.

Lemma 2. *$\gamma^c(A) = \nu(\alpha) \neq 0, 1$ if and only if $\forall E, \gamma_\#(E) > \alpha$ implies $E \cap A \neq \emptyset$ and $\exists E, E \cap A = \emptyset$ such that $\gamma_\#(E) = \alpha$.*

Proof: $\gamma^c(A) \geq \nu(\alpha)$ if and only if $\gamma(A^c) \leq \alpha$ if and only if $\forall E, \gamma_\#(E) > \alpha$ implies $E \not\subseteq A^c$. Besides, the equality $\gamma^c(A) = \nu(\alpha)$ is attained if moreover there is a focal set $E \subseteq A^c$ such that $\gamma_\#(E) = \alpha$.

Proposition 9. *A is a focal element of γ^c such that $\gamma^c_{\#}(A) = \nu(\alpha) > 0$ if and only if it is a minimal element of the family $\{E = \{s_E : \gamma_{\#}(E) > \alpha\}, E \cap F = \emptyset$ for some $F \in \mathcal{F}^{\gamma}_{\alpha}\}$, where $s_E \in E$.*

Proof: A direct consequence of the lemma, since by construction $\gamma^c_{\#}(A) = \nu(\alpha)$ means that A is a minimal set such that $\gamma^c(A) = \nu(\alpha)$.

These results show how the inner qualitative Moebius transform of a capacity can be computed from the one of its conjugate. It is easy to see that also in the general case, if a capacity has k weighted focal sets, its conjugate will be k-maxitive, since the largest focal elements of γ^c (they have weight equal to 1) are obtained by picking one element in each focal set of γ. Another issue is now to compute the n possibility distributions such that γ is n-adjunctive in terms of the m possibility distributions such that γ is m-max-dominant. For instance, while a necessity measure N is 1-adjunctive w.r.t. its associated possibility distribution π, it is also n-max-dominant with respect to n possibility measures, where n is the number of (nested) focal sets of the necessity measure N. They are all distinct sets $A_{\alpha_i} = \{s : \pi(s) \geq \alpha_i\}$ such that $N_{\#}(A_{\alpha_i}) = \nu(\alpha_{i+1})$, where $\alpha_1 = 1 > \alpha_2 > \cdots > \alpha_n > \alpha_{n+1} = 0$. Then $N = \min_{i=1}^{n} \Pi_i$, where $\pi_i(s) = \nu(\alpha_{i+1}), \forall s \in A_{\alpha_i}$ and 1 otherwise.

3 The Modal Logic View of Capacities

In this section, we show that our previous results suggest a new semantics for general modal logics. Consider a propositional language $\mathcal{L}$ with Boolean variables $\{a, b, c...\}$ and standard connectives $\wedge, \vee, \neg, \rightarrow$. Let S be the set of interpretations of this language (assigning 1 or 0 to all variables). Given a proposition $p \in \mathcal{L}$, necessity measure N on S based on possibility distribution π, we denote by $\Box p$ the statement $N(A) \geq \lambda > 0$, where $A = [p]$ is the set of models of p. $\Box p$ corresponds to a Boolean necessity measure based on a possibility distribution that is the characteristic function of $E = \{s | \pi(s) > \nu(\lambda)\}$. Consider a higher level propositional language $\mathcal{L}_{\Box}$ defined by: $\forall p \in \mathcal{L}, \Box p \in \mathcal{L}_{\Box}$, and if $\phi, \psi \in \mathcal{L}_{\Box}$, then $\neg\phi \in \mathcal{L}_{\Box}$, and $\phi \wedge \psi \in \mathcal{L}_{\Box}$. The variables of $\mathcal{L}_{\Box}$ are thus $\{\Box p : p \in \mathcal{L}\}$. Let $\Diamond p$ be short for $\neg\Box\neg p$. Then $\models \Diamond p$ stands for $\Pi(A) \geq \nu(\lambda)$ where Π is the conjugate of N. It defines a very elementary fragment of a KD modal logic known as MEL [1]. Indeed, the following KD axioms are valid

- $(K) : \Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$
- $(N) : \ \Box\top$
- $(D) : \ \Box p \rightarrow \Diamond p$

and imply axiom (C) : $\Box(p \wedge q) \equiv (\Box p \wedge \Box q)$, which is the Boolean form of the minitivity axiom.

A "model" of a formula in $\phi \in \mathcal{L}_{\Box}$ is a nonempty subset $E \subseteq S$ of propositional models. The set E is understood as an epistemic state (a *meta-model*). The satisfaction of MEL-formulae is then defined recursively given $\phi, \psi \in \mathcal{L}_{\Box}$:

- $E \models \Box p$, if and only if $E \subseteq [p]$
- $E \models \neg\phi$, if and only if $E \not\models \phi$,
- $E \models \phi \wedge \psi$, if and only if $E \models \phi$ and $E \models \psi$,
- So, $E \models \Diamond p$ if and only if $E \cap [p] \neq \emptyset$

For any set $\Gamma \cup \{\phi\}$ of $\mathcal{L}_\Box$-formulae, ϕ is a semantic consequence of Γ, written $\Gamma \models \phi$, provided for every epistemic state E, $E \models \Gamma$ implies $E \models \phi$. This Boolean possibilistic logic, equipped with modus ponens, (the $\mathcal{L}_\Box$-fragment of KD) is sound and complete w.r.t. this semantics [1]. In fact, if N is the Boolean necessity measure induced by E, it defines precisely a classical interpretation of $\mathcal{L}_\Box$, of the form $\bigwedge_{p\in\mathcal{L}:N([p])=1} \Box p \wedge \bigwedge_{p\in\mathcal{L}:N([p])=0} \neg\Box p$ obeying axioms K, D, N. In particular the semantics does not rely on the use of accessibility relations.

Using the same language, denote now $\models \Box p$ as standing for $\gamma([p]) \geq \lambda > 0$ for any qualitative capacity γ. $\Box p$ now corresponds to a Boolean capacity defined by $\gamma_\lambda(A) = 1$ if $\gamma([p]) \geq \lambda > 0$ and 0 otherwise. The following axioms are then verified [8]:

- (RE) : $\Box p \equiv \Box q$ whenever $\vdash p \equiv q$.
- (RM) : $\Box p \rightarrow \Box q$, whenever $\vdash p \rightarrow q$.
- (N) : $\Box\top$; (P) : $\Diamond\top$.

It is a non-regular modal logic. It is a fragment of the *monotonic modal* logic EMN, Chellas [4], where modalities only apply to propositions. Its usual semantics is based on so-called neighborhoods (families of subsets of possible worlds having some properties). This logic no longer satisfies axioms K, C nor D. This modal logic is the natural logical account of qualitative capacities. Indeed, any classical interpretation of $\mathcal{L}_\Box$ that satisfies the above axioms defines and is defined by a Boolean capacity β and is of the form $\bigwedge_{p\in\mathcal{L}:\beta([p])=1} \Box p \wedge \bigwedge_{p\in\mathcal{L}:\beta([p])=0} \neg\Box p$.

Interestingly, we can capture the n-adjunction axiom in the modal setting (see [8] for $n = 2$). Let n be the smallest integer for which $\gamma(A) = \max_{i=1}^n N_i(A)$. Denoting by $\Box_i p$ the statement $N_i([p]) \geq \lambda > 0$, it is clear that $\gamma([p]) \geq \lambda > 0$ stands for $\Box p \equiv \vee_{i=1}^n \Box_i p$, where $\Box_i$ are KD modalities. By duality we can define $\Diamond p$ as short for $\neg\Box\neg p$, that is, $\Diamond p \equiv \wedge_{i=1}^n \Diamond_i p$. So, applying the characterisation of n-minitivity to the restriction of the modal logic EMN yields the axiom

$$(n\text{-C}) : \vdash (\wedge_{i=1}^{n+1} \Box p_i) \rightarrow \vee_{i\neq j=1}^{n+1} \Box(p_i \wedge p_j)$$

It implies that if $p_i, i = 1 \ldots, n+1$ are mutually inconsistent, then $\vdash \neg \wedge_{i=1}^{n+1} \Box p_i$. This property claims that we cannot have $\gamma([p_i]) \geq \lambda > 0$ for all $i = 1 \ldots, n+1$.

The semantics of the EMNP+n-C logic can be expressed in two ways:

- In terms of n-tuple of epistemic states (subsets of S) : $(E_1, \ldots, E_n) \models \Box p$ if $\exists i \in [1, n], E_i \models \Box_i p$. By construction, $E_1, \ldots, E_n$ are the focal sets of the Boolean capacity defined by $\gamma_\lambda(A) = 1$ if $\gamma([p]) \geq \lambda > 0$ and 0 otherwise.
- More classically, in terms of neigborhoods: they are non-empty subsets $\mathcal{N}$ of 2^S such that $\mathcal{N} \models \Box p$ if and only if $[p] \in \mathcal{N}$ and $\mathcal{N} \models \Diamond p$ if and only if $[\neg p] \notin \mathcal{N}$.

For a KD modality, it is obvious that $\mathcal{N} = \{A, N(A) \geq \lambda\} = \{A | A \supseteq E\}$ for some non-empty $E \subseteq S$ ($\mathcal{N}$ is a proper filter). For an EMNP modality $\mathcal{N} = \{A, \gamma(A) \geq \lambda > 0\} \neq 2^S$ is closed under inclusion and not empty). For an EMNP+n-C modality, $\mathcal{N} = \{A, \gamma(A) \geq \lambda > 0\}$ is the union of n proper filters of the form $\{A, N_i(A) \geq \lambda\} = \{A | A \supseteq E_i\}$.

In the extreme case when the sets $(E_1, \ldots, E_n)$ are singletons (i.e., fully informed conflicting sources), the necessity modality $\Box p$ satisfies distributivity w.r.t. disjunction: $\vdash \Box(p \vee q) \equiv \Box p \vee \Box q$ (but no longer w.r.t. conjunction !) and the opposite of axiom D : $\vdash \Diamond p \rightarrow \Box p$. In other words, necessity and possibility modalities are exchanged.

We go back to the MEL logic exchanging the basic modalities $\Box$ and $\Diamond$. In fact, the swapping of modalities is a simple instance of the more general question, considered in the previous section, of computing the focal sets of a capacity from the ones of its conjugate. It comes down at the semantic level to the transformation of a logic based on the epistemic states of k agents into the dual situation of multiple source epistemic logic underlying a set of agents whose knowledge has limited imprecision (i.e., each epistemic state involves at most k possible worlds).

4 Conclusion

We have studied the representation of capacities having values on a finite totally ordered scale by families of qualitative possibility distributions. It turns out that any capacity can be viewed either as a lower possibility measure or as an upper necessity measure with respect to two distinct families of possibility distributions. This remark has led to propose a generalisation of maxitivity and minitivity properties of possibility theory, thus offering a classification of qualitative capacities in terms of increasing levels of complexity and generality, based on the minimal number of possibility distributions needed to represent them. In particular, it has been shown that a Sugeno integral is a lower possibility integral [7]. Then the computation of Sugeno integral can be reduced for k-adjunctive or k-max dominant capacities. Moreover, the study of relationships between the focal sets of a capacity and the focal sets of its conjugate has shown the links between k-adjunction and k-maxitive capacities. We have finally shown a connection between qualitative capacities and non-regular modal logics, which generalize KD-style modal logics in the same sense as capacities generalize necessity measures.

Numerous alleys of research are opened by the above results:

- On the logical side, we may reconsider the study of non-regular modal logics in the light of capacity-based semantics. The fact that they lead to disjunctions of KD necessity operators is clearly reminding of Belnap epistemic set-up [3], and paraconsistent logics. The fact that an extreme case of the EMN logic comes down to a modal logic similar to a KD one where possibility and necessity are exchanged reflects the fact that in Belnap bilattices, the epistemic values representing conficting information and absence thereof play symmetric roles
- One may also wish to evaluate the quantity of information (or uncertainty) contained in a qualitative capacity [16]. In [7], the maximal specific possibility distribution dominating a capacity was studied and shown to be the counterpart of the contour function of belief functions for qualitative capacities. This notion could suggest one approach based on the comparison of contour functions.
- The analogy between belief functions and qualitative capacities was discussed in [18] and a qualitative counterpart of information ordering based on specialisation (inclusion of focal sets) was also proposed, as well as counterparts to Dempster rule of combination. These lines should be pursued in the scope of qualitative information fusion techniques going beyond those based on possibility theory.

References

1. Banerjee, M., Dubois, D.: A simple modal logic for reasoning about revealed beliefs. In: Sossai, C., Chemello, G. (eds.) ECSQARU 2009. LNCS (LNAI), vol. 5590, pp. 805–816. Springer, Heidelberg (2009)
2. Banon, G.: Constructive decomposition of fuzzy measures in terms of possibility and necessity measures. In: Proc. VIth IFSA World Congress, São Paulo, Brazil, vol. I, pp. 217–220 (1995)
3. Belnap, N.D.: How a computer should think. In: Ryle, G. (ed.) Contemporary Aspects of Philosophy, pp. 30–56. Oriel Press, Boston (1977)
4. Chellas, B.F.: Modal logic: an Introduction. Cambridge University Press, Cambridge (1980)
5. Coletti, G., Scozzafava, R., Vantaggi, B.: Inferential processes leading to possibility and necessity. Information Sciences (2012)
6. Dempster, A.P.: Upper and lower probabilities induced by a multivalued mapping. Annals of Mathematical Statistics 38, 325–339 (1967)
7. Dubois, D.: Fuzzy Measures on Finite Scales as Families of Possibility Measures. In: Proc. European Society for Fuzzy Logic and Technology (EUSFLAT-LFA), Aix-Les-Bains, France (July 2011)
8. Dubois, D.: Reasoning about ignorance and contradiction: many-valued logics versus epistemic logic. Soft Computing 16(11), 1817–1831 (2012)
9. Dubois, D., Fargier, H.: Making Discrete Sugeno Integrals More Discriminant. Int. J. of Approximate Reasoning 50, 880–898 (2009)
10. Dubois, D., Fargier, H.: Capacity refinements and their application to qualitative decision evaluation. In: Sossai, C., Chemello, G. (eds.) ECSQARU 2009. LNCS (LNAI), vol. 5590, pp. 311–322. Springer, Heidelberg (2009)
11. Dubois, D., Prade, H.: Upper and lower possibilities induced by a multivalued mapping, In: Proc. IFAC Symp. on Fuzzy Information, Knowledge Representation and Decision Analysis, Marseille, July 19-21, pp. 174–152 (1983)
12. Dubois, D., Prade, H.: Evidence measures based on fuzzy information. Automatica 21, 547–562 (1985)
13. Dubois, D., Prade, H.: Aggregation of possibility measures. In: Kacprzyk, J., Fedrizzi, M. (eds.) Multiperson Decision Making using Fuzzy Sets and Possibility Theory, pp. 55–63. Kluwer, Dordrecht (1990)
14. Grabisch, M.: On the representation of k-decomposable measures. In: Proc. 7th IFSA World Congress, Prague, vol. 1, pp. 478–483 (1997)
15. Grabisch, M.: The Moebius transform on symmetric ordered structures and its application to capacities on finite sets. Discrete Mathematics 287, 17–34 (2004)
16. Marichal, J.-L., Roubens, M.: Entropy of Discrete Fuzzy Measures. Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems 8(6), 625–640 (2000)
17. Mesiar, R.: k-order pan-discrete fuzzy measures. In: Proc. 7th IFSA World Congress, Prague, vol. 1, pp. 488–490 (1997)
18. Prade, H., Rico, A.: Possibilistic Evidence. In: Liu, W. (ed.) ECSQARU 2011. LNCS (LNAI), vol. 6717, pp. 713–724. Springer, Heidelberg (2011)
19. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, Princeton (1976)
20. Schmeidler, D.: Core of exact games I. J. Math. Analysis and Appl. 40, 214–225 (1972)
21. Tsiporkova, E., De Baets, B.: A General Framework for Upper and Lower Possibilities and Necessities. Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems 6(1), 1–34 (1998)
22. Walley, P.: Statistical Reasoning with Imprecise Probabilities. Chapman and Hall (1991)

Conditional Preference Nets and Possibilistic Logic

Didier Dubois, Henri Prade, and Fayçal Touazi

IRIT, University of Toulouse, 118 rte de Narbonne, Toulouse, France
{dubois,prade,faycal.touazi}@irit.fr

Abstract. CP-nets (Conditional preference networks) are a well-known compact graphical representation of preferences in Artificial Intelligence, that can be viewed as a qualitative counterpart to Bayesian nets. In case of binary attributes it captures specific partial orderings over Boolean interpretations where strict preference statements are defined between interpretations which differ by a single flip of an attribute value. It respects preferential independence encoded by the ceteris paribus property. The popularity of this approach has motivated some comparison with other preference representation setting such as possibilistic logic. In this paper, we focus our discussion on the possibilistic representation of CP-nets, and the question whether it is possible to capture the CP-net partial order over interpretations by means of a possibilistic knowledge base and a suitable semantics. We show that several results in the literature on the alleged faithful representation of CP-nets by possibilistic bases are questionable. To this aim we discuss some canonical examples of CP-net topologies where the considered possibilistic approach fails to exactly capture the partial order induced by CP-nets, thus shedding light on the difficulties encountered when trying to reconcile the two frameworks.

1 Introduction

The representation and the handling of preferences has been extensively studied in artificial intelligence (AI), operations research, and data bases; see [1] for an introductory survey. "CP-nets" [2] have been especially popular in AI as a framework for expressing conditional preferences, based on a graphical representation. CP-nets express that in a given context, a partially described situation is strictly preferred to another partially described situation, every other variable having the same value in both situations; this is the *ceteris paribus* condition.

However the systematic application of the ceteris paribus principle introduces restrictions in the expression of preferences. This has motivated the comparison between CP-nets and possibilistic logic [3] since the latter provides another flexible setting for representing preferences [4, 5]. In possibilistic logic, classical propositions state goals, and weights are priority levels that express how imperative are these goals. A merit of a logic-based representation of preferences is also the capability of reasoning about preferences and in particular to deal with their possible inconsistency. A series of publications [6–10] have dealt with the question of representing CP-nets by means of a possibilistic logic base. Since CP-nets may leave some interpretations non comparable, a possibilistic logic

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 181–193, 2013.

representation of them should use partially ordered symbolic weights [11] that leave room for incomparability. It has been also noticed that CP-nets implicitly privilege the preference constraints associated with father nodes with respect to the ones associated to children nodes in the graphical representation.

However, the possibilistic logic representation of CP-nets advocated in [8–10] is not always completely faithful and may remain locally approximate. The aim of this paper is to fully investigate this state of facts, also highlighting when the existing approach does provide an exact representation for CP-nets.

The paper is organized as follows. First, a short background on possibilistic logic, on CP-nets and its encoding with possibilistic logic formulas having symbolic weights is provided in Sections 2 and 3. Then in Section 4 we discuss the different partial orders that can be used for comparing the vectors of symbolic weights which reflect the violation of preferences and are associated with each interpretation. Used as such, each of the considered orders are successful for retrieving the CP-net ordering on specific graphical structures and fail on others, as shown in Section 5. Section 6 identifies on which particular structures the existing possibilistic representation is exact, and shows more generally how lower and upper representations can be obtained. Section 7 briefly discusses the related work and exhibits a final example that points out the difficulty of capturing the CP-net ordering exactly in a logical way.

2 Possibilistic Logic

We consider a propositional language where formulas are denoted by p_1, ..., p_n, and Ω is its set of interpretations. Let $B^N = \{(p_j, \alpha_j) \mid j = 1, \ldots, m\}$ be a possibilistic logic base where p_j is a propositional logic formula and $\alpha_j \in \mathcal{L} \subseteq [0, 1]$ is a priority level [3]. The logical conjunctions and disjunctions are denoted $\wedge$ and $\vee$. Each formula (p_j, α_j) means that $N(p_j) \geq \alpha_j$, where N is a necessity measure, i.e., a set function satisfying the property $N(p \wedge q) = \min(N(p), N(q))$. A necessity measure is associated to a possibility distribution π (a mapping $\Omega \to [0, 1]$ here expressing preference) as follows:

$$N(p) = \min_{\omega \notin M(p)}(1 - \pi(\omega)) = 1 - \Pi(\neg p),$$

where Π is the possibility measure associated to N and $M(p)$ is the set of models induced by the underlying propositional language for which p is true.

The base B^N is associated to the possibility distribution

$$\pi_B^N(\omega) = \min_{j=1,\ldots,m} \pi_{(p_j,\alpha_j)}(\omega)$$

on the set of interpretations, where $\pi_{(p_j,\alpha_j)}(\omega) = 1$ if $\omega \in M(p_j)$, and $\pi_{(p_j,\alpha_j)}(\omega) = 1 - \alpha_j$ if $\omega \notin M(p_j)$. An interpretation ω is all the more possible as it does not violate any formula p_j having a higher priority level α_j. So, if $\omega \notin M(p_j)$, $\pi_B^N(\omega) \leq 1 - \alpha_j$, and if $\omega \in \bigcap_{j \in J} M(\neg p_j)$, $\pi_B^N(\omega) \leq \min_{j \in J}(1 - \alpha_j)$. It is a description "from above" of π_B^N, which is the least specific possibility distribution in agreement with the knowledge base B^N. A possibilistic base B^N can be transformed in a base where the formulas p_i are clauses (without altering the distribution π_B^N). We can still see B^N as a conjunction of weighted clauses, i.e., as an extension of the conjunctive normal form.

3 CP-Nets and Their Encoding in Possibilistic Logic

A CP-net [2] is graphical in nature, and exploits conditional preferential independence in structuring the preferences provided by a user. The model is reminiscent of a Bayes net; however, the nature of the relation between nodes within a network is generally quite weak, compared with the probabilistic relations in Bayes nets. The aim in using the graph is to capture statements of qualitative conditional preferential independence.

Definition 1. *A CP-net $\mathcal{N}$ over the set of Boolean variables $V = \{X_1, \cdots, X_n\}$ is a directed graph over the nodes $X_1, \cdots, X_n$, and there is a directed edge from X_i to X_j if the preference over the value X_j is conditioned on the value of X_i. Each node $X_i \in V$ is associated with a conditional preference table $CPT(X_i)$ that associates a strict preference ($x_i > \neg x_i$ or $\neg x_i > x_i$) with each possible instantiation u_i of the parents of X_i (if any).*

A complete (preference) ordering of interpretations satisfies a CP-net $\mathcal{N}$ iff it satisfies each conditional preference expressed in $\mathcal{N}$. In this case, the ordering is said to be *consistent* with $\mathcal{N}$. We denote by $Pa(X)$ the set of direct parent variables of X, and by $Ch(X)$ the set of direct successors (children) of X. The set of interpretations of a group of variables $S \subseteq V$ is denoted by $Ast(S)$, with $\Omega = Ast(V)$. Given a CP-net $\mathcal{N}$, for each node $X_i, i = 1, \ldots, n$, each entry in a conditional preference table CPT_i is of the form $\phi = u : \star x_i > \star \neg x_i$, where $u \in Ast(Pa(X_i))$, $\star$ is blank if the preference is $x_i > \neg x_i$ and is $\neg$ otherwise. This is encoded by a constraint of the form $N(\neg u \vee \star x_i) \geq \alpha_i > 0$, in possibility theory, where N is a necessity measure [3]. The weight α_i stands for the priority of the formula $\neg u \vee \star x_i$. Although valued on [0, 1] this priority is not instantiated, that is, α_i is a variable attached to node i. It expresses that having $\neg \star x_i$ is somewhat not satisfactory in context u, as the possibility of $\neg \star x_i \wedge u$ is upper bounded by $1 - \alpha_i$. Clearly, satisfying $\neg \star x_i \wedge u$ is all the more impossible as α_i is large.

The encoding of a CP-net in possibilistic logic is performed as follows:

- According to the above conventions, each entry of the form $u : \star x_i > \star \neg x_i$ in the conditional preference table CPT_i of each node $X_i, i = 1, \ldots, n$ is encoded by the possibilistic logic clause $(\neg u \vee \star x_i, \alpha_i)$, where $\alpha_i > 0$ is a symbolic weight.
- Since the same weight is attached to each clause built from CPT_i, the set of weighted clauses induced from CPT_i is thus equivalent to the weighted conjunction $\phi_i = (\bigwedge_{u \in Ast(Pa(X_i))} (\neg u \vee \star x_i), \alpha_i)$, one per variable, or to the pair of weighted clauses (ϕ_i^+, ϕ_i^-) of the form:
$$(\neg(\vee_{u \in A_i^+} u) \vee x_i, \alpha_i), (\neg(\vee_{u \in A_i^-} u) \vee \neg x_i, \alpha_i),$$
where $\{A_i^+, A_i^-\}$ is a partition of $Ast(Pa(X_i))$, such that $x_i > \neg x_i$ on A_i^+ and $\neg x_i > x_i$ on A_i^-.
- Additional constraints over weights are added. The weight α_i attached to each node X_i, is supposed to be strictly smaller than the weight of each of its parents α_i^* (thus leading to constraints of the form $\max(\{\alpha_i\}) < \alpha_i^*$).

A partially ordered possibilistic base $(\Sigma, \succeq_\Sigma)$ is built from a CP-net in this way, where $\succeq_\Sigma$ stands for the order relation over weights. Let us denote by $\mathcal{F}_\omega \subseteq \Sigma$, the set of formulas falsified by the interpretation $\omega \in \Omega$. For each interpretation ω, we associate a vector $\boldsymbol{\omega}(\Sigma)$ obtained as follows. For each weighted formula $\phi_i^+ \wedge \phi_i^-$ in the possibilistic base Σ satisfied by ω, we put 1 in the i^{th} component of the vector, and $1-\alpha_i$ otherwise, in agreement with possibilistic logic semantics [3]. By construction, $L = \{1, 1-\alpha_i, i = 1\ldots, n\}$, with $1 > 1-\alpha_i, \forall i$. Vector $\boldsymbol{\omega}(\Sigma)$ has a specific format. Namely its component v_i (one per CP-net node) lies in $\{1, 1-\alpha_i\}$ for $i = 1, \ldots, n$. We consider different possible partial orders for comparing such vectors in the next section.

Example 1: [2]. Fig. 1(a) illustrates a CP-net about preferences for evening dress. It involves variables J, P, and S, standing for the jacket, pants, and shirt:

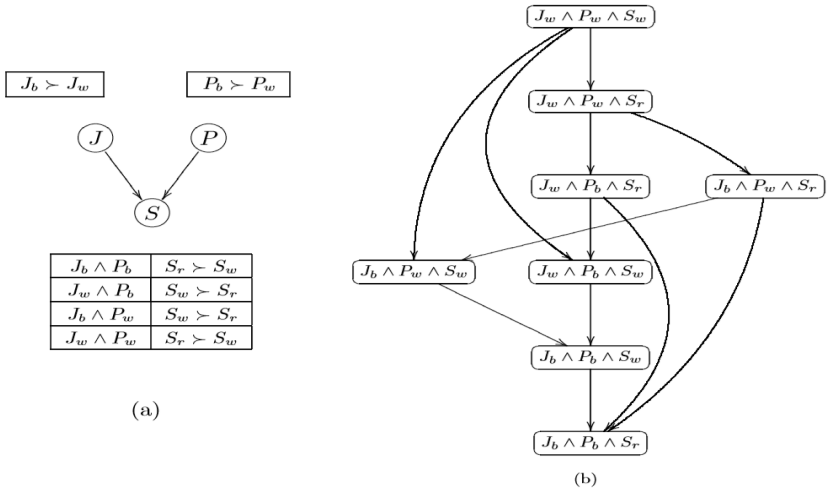


Fig. 1. CP-net and partial order induced by it

- preferred color is black (b) rather than white (w) for J and P: $P_b > P_w$, which yields formula $\phi_P = (P_b, \alpha)$, and $J_b > J_w$, which yields formula $\phi_J = (J_b, \beta)$.
- the preference between the red and white shirts is conditioned on the combination of jacket and pants: if they have the same color, then a white shirt will make my outfit too colorless, thus a red shirt is preferred: $P_b \wedge J_b : S_r > S_w$; $P_w \wedge J_w : S_r > S_w$, which yields formula $\phi_S^- = (\neg(J = P) \vee S_r, \gamma)$.
- Otherwise, if the jacket and the pants are of different colors, then a red shirt will probably make the outfit too flashy, thus a white shirt is preferred. $P_b \wedge J_w : S_w > S_r$; $P_w \wedge J_b : S_w > S_r$, which yields formula $\phi_S^+ = ((J = P) \vee S_w, \gamma)$. Moreover, we assume $\alpha > \gamma$ and $\beta > \gamma$ since P and S are father nodes of J.

4 Partial Order Relations over Vectors

In this section we will present a number of partial order relations with the purpose to use them to generate a particular ordering over interpretations.

In Section 3, we have shown how to encode a CP-net in a possibilistic logic format. Since we can associate a vector to each interpretation with respect to formulas in the possibilistic base, comparing two interpretations amounts to comparing their associated vectors. We first give definitions of some order relations over vectors, and then discuss how to capture CP-net orderings when we interpret possibilistic logic bases based on these vector comparison techniques. Let $\boldsymbol{v} = (v_1, ..., v_k), \boldsymbol{v'} = (v'_1, ..., v'_k) \in L^k$ be two vectors, where L is a scale partially ordered by $>$:

Definition 2 (Pareto). $\boldsymbol{v} \succ_{Pareto} \boldsymbol{v'}$ *if and only if* $\forall i, v_i \geq v'_i$ *and* $\exists j, v_j > v'_j$.

Definition 3 (symmetric Pareto). $\boldsymbol{v} \succ_{SP} \boldsymbol{v'}$ *if and only if there exists a permutation* σ *the components of* $\boldsymbol{v'}$*, yielding vector* $\boldsymbol{v'}^{\sigma}$*, such that* $\boldsymbol{v} \succ_{Pareto} \boldsymbol{v'}^{\sigma}$.

The discrimin order, denoted by $\succ_{discrimin}$ is defined for totally ordered scales in the following way: identical vector components are discarded, and the minimum of the remaining components for each vector are compared. Since here the minimum does not always correspond to a single value, but to subsets of L^k, we propose the following procedure for comparing the vectors:

Definition 4 (discrimin). *Let* $\mathfrak{D}(\boldsymbol{v}, \boldsymbol{v'}) = \{j | v_j \neq v'_j\}$ *be the set of component indices where the two vectors* $\boldsymbol{v}$ *and* $\boldsymbol{v'}$ *differ. Then* $v \succ_{discrimin} v'$ *iff* $\min(\{v_i | i \in \mathfrak{D}(\boldsymbol{v}, \boldsymbol{v'})\} \cup \{v'_i | i \in \mathfrak{D}(\boldsymbol{v}, \boldsymbol{v'})\}) \subseteq \{v'_i | i \in \mathfrak{D}(\boldsymbol{v}, \boldsymbol{v'})\} \setminus \{v_i | i \in \mathfrak{D}(\boldsymbol{v}, \boldsymbol{v'})\}$. *where* min *here returns the subset of the smallest incomparable values (wrt* $>$*).*

In the standard case of a totally ordered scale, the leximin order is defined by first reordering the vectors in an increasing way and then applying the discrimin order to the reordered vectors. Since we deal with a partial order, the reordering of vectors is no longer unique, and we have to generalize the definition:

Definition 5 (leximin). *First, delete all pairs* (v_i, v'_j) *such that* $v_i = v'_j$ *in* $\boldsymbol{v}$ *and* $\boldsymbol{v'}$ *(each deleted component can be used only one time in the deletion process). Thus, we get two non overlapping sets* $r(\boldsymbol{v})$ *and* $r(\boldsymbol{v'})$ *of remaining components, namely* $r(\boldsymbol{v}) \cap r(\boldsymbol{v'}) = \emptyset$*. Then,* $v \succ_{lex} v'$ *iff* $\min(r(\boldsymbol{v}) \cup r(\boldsymbol{v'})) \subseteq r(\boldsymbol{v'})$.

In the following, we shall apply these relations to the particular vectors associated to the possibilistic encoding of CP-nets, as explained in Section 3, where the possible values of a vector component i are either 1 or $1 - \alpha_i$ (the α_i being distinct variables), and $L = \{1, 1 - \alpha_i, i = 1, \ldots, n\}$ such that $1 > 1 - \alpha_i$.

Proposition 1. Leximin *and* discrimin *orders coincide on these particular vectors.*

Proof. Indeed, since the value of a vector component is either '1' or '$1 - \alpha_i$', and since each possibilistic formula attached to a node in the CP-net is associated with a different weight α_i, we are sure that a given '$1 - \alpha_i$' is present only in one component position. With these hypotheses, the difference between *leximin* and *discrimin* procedures is that *leximin* deletes some components with value '1' because it is the only component value that can be in different ranks. But we

know that '1' is the greatest component value, so this cannot affect the result of the final application of min operator in each case. Thus, *leximin* and *discrimin* orders coincide on these particular vectors.

These relations have been previously used for capturing the CP-nets ordering: symmetric Pareto (SP), discrimin in [8, 9], or leximin in [10] or min order in [6, 7]. In the next section, we provide a comparative discussion of these proposals and we point out when each ordering fails to exactly retrieve the CP-net ordering.

5 CP-Nets vs. Possibilistic Logic: Counterexamples

It has been claimed that CP-net orderings can be captured by using the encoding explained in Section 3 and applying the *symmetric Pareto* order [8, 9] recalled in Section 4, or the *leximin* order [10], to vectors $\boldsymbol{\omega}(\Sigma)$. This is in fact true only for special families of CP-nets, as shown in the example below. But the possibilistic encoding of CP-nets together with the use of one of the previously cited orders do not always lead to an exact representation of CP-nets in the general case, as we shall see on further examples.

Considering Ex. 1 again, Table 2 gives the satisfaction levels for the possibilistic clauses encoding the 3 elementary preferences, and the 8 possible interpretations (choices), where α, β, γ are the weights of nodes J, P, S respectively.

Table 1. Possible alternative choices in Example 1

Ω	ϕ_P	ϕ_J	ϕ_S
$P_b J_b S_r$	1	1	1
$P_b J_b S_w$	1	1	1- γ
$P_b J_w S_w$	1	1-β	1
$P_w J_b S_w$	1-α	1	1
$P_b J_w S_r$	1	1-β	1-γ
$P_w J_b S_r$	1-α	1	1-γ
$P_w J_w S_r$	1-α	1-β	1-γ
$P_w J_w S_w$	1-α	1-β	1

We introduce the following constraints, $\alpha > \gamma$ and $\beta > \gamma$ between the symbolic weights, which give priority to the constraint associated to father nodes J, P over the ones corresponding to the child node S. Then, the application of symmetric Pareto order or leximin order, allows us to rank-order interpretations. It can be checked that the ordering of interpretations obtained by these two orders applied to vectors $\boldsymbol{\omega}(\Sigma)$ coincide with the ordering $\succ_{\mathcal{N}}$ induced by the CP-net $\mathcal{N}$, as indicated in Fig. 1(b) (for short, $P_b J_b S_r$ is denoted bbr, etc.):

- $bbr \succ_{\mathcal{N}} bbw \succ_{\mathcal{N}} bww \succ_{\mathcal{N}} bwr \succ_{\mathcal{N}} bwr \succ_{\mathcal{N}} wwr \succ_{\mathcal{N}} www$.
- $bbr \succ_{\mathcal{N}} bbw \succ_{\mathcal{N}} wbw \succ_{\mathcal{N}} wbr \succ_{\mathcal{N}} wwr \succ_{\mathcal{N}} www$.

In order to provide a clear discussion about the possibilistic logic representation, we first establish that a preference between interpretation vectors differing by a single variable flip only depends on the instantiations of the corresponding variable and its children:

Proposition 2. *Let X_i be a node in a CP-net $\mathcal{N}$ and $Y_i = V \setminus \{\{X_i\} \cup Pa(X_i)\}$. Let $(\Sigma, \succeq_\Sigma)$ be the partially ordered possibilistic base associated with $\mathcal{N}$ using the procedure of Section 3. If the CP-net contains the statement $u : x_i > \neg x_i$ (resp: $u : \neg x_i > x_i$), the preference only depends on the instantiations of variable x_i and its children nodes.*

Proof: Let $\omega^+ = u_i x_i y_i$ and $\omega^- = u_i \neg x_i y_i$, $u_i \in A_i^+$. Since they share the same assignment of variables in $Pa(X_i)$, both models satisfy either ϕ_j^+ or $\phi_j^-, \forall X_j \in Pa(X)$. We denote by $\mathcal{F}^{Pa}$ the set of formulas $\phi_j^+, \phi_j^-, X_j \in Pa(X_i)$ falsified by ω^+, ω^- (they are the same); and by $\mathcal{F}^Y$ the set of formulas $\phi_j^+, \phi_j^-, X_j \in Y_i \setminus Ch(X_i)$, (i.e. X_j is a neither a direct descendant of X_i nor one of its parents) and falsified by ω^+, ω^-; and by $\mathcal{F}^{Ch}_{\omega^+}$ the set of formulas $\phi_j^+, \phi_j^-, X_j \in Ch(X_i)$ falsified by ω^+ and $\mathcal{F}^{Ch}_{\omega^-}$ the set of formulas falsified by ω^-. Then, $\mathcal{F}_{\omega^+} = \mathcal{F}^{Pa} \cup \mathcal{F}^Y \cup \mathcal{F}^{Ch}_{\omega^+}$ and $\mathcal{F}_{\omega^-} = \mathcal{F}^{Pa} \cup \{\phi_i^+\} \cup \mathcal{F}^Y \cup \mathcal{F}^{Ch}_{\omega^-}$. So we have $\mathcal{F}_\omega \setminus \mathcal{F}_{\omega'} = \mathcal{F}^{Ch}_{\omega^+}$ and $\mathcal{F}_{\omega'} \setminus \mathcal{F}_\omega = \{\phi_i^+\} \cup \mathcal{F}^{Ch}_{\omega^-}$. Following the construction of $(\Sigma, \succeq_\Sigma)$ we have that ϕ_i^+ is strictly preferred to all formulas in $\mathcal{F}^{Ch}_{\omega^+} \cup \mathcal{F}^{Ch}_{\omega^-}$. Then $\forall \phi \in \mathcal{F}_\omega \setminus \mathcal{F}_{\omega'}, \phi_i^+ \succ_\Sigma \phi$.

Let X_k be a child of X_i. Note that by construction, $\omega^+ \models \phi_k^+$ and $\omega^- \models \phi_k^-$. Besides, $\omega^+ \models \neg\phi_k^-$ if and only if $\omega^+ \models u_k$, and $\omega^- \models \neg\phi_k^+$ if and only if $\omega^- \models u_k$. Hence there are three cases for the child X_k:

- either $\omega^+ \models u_k$ and $\omega^- \models \neg u_k$ (then $\phi_k^- \in \mathcal{F}^{Ch}_{\omega^+}$, but $\phi_k^+ \notin \mathcal{F}^{Ch}_{\omega^+}$);
- or $\omega^+ \models \neg u_k$ and $\omega^- \models u_k$ (then $\phi_k^+ \in \mathcal{F}^{Ch}_{\omega^-}$, but $\phi_k^- \notin \mathcal{F}^{Ch}_{\omega^-}$);
- or $\omega^+ \models \neg u_k$ and $\omega^- \models \neg u_k$, and $\mathcal{F}^{Ch}_{\omega^-} \cup \mathcal{F}^{Ch}_{\omega^+}$ does not contain any formula pertaining to variable X_k.

Now, it becomes clear that $\boldsymbol{\omega}^+(\Sigma)$ and $\boldsymbol{\omega}^-(\Sigma)$ only differ on components pertaining to children nodes of X_i and to X_i itself. □

Due to the specific structure of CP-nets, and since we have shown that a preference is only related to a variable node and their children nodes (Proposition 2), we have to consider the three following elementary cases:

- *Case a*: Two father nodes and a child node (see Fig 2(a)) (also Fig. 1);
- *Case b*: A father node and two children nodes (see Fig 2(b));
- *Case c*: A grandfather node, a father node and a child node (see Fig 2(c)).

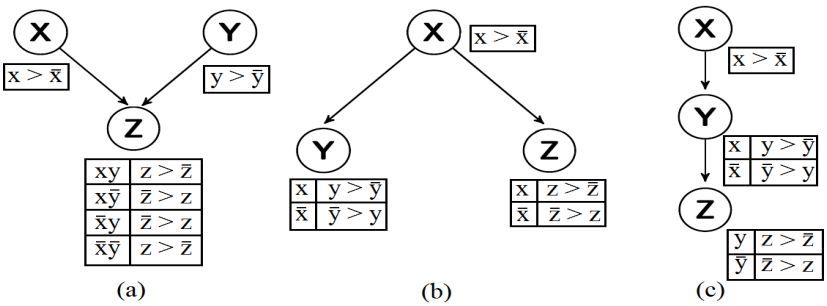


Fig. 2. Elementary cases of CP-nets

Then, any CP-net is a combination of these three elementary cases (with possibly more fathers or children). Considering these three basic structures, the following examples show in which case a particular order induced by $(\Sigma, \succ_\Sigma)$ fails to capture the ordering of interpretations induced by the CP-net.

Example 2: $V = \{X, Y, Z\}$ is the set of variables involved in the examples on Fig. 2. In these examples, preference constraints are as follows: $\phi_1 = x > \bar{x}, \phi_2 = y > \bar{y}, \phi_3 = (X \iff Y : z > \bar{z}, \neg(X \iff Y) : \bar{z} > z), \phi_4 = (x : z > \bar{z}, \bar{x} : \bar{z} > z)$, $\phi_5 = (x : y > \bar{y}, \bar{x} : \bar{y} > y)$ and $\phi_6 = (y : z > \bar{z}, \bar{y} : \bar{z} > z)$. The possibilistic logic bases obtained in the different examples in Fig 2 are:

- $\Sigma_a = \{\phi_1, \phi_2, \phi_3\}$: $\phi_1 = (x, \alpha_1), \phi_2 = (y, \alpha_2), \phi_3 = (((\neg(x \wedge y) \wedge \neg(\neg x \wedge \neg y)) \vee z) \wedge (\neg(x \wedge \neg y) \wedge \neg(\neg x \wedge y)) \vee \neg z), \alpha_3)$, and $\min(\alpha_1, \alpha_2) \succ_{\Sigma_a} \alpha_3$,
- $\Sigma_b = \{\phi_1, \phi_4, \phi_5\}$ with $\phi_4 = ((\neg x \vee z) \wedge (x \vee \neg z), \alpha_4), \phi_5 = ((\neg x \vee y) \wedge (x \vee \neg y), \alpha_5)$, and is such that $\alpha_1 \succ_{\Sigma_b} \max(\alpha_4, \alpha_5)$,
- $\Sigma_c = \{\phi_1, \phi_5, \phi_6\}$ with $\phi_6 = ((\neg y \vee z) \wedge (y \vee \neg z), \alpha_6)$ and $\alpha_1 \succ_{\Sigma_c} \alpha_5 \succ_{\Sigma_c} \alpha_6$.

Table 2. Possible alternative choices in Example 2

Ω	ϕ_1	ϕ_2	ϕ_3	ϕ_1	ϕ_4	ϕ_5	ϕ_1	ϕ_5	ϕ_6
xyz	1	1	1	1	1	1	1	1	1
$xy\bar{z}$	1	1	1-α_3	1	1-α_4	1	1	1	1-α_6
$x\bar{y}z$	1	1-α_2	1-α_3	1	1	1-α_5	1	1-α_5	1-α_6
$x\bar{y}\bar{z}$	1	1-α_2	1	1	1-α_4	1-α_5	1	1-α_5	1
$\bar{x}yz$	1-α_1	1	1-α_3	1-α_1	1-α_4	1-α_5	1-α_1	1-α_5	1
$\bar{x}y\bar{z}$	1-α_1	1	1	1-α_1	1	1-α_5	1-α_1	1-α_5	1-α_6
$\bar{x}\bar{y}z$	1-α_1	1-α_2	1	1-α_1	1-α_4	1	1-α_1	1	1-α_6
$\bar{x}\bar{y}\bar{z}$	1-α_1	1-α_2	1-α_3	1-α_1	1	1	1-α_1	1	1

Results are as follows:

- In the 1st case ($\mathcal{N}_a$), *symmetric Pareto* and *leximin* orders are able to capture the ordering of the CP-net exactly. Otherwise, the min order fails to distinguish between the interpretations $\{\bar{x}yz, \bar{x}y\bar{z}, \bar{x}\bar{y}z, \bar{x}\bar{y}\bar{z}\}$ and between $\{x\bar{y}\bar{z}, x\bar{y}z\}$.
- In the 2nd case ($\mathcal{N}_b$), *symmetric Pareto* order fails to capture the CP-net ordering exactly by leaving the two interpretations $\omega = x\bar{y}\bar{z}$ and $\omega' = \bar{x}\bar{y}\bar{z}$ non compared (while node X in the CP-net $\mathcal{N}_\lfloor$ ensures $x\bar{y}\bar{z} \succ_{\mathcal{N}} \bar{x}\bar{y}\bar{z}$). Otherwise the representation is exact. The associated vectors $\boldsymbol{\omega}(\Sigma) = (1, 1-\alpha_4, 1-\alpha_5)$ and $\boldsymbol{\omega'}(\Sigma) = (1-\alpha_1, 1, 1)$ are not comparable by *symmetric Pareto*. Indeed $\nexists\ \sigma$ s.t. $\boldsymbol{\omega}(\Sigma) \succ_{SP} \boldsymbol{\omega'}^{\sigma}(\Sigma)$, since $1-\alpha_1 < \min(1-\alpha_4, 1-\alpha_5)$ while $1 > \max(1-\alpha_4, 1-\alpha_5)$. Otherwise, the min order is able to compare these two interpretations $x\bar{y}\bar{z} \succ_{\min} \bar{x}\bar{y}\bar{z}$, but it fails to distinguish between the interpretations $\{\bar{x}yz, \bar{x}y\bar{z}, \bar{x}\bar{y}z, \bar{x}\bar{y}\bar{z}\}$ and between $\{x\bar{y}\bar{z}, xy\bar{z}\}$. But *leximin* is able here to capture the CP-net ordering exactly.
- In the 3rd case ($\mathcal{N}_c$), both *leximin* and min orders fail to capture the CP-net ordering: the two interpretations $\omega = x\bar{y}z$ and $\omega' = \bar{x}\bar{y}\bar{z}$ become comparable while the CP-net *cannot compare them*. Since $\boldsymbol{\omega}(\Sigma) = (1, 1-\alpha_5, 1-\alpha_6)$ and

$\boldsymbol{\omega'}(\Sigma) = (1-\alpha_1, 1, 1)$, with $\min(\boldsymbol{\omega}(\Sigma)) = 1-\alpha_5$, $\min(\boldsymbol{\omega'}(\Sigma)) = 1-\alpha_1$ and $1-\alpha_1 < 1-\alpha_5$, we have $\omega \succ_{lex} \omega'$ and $\omega \succ_{\min} \omega'$. But *symmetric Pareto* can capture the CP-net ordering exactly in this case.

To summarize, as observed in the Example, the *symmetric Pareto* order fails to compare two interpretations when the concerned variable has more than one child node as in *Case b* (Fig.2 (b)). Besides, in *Case c* (Fig.2 (c)) *leximin* and min break the incomparability of some interpretations in the CP-net.

6 Approaching CP-Net Preferences by Possibilistic Logic

As seen in Ex. 2 of Section 5, the *symmetric Pareto* relation is not fine-grained enough to capture the CP-net partial order in general, while the *lexi-min* order may make some CP-net-incomparable interpretations comparable. In this Section, we point out a class of CP-nets for which possibilistic logic with symbolic weights can capture the CP-net partial order exactly. First, we prove that any strict comparison obtained by *symmetric Pareto* is true for the CP-net order.

Proposition 3. *Let $\mathcal{N}$ be an acyclic CP-net and $(\Sigma, \succeq_\Sigma)$ be its associated partially ordered base. Let $\succeq_{SP}$ be the partial order associated to $(\Sigma, \succeq_\Sigma)$.*

$$\forall \omega, \omega' \in \Omega, \omega \succ_{SP} \omega' \;\Rightarrow\; \omega \succ_{\mathcal{N}} \omega'$$

Proof of Proposition 3
Suppose that $\omega \succ_{SP} \omega'$. This means that there exists a permutation σ of $\boldsymbol{\omega'}(\Sigma)$ such that when comparing the result of this permutation with $\boldsymbol{\omega}(\Sigma)$, the second vector is greater than or equal to, componentwise, the reordered one. There are two cases: either for any component, where there is no equality, the comparison between the two vectors is of the form $1 > 1-\alpha_{\sigma(i)}$, or there is at least one component where the comparison takes the form $1-\alpha_j > 1-\alpha_{\sigma(k)}$. This corresponds respectively to two different situations:

i) ω' falsifies more formulas in Σ than ω, and $\mathcal{F}_\omega \subset \mathcal{F}_{\omega'}$, where $\mathcal{F}_\omega$ (resp. $\mathcal{F}'_\omega$) denotes the set of nodes falsified by interpretation ω (resp. ω'). This corresponds to the first case above, where $\mathcal{F}_{\omega'} \setminus \mathcal{F}_\omega$ corresponds precisely to the violated formulas whose priority $\alpha_{\sigma(i)}$ is involved in the observed inequalities $1 > 1-\alpha_{\sigma(i)}$; it is known that $\mathcal{F}_\omega \subset \mathcal{F}_{\omega'}$ entails $\omega \succ_{\mathcal{N}} \omega'$.

ii) ω' falsifies at least one formula whose priority is greater than the one of another formula violated by ω, namely $1-\alpha_j > 1-\alpha_{\sigma(k)}$, equivalent to $\alpha_j < \alpha_{\sigma(k)}$. In fact, there is at least one component in $\boldsymbol{\omega'}(\Sigma)$ of the form $1-\alpha_{\sigma(r)}$ which is a minimal component among those in the two subvectors on which $\boldsymbol{\omega}(\Sigma)$ and $\boldsymbol{\omega'}(\Sigma)$ differ. It corresponds to a formula having maximal priority ($\alpha_{\sigma(r)}$) violated by ω' and not by ω. Now, the constraints $\alpha_j < \alpha_{\sigma(k)} \leq \alpha_{\sigma(r)}$ reveal that the nodes corresponding in the CP-nets to these priorities are related by a path in the CP-net linking an ancestor $X_{\sigma(r)}$ (having maximal priority) to a descendent X_j. The set of such paths can be associated with a chain of improving flips from ω' to ω, and thus $\omega \succ_{\mathcal{N}} \omega'$. □

We have noticed that there are cases where the *symmetric Pareto* order together with the possibilistic logic encoding does capture the CP-net ordering exactly. The following proposition indicates a class of CP-nets where it is indeed the case.

Proposition 4. *Let $\mathcal{N}$ be an acyclic CP-net with every node have at most one child node. Let $(\Sigma, \succeq_\Sigma)$ be its associated partially ordered base. Let $\succeq_{SP}$ be the partial order associated to $(\Sigma, \succeq_\Sigma)$. Then, $\forall \omega, \omega' \in \Omega, \omega \succ_{SP} \omega'$ iff $\omega \succ_{\mathcal{N}} \omega'$.*

Proof of Proposition 4

i) Suppose that $\omega \succ_{\mathcal{N}} \omega'$. We know that ω dominates ω' (i.e. $\omega \succ_{\mathcal{N}} \omega'$) if and only if there is a chain of worsening flips which consists of a change of the instantiation of one variable each time. This means that there exists a sequence $\omega_0, \cdots, \omega_k$ such that $\omega \succ \omega_0 \succ \cdots \succ \omega_k \succ \omega'$, where $\omega \succ \omega_0, \ldots, \omega_k \succ \omega'$ are ceteris paribus preferences. We have shown in Proposition 1 that such preference statements are related to the concerned variable (which corresponds here to the flip) and its children. Since we have supposed that each node has at most one child node, the associated evaluation vectors for every two interpretations in a chain of worsening flips differ on at most two components corresponding to the flipped variable and its child node. Since we give the priority to father node over the child node, the two interpretations are ordered by $\succ_{SP}$. So we have $\omega \succ_{SP} \omega_0 \succ_{SP} \cdots \succ_{SP} \omega_k \succ_{SP} \omega'$, and finally $\omega \succ_{SP} \omega'$ by transitivity.

ii) By Proposition 3, we have: if $\omega \succ_{SP} \omega'$ then $\omega \succ_{\mathcal{N}} \omega'$. □

We have also noticed on some examples that *leximin* order is more refined than the order induced by the considered CP-net. The following proposition establishes that any strict comparison obtained by a CP-net is also true in its possibilistic logic counterpart using *leximin* order:

Proposition 5. *Let $\mathcal{N}$ be an acyclic CP-net. Let $(\Sigma, \succeq_\Sigma)$ be its associated partially ordered base. Then: $\forall \omega, \omega' \in \Omega, \omega \succ_{\mathcal{N}} \omega' \Rightarrow \boldsymbol{\omega}(\Sigma) \succ_{lex} \boldsymbol{\omega'}(\Sigma)$*

Proof of Proposition 5

Since $\succ_{\mathcal{N}}$ is transitive, it is enough to prove that this is true for $\omega \succ_{\mathcal{N}} \omega'$ where there is one worsening flip which consists in a change of the instantiation of one variable, in the ceteris paribus preference style. By transitivity we get the general case where there is a chain of worsening flips since *leximin* order is also transitive. We have shown in Proposition 2 that such a ceteris paribus preference pertains to the concerned variable and its children. So for ω and ω', $\min(\{v_i \in \boldsymbol{\omega}(\Sigma)\} \cup \{v_i \in \boldsymbol{\omega'}(\Sigma)\}) \subseteq \{v_j \in (\boldsymbol{\omega'}(\Sigma)\} \setminus \{v_j \in (\boldsymbol{\omega}(\Sigma)\})$. Indeed the evaluation associated to the father node is smaller than any other evaluation associated with its children, and then the min will downrank the interpretation that violates the father node. So we have $\omega \succ_{lex} \omega'$. □

7 Related Work and Final Discussion

Possibilistic logic for preferences representation has been first advocated in [4, 5]. Its use with symbolic weights for approximating acyclic Boolean CP-nets [2] and

TCP-nets [12], has been discussed in [6, 7, 13]. Then, a representation of CP-net has been proposed using the symmetric Pareto order in [8, 9], and recalled in [10, 14] using leximin order. This representation has been presented as being faithful in the general case (without providing the proof). It turns out that the representation using the symmetric Pareto order is exact only in special cases. We have shown that it is indeed the case for the particular CP-nets where nodes have at most one child. We have also proved that in general it is a lower approximation, while the use of leximin order leads to an upper approximation.

Thus, the semantics of possibilistic logic that could lead to an exact representation of any (acyclic) CP-net in the general case is still to be found (if it exists). However, the partial ordering induced by the CP-net approach may appear somewhat questionable, as exemplified now, which in turn questions the possibility of an exact representation of the latter by means of an approach that handles preferences in a more global way.

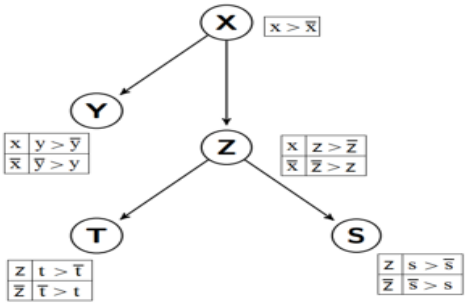


Fig. 3. CP-net related to Example 3

Example 3: Let us consider the CP-net of Fig. 3 on variables $V = \{X, Y, S, Z, T\}$. Let us consider the interpretations $\omega = xyz\bar{s}\bar{t}, \omega' = x\bar{y}\bar{z}\bar{s}\bar{t}, \omega'' = \bar{x}\bar{y}\bar{z}\bar{s}\bar{t}$ and $\omega''' = xy\bar{z}\bar{s}t$. We notice that ω violates the preferences at two grandson nodes S, T, but ω' violates the preferences at children nodes Y, Z. Moreover, ω'' violates the preference at the father node X and ω''' violates preference at a child Z and a grandson T. The CP-net order is such that $\omega \succ_{\mathcal{N}} \omega' \succ_{\mathcal{N}} \omega''$, $\omega \succ_{\mathcal{N}} \omega'''$, but it tells nothing on ω''' vs. ω'' and ω'. Thus, violating preferences at grandsons S, T (ω) is better than violating preferences at children nodes Y, Z (ω'), which is better than violating preferences at the father node X (ω''), in agreement with the CP-net implicit priorities. But it is troublesome that violating CP-net preferences at one child node Z and one grandson node T (ω''') is neither comparable with the violation of preference at the two children nodes Y, Z (ω'), let alone the father node X (ω''). This is not acknowledged by the possibilistic approach using *leximin* ordering.

8 Concluding Remarks

The interest for preference representation of the possibilistic logic framework relies first on the logical nature of the representation and constitutes an alternative

to the introduction of a preference relation inside the representation language, as in, e.g., [15]. Moreover, the possibilistic representation is expressive (see [10] for an introductory survey), and can capture partial orders thanks to the use of symbolic weights, without being obliged to impose greater priority weights to any preference (as it is the case for father node preferences in CP nets). Still much remains to be done. First, the question of an exact representation of any CP-net remains open. Moreover, an attempt has been made recently [10] for representing more general CP-theories [16] in the possibilistic logic approach (by introducing further inequalities between symbolic weights in order to take into account the CP-theory idea that some preferences hold irrespective of the values of some variables), where the leximin order seems to provide an upper approximation. This remains to be confirmed and developed further. Comparing CP-nets with Bayesian possibilistic nets would be also of interest.

Acknowledgements. The authors are grateful to Nic Wilson for useful comments on their previous workshop paper [10].

References

1. Domshlak, C., Hüllermeier, E., Kaci, S., Prade, H.: Preferences in AI: An overview. Artif. Intell. 175, 1037–1052 (2011)
2. Boutilier, C., Brafman, R.I., Domshlak, C., Hoos, H., Poole, D.: CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. J. Artificial Intelligence Research (JAIR) 21, 135–191 (2004)
3. Dubois, D., Prade, H.: Possibilistic logic: a retrospective and prospective view. Fuzzy Sets and Systems 144, 3–23 (2004)
4. Benferhat, S., Dubois, D., Prade, H.: Towards a possibilistic logic handling of preferences. Applied Intelligence 14, 303–317 (2001)
5. Benferhat, S., Dubois, D., Kaci, S., Prade, H.: Possibilistic logic representation of preferences: relating prioritized goals and satisfaction levels expressions. In: Proc. 15th. Europ. Conf. on Artificial Intelligence (ECAI 2002), Lyon, July 21-26, pp. 685–689. IOS Press (2002)
6. Dubois, D., Kaci, S., Prade, H.: CP-nets and possibilistic logic: Two approaches to preference modeling. Steps towards a comparison. In: Proc. of IJCAI 2005 Workshop on Advances in Preference Handling, Edinburg, July 31-August 1 (2005)
7. Dubois, D., Kaci, S., Prade, H.: Approximation of conditional preferences networks "CP-nets" in possibilistic logic. In: Proc. Inter. Conf. on Fuzzy Systems (FUZZ-IEEE), Vancouver, July 16-21, pp. 2337–2342 (2006)
8. Kaci, S., Prade, H.: Mastering the processing of preferences by using symbolic priorities. In: Proc. 18th Europ. Conf. on Artif. Intel. (ECAI 2008), pp. 376–380 (2008)
9. Kaci, S.: Working With Preferences: Less Is More. Springer (2012)
10. Dubois, D., Prade, H., Touazi, F.: Handling partially ordered preferences in possibilistic logic - A survey discussion. In: Proc. ECAI 2012 Workshop on Weighted Logics for AI, pp. 91–98 (2012)
11. Benferhat, S., Prade, H.: Encoding formulas with partially constrained weights in a possibilistic-like many-sorted propositional logic. In: Kaelbling, L.P., Saffiotti, A. (eds.) Proc. 19th Int. Joint Conf. on Artif. Intellig. (IJCAI 2005), Edinburgh, July 30-August 5, pp. 1281–1286 (2005)

12. Wilson, N.: Extending CP-nets with stronger conditional preference statements. In: Proc. 19th National Conference on Artificial Intelligence (AAAI 2004), pp. 735–741 (2004)
13. Kaci, S., Prade, H.: Relaxing ceteris paribus preferences with partially ordered priorities. In: Mellouli, K. (ed.) ECSQARU 2007. LNCS (LNAI), vol. 4724, pp. 660–671. Springer, Heidelberg (2007)
14. HadjAli, A., Kaci, S., Prade, H.: Database preference queries - A possibilistic logic approach with symbolic priorities. Ann. Math. Artif. Intell. 63, 357–383 (2011)
15. Bienvenu, M., Lang, J., Wilson, N.: From preference logics to preference languages, and back. In: Lin, F.Z., Sattler, U., Truszczynski, M. (eds.) Proc. 12th Int. Conf. on Principles of Knowled. Represent. and Reas. (KR 2010), Toronto, pp. 414–424. AAAI (2010)
16. Wilson, N.: Computational techniques for a simple theory of conditional preferences. Artif. Intell. 175, 1053–1091 (2011)

Many-Valued Modal Logic and Regular Equivalences in Weighted Social Networks*

Tuan-Fang Fan[1] and Churn-Jung Liau[2]

[1] Department of Computer Science and Information Engineering
National Penghu University of Science and Technology
Penghu 880, Taiwan
dffan@npu.edu.tw
[2] Institute of Information Science
Academia Sinica, Taipei 115, Taiwan
liaucj@iis.sinica.edu.tw

Abstract. Social network analysis is a methodology used extensively in social sciences. While classical social networks can only represent the qualitative relationships between actors, weighted social networks can describe the degrees of connection between actors. In classical social network, regular equivalence is used to capture the similarity between actors based on their linking patterns with other actors. Specifically, two actors are regularly equivalent if they are equally related to equivalent others. The definition of regular equivalence has been extended to regular similarity and generalized regular equivalence for weighted social networks. Recently, it was shown that social positions based on regular equivalence can be syntactically expressed as well-formed formulas in a kind of modal logic. Thus, actors occupying the same social position based on regular equivalence will satisfy the same set of modal formulas. In this paper, we will present analogous results for regular similarity and generalized regular equivalence based on many-valued modal logics.

Keywords: Weighted social network, regular similarity, generalized regular equivalence, many-valued modal logic.

1 Introduction

Social network analysis (SNA) is a methodology used extensively in social and behavioral sciences, as well as in political science, economics, organization theory, and industrial engineering [25,14,27]. Positional analysis of a social network tries to find similarities between nodes in the network [3,4,10,16,28]. In SNA, a category, called a *social role* or *social position*, is defined in terms of the similarities of the patterns of relations among the nodes, rather than the attributes of the nodes. One of the the most studied notions in the positional analysis of social networks is called *regular equivalence* [3,6,23,24]. According to Borgatti

* This work was partially supported by NSC (Taiwan) Grants: 101-2410-H-346-004-MY2 (T.F. Fan) and 99-2221-E-001-008-MY3 (C.J. Liau).

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 194–205, 2013.

and Everett [3], two actors are regularly equivalent if they are equally related to equivalent others. Interestingly, it was shown that social positions based on regular equivalence can be syntactically expressed as well-formed formulas (wff) in a kind of modal logic [21]. Thus, actors occupying the same social position based on regular equivalence will satisfy the same set of modal formulas.

In recent years, weighted social networks have also received considerable attention because they can represent both the qualitative relationships and the degrees of connection between nodes [2,11,12,15,22,26]. In [12], the notion of regular equivalence is extended to weighted social networks based on two alternative definitions of regular equivalence. While the two definitions are equivalent for ordinary networks, they induce different generalizations for weighted networks. The first generalization, called *regular similarity*, is based on the definition of regular equivalence as an equivalence relation that commutes with the underlying graph edges [4]. By the definition, regular similarity is a fuzzy relation that describes the degree of similarity between actors in the network. The second generalization, called *generalized regular equivalence*, is based on the definition of role assignment or coloring [16]. A role assignment (resp. coloring) is a mapping from the set of actors to a set of roles (resp. colors). The mapping is regular if actors assigned to the same role have the same roles in their neighborhoods. Consequently, generalized regular equivalence is an equivalence relation that can determine the role partition of actors in a weighted social network.

Due to the importance of weighted social networks, we would like to explore the logical characterizations of regular similarity and generalized regular equivalence. In this paper, many-valued modal logics are used to characterize these two kinds of relations. On one hand, we show that the truth values of many-valued modal logic formulas are invariant with respect to generalized regular equivalence. On the other hand, we show that the degree of the maximum regular similarity between any two actors is equal to the infimum equivalence degree of truth values of many-valued modal logic formulas in these two actors.

The remainder of this paper is organized as follows. In Section 2, we review some basic concepts about social networks, fuzzy relations, and positional analysis. In Sections 3 and 4, we present the logical characterizations of regular similarity and generalized regular equivalence respectively. Finally, we summarize the results in Section 5.

2 Preliminaries

2.1 Social Networks

Social networks are defined by actors and relations (or nodes and edges in terms of graph theory) [14]. A social network is defined as a relational structure $\mathfrak{N} = (U, (P_i)_{i\in I}, (R_j)_{j\in J})$, where the universe U is a *finite set* of actors, $P_i \subseteq U$ for all $i \in I$, and $R_j \subseteq U \times U$ for all $j \in J$. For each $x \in U$, the out-neighborhood and in-neighborhood of x with respect to a binary relation R, denoted respectively by Rx and $R^{-}x$, are defined as follows:

$$Rx = \{y \in U \mid (x, y) \in R\}, \tag{1}$$

$$R^{-}x = \{y \in U \mid (y, x) \in R\}. \tag{2}$$

If E is an equivalence relation on U and x is an actor, the E-equivalence class of x is equal to its neighborhood, i.e., $[x]_E = Ex = E^{-}x$. Note that the latter equality holds because of the symmetry of E. For any $X \subseteq U$, we denote by $[X]_E$ the set $\{[x]_E \mid x \in X\}$.

Several equivalence relations have been proposed for exploring the structural similarity between actors. Among them, regular equivalence has been extensively studied [3,4,10,16,28]. Although there are several definitions of regular equivalence, we only consider two of them in this paper. The first is given by Boyd and Everett [4], which states that an equivalence relation E is a *regular equivalence* with respect to a binary relation R if it commutes with R; i.e.,

$$E \cdot R = R \cdot E, \tag{3}$$

where $E \cdot R = \{(x, y) \mid \exists z \in U, (x, z) \in E \wedge (z, y) \in R\}$ is the composition of E and R. By this definition, if E is a regular equivalence with respect to R and $(x, y) \in E$, then for each $z \in Rx$ (resp. $R^{-}x$), there exists $z' \in Ry$ (resp. $R^{-}y$) such that $(z, z') \in E$. The property naturally leads to an alternative definition of regular equivalence based on role assignment [16], which states that an equivalence relation E is a regular equivalence with respect to a binary relation R if for $x, y \in U$,

$$(x, y) \in E \Rightarrow ([Rx]_E = [Ry]_E \text{ and } [R^{-}x]_E = [R^{-}y]_E). \tag{4}$$

According to this definition, if x and y are regularly equivalent, then they are connected to equivalent neighborhoods. Obviously, the above definitions are equivalent. Thus, we have the following definition.

Definition 1. *Let $\mathfrak{N} = (U, (P_i)_{i \in I}, (R_j)_{j \in J})$ be a social network and E be an equivalence relation on U; then E is a regular equivalence with respect to $\mathfrak{N}$ if*

1. *$(x, y) \in E$ implies $x \in P_i$ iff $y \in P_i$ for all $i \in I$; and*
2. *E is a regular equivalence with respect to R_j for all $j \in J$.*

By the definition, there may exist more than one regular equivalence for a given network. However, it has been shown that there always exists a maximum (i.e., coarsest) regular equivalence for a network [16].

2.2 Weighted Social Networks

Social networks can model the interactions and connections between actors. However, in most real-world networks, not all ties in a network have the same capacity. In fact, ties are often associated with weights that differentiate them in terms of their strength, intensity, or capacity [2]. Mathematically, we can use fuzzy sets and relations to model weighted social networks. Fuzzy sets are sets whose elements have degrees of membership [29]. The membership degrees are typically drawn from the unit interval $[0, 1]$. A *t-norm* operation on $[0, 1]$ is usually used

to define the intersection of fuzzy sets. A t-norm is a binary operation $\otimes$ on $[0,1]$ satisfying commutativity, associativity, non-decreasing in both arguments, and $1 \otimes c = c$ and $0 \otimes c = 0$ for all $c \in [0,1]$ [13]. The *residuum* of a t-norm $\otimes$ is a binary operation $\Rightarrow$ on $[0,1]$ defined as $a \Rightarrow b = \sup\{c \mid a \otimes c \leq b\}$ for all $a, b \in [0,1]$. Furthermore, the residuum defines its corresponding unary operation of *precomplement* $-c = c \Rightarrow 0$. In this paper, we mainly use the well-known Gödel t-norm $a \otimes b = \min(a, b)$. Hence, its corresponding residuum is defined by

$$a \Rightarrow b = \begin{cases} 1, \text{ if } a \leq b, \\ b, \text{ otherwise.} \end{cases} \tag{5}$$

and its corresponding precomplement is the Gödel negation defined by

$$-a = \begin{cases} 1, \text{ if } a = 0, \\ 0, \text{ otherwise.} \end{cases} \tag{6}$$

In addition, we use $a \Leftrightarrow b$ to denote $\min(a \Rightarrow b, b \Rightarrow a)$. It is easy to see that

$$a \Leftrightarrow b = \begin{cases} 1, & \text{if } a = b, \\ \min(a, b), & \text{otherwise.} \end{cases} \tag{7}$$

In fuzzy set theory, a fuzzy binary relation R on U can be characterized by its membership function $\mu_R : U \times U \mapsto [0,1]$. Obviously, a fuzzy binary relation is a generalization of a binary relation, so the upper-case letters R, S, T, etc., are used to denote both fuzzy and crisp relations. Since we only consider fuzzy binary relations in this paper, we call them fuzzy relations hereafter, and the term "binary relation" means crisp relations only. A fuzzy relation R is included in another fuzzy relation S, denoted by $R \subseteq S$, if $\mu_R(x, y) \leq \mu_S(x, y)$ for all $x, y \in U$. Several basic operations for binary relations can be easily generalized to fuzzy relations.

Definition 2. *Given two fuzzy relations R and S on U, the following fuzzy relations can be derived:*

1. *the identity relation Id:*

$$\mu_{Id}(x, y) = \begin{cases} 1, \text{ if } x = y, \\ 0, \text{ otherwise;} \end{cases} \tag{8}$$

2. *the converse of R, R^-:*

$$\mu_{R^-}(x, y) = \mu_R(y, x); \tag{9}$$

3. *the composition of R and S, $R \cdot S$:*

$$\mu_{R \cdot S}(x, y) = \sup_{z \in U} \min(\mu_R(x, z), \mu_S(z, y)); \tag{10}$$

4. *the union of R and S, $R \cup S$:*

$$\mu_{R \cup S}(x, y) = \max(\mu_R(x, y), \mu_S(x, y)); \tag{11}$$

5. the intersection of R and S, $R \sqcap S$:

$$\mu_{R \sqcap S}(x, y) = \min(\mu_R(x, y), \mu_S(x, y)). \tag{12}$$

The composition of R with itself k times is denoted by R^k and the transitive closure of R is defined as $R^\infty = \bigcup_{k \geq 1} R^k$. Based on these definitions, the equivalence relation is generalized to the similarity relation in fuzzy set theory.

Definition 3. *A fuzzy relation R is called a similarity relation if it satisfies the following properties:*

- *reflexivity: $Id \subseteq R$,*
- *symmetry: $R = R^-$, and*
- *(sup-min) transitivity: $R^2 \subseteq R$.*

Intuitively, if R is a similarity relation, then $R(x, y)$ specifies the degree of similarity between x and y. The set of all similarity relations on a domain U forms a lattice. The meet and join of two similarity relations R and S in the lattice are defined as $R \sqcap S = R \cap S$ and $R \sqcup S = (R \cup S)^\infty$ respectively.

Given the basic notations of fuzzy sets and relations, a *weighted social network* can be defined as a structure $\mathfrak{N} = (U, (P_i)_{i \in I}, (R_j)_{j \in J})$, where U is a *finite set* of actors, P_i is a fuzzy subset of U for all $i \in I$, and R_j is a fuzzy relation on U for all $j \in J$. Although the membership degrees of fuzzy sets and relations may be any real numbers from the unit interval, in practice, the weights in a weighted social network are rarely irrational numbers. Thus, we further assume that the membership degrees of P_i's and R_j's are all rational. Hereafter, when we mention the unit interval $[0, 1]$, it really means $[0, 1] \cap \mathbb{Q}$.

We have presented the definition of regular equivalence in two ways. While these two definitions coincide for classical social networks, they behave quite differently in weighted social networks. Based on the commutativity between the similarity relation and the underlying fuzzy relations, we can induce a kind of structural similarity between actors. Such similarity is called a regular similarity. Formally, a similarity relation S is called a *regular similarity* with respect to a fuzzy relation R if it commutes with R, i.e., $S \cdot R = R \cdot S$. Hence, the regular similarity of a weighted social network can be defined as follows.

Definition 4. *Let $\mathfrak{N} = (U, (P_i)_{i \in I}, (R_j)_{j \in J})$ be a weighted social network and S be a similarity relation on U; then S is a regular similarity with respect to $\mathfrak{N}$ if*

1. *for all $x, y \in U$, $\mu_S(x, y) \leq \min_{i \in I}(\mu_{P_i}(x) \Leftrightarrow \mu_{P_i}(y))$; and*
2. *S is a regular similarity with respect to R_j for all $j \in J$.*

In [12], it is shown that regular similarities are closed with respect to the usual join of similarity relations. Thus, we can define the maximum (with respect to fuzzy inclusion) regular similarity of a weighted social network.

On the other hand, based on the notion of role assignment, we can derive the concept of *generalized regular equivalence* (GRE). Although regular similarity is a fuzzy relation, GRE gives a crisp partition of actors in a weighted network. To define GRE, we need to consider the neighborhoods of the nodes in

weighted networks. Let R be a fuzzy relation on U. Then, for each $x \in U$, the out-neighborhood and in-neighborhood of x, still denoted by Rx and $R^{-}x$ respectively, are two fuzzy subsets of U with the following membership functions:

$$\mu_{Rx}(y) = \mu_R(x, y), \tag{13}$$

$$\mu_{R^{-}x}(y) = \mu_R(y, x), \tag{14}$$

for any $y \in U$. Let F be a fuzzy subset of U and E be an equivalence relation on U. Then, $[F]_E$ is a fuzzy subset of the quotient set $U/E = \{[x]_E \mid x \in U\}$ with the following membership function:

$$\mu_{[F]_E}(X) = \max_{y \in X} \mu_F(y) \tag{15}$$

for any $X \in U/E$. Then, an equivalence relation E is a GRE with respect to a fuzzy relation R if $(x, y) \in E$ implies that

$$[Rx]_E = [Ry]_E \text{ and } [R^{-}x]_E = [R^{-}y]_E. \tag{16}$$

Hence, the GRE of a weighted social network can be defined as follows.

Definition 5. *Let $\mathfrak{N} = (U, (P_i)_{i \in I}, (R_j)_{j \in J})$ be a weighted social network and E be an equivalence relation on U; then E is a GRE with respect to $\mathfrak{N}$ if*

1. *$(x, y) \in E$ implies $\mu_{P_i}(x) = \mu_{P_i}(y)$ for all $i \in I$; and*
2. *E is a GRE with respect to R_j for all $j \in J$.*

Like regular equivalences, GRE is also closed with respect to the usual join of equivalence relations. Thus, we can define the maximum (the coarsest) GRE of a weighted social network. In addition, we use $x \equiv^{g}_{\mathfrak{N}} y$ to denote that (x, y) is in the maximum GRE of the network.

3 Regular Similarity and Modal Logic

As in the case of ordinary social networks, we would like to find a logical language that can characterize regular similarity in weighted social networks. One candidate for such logic is the many-valued modal logic since its formulas typically have a degree of truth in the unit interval. Many-valued modal logic is the extension of modal logic with the underlying propositional logic being replaced by many-valued logic. There exist a variety of many-valued logics depending on their choices of syntax and semantics. In particular, a family of [0,1]-valued logics is introduced in [13], where the most important instances are Łukasiewicz, Gödel and product logics. These logic systems are interpreted in algebraic structures called residuated lattices such that continuous t-norms and their corresponding residua in the algebras are taken as the truth functions of the conjunction and the implication respectively. For the purpose of this paper, we mainly consider the Gödel logic. Hence, we introduce the many-valued modal logic $G(\Diamond)$ as follows.

The alphabet of $G(\diamond)$ is close to that of classical modal logic. However, to represent partial truth, $G(\diamond)$ is extended with the set of truth constants $\overline{c}$ for each rational $c \in [0,1]$. Thus, the alphabet of $G(\diamond)$ consists of a set of propositional symbols PV, the set of truth constant $\{\overline{c} \mid c \text{ is a rational in } [0,1]\}$, a set of relational symbols REL, the Boolean connectives $\wedge$ and $\rightarrow$, the relational converse symbol $^-$, and the modality-forming symbol $\langle\rangle$. The set of wffs of $G(\diamond)$ is the smallest set containing PV and the set of truth constants that satisfies the following conditions:

- if φ is a wff and α is a relational symbol, then $\langle\alpha\rangle\varphi$ and $\langle\alpha^-\rangle\varphi$ are wffs;
- if φ and ψ are wffs, then $\varphi \wedge \psi$ and $\varphi \rightarrow \psi$ are wffs.

We abbreviate $\varphi \rightarrow \overline{0}$ as $\neg\varphi$, $(\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$ as $\varphi \leftrightarrow \psi$, and $((\varphi \rightarrow \psi) \rightarrow \psi) \wedge ((\psi \rightarrow \varphi) \rightarrow \varphi)$ as $\varphi \vee \psi$.

A Kripke model for $G(\diamond)$ is $\mathfrak{M} = (W, (R_\alpha)_{\alpha \in REL}, V)$, where W is a set of possible worlds, for each $\alpha \in REL$, R_α is a fuzzy relation on W, and $V : W \times PV \mapsto [0,1]$ is a truth assignment for evaluating the truth value of each propositional symbol in each world. Let Φ denote the set of all $G(\diamond)$ wffs. Then, the truth assignment V can be iteratively extended to a function $V : W \times \Phi \mapsto [0,1]$ in the following way:

1. $V(w, \overline{c}) = c$
2. $V(w, \varphi \wedge \psi) = \min(V(w,\varphi), V(w,\psi))$,
3. $V(w, \varphi \rightarrow \psi) = (V(w,\varphi) \Rightarrow V(w,\psi))$,
4. $V(w, \langle\alpha\rangle\varphi) = \sup_{u \in W} \min(R_\alpha(w,u), V(u,\varphi))$,
5. $V(w, \langle\alpha^-\rangle\varphi) = \sup_{u \in W} \min(R_\alpha^-(w,u), V(u,\varphi))$.

Obviously, we can derive the following results:

1. $V(w, \neg\varphi) = -V(w,\varphi)$,
2. $V(w, \varphi \leftrightarrow \psi) = (V(w,\varphi) \Leftrightarrow V(w,\psi))$,
3. $V(w, \varphi \vee \psi) = \max(V(w,\varphi), V(w,\psi))$.

For a given weighted social network $\mathfrak{N} = (U, (P_i)_{i \in I}, (R_j)_{j \in J})$, we define a $G(\diamond)$ language with the basic symbols $PV = \{p_i \mid i \in I\}$ and $REL = \{\alpha_j \mid j \in J\}$. The weighted social network $\mathfrak{N}$ is then transformed into a Kripke model for the language $\mathfrak{M}_\mathfrak{N} = (U, (R_j)_{j \in J}, V)$, where V is defined by $V(x, p_i) = \mu_{P_i}(x)$ for all $x \in U$ and $i \in I$ and R_j denotes R_{α_j} for $j \in J$. Let $\Phi_\mathfrak{N}$ denote the set of wffs of this $G(\diamond)$ language. Then, the logical characterization of regular similarity is presented as the following theorem.

Theorem 1. *Let $\mathfrak{N} = (U, (P_i)_{i \in I}, (R_j)_{j \in J})$ be a weighted social network, S be its maximum regular similarity, and $\mathfrak{M}_\mathfrak{N} = (U, (R_j)_{j \in J}, V)$ be the corresponding Kripke model. Then, for any $x, y \in U$, we have*

$$\mu_S(x,y) = \inf_{\varphi \in \Phi_\mathfrak{N}} (V(x,\varphi) \Leftrightarrow V(y,\varphi)). \tag{17}$$

The theorem represents a fuzzified version of the semantic invariance of any wffs with respect to the regular similarity. Intuitively, a wff can be seen as the descriptive property of actors. Thus, $V(x, \varphi)$ is the degree that the property φ is true of the actor x. Hence, the theorem essentially means that the more similar two actors are, the more equivalent their descriptive properties are. While regular similarity is characterized by fuzzy equivalence, there is a special case that is useful for the crisp partition of the network. We say that two actors, x and y, are $G(\Diamond)$-equivalent with respect to $\mathfrak{N}$ if for all $\varphi \in \Phi_{\mathfrak{N}}$, $V(x, \varphi) = V(y, \varphi)$.

Corollary 1. *Let S be the maximum regular similarity of $\mathfrak{N}$. Then, for any actors $x, y \in U$, $\mu_S(x, y) = 1$ iff x and y are $G(\Diamond)$-equivalent with respect to $\mathfrak{N}$.*

4 Generalized Regular Equivalence and Modal Logic

To present the logical characterization of GRE, we have to extend $G(\Diamond)$ with the projection operator Δ. As the operator was first used by Baaz for Gödel logic, it is also called the Baaz Delta [1]. The projection operator $\Delta : [0, 1] \mapsto \{0, 1\}$ is defined by

$$\Delta a = \begin{cases} 1, \text{ if } a = 1, \\ 0, \text{ otherwise.} \end{cases} \tag{18}$$

Let $G_{\triangle}(\Diamond)$ be the extension of $G(\Diamond)$ with the unary projection connective $\triangle$. Then, the formation rules for the wffs of $G_{\triangle}(\Diamond)$ are those for $G(\Diamond)$ and the following one:

– if φ is a wff, then $\triangle\varphi$ is a wff.

The definition of Kripke models for $G_{\triangle}(\Diamond)$ remains the same as that for $G(\Diamond)$, but, for a model $\mathfrak{M} = (W, (R_\alpha)_{\alpha \in REL}, V)$, the truth assignment V satisfies the additional condition:

$$V(w, \triangle\varphi) = \Delta(V(w, \varphi)).$$

The projection connective can be also defined by an involutive negation. A negation operator is involutive if it satisfies the double negation law. As it can be easily seen, the negation in $G(\Diamond)$ is not involutive. That is, $\neg\neg\varphi \leftrightarrow \varphi$ is not a 1-tautology in $G(\Diamond)$. The extensions of many-valued logic with an additional involutive negation have been extensively studied in [5,9]. Let $G_{\sim}(\Diamond)$ be the extension of $G(\Diamond)$ with the involutive negation $\sim$. Then, in addition of the formation rules for wffs of $G(\Diamond)$, we also have the following rule:

– if φ is a wff, then $\sim\varphi$ is a wff,

and, for a model $\mathfrak{M} = (W, (R_\alpha)_{\alpha \in REL}, V)$, the truth assignment V satisfies the following condition:

$$V(w, \sim\varphi) = 1 - V(w, \varphi).$$

In $G_{\sim}(\Diamond)$, we can define $\triangle\varphi$ as the abbreviation of $\neg\sim\varphi$. Thus, $G_{\sim}(\Diamond)$ is more expressive than $G_{\triangle}(\Diamond)$. However, they can both characterize GRE in the same way.

Given a weighted social network $\mathfrak{N} = (U, (P_i)_{i\in I}, (R_j)_{j\in J})$, we can define the basic symbols in PV and REL from $\mathfrak{N}$ and the Kripke model $\mathfrak{M}_{\mathfrak{N}}$ in the same way as in the preceding section. Let $\Phi^{\Delta}_{\mathfrak{N}}$ and $\Phi^{\sim}_{\mathfrak{N}}$ denote the set of such $G_{\Delta}(\Diamond)$ and $G_{\sim}(\Diamond)$ wffs respectively. We say that two actors $x, y \in U$ are $G_{\Delta}(\Diamond)$-equivalent (resp. $G_{\sim}(\Diamond)$-equivalent) with respect to $\mathfrak{N}$ iff for any $\varphi \in \Phi^{\Delta}_{\mathfrak{N}}$ (resp. $\varphi \in \Phi^{\sim}_{\mathfrak{N}}$), $V(x, \varphi) = V(y, \varphi)$. Then, we have the following theorem.

Theorem 2. *Let $\mathfrak{N} = (U, (P_i)_{i\in I}, (R_j)_{j\in J})$ be a weighted social network. Then, for any $x, y \in U$, the following three statements are equivalent:*

1. $x \equiv^{g}_{\mathfrak{N}} y$;
2. *x and y are $G_{\sim}(\Diamond)$-equivalent with respect to $\mathfrak{N}$;*
3. *x and y are $G_{\Delta}(\Diamond)$-equivalent with respect to $\mathfrak{N}$.*

Combining this theorem with Corollary 1 establishes a relationship between GRE and regular similarity.

Corollary 2. *Let S be the maximum regular similarity of $\mathfrak{N}$. Then, for any actors $x, y \in U$, $x \equiv^{g}_{\mathfrak{N}} y$ implies $\mu_S(x, y) = 1$.*

4.1 Special Case: Hybrid Social Network

In weighted social networks, each actor is associated with fuzzy attributes and connected with other actors by fuzzy relations. However, there is a special kind of weighted social networks in which the attributes are crisp although the relations between actors are still weighted. For example, in a friendship network, the strength of ties determines a degree of friendship between actors. Hence, the friendship relation is modeled as a fuzzy relation. However, the personal attributes of each actor, such as gender, age, and occupation, etc. may be all crisp. To model such networks, we say that a weighted social network $\mathfrak{N} = (U, (P_i)_{i\in I}, (R_j)_{j\in J})$ is a *hybrid social network* if for each $i \in I$, P_i is a crisp subset of U.

For GRE of hybrid social networks, in addition to the characterizations above, we can provide an alternative characterization based on quantitative modal logic (QML) [17,18,19,20]. QML is a modal version of the possibilistic logic, which is a logic for reasoning about uncertainty based on possibility theory [7,8]. In the theory, a *possibility distribution* on the universe U is a function $\pi : U \mapsto [0, 1]$ and two measures on U, called possibility and necessity measures and denoted by Π and N respectively, can be derived from π. Formally, $\Pi, N : 2^U \mapsto [0, 1]$ are defined as

$$\Pi(X) = \sup_{u\in X} \pi(u), \tag{19}$$

$$N(X) = 1 - \Pi(\overline{X}), \tag{20}$$

where $\overline{X}$ is the complement of X with respect to U. In a weighted social network, each actor's out-neighborhood and in-neighborhood with respect to a fuzzy relation can be seen as possibility distributions. In other words, the membership

functions in equations (13) and (14) correspond to these possibility distributions. The modalities in QML can represent lower bounds of the possibility measures of propositions, where each proposition is interpreted as a subset of possible worlds (or actors).

The alphabet of QML is the same as that of classical modal logic. However, the formation rule for modal formulas is modified as follows:

– if φ is a wff, α is a relational symbol, and c is a rational in $[0,1]$ then $\langle\alpha_{\geq c}\rangle\varphi$, $\langle\alpha_{>c}\rangle\varphi$, $\langle\alpha^{-}_{\geq c}\rangle\varphi$, and $\langle\alpha^{-}_{>c}\rangle\varphi$ are all wffs.

For the semantics, a Kripke model of QML is a special kind of $G_{\Delta}(\Diamond)$ model $\mathfrak{M} = (W, (R_\alpha)_{\alpha\in REL}, V)$ with the restriction of $V : W \times PV \mapsto \{0,1\}$. Then, the satisfaction of modal formulas are defined as follows:

1. $\mathfrak{M}, w \models \langle\alpha_{\geq c}\rangle\varphi$ iff $\sup_{u\in|\varphi|} \mu_{R_\alpha}(w,u) \geq c$,
2. $\mathfrak{M}, w \models \langle\alpha_{>c}\rangle\varphi$ iff $\sup_{u\in|\varphi|} \mu_{R_\alpha}(w,u) > c$,
3. $\mathfrak{M}, w \models \langle\alpha^{-}_{\geq c}\rangle\varphi$ iff $\sup_{u\in|\varphi|} \mu_{R^{-}_\alpha}(w,u) \geq c$,
4. $\mathfrak{M}, w \models \langle\alpha^{-}_{>c}\rangle\varphi$ iff $\sup_{u\in|\varphi|} \mu_{R^{-}_\alpha}(w,u) > c$,

where $|\varphi| = \{x \in W \mid \mathfrak{M}, x \models \varphi\}$ is the truth set of φ. According to equations (13) and (14), $\sup_{u\in|\varphi|} \mu_{R_\alpha}(w,u)$ and $\sup_{u\in|\varphi|} \mu_{R^{-}_\alpha}(w,u)$ represent the possibility measures of φ derived from the out-neighborhood and in-neighborhood of w respectively.

Unlike many-valued modal logics, QML is a two-valued multi-modal logic. The main feature is that the numerical possibility measures are internalized by using modal operators. For a given hybrid social network $\mathfrak{N} = (U, (P_i)_{i\in I}, (R_j)_{j\in J})$, we define a QML language with the basic symbols $PV = \{p_i \mid i \in I\}$ and $REL = \{\alpha_j \mid j \in J\}$. Then, $\mathfrak{N}$ is transformed into a QML model $\mathfrak{M}_\mathfrak{N} = (U, (R_j)_{j\in J}, V)$, where V is defined by $V(x, p_i) = 1$ iff $x \in P_i$ for all $x \in U$ and $i \in I$ and R_j denotes R_{α_j} for $j \in J$. We say that two actors, x and y, are QML-equivalent with respect to $\mathfrak{N}$ if for all φ in the given QML language, $(\mathfrak{M}_\mathfrak{N}, x \models \varphi$ iff $\mathfrak{M}_\mathfrak{N}, y \models \varphi)$.

Theorem 3. *Let $\mathfrak{N} = (U, (P_i)_{i\in I}, (R_j)_{j\in J})$ be a hybrid social network. Then, for any $x, y \in U$, $x \equiv^{g}_{\mathfrak{N}} y$ iff x and y are QML-equivalent with respect to $\mathfrak{N}$.*

5 Conclusion

The notion of regular equivalence has been studied extensively in social network analysis and found many applications in block modeling, network clustering, role or position identification, and so on. To represent the intensity of ties and interactions between actors, traditional social networks have been generalized to weighted social networks. There exist different, but equivalent, definitions of regular equivalences in the literature. However, when generalized to weighted social networks, these definitions may result in different notions of similarity. Two kinds of generalizations based on Gödel t-norm, called regular similarity and generalized regular equivalence (GRE), have been proposed in [12].

In this paper, we show that many-valued modal logic can characterize regular similarity or GRE in a weighted social network. By viewing a weighted social network as a model of the many-valued modal logic, similar or equivalent actors satisfy the set of modal logic formulas to the same degree. Specifically, the many-valued modal logic $G(\Diamond)$ based on the Gödel t-norm characterizes regular similarity in the sense that the degree of similarity between two actors is equal to the fuzzy equivalence of the actors' truth degrees for any formulas of the logic. Also, the extensions of $G(\Diamond)$ with the involutive negation or projection operators characterize GRE in the sense that two actors are equivalent iff they satisfy any formula of the logics to the same degree. For a special kind of weighted social network, called hybrid social network, where the actors' attributes are all crisp although their ties may be weighted, we also show its logical characterization by a modal version of possibilistic logic, QML.

References

1. Baaz, M.: Infinite-valued Gödel logics with 0-1-projections and relativizations. In: Hájek, P. (ed.) Gödel 1996: Logical Foundations of Mathematics, Computer Science, and Physics. Lecture Notes Logic, vol. 6, pp. 23–33. Springer (1996)
2. Barrat, A., Barthelemy, M., Pastor-Satorras, R., Vespignani, A.: The architecture of complex weighted networks. Proceedings of the National Academy of Sciences 101(11), 3747–3752 (2004)
3. Borgatti, S.P., Everett, M.G.: The class of all regular equivalences: Algebraic structure and computation. Social Networks 11(1), 65–88 (1989)
4. Boyd, J.P., Everett, M.G.: Relations, residuals, regular interiors, and relative regular equivalence. Social Networks 21(2), 147–165 (1999)
5. Cintula, P., Klement, E.P., Mesiar, R., Navara, M.: Fuzzy logics with an additional involutive negation. Fuzzy Sets and Systems 161(3), 390–411 (2010)
6. Doreian, P.: Measuring regular equivalence in symmetric structures. Social Networks 9(2), 89–107 (1987)
7. Dubois, D., Lang, J., Prade, H.: Possibilistic logic. In: Gabbay, D.M., Hogger, C.J., Robinson, J.A. (eds.) Handbook of Logic in Artificial Intelligence and Logic Programming. Nonmonotonic Reasoning and Uncertain Reasoning, vol. 3, pp. 439–513. Clarendon Press, Oxford (1994)
8. Dubois, D., Prade, H.: An introduction to possibilistic and fuzzy logics. In: Smets, P., Mamdani, A., Dubois, D., Prade, H. (eds.) Non-Standard Logics for Automated Reasoning, pp. 253–286. Academic Press (1988)
9. Esteva, F., Godo, L., Hájek, P., Navara, M.: Residuated fuzzy logics with an involutive negation. Archive for Mathematical Logic 39(2), 103–124 (2000)
10. Everett, M.G., Borgatti, S.P.: Regular equivalences: General theory. Journal of Mathematical Sociology 18(1), 29–52 (1994)
11. Fan, T.F., Liau, C.J., Lin, T.Y.: Positional analysis in fuzzy social networks. In: Proceedings of the 3rd IEEE International Conference on Granular Computing, pp. 423–428 (2007)
12. Fan, T.F., Liau, C.J., Lin, T.Y.: A theoretical investigation of regular equivalences for fuzzy graphs. International Journal of Approximate Reasoning 49(3), 678–688 (2008)
13. Hájek, P.: Metamathematics of Fuzzy Logic. Kluwer Academic Publisher (1998)

14. Hanneman, R.A., Riddle, M.: Introduction to Social Network Methods. University of California, Riverside (2005)
15. Kumpula, J.M., Onnela, J.-P., Saramäki, J., Kaski, K., Kertész, J.: Emergence of communities in weighted networks. Physical Review Letters 99, 228701-1-228701-4 (2007)
16. Lerner, J.: Role assignments. In: Brandes, U., Erlebach, T. (eds.) Network Analysis. LNCS, vol. 3418, pp. 216–252. Springer, Heidelberg (2005)
17. Liau, C.J., Lin, I.P.: Quantitative modal logic and possibilistic reasoning. In: Proceedings of the 10th European Conference on Artificial Intelligence, pp. 43–47 (1992)
18. Liau, C.J., Lin, I.P.: Proof methods for reasoning about possibility and necessity. International Journal of Approximate Reasoning 9(4), 327–364 (1993)
19. Liau, C.J., Lin, I.P.: Reasoning about higher order uncertainty in possibilistic logic. In: Komorowski, J., Raś, Z.W. (eds.) ISMIS 1993. LNCS (LNAI), vol. 689, pp. 316–325. Springer, Heidelberg (1993)
20. Liau, C.J., Lin, I.P.: Possibilistic reasoning - a mini-survey and uniform semantics. Artificial Intelligence 88(1-2), 163–193 (1996)
21. Marx, M., Masuch, M.: Regular equivalence and dynamic logic. Social Networks 25(1), 51–65 (2003)
22. Nair, P.S., Sarasamma, S.: Data mining through fuzzy social network analysis. In: Proc. of the 26th International Conference of North American Fuzzy Information Processing Society, San Diego, California, pp. 251–255 (2007)
23. Pattison, P.E.: The analysis of semigroups of multirelational systems. Journal of Mathematical Psychology 25, 87–117 (1982)
24. Pattison, P.E.: Algebraic Models for Social Networks. Cambridge University Press (1993)
25. Scott, J.: Social Network Analysis: A Handbook, 2nd edn. SAGE Publications (2000)
26. Toivonen, R., Kumpula, J.M., Saramäki, J., Onnela, J.-P., Kertész, J., Kask, K.: The role of edge weights in social networks: modelling structure and dynamics. In: Proceedings of SPIE 6601(1): Noise and Stochastics in Complex Systems and Finance, pp. B1–B8 (2007)
27. Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications. Cambridge University Press (1994)
28. White, D.R., Reitz, K.P.: Graph and semigroup homomorphisms on netwoks and relations. Social Networks 5(1), 143–234 (1983)
29. Zadeh, L.A.: Fuzzy sets. Information and Control 8(3), 338–353 (1965)

Zero-Probability and Coherent Betting: A Logical Point of View

Tommaso Flaminio[1], Lluis Godo[2], and Hykel Hosni[3]

[1] Dipartimento di Scienze Teoriche e Applicate, Università dell'Insubria
Via Mazzini 5, 21100 Varese, Italy
tommaso.flaminio@uninsubria.it
[2] Artificial Intelligence Research Institute (IIIA - CSIC)
Campus de la Univ. Autònoma de Barcelona s/n, 08193 Bellaterra, Spain
godo@iiia.csic.es
[3] Scuola Normale Superiore,
Piazza dei Cavalieri 7 Pisa, Italy and
CPNSS – London School of Economics, UK
h.hosni@gmail.com

Abstract. The investigation reported in this paper aims at clarifying an important yet subtle distinction between (i) the logical objects on which measure theoretic probability can be defined, and (ii) the interpretation of the resulting values as rational degrees of belief. Our central result can be stated informally as follows. Whilst all subjective degrees of belief can be expressed in terms of a probability measure, the converse doesn't hold: probability measures can be defined over linguistic objects which do not admit of a meaningful betting interpretation. The logical framework capable of expressing this will allow us to put forward a precise formalisation of de Finetti's notion of *event* which lies at the heart of the Bayesian approach to uncertain reasoning.

1 Introduction: The Epistemic Structure of De Finetti's Betting Interpretation

De Finetti's theory of subjective probability is well-known and widely scrutinized in the literature (cf. [2]), so we will review only on those aspects which are directly relevant to our present purposes[1].

Let $\theta_1, \ldots, \theta_n$ be events of interest. De Finetti's *betting problem* is the choice that an idealised agent called *bookmaker* must make when publishing a *book*, i.e. when making an assignment $B = \{(\theta_i, \beta_i) : i = 1, \ldots, n\}$ in which each event of interest θ_i is given value $\beta_i \in [0, 1]$. Once a book has been published, a *gambler* can place bets $S_i \in \mathbb{R}$ on any event θ_i by paying $S_i\beta_i$ to the bookmaker. In return for this payment, the gambler will receive S_i, if θ_i obtains and nothing otherwise. Note that "betting on θ_i" effectively amounts, for the gambler, to choosing a real-valued S_i which determines the amount payable to the bookmaker[2].

[1] The reader who wishes to consult the originals is referred to [2,3,5,6].

[2] In order to avoid potential distortions arising from the diminishing value of money, de Finetti invokes the "rigidity hypothesis" to the effect that S_i should be small.

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 206–217, 2013.

De Finetti's construction of the betting problem proceeds by forcing the bookmaker to write *fair betting odds* for any given book B. To this end, two modelling assumptions are built into the problem, namely (i) the bookmaker must accept any number of bets on B and (ii) when betting on θ_i, gamblers can choose the sign of the stakes S_i, thereby possibly (and unilaterally) imposing a payoff swap to the bookmaker. Taken jointly, conditions (i-ii) force the bookmaker to publish books with zero-expectation, for doing otherwise may offer gamblers the possibility of making a sure profit, possibly by swapping payoffs. As the game is zero-sum, this is equivalent to forcing the bookmaker into sure loss. The Dutch Book theorem states that this possibility is avoided exactly when the bookmaker chooses betting odds which are probabilities.

This line of argument presupposes an *epistemic structure* which de Finetti mentions only in passing in his major contributions to this topic [3,4,5]. A more direct, albeit very informal, reference to the point appears in [6]. For reasons that will be apparent in a short while, the underlying epistemic structure of the betting problem is fundamental to understanding the notion of *event*:

> [T]he characteristic feature of what I refer to as an "event" is that the circumstances under which the event will turn out to be "verified" or "disproved" have been fixed in advance. [6] (p. 150)

This very informal characterisation echoes the characterisation de Finetti gives of random quantities –of which events are special cases. A random quantity is a "well-determined" unknown, namely one which is so formulated as "to rule out any possible disagreement on its actual value, for instance, as it might arise when a bet is placed on it." ([5], Section 2.10.4).

The epistemic structure implicit in the betting framework clearly builds on the presupposition that at the time of betting bookmakers *and* gamblers ignore the truth value of the event on which they are betting, i.e. they agree that, say $v(\theta)$ is undefined. Yet, for the bet to be meaningful, i.e. payable at all, players must also agree on the conditions which will *decide* the truth value of θ. This implies that a betting interpretation of probability is meaningful only for those sentences whose truth value is presently (at the time of betting) undecided, but which the players know that will eventually be decided. Now, there are certainly well-formed formulas escaping this restriction, so probability functions defined on them cannot have a betting interpretation.

Before introducing the logical framework that will formalise this in Section 2, let us pause for a second to appreciate why the interpretation of probability which arises in this context is clearly subjective. Whether a sentence qualifies as an *event* depends crucially on the *state of information* of the individuals involved in the betting problem. Compare this with the logical, measure-theory inspired, characterisation of probability functions which is derived under the tacit assumption that the agent's state of information is empty, that is to say the set of events includes all possible sentences. This assumption will be relaxed in our framework and indeed this will lead us to generalise the scope of the representation theorem of probability functions on sentences by introducing a refinement of the notion of probability functions which we call *bet functions*

and we denote by $Bet(\cdot)$. In particular, we shall be interested in characterising sentences of SL in such a way that the resulting definitions of *facts* and *events* (Section 3.2) will give us enough structure to prove that $Bet(\cdot)$ so defined is *consistent* in the sense of de Finetti (Section 4) and to show that its extension to *inaccessible sentences* preserves consistency (Section 5). Section 6 concludes by pointing to the future work which we envisage within the framework fleshed out in this paper.

2 Background

Let $L = \{p_1, \ldots, p_n\}$ be a finite set of propositional variables, and let $SL = \{\theta, \phi, \ldots\}$ be the set of sentences built as usual from L in the language of classical propositional logic. Denote by AT^L be the set of maximally elementary conjunctions of L, that is the set of sentences of the form $\alpha = p_1^{\epsilon_1} \wedge p_2^{\epsilon_2} \wedge \ldots \wedge p_n^{\epsilon_n}$, with $\epsilon_i \in \{0, 1\}$ and where $p_i^1 = p_i$ and $p_i^0 = \neg p_i$, for $i = 1, \ldots, n$.

Note that the Lindenbaum algebra[3] on SL is a finite Boolean algebra and hence it is atomic. In particular the elements of AT^L exactly correspond the atoms of the Lindenbaum algebra.

AT^L is in 1-1 correspondence with the set $\mathbb{V}$ of (classical) valuations on L. This implies that there is a unique valuation satisfying $v(\alpha) = 1$ namely $v_\alpha(p_i^{\epsilon_i}) = \epsilon_i$ for $1 \leq i \leq n$. Conversely, given a valuation $v \in \mathbb{V}$ there exists a unique atom $\alpha \in AT^L$ such that $v(\alpha) = 1$. Now let

$$M_\theta = \{\alpha \in AT^L \mid \alpha \models \theta\},$$

where $\models$ denotes the classical Tarskian consequence. Since there exists a unique valuation satisfying α, say v_α, by definition of $\models$ it must be the case that $v_\alpha(\theta) = 1$. Thus

$$M_\theta = \{\alpha \in AT^L \mid v_\alpha(\theta) = 1\}.$$

This framework is sufficient to provide a very general representation theorem for probability functions.

Theorem 1 (Paris 1994)

1. *Let P be a probability function on SL.*[4] *Then the values of P are completely determined by the values it takes on $AT^L = \{\alpha_1, \ldots, \alpha_J\}$, as fixed by the vector*

$$\langle P(\alpha_1), P(\alpha_2), \ldots, P(\alpha_J)\rangle \in \mathbb{D}^L = \{\boldsymbol{a} \in \mathbb{R}^J \mid \boldsymbol{a} \geq 0, \sum_{i=1}^{J} a_i = 1\}.$$

[3] Recall that the Lindenbaum algebra over L is the quotient set $SL/\equiv$, where $\equiv$ is the logical equivalence relation (defined as $\theta_1 \equiv \theta_2$ iff $\models \theta \leftrightarrow \theta_2$), with the operations induced by the classical conjunction, disjunction and negation connectives.

[4] $P : SL \to [0, 1]$ is a probability function on sentences if (i) $P(\top) = 1$, (ii) $P(\theta_1 \vee \theta_2) = P(\theta_1) + P(\theta_2)$ if $\models \neg(\theta_1 \wedge \theta_2)$, and (iii) $P(\theta_1) = P(\theta_2)$ if $\models \theta_1 \leftrightarrow \theta_2$.

2. *Conversely, fix* $\boldsymbol{a} = \langle a_1, \ldots, a_J \rangle \in \mathbb{D}^L$ *and let* $P' : SL \to [0,1]$ *be defined by*

$$P'(\theta) = \sum_{i:\alpha_i \in M_\theta} a_i. \tag{1}$$

Then P' *is a probability function.*

In words, Theorem 1 shows that every probability function arises from distributing the unit mass of probability across the $J = 2^n$ atoms of the Lindenbaum algebra generated by $L = \{p_1, \ldots, p_n\}$.

Our goal is to refine this result by isolating a class of sentences on which, we argue, there should be no distribution of epistemically significant mass. More specifically, we aim at building a framework in which those probabilities which bear a meaning as *betting quotients* can be formally distinguished from those which do not. Central to achieving this will be a rigorous definition of de Finetti's notion of *event*, which will be distinguished from the related notion of *fact*. Under certain conditions, all sentences in SL will either be events or facts. Under more general conditions a third class of inaccessible sentences will feature in SL. The central result of this paper can be intuitively phrased as establishing that *probabilities which are defined on sentences which are not events can only be given trivial values*. Trivial, as we will shortly see, means one of two things. Either a sentence can (coherently) be given only its truth value (and this characterises betting on facts), or it should be given 0. This means that the "uncertainty mass" is really concentrated only on events, for which we provide a formal definition.

3 Formal Preliminaries: Information Frames, Facts and Events

In what follows, we denote subsets of SL by capital Greek letters $\Gamma, \Delta, \ldots$, and the classical Tarskian consequence is denoted by either $\models$ or Cn depending on whether its relational or operational definition is more suited to the specific to the context. Recall that a (total, classical) valuation is a function $v : L \to \{0,1\}$ which extends uniquely to the sentences in SL. A total valuation represents a "fully informed" epistemic state since it allows agents to assign a truth-value (either 1 or 0) to any sentence of SL. However, an epistemic state determined by a set Γ of sentences (the ones known to be true), will permit an assignment of truth-values 1 or 0 only to some subset of sentences. In fact, each Γ uniquely determines a three-valued map on SL, $e_\Gamma : SL \to \{0,1,u\}$, defined as

$$e_\Gamma(\theta) = \begin{cases} 1 & \text{if } \theta \in Cn(\Gamma), \\ 0 & \text{if } \neg\theta \in Cn(\Gamma), \\ u & \text{otherwise.} \end{cases} \tag{2}$$

where the new value u reads as *unknown*.

Notice that partial evaluations are not truth-functional. Note also that, if $\Gamma \subseteq \Gamma'$ then $Cn(\Gamma) \subseteq Cn(\Gamma')$. From now on, we will say that a mapping

$e : SL \to \{0, 1, u\}$ is a *partial evaluation* whenever there exists $\Gamma \subseteq SL$ such that $e = e_\Gamma$.

Given two partial valuations e, e', we say that e' extends e, written $e \subseteq e'$, when the class of formulas which e sends into $\{0, 1\}$ is included into that one which e' sends into $\{0, 1\}$. Note that if $e = e_\Gamma$ and $e' = e_{\Gamma'}$ then

$$e \subseteq e' \Leftrightarrow \Gamma \subseteq \Gamma'. \tag{3}$$

By a *theory* we mean a deductively closed subset of SL. So, Γ is a theory if and only if $Cn(\Gamma) = \Gamma$. We denote the set of theories on L by $\mathbf{T}$. Let us finally recall that a theory $\Gamma \in \mathbf{T}$ is *maximally consistent* iff for every $\theta \in SL$, either $\Gamma \models \theta$, or $\Gamma \models \neg\theta$. Note also that for any maximally consistent $\Gamma \in \mathbf{T}$, there exists a (total) valuation $v \in \mathbb{V}$ such that for all $\theta \in SL$, $e_\Gamma(\theta) = v(\theta)$.

Definition 1 (Determined sentences). *We say that $\Gamma \subseteq SL$* determines *$\theta \in SL$, written $\Gamma \succ \theta$ if and only if, $\forall p_i \in Var(\theta)$, $e_\Gamma(p_i) \in \{0, 1\}$.*

Definition 2 (Decided sentences). *We say that $\Gamma \subseteq SL$* decides *$\theta \in SL$, written $\Gamma \rhd \theta$ if and only if $e_\Gamma(\theta) \in \{0, 1\}$.*

It is clear that for all $\Gamma \subseteq SL$ and $\theta \in SL$, if $\Gamma \succ \theta$ then $\Gamma \rhd \theta$ as well. Furthermore, as remarked above, if $\Gamma \in \mathbf{T}$ is maximally consistent, then $\Gamma \succ \theta \Leftrightarrow \Gamma \rhd \theta$. The following are immediate consequences of the above definitions.

Proposition 1. *For all $\Gamma \subseteq SL$, and for all $\theta, \varphi \in SL$, the following hold:*

1. *$\Gamma \succ \theta$ iff $\Gamma \succ \neg\theta$; $\Gamma \rhd \theta$ iff $\Gamma \rhd \neg\theta$.*
2. *If $\Gamma \rhd \theta$, and $\Gamma \rhd \varphi$, then $\Gamma \rhd \theta \circ \varphi$ for all $\circ \in \{\wedge, \vee, \to\}$.*
3. *If $\Gamma \rhd \theta$, $\Gamma \not\rhd \varphi$, and $e_\Gamma(\theta) = 0$ then $\Gamma \not\succ \theta \circ \varphi$ for every $\circ \in \{\wedge, \vee, \to\}$, but $\Gamma \rhd \theta \wedge \varphi$ and $\Gamma \rhd \theta \to \varphi$, and in particular $e_\Gamma(\theta \wedge \varphi) = 0$, $e_\Gamma(\theta \to \varphi) = 1$.*
4. *If $\Gamma \rhd \theta$, $\Gamma \not\rhd \varphi$, and $e_\Gamma(\theta) = 1$ then $\Gamma \not\succ \theta \circ \varphi$ for every $\circ \in \{\wedge, \vee, \to\}$, but $\Gamma \rhd \theta \vee \varphi$, $\Gamma \rhd \varphi \to \theta$ and $\Gamma \rhd \theta \to \varphi$, and in particular $e_\Gamma(\theta \vee \varphi) = e_\Gamma(\varphi \to \theta) = 1$.*

3.1 Information Frames

Definition 3 (Information frame). *An* information frame *$\mathcal{F}$ is a pair $\langle W, R\rangle$ where W is a non-empty subset of partial valuations defined as in Equation (2) and R is a binary transitive relation on W.*

Remark 1. Since each partial valuation is uniquely determined by a $\Gamma \subseteq SL$, we can freely use $w_1, w_2, \ldots$ to denote either subsets of SL or their associated partial valuations, depending on which interpretation suits best the specific context. As a consequence of Equation (3) the inclusion $w \subseteq w'$ is always defined.

We interpret $w_i \in W$ as an agent's *state of information*, i.e. the sentences (equivalently, the partial valuation) which capture all and only the information available to an agent who finds itself in state w_i. Under this interpretation the relation R models the agent's *possible transitions* among information states. For reasons that will soon be apparent, we always require R to be transitive. As more structure is needed further restrictions on R will be considered.

Definition 4. *Let $\mathcal{F} = \langle W, R\rangle$ be an information frame. We say that $\mathcal{F}$ is*

- Monotone *if $(w, w') \in R$ implies $w \subseteq w'$.*
- Complete *if $w \subseteq w'$ implies $(w, w') \in R$.*

Under our interpretation, monotonicity captures the idea that agents can only learn new information, but never "unlearn" the old one. In addition, monotonicity implies that the dynamics of information is stable in the sense that once a formula is either determined or decided at state w (i.e. it is given a binary truth-value), this remains fixed at any information state reachable from w. Hence if $w \succ \phi$, then there cannot exist $(w, w') \in R$ such that $w' \not\succ \phi$. Completeness ensures that the agent will learn all the possible consistent refinements to its current information state. So, if $(w, w') \notin R$, there exists θ such that $w' \succ \theta$ and $w \succ \neg\theta$. Finally, note that if $\mathcal{F}$ is monotonic and complete then obviously R coincides with set-inclusion among states (equivalently, sets of sentences).

3.2 Facts and Events

The following definition captures the differences among facts, events and inaccessible sentences in a monotone information frame.

Definition 5. *Let $\langle W, R\rangle$ be a monotone information frame, let $w \in W$, and let $\theta \in SL$. We say that θ is a* w-fact *if $w \rhd \theta$.*

On the other hand, if $w \not\rhd \theta$, we say that θ is:

- *a* w-event *if for every (total) valuation V extending w there exists w' with $(w, w') \in R$ such that $w' \rhd \theta$ and $w'(\theta) = V(\theta)$.*
- w-inaccessible *if for every (total) valuation V and every world w' such that $w'(\theta) = V(\theta)$, $(w, w') \notin R$.*

We shall respectively denote by $\mathcal{F}(w)$, $\mathcal{E}(w)$ and $\mathcal{I}(w)$ the class of w-facts, w-events, and w-inaccessible sentences, for some information frame $\langle W, R\rangle$ and some $w \in W$.

The following proposition sums up some key properties of the sets $\mathcal{F}(w)$, $\mathcal{E}(w)$ and $\mathcal{I}(w)$.

Proposition 2. *Let $\langle W, R\rangle$ be a monotone information frame, and let $w \in W$. Then the following hold:*

1. *The structure $\langle \mathcal{F}(w), \wedge, \neg, \bot\rangle$ is a Boolean algebra.*
2. *If w is a total valuation, then $SL = \mathcal{F}(w)$, while if $w = \emptyset$ is the empty valuation, then $\mathcal{F}(w) = \emptyset$.*
3. *If $\langle W, R\rangle$ is complete, then $\langle \mathcal{E}(w), \wedge, \neg, \bot\rangle$ is a Boolean algebra.*
4. *If $\langle W, R\rangle$ is complete, then for all $w \in W$, $SL = \mathcal{F}(w) \cup \mathcal{E}(w)$. Therefore, in particular, if $\langle W, R\rangle$ is complete, then $\mathcal{I}(w) = \emptyset$.*
5. *If $\mathcal{I}(w) \neq \emptyset$, then for every w' such that its corresponding valuation is total, $(w, w') \notin R$.*

It is worth noticing that in arbitrary monotone information frameworks one cannot ensure that sentences which are neither w-facts nor w-events, are w-inaccesible, so that the sets $\mathcal{F}(w)$, $\mathcal{E}(w)$, $\mathcal{I}(w)$ form a partition of SL. As we will discuss in further detail in the concluding section, it is surprisingly difficult to find natural properties on frames which ensure the rather desirable property that $SL = \mathcal{F}(w) \cup \mathcal{E}(w) \cup \mathcal{I}(w)$. When the information framework is also complete then we trivially get this condition since $\mathcal{I}(w) = \emptyset$.

4 Formalising the Betting Problem

Next we formalise a notion of Dutch book in our generalised framework.

Definition 6. *Let $\langle W, R\rangle$ be an information frame, and let $\Gamma = \{\theta_1, \ldots, \theta_n\}$. A* book *is any mapping $B : \Gamma \to [0, 1]$. Then we further define:*

- *for $w \in W$, the book B is said to be* w-Dutch *iff there exist $S_1, \ldots, S_n \in \mathbb{R}$ such that for every $w' \in W$ such that $w' \rhd \theta_i$ for every i, and $(w, w') \in R$,*

$$\sum_{i=1}^{n} S_i(w'(\theta_i) - B(\theta_i)) < 0;$$

- *the book B is said to be* w-coherent*, or* non-w-Dutch*, if B is not w-Dutch;*
- *the book B is said to be a* w-book*, if each formula $\theta_i \in \Gamma$ is a w-event.*

For w-books, being w-Dutch is a notion that collapses to the usual case. In fact if all the θ_i's are w-events, by definition, each possible evaluation of θ_i is accessible from w, and hence the *extra* requirement that the book be w-Dutch is redundant. On the other hand, a w-coherent w-book can be extended to more general books satisfying w-coherence, as shown by the following result.

Theorem 2. *Let (W, R) be a monotone information frame, let $w \in W$ and let $B : \theta_i \in \Gamma \mapsto \beta_i \in [0, 1]$ be a w-coherent w-book. Let φ be a sentence which is not a w-event and consider the book $B' = B \cup \{(\varphi, \alpha)\}$. Then:*

(1) B' is w-coherent iff $\alpha = w(\varphi)$, in case φ is a w-fact.
(2) B' is w-coherent iff $\alpha = 0$, in case φ is w-inaccessible.

Proof: (1). ($\Rightarrow$). Suppose, to the contrary, that $\alpha \neq w(\varphi)$, and in particular suppose that $w(\varphi) = 1$, so that $\alpha < 1$. Then, the gambler can secure a sure win by betting a positive S on φ. In this case in fact, since the information frame is monotonic by the definition of w-book, $w(\varphi) = 1$ holds in every world w' accessible from w. Thus the gambler pays $S \cdot \alpha$ in order to surely receive S in any such w'. Conversely, if $w(\varphi) = 0$, then, under the absurd hypothesis, $\alpha > 0$ and in that case it is easy to see that a sure-winning choice for the gambler consists in swapping payoffs with the bookmaker, i.e. to bet a negative amount of money on φ.

($\Leftarrow$). Let $S_1, \ldots, S_n, S$ be a system of bets on on $\theta_1, \ldots, \theta_n, \varphi$. Since B is coherent, there exists a w' accessible from w that realizes every θ_i, and such that

$$\sum_{i=1}^{n} S_i(\beta_i - w'(\theta_i)) = 0.$$

Since φ is a w-fact and w' is accessible from w, it follows that $w'(\varphi) = w(\varphi) = \alpha$. Therefore one also has

$$\left(\sum_{i=1}^{n} S_i(\beta_i - w'(\theta_i))\right) + S(\alpha - w'(\varphi)) = 0$$

and hence B' is also w-coherent.

(2). ($\Rightarrow$). Suppose that $\alpha > 0$. By contract, the bettor is accepting to pay a positive stake $S > 0$ on φ, and this means that the he must pay $\alpha \cdot S$ to the bookmaker, thus occurring in a sure loss since φ will not be decided in any world w' accessible from w.
($\Leftarrow$). Since B is w-coherent and since by hypothesis $\alpha = 0$, B' extends B in way which is trivial in the following sense: any gambler betting strictly positive stakes $S_1, \ldots, S_n, S$ on B' will pay to the bookmaker $\sum_i S_i\alpha_i + S\alpha = \sum_i S_i\alpha_i + 0$. And since φ is w inaccessible, in every world w' accessible from w, she will receive $\sum_i S_i w'(\theta_i)$. Hence the coherence of B' follows from the coherence of B. □

The following example illustrates that w-coherent w-books cannot be characterised, in general, within the standard axiomatic framework for probability.

Example 1. Let $L = \{p, q\}$ with the following intuitive interpretation:

- p reads "the electron ε has position π";
- q reads "the electron ε has energy η".

Suppose further that our agent is in a state w such that the truth value of both p and q are unknown. In the usual quantum mechanics interpretation, an agent in w may either learn the position of ε, or its energy, but not both. This gives rise to the information frame depicted in Figure 1 where we may assume the following conditions hold:

$w_1 \rhd p,\ w_1 \not\rhd q$, and $w_1(p) = 0$;	$w_2 \rhd p,\ w_2 \not\rhd q$, and $w_2(p) = 1$;
$w_3 \rhd q,\ w_3 \not\rhd p$, and $w_3(q) = 0$;	$w_4 \rhd q,\ w_4 \not\rhd p$, and $w_4(q) = 1$;
$w_5 \rhd p, q$, and $w_5(p) = w_5(q) = 1$	$w_6 \rhd p, q$, and $w_5(p) = w_5(q) = 0$.
$w_7 \rhd p, q$, and $w_7(p) = 0, w_7(q) = 1$	$w_8 \rhd p, q$, and $w_8(p) = 1, w_8(q) = 0$.

It is immediate to see that p and q are w-events, but $p \wedge q$ is not. In fact, for instance, due to the inaccessibility of w_5, the valuation v mapping p and q to 1 has no correspondence in the worlds which are accessible from w. Analogously, $\neg p \wedge q$, $p \wedge \neg q$ and $\neg p \wedge \neg q$ are not w-events either.

Each probability assignement which coherently assigns a value to $p \wedge q$ returns $P(p \wedge q) = 0$. In fact either $p \wedge q$ turns out to be realized in an accessible state

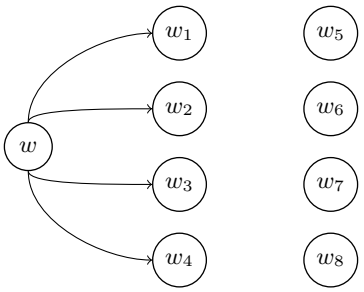


Fig. 1. Heisenberg's principle allows for the information frame to be such that states w_1, w_2, w_3, w_4 are reachable from w. Any world in which both variables are decided, namely w_5, w_6, w_7 and w_8, are not accessible from w.

(i.e. in w_1, or in w_3) in which it turns out to be false, or it turns out to be true, but in the world w_5 which is not accessible. Therefore, by an argument entirely analogous to the proof of Theorem 2, every assignment giving a positive value β to $p \wedge q$ would lead to a sure loss for the bookmaker.

Compare this with the standard measure-theoretic approach. In particular let $\mathbf{L}_2$ be the 16 element Lindenbaum algebra generated by the variables p and q with atoms $p \wedge q$, $\neg p \wedge q$, $p \wedge \neg q$, and $\neg p \wedge \neg q$. In the absence of the structure imposed by information frames, it would be very natural to assume a uniform probability distribution over the atoms of $\mathbf{L}_2$, thereby mapping $p \wedge q$ into a strictly positive value and therefore exposing the bookmaker to sure loss for the bookmaker.

5 Betting on Inaccessible Sentences

Example 1 illustrates that an otherwise standard probability assignment on the atoms of $\mathbf{L}_2$ may lead to sure loss because of the *inaccessibility* of w_5. The purpose of this section is to show that de Finetti's own coherence criterion fully applies when the information frame shared by the bookmaker and gamblers are *complete*, so that no sentence is inaccessible.

Definition 7 (Bet functions). *Let $\langle W, R\rangle$ be a monotone information frame, and $w \in W$. We say that a partial function $Bet : SL \to [0,1]$ satisfying:*

$$Bet(\theta) = \begin{cases} w(\theta) \in \{0,1\}, & \text{if } \theta \in \mathcal{F}(w) \\ 0, & \text{if } \theta \in \mathcal{I}(w) \end{cases} \tag{4}$$

*is a w-*bet function *if in addition it satisfies:*

- *$Bet(\theta) = Bet(\varphi)$, for all $\theta, \varphi \in \mathcal{E}(w)$ such that $\models \theta \leftrightarrow \varphi$,*
- *for all $\theta, \varphi, (\theta \vee \varphi) \in \mathcal{E}(w) \cup \mathcal{F}(w)$ in the domain of Bet such that $\theta \models \neg\varphi$,*

$$Bet(\theta \vee \varphi) = Bet(\theta) + Bet(\varphi) \tag{5}$$

– *$Bet(\theta)$ is not defined on each $\theta \in SL \setminus (\mathcal{E}(w) \cup \mathcal{F}(w) \cup \mathcal{I}(w))$.*

The conditions in (4) capture the (obvious) formalisation of the intuitive remarks put forward at the end of Section 2 which we now generalise to possibly incomplete frames, i.e. such that for some $w \in W$, $\mathcal{I}(w) \neq \emptyset$. The condition expressed by (5) clearly captures the additivity of the betting functions.

In order to characterise inaccessible sentences, we will be working with the corresponding partial valuations and we will identify, for the sake of notational simplicity, states with (partial) valuations.

Definition 8. *Let w, w' be partial valuations. We say that w and w' are* incompatible *(and we will write $w \perp w'$) if $\exists p$ such that $w \rhd p, w' \rhd p$, and $w(p) \neq w'(p)$.*

For a fixed w and $\Gamma \subseteq SL$ let $Var(\Gamma)$ be the set of propositional variables occurring in Γ, we define $S(\Gamma, w)$ to be the set of worlds $w' \in W$ such that:

(1) $(w, w') \in R$
(2) for all $p \notin Var(\Gamma)$, $w'(p) = u$
(3) there exists a total valuation v such that $\forall \theta \in \Gamma$, $v(\theta) = w'(\theta)$

We call the set $S(\Gamma, w)$ the *w-decisive set for Γ*. The idea is that $S(\Gamma, w)$ captures the minimal set of accessible worlds from w where all sentences of Γ are decided, and no other sentences except for those that necessarily follow from Γ. States belonging to the w-decisive set for Γ are *logically independent* in the following sense: for any set of formulas Γ, and for every $w \in W$, either $S(\Gamma, w)$ is empty, or $w' \perp w''$ for each $w', w'' \in S(\Gamma, w)$, i.e., by Definition 8, $w' \cup w'' \vdash \perp$.[5]

The following easily proved proposition sums up interesting properties of w-decisive sets.

Proposition 3. *Let $\langle W, R \rangle$ be a monotone information frame, $w \in W$, $\Gamma \subseteq SL$. Then the following hold:*

1. *If $\Gamma \cap \mathcal{I}(w) \neq \emptyset$, then $S(\Gamma, w) = \emptyset$;*
2. *If $\Gamma \subseteq \mathcal{E}(w)$, then $S(\Gamma, w) \neq \emptyset$;*

Let $\langle W, R \rangle$ be an information frame, $w \in W$, and $\Gamma \subseteq \mathcal{E}(w) \cup \mathcal{F}(w) \cup \mathcal{I}(w)$. Further, let $\Gamma' = \Gamma \cap (\mathcal{E}(w) \cup \mathcal{F}(w))$. Finally, let $\pi : S(\Gamma', w) \to [0, 1]$ satisfy $\sum_{w' \in S(\Gamma', w)} \pi(w') = 1$, and define $Bet'_\pi(\cdot) : \Gamma' \subseteq SL \to [0, 1]$ by

$$Bet'_\pi(\theta) = \sum_{w' \in S(\Gamma', w)} \pi(w') \cdot w'(\theta),$$

for all $\theta \in \Gamma'$.

[5] Note that if $\mathcal{I}(w) \neq \emptyset$, w-bets cannot be characterised as distributions on AT^L. As pointed out in Section 2 above, in fact, the formulas in AT^L correspond to *total* valuations. But by Proposition 2, whenever $\mathcal{I}(w) \neq \emptyset$, each w' corresponding to a total valuation must be such that $(w, w') \notin R$.

The map Bet'_π is extended to a partial map Bet_π over Γ by the coherence criterion we proved in Theorem 2. Hence, for each $\theta \in \Gamma$,

$$Bet_\pi(\theta) = \begin{cases} Bet'_\pi(\theta) & \text{if } \theta \in \Gamma', \\ 0 & \text{if } \theta \in \mathcal{I}(w). \end{cases} \tag{6}$$

The following is then easily proved.

Theorem 3. *Let Γ, w and π be as above, and let Bet_π be defined by (6). Then Bet_π is a w-bet function.*

Proof: Bet_π restricted to w-events and w-facts of Γ is clearly normalised and additive in the sense of Definition 7. In addition, for $\theta \in \mathcal{F}(w)$, $w(\theta) = w'(\theta)$ for each $w' \in S(\Gamma, w)$, and hence we have: (i) if $w(\theta) = 1$, then $Bet_\pi(\theta) = \sum_{w' \in S(\Gamma,w)} \pi(w') \cdot w'(\theta) = \sum_{w' \in S(\Gamma,w)} \pi(w') = 1$; (ii) if $w(\theta) = 0$, $Bet_\pi(\theta) = \sum_{w' \in S(\Gamma,w)} \pi(w') \cdot 0 = 0$. Therefore in any case, $Bet_\pi(\theta) = w(\theta)$ for each $\theta \in \mathcal{F}(w)$, and hence $Bet_\pi(\top) = 1$ holds. □

We close the section by stating two easily proved results which illustrate how the notion of w-coherence arises from w-bets. The notion of w-coherence will be the focus of future work.

Theorem 4. *Let Γ be any set of formulas, and let $B : \Gamma \to [0, 1]$ be a book. Then the following are equivalent:*

(1) B is w-coherent,
(2) There exists a w-bet function Bet on SL extending B.
(3) There exists a probability measure P on on the Lindenbaum algebra generated by $\Gamma \cap (\mathcal{E}(w) \cup \mathcal{F}(w))$ extending B on $\Gamma \cap (\mathcal{E}(w) \cup \mathcal{F}(w))$.

Proof: We are going to sketch the proof of (1) ⇔ (2).

(1) ⇒ (2). If B is w-coherent, then so is the book B^- obtained by restricting B to the formulas in $\Gamma' = \Gamma \cap (\mathcal{E}(w) \cup \mathcal{F}(w))$. Since Γ' does not contain w-inaccessible formulas, B^- is coherent and hence a standard argument (see for instance [8, Theorem 2]) shows that B' is coherent iff one can find a probability distribution π on $S(\Gamma', w)$. Then the map Bet_π defined through (6) satisfies (2).
(2) ⇒ (1). Let Bet' the partial mapping on SL defined by restricting Bet on $\mathcal{E}(w)$. Then the claim easily follows from Theorem 2. □

The above theorem shows that the usual characterization of coherence can be recovered asking for the information frame to be monotone and complete.

Corollary 1. *Let $\langle W, R\rangle$ be monotone and complete, with $w \in W$. Let $\Gamma \subseteq SL$, and let $B : \Gamma \to [0, 1]$. Then the following are equivalent:*

(1) B is w-coherent,
(2) B is coherent,
(3) There exists a w-bet function Bet on SL extending B,
(4) There exists a probability P on SL extending B.

6 Conclusions and Future Work

We have introduced a logical framework capable of making explicit the implicit epistemic structure that lies at the very heart of the Bayesian representation of uncertainty. As a central step towards achieving this we distinguished facts, events and inaccessible sentences with the understanding that the betting framework underlying the subjective interpretation of probability demands that genuine uncertainty be expressed only on events. The ensuing logical framework leads to a significant refinement of the classical (logical) representation of probability functions recalled in Section 1. In this spirit, Theorem 3 shows that consistent subjective degrees of belief are the subset of probability values which arise from what we call betting functions.

In further work we will tackle the question at a higher level of generality, namely by showing how Theorem 1 can be in fact derived within our framework as a special case of a more general result which involves defining bet functions over suitable quotient algebras. The idea, roughly speaking, is to capture the requirement that a specific set of sentences (events) should be given all the unit mass by factoring a Lindenbaum algebra over the ideal generated by the set of w-facts, for some $w \in W$. This will provide a suitable basis for giving a pure measure-theoretic account of subjective probability with its underlying epistemic structure. One obstacle to achieving this full generality is currently represented by our unsuccessful attempts to provide natural conditions under which SL is partitioned by facts, events and inaccessible formulas.

Acknowledgements. Flaminio acknowledges partial support of the Italian project FIRB 2010 (RBFR10DGUA_002). Godo acknowledges partial support of the Spanish projects AT (CONSOLIDER CSD2007-0022, INGENIO 2010) and EdeTRI (TIN2012-39348-C02-01). Flaminio and Godo acknowledge partial support of the IRSES project MaToMUVI (PIRSES-GA-2009-247584).

References

1. Burris, S., Sankappanavar, H.P.: A course in Universal Algebra. Springer, New York (1981)
2. Coletti, G., Scozzafava, R.: Probabilistic Logic in a Coherent Setting. Trends in Logic, vol. 15. Kluwer (2002)
3. de Finetti, B.: Sul significato soggettivo della probabilità. Fundamenta Mathematicae 17, 289–329 (1931)
4. de Finetti, B.: La prévision: ses lois logiques, ses sources subjectives. Annales de l'Institut Henri Poincaré 7(1), 1–68 (1937)
5. de Finetti, B.: Theory of Probability, vol. 1. John Wiley and Sons (1974)
6. de Finetti, B.: Philosophical Lectures on Probability. Synthese Library, vol. 340. Springer (2008)
7. Paris, J.B.: The Uncertain Reasoner's Companion: A Mathematical Perspective. Cambridge University Press (1994)
8. Paris, J.B.: A note on the Dutch Book method. In: De Cooman, G., Fine, T., Seidenfeld, T. (eds.) Proc. of ISIPTA 2001, Ithaca, NY, USA, pp. 301–306. Shaker Pub. Company (2001), `http://www.maths.manchester.ac.uk/~jeff/`

Conditional Random Quantities and Iterated Conditioning in the Setting of Coherence

Angelo Gilio[1] and Giuseppe Sanfilippo[2]

[1] Dipartimento di Scienze di Base e Applicate per l'Ingegneria,
University of Rome "La Sapienza", Italy
angelo.gilio@sbai.uniroma1.it
[2] Dipartimento di Matematica e Informatica, University of Palermo, Italy
giuseppe.sanfilippo@unipa.it

Abstract. We consider conditional random quantities (c.r.q.'s) in the setting of coherence. Given a numerical r.q. X and a non impossible event H, based on betting scheme we represent the c.r.q. $X|H$ as the unconditional r.q. $XH + \mu H^c$, where μ is the prevision assessed for $X|H$. We develop some elements for an algebra of c.r.q.'s, by giving a condition under which two c.r.q.'s $X|H$ and $Y|K$ coincide. We show that $X|HK$ coincides with a suitable c.r.q. $Y|K$ and we apply this representation to Bayesian updating of probabilities, by also deepening some aspects of Bayes' formula. Then, we introduce a notion of iterated c.r.q. $(X|H)|K$, by analyzing its relationship with $X|HK$. Our notion of iterated conditional cannot formalize Bayesian updating but has an economic rationale. Finally, we define the coherence for prevision assessments on iterated c.r.q.'s and we give an illustrative example.

Keywords: Coherence, betting scheme, conditional random quantities, conditional previsions, Bayesian updating, iterated conditioning.

1 Introduction

Probabilistic reasoning under coherence allows a consistent treatment of uncertainty in many applications of statistical analysis, economy, decision theory, fuzzy set theory, psychology and artificial intelligence. This probabilistic approach allows to manage incomplete probabilistic assignments in a situation of vague or partial knowledge, see e.g. [9, 11, 13–15, 32]; see also [18, 21, 22, 24–28, 36] where a flexible probabilistic approach to inference rules in nonmonotonic reasoning and to the psychology of uncertain reasoning is developed. Based on coherence, we can develop a numerical approach to conditional events consistent with the three-valued logic proposed in the pioneering paper [16] by de Finetti; in this work we extend the approach to conditional random quantities (c.r.q.'s). Based on the betting scheme ([17], see also [31]), if for a numerical r.q. X we evaluate μ its prevision $\mathbb{P}(X)$, then we agree to pay (resp., to receive) an amount μ and to receive (resp., to pay) the random amount X. Analogously, given any non impossible event H, if we assess $\mathbb{P}(X|H) = \mu$ for the prevision

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 218–229, 2013.

of X *conditional on* H, then we agree to pay (resp., to receive) μ and to receive (resp., to pay) an amount, denoted $X|H$, which coincides with X, or μ, according to whether H is true, or false, i.e. $H = 1$, or $H = 0$ (in terms of indicators); then, *operatively*, $X|H = XH + \mu(1 - H)$. Thus, one of the values of $X|H$ is the prevision $\mathbb{P}(X|H) = \mu$, which is *subjectively evaluated*. In particular, if for a conditional event $A|H$ we assess $P(A|H) = p$, then (the indicator of) $A|H$ is the r.q. $AH + p(1 - H)$, with set of possible values $\{1, 0, p\}$. The problem of suitably defining the third value for (the indicators of) conditional events has been studied in some works by Coletti and Scozzafava (see, e.g., [13]).

We point out that, differently from other authors (see, e.g., ([37]; see also [34]), in our approach a c.r.q. $X|H$ is explicitly managed as an 'unconditional object' which among its possible values admits (the conditional prevision) μ. We also observe that the generalization of our results to imprecise conditional prevision assessments is out of the scope of the paper.

By exploiting this representation of c.r.q.'s, we obtain some basic results which concern an algebra of c.r.q.'s. Among other things, given any events H, K and any r.q.'s X, Y, we examine the condition under which $X|H$ and $Y|K$ coincide; in particular, we show that $X|HK$ can be represented as a suitable c.r.q. $Y|K$. Then, we use this representation in the context of Bayesian updating of probabilities and we deepen some aspects of Bayes' formula in the setting of coherence. As a natural consequence, we introduce the iterated c.r.q. $(X|H)|K$, which is defined as a suitable c.r.q. $Y|K$; then, we analyze its relationship with $X|HK$. However, the Bayesian updating for the probability of any hypothesis H cannot be formalized by our notion of iterated conditioning. Finally, we define the coherence for prevision assessments on iterated c.r.q.'s and we illustrate this notion by an example.

2 Preliminary Notions and Results

In our approach an event A represents an uncertain fact described by a (non ambiguous) logical proposition; hence we look at A as a two-valued logical entity which can be true (T), or false (F). The indicator of A, denoted by the same symbol, is a two-valued numerical quantity which is 1, or 0, according to whether A is true, or false. The sure event is denoted by Ω and the impossible event is denoted by $\emptyset$. Moreover, we denote by $A \wedge B$ (resp., $A \vee B$) the logical conjunction (resp., logical disjunction). In many cases we simply denote the conjunction between A and B as the product AB. By the symbol A^c we denote the negation of A. Given any events A and B, we simply write $A \subseteq B$ to denote that A logically implies B, i.e. $AB^c = \emptyset$. We recall that n events are logically independent when the number of atoms, or constituents, generated by them is 2^n. In case of some logical dependencies among the events, the number of atoms is less than 2^n. Given any events A and B, with $A \neq \emptyset$, the conditional event $B|A$ is looked at as a three-valued logical entity which is true (T), or false (F), or void (V), according to whether AB is true, or AB^c is true, or A^c is true.

Given an event $H \neq \emptyset$ and a r.q. X, we denote by V_H, the set of possible values of X restricted to H and, if X is finite, we set $V_H = \{x_1, x_2, \ldots, x_r\}$. In the setting of coherence, agreeing to the betting metaphor the prevision of $''X$ *conditional on* H'' (also named $''X$ *given* H''), $\mathbb{P}(X|H)$, is defined as the amount μ you agree to pay (resp., to receive), by knowing that you will receive (resp., to pay) the amount X if H is true, or you will receive back (resp., to pay back) the amount μ if H is false (bet called off). Agreeing with the operational subjective approach given in [31], we denote by $X|H$ the amount that you receive when a conditional bet is stipulated on $''X$ *given* H''. Then, it holds that $X|H = XH + \mu H^c$, where $\mu = \mathbb{P}(X|H)$, so that *operatively* we can look at the c.r.q. $X|H$ as the *unconditional* r.q. $XH + \mu H^c$. If X is finite and $\mu \notin V_H$, then $X|H \in \{x_1, x_2, \ldots, x_r, \mu\}$. Moreover, denoting by A_i the event $(X = x_i)$, $i \in J_r$, the family $\{A_1H, \ldots, A_rH, H^c\}$ is a partition of Ω and we have $X|H = XH + \mu H^c = x_1A_1H + \cdots + x_rA_rH + \mu H^c$. In particular, when X is an event A, the prevision of $X|H$ is the probability of $A|H$ and, if you assess $P(A|H) = p$, then for the indicator of $A|H$, denoted by the same symbol, we have $A|H = AH + pH^c \in \{1, 0, p\}$. The choice of p as the third value of $A|H$ has been proposed in some previous works, see e.g. [13, 19, 31].

Coherence for Conditional Prevision Assessments
Given a prevision function $\mathbb{P}$ defined on an arbitrary family $\mathcal{K}$ of c.r.q.'s, let $\mathcal{F}_n = \{X_i|H_i,\, i \in J_n\}$ be any finite subfamily of $\mathcal{K}$; we set $\mathcal{M}_n = (\mu_i,\, i \in J_n)$, where $\mu_i = \mathbb{P}(X_i|H_i)$. With the pair $(\mathcal{F}_n, \mathcal{M}_n)$ we associate the random gain $\mathcal{G} = \sum_{i \in J_n} s_iH_i(X_i - \mu_i)$; moreover, we set $\mathcal{H}_n = H_1 \vee \cdots \vee H_n$ and we denote by $G_{\mathcal{H}_n}$ the set of values of $\mathcal{G}$ restricted to the disjunction $\mathcal{H}_n$ of the conditioning events $H_1, \ldots, H_n$. Then, by de Finetti's *betting scheme*, we have

Definition 1. The function $\mathbb{P}$ defined on a finite family $\mathcal{K}$ is coherent if and only if, $\forall n \geq 1$, $\forall\, \mathcal{F}_n \subseteq \mathcal{K}$, $\forall\, s_1, \ldots, s_n \in \mathbb{R}$, it holds that: $\inf G_{\mathcal{H}_n} \leq 0 \leq \sup G_{\mathcal{H}_n}$. When $\mathcal{K}$ is infinite, we say that $\mathbb{P}$ is coherent if its restriction $\mathcal{M}_n$ on $\mathcal{F}_n$ is coherent, for every $\mathcal{F}_n \subset \mathcal{K}$.

Remark 1. Given a finite c.r.q. $X|H$, with $\mathbb{P}(X|H) = \mu$ and $V_H = \{x_1, \ldots, x_r\}$, we have that μ is coherent if and only if $min\, V_H \leq \mu \leq max\, V_H$. In particular, if $V_H = \{c\}$, then $X|H = cH + \mu H^c$; in this case μ is coherent if and only if $\mu = c$. Of course, for $X = H$ (resp. $X = H^c$) it holds that $\mu = 1$ (resp. $\mu = 0$) and hence $H|H = 1$, $H^c|H = 0$.

Checking of Coherence for Conditional Prevision Assessments
Given a family of n finite c.r.q.'s $\mathcal{F}_n = \{X_1|H_1, \ldots, X_n|H_n\}$, for each $i \in J_n$ we denote by $\{x_{i1}, \ldots, x_{ir_i}\}$ the set of possible values for the restriction of X_i to H_i; then, for each $i \in J_n$ and $j = 1, \ldots, r_i$, we set $A_{ij} = (X_i = x_{ij})$. Of course, for each $i \in J_n$, the family $\{H_i^c, A_{ij}H_i\,,\ j = 1, \ldots, r_i\}$ is a partition of the sure event Ω. Then, the constituents generated by the family $\mathcal{F}_n$ are (the elements of the partition of Ω) obtained by expanding the expression $\bigwedge_{i \in J_n}(A_{i1}H_i \vee \cdots \vee A_{ir_i}H_i \vee H_i^c)$. We set $C_0 = H_1^c \cdots H_n^c$ (it may be $C_0 = \emptyset$); moreover, we denote by $C_1, \ldots, C_m$ the constituents contained in $\mathcal{H}_n = H_1 \vee \cdots \vee H_n$. Hence

$\bigwedge_{i\in J_n}(A_{i1}H_i \vee \cdots \vee A_{ir_i}H_i \vee H_i^c) = \bigvee_{h=0}^{m} C_h$. With each C_h, $h \in J_m$, we associate a vector $Q_h = (q_{h1}, \ldots, q_{hn})$, where

$$q_{hi} = x_{ij}, \text{ if } C_h \subseteq A_{ij}H_i\,,\ j = 1, \ldots, r_i;\ q_{hi} = \mu_i, \text{ if } C_h \subseteq H_i^c\,.$$

In more explicit terms, for each $j \in \{1, \ldots, r_i\}$ the condition $C_h \subseteq A_{ij}H_i$ amounts to $C_h \subseteq A_{i1}^c \cdots A_{i,j-1}^c A_{ij} A_{i,j+1}^c \cdots A_{ir}^c A_{ir_i}^c H_i$. We observe that the vector $Q_h = (q_{h1}, \ldots, q_{hn})$ is the value of the random vector $(X_1|H_1, \ldots, X_n|H_n)$ when C_h is true; moreover, if C_0 is true, then the value of such a random vector is $\mathcal{M}_n = (\mu_1, \ldots, \mu_n)$. Denoting by $\mathcal{I}_n$ the convex hull of $Q_1, \ldots, Q_m$, the condition $\mathcal{M}_n \in \mathcal{I}_n$ amounts to the existence of a vector $(\lambda_1, \ldots, \lambda_m)$ such that: $\sum_{h\in J_m} \lambda_h Q_h = \mathcal{M}_n\,,\ \sum_{h\in J_m} \lambda_h = 1\,,\ \lambda_h \geq 0\,,\ \forall h$; in other words, $\mathcal{M}_n \in \mathcal{I}_n$ is equivalent to solvability of the following system Σ associated with the pair $(\mathcal{F}_n, \mathcal{M}_n)$, in the nonnegative unknowns $\lambda_1, \ldots, \lambda_m$,

$$\Sigma: \quad \sum_{h\in J_m} \lambda_h q_{hi} = \mu_i\,,\ i \in J_n\,;\ \sum_{h\in J_m} \lambda_h = 1\,;\ \lambda_h \geq 0\,,\ h \in J_m\,. \tag{1}$$

Given a subset $J \subseteq J_n$, we set $\mathcal{F}_J = \{X_i|H_i\,,\ i \in J\}\,,\ \mathcal{M}_J = (\mu_i\,,\ i \in J)$; then, we denote by Σ_J, where $\Sigma_{J_n} = \Sigma$, the system like (1) associated with the pair $(\mathcal{F}_J, \mathcal{M}_J)$. Then, it can be proved the following ([7])

Theorem 1. [*Characterization of coherence*]. Given a family of n finite c.r.q.'s $\mathcal{F}_n = \{X_1|H_1, \ldots, X_n|H_n\}$ and a vector $\mathcal{M}_n = (\mu_1, \ldots, \mu_n)$, the conditional prevision assessment $\mathbb{P}(X_1|H_1) = \mu_1, \ldots, \mathbb{P}(X_n|H_n) = \mu_n$ is coherent if and only if, for every subset $J \subseteq J_n$, defining $\mathcal{F}_J = \{X_i|H_i\,,\ i \in J\}$, $\mathcal{M}_J = (\mu_i\,,\ i \in J)$, the system Σ_J associated with the pair $(\mathcal{F}_J, \mathcal{M}_J)$ is solvable.

A characterization of coherence of conditional prevision assessments by non dominance with respect to proper scoring rules has been given in [8].

3 Deepenings on Conditional Random Quantities and Bayes Theorem

In this section, by exploiting the representation $X|H = XH + \mu H^c$, where $\mu = \mathbb{P}(X|H)$, we develop some elements of *an algebra of c.r.q.'s*. In particular, we recall a result which also concerns the general compound prevision theorem; then, we give some comments on the Bayesian updating of probabilities. We have

Theorem 2. Given any real quantities $a_1, \ldots, a_n$, any event $H \neq \emptyset$, any random quantities $X_1, \ldots, X_n$ and any coherent assessment $(\mu_1, \ldots, \mu_n, \nu)$ on $\{X_1|H, \ldots, X_n|H, (\sum_{i=1}^n a_i X_i)|H\}$, we have: $\sum_{i=1}^n a_i(X_i|H) = (\sum_{i=1}^n a_i X_i)|H$.

Proof. We have $(\sum_{i=1}^n a_i X_i)|H = (\sum_{i=1}^n a_i X_i)H + \nu H^c$; moreover, it holds that $\mathbb{P}[(\sum_{i=1}^n a_i X_i)|H] = \sum_{i=1}^n a_i \mathbb{P}(X_i|H)$; that is $\nu = \sum_{i=1}^n a_i \mu_i$. Then

$$\sum_{i=1}^n a_i(X_i|H) = \sum_{i=1}^n a_i(X_i H + \mu_i H^c) = \big(\sum_{i=1}^n a_i X_i\big)H + \nu H^c = \big(\sum_{i=1}^n a_i X_i\big)|H\,.$$

In particular: $a(X|H) = (aX)|H = aX|H$.

Theorem 3. Given any c.r.q.'s $X_1|H_1, \ldots, X_n|H_n$, with $\mathbb{P}(X_i|H_i) = \mu_i$, $\forall\, i$, and with $(\mu_1, \ldots, \mu_n)$ coherent, we have: $\mathbb{P}(\sum_{i=1}^n X_i|H_i) = \sum_{i=1}^n \mathbb{P}(X_i|H_i)$.

Proof. By linearity of prevision, we have

$$\mathbb{P}\Big(\sum_{i=1}^n X_i|H_i\Big) = \mathbb{P}\Big[\sum_{i=1}^n (X_iH_i + \mu_iH_i^c)\Big] = \sum_{i=1}^n \mathbb{P}\Big(X_iH_i + \mu_iH_i^c\Big) = \sum_{i=1}^n \mathbb{P}(X_i|H_i)\,.$$

We now consider the following questions:
(a) given two r.q. $X|H, Y|K$, with $H \neq K$, may it happen that $X|H = Y|K$?
(b) given any events H, K, with $HK \neq \emptyset$, and any r.q. X, with $\mathbb{P}(X|HK) = \mu$, does there exist a r.q. Y such that $X|HK = Y|K$?
We recall two results ([23, Theorems 7 and 9]) which show that the answers to both questions are positive. Concerning question (a) we have

Theorem 4. Given two c.r.q.'s $X|H, Y|K$, let (μ, ν) be a coherent prevision assessment on $\{X|H, Y|K\}$, with $\mathbb{P}(X|H) = \mu$, $\mathbb{P}(Y|K) = \nu$. Moreover, assume that $X|H = Y|K$ when the disjunction $H \vee K$ is true. Then $X|H = Y|K$.

The answer to question (b) is given in condition (i) of the result below, where (by Theorem 4) it is shown that $X|HK = Y|K$, where $Y = XH + yH^c$ and $y = \mathbb{P}(X|HK)$. The condition (ii), i.e. the general compound prevision theorem, is directly obtained by condition (i), by exploiting the linearity of prevision.

Theorem 5. Given two events $H \neq \emptyset, K \neq \emptyset$ and a r.q. X, let (x, y, z) be a coherent prevision assessment on $\{H|K, X|HK, XH|K\}$. Then:
(i) $X|HK = (XH + yH^c)|K$;
(ii) $z = xy$; that is: $\mathbb{P}(XH|K) = P(H|K)\mathbb{P}(X|HK)$.

In the next subsection, condition (i) of Theorem 5 will be applied to Bayesian updating of conditional probabilities.

3.1 An Application to Bayesian Inference

Given a hypothesis H, with $P(H) = p_0$, and a sequence of evidences $E_1, \ldots, E_n$, we set: $E_1 \cdots E_k = A_k$, $P(H|A_k) = p_k$, $Y_k = HA_{k-1} + p_kA_{k-1}^c$, $k = 1, \ldots, n$. By applying condition (i) of Theorem 5, with X, H, K replaced respectively by H, A_{k-1}, and E_k, we obtain

$$H|E_1 \cdots E_k = H|A_{k-1}E_k = Y_k|E_k = (HA_{k-1} + p_kA_{k-1}^c)|E_k\,,\ k = 1, \ldots, n\,.$$

We can verify that, in the previous equality, the prevision on the right-hand side coincides with that one on the left-hand side, which is the probability $P(H|E_1 \cdots E_k) = p_k$. Indeed, we have

$$\mathbb{P}(Y_k|E_k) = \mathbb{P}[(HA_{k-1} + p_kA_{k-1}^c)|E_k] = P(HA_{k-1}|E_k) + p_kP(A_{k-1}^c|E_k) =$$
$$= p_kP(A_{k-1}|E_k) + p_kP(A_{k-1}^c|E_k) = p_k\,,\ k = 1, \ldots, n\,.$$

As we can see, the updating of the probability of H, on the basis of evidences $E_1, \ldots, E_n$, consists at each step in replacing a probability by the next one in the

following sequence: $P(H)\,,\ P(H|E_1)\,,\ P(H|E_1E_2)\,,\ \cdots,\ P(H|E_1\cdots E_k)\,,\ \cdots$; that is, using the Bayesian mechanism, at each step we replace $P(H|A_{k-1}) = p_{k-1}$ by $P(H|A_k) = p_k$ when the new evidence E_k is obtained. Of course, in order to compute p_k by Bayes' formula

$$p_k = P(H|A_k) = P(H|A_{k-1}E_k) = \frac{P(E_k|A_{k-1}H)P(H|A_{k-1})}{P(E_k|A_{k-1})}\,,$$

all the needed probabilities must be assigned and $P(E_k|A_{k-1})$ must be positive. If $P(E_k|A_{k-1}) = 0$, by the methods of coherence, e.g. by using the Algorithm 1 in [2], or the zero-layers procedure in [13], it easily follows $p_k \in [0,1]$. More in general, if some of the values in Bayes' formula are not specified, then p_k is not uniquely determined and for its lower and upper bounds, l, u, there are different cases considered in the next subsection.

3.2 Lower/Upper Bounds on the Probability of $H|A_k$

We assume H, A_{k-1}, E_k logically independent and we set $A_{k-1} = A, E_k = E$; then $\{H|A_{k-1}, E_k|A_{k-1}, E_k|A_{k-1}H, H|A_{k-1}E_k\} = \{H|A, E|A, E|AH, H|AE\}$. As a preliminary remark we note that, given any sub-family $\Gamma = \{E_1|H_1, E_2|H_2\}$ of the family $\{H|A, E|A, E|AH, H|AE\}$, it can be verified that the set of coherent assessments (x, y) on Γ coincides with the unit square $[0,1]^2$. Then, if we assign only one of the quantities $P(E|A)$, or $P(E|AH)$, or $P(H|A)$, and we want to propagate it to $H|AE$, it holds that each value $z = P(H|AE) \in [0,1]$ is a coherent extension of the given assignment; that is $l = 0, u = 1$.
We now consider the cases where we assign only two of the quantities $P(E|A)$, $P(E|AH)$, $P(H|A)$, by giving the lower/upper bounds on $P(H|AE)$. We have three cases (which, due to the lack of space, are discussed without proof):
(i) only $x = P(E|A)$ and $y = P(E|AH)$ are assigned; then, the assessment $\mathcal{P} = (x, y, z)$ on $\mathcal{F} = \{E|A, E|AH, H|AE\}$ is coherent if and only if $z \in [0, u]$, with $u = \frac{y(1-x)}{x(1-y)}$, or $u = 1$, according to whether $y < x$, or $y \geq x$.
(ii) only $x = P(H|A)$ and $y = P(E|AH)$ are assigned; then, the assessment $\mathcal{P} = (x, y, z)$ on $\mathcal{F} = \{H|A, E|AH, H|AE\}$ is coherent if and only if $z \in [l, 1]$, with $l = 0$, or $l = \frac{xy}{1-x+xy}$, according to whether $(x, y) = (1, 0)$, or $(x, y) \neq (1, 0)$.
(iii) only $x = P(H|A)$ and $y = P(E|A)$ are assigned; then, based on the probabilistic analysis of the *CM rule* given in [20], the assessment $\mathcal{P} = (x, y, z)$ on $\mathcal{F} = \{H|A, E|A, H|AE\}$ is coherent if and only if $l \leq z \leq u$, with

$$l \;=\; \begin{cases} \frac{x+y-1}{y}\,, \text{ if } x+y>1\,, \\ 0\,, \qquad \text{if } x+y\leq 1\,, \end{cases} \qquad u \;=\; \begin{cases} \frac{x}{y}\,, \text{ if } x<y\,, \\ 1\,, \text{ if } x\geq y\,. \end{cases}$$

Remark 2. Given n logically independent events $E_1, \ldots, E_{n-1}, H$, and any assessment $\mathcal{P} = (x_1, \ldots, x_{n-1}, p_0)$ on the family $\mathcal{F} = \{E_1, \ldots, E_{n-1}, H\}$, the extension $p_{n-1} = P(H|E_1 \cdots E_{n-1})$ is coherent if and only if: $l \leq p_{n-1} \leq u$, where

$$l \;=\; \begin{cases} max\left\{0, \frac{x_1+\cdots+x_{n-1}+p_0-(n-1)}{x_1+\cdots+x_{n-1}-(n-2)}\right\}\,, \text{ if } x_1+\cdots+x_{n-1} > n-2\,, \\ 0\,, \qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{if } x_1+\cdots+x_{n-1} \leq n-2\,; \end{cases}$$

$$u = \begin{cases} min\left\{1, \frac{p_0}{x_1+\cdots+x_{n-1}-(n-2)}\right\}, & \text{if } x_1+\cdots+x_{n-1} > n-2\,, \\ 1\,, & \text{if } x_1+\cdots+x_{n-1} \leq n-2\,. \end{cases}$$

The previous formulas are obtained from (and better represent) the lower and upper bounds, l and u, given for the *generalized Cautious Monotonicity rule* in [21]; indeed, the representation of the probability bounds given in [21, Theorem 11] only concerns the case where the condition $x_1+\cdots+x_{n-1} > n-2$ is satisfied. A similar comment applies to [21, Theorem 10].

Other aspects of Bayes' theorem have been analyzed in [13] and [39]. Theoretical aspects and algorithms concerning the set of probability assessments which are compatible with given initial ones have been studied in several fields, such as probabilistic reasoning under coherence, model-theoretic probabilistic logic, probabilistic satisfiability, credal networks, and others; see, e.g., [2–6, 9, 10, 13, 29, 35, 38]. In the next section we give some results on iterated conditioning and we make a critical comparison with Bayesian updating.

4 Iterated Conditioning

The notion of iterated conditioning for c.r.q.'s was introduced in [23] and is consistent with that one given for conditional events in [26]. The basic intuition for our notion of iterated c.r.q. follows by the representation $X|H = XH + \mu H^c$, where $\mu = \mathbb{P}(X|H)$. After the definition we briefly discuss the meaning of the 'new object' $(X|H)|K$; then we give some results.

Definition 2. Given any events H, K, with $H \neq \emptyset, K \neq \emptyset$, and a finite r.q. X, with $\mathbb{P}(X|H) = \mu$, we define $(X|H)|K = (XH + \mu H^c)|K$.

From the previous definition, as $H^c|H = 0$, it follows:
$(X|H)|H = (XH + \mu H^c)|H = XH|H + \mu H^c|H = XH|H = X|H$; then, if we set $Y = X|H = XH + \mu H^c$, from $(X|H)|H = X|H$ it follows $Y|H = Y$.

Remark 3. Does there exist a reasonable justification for Definition 2 ?
We can provide a rationale for Definition 2, by imagining a decision problem involving two prevision assessments:

1) an agent evaluates $\mathbb{P}(X|H) = \mu$, by accepting then any transaction where, by paying an amount μ, one receives the uncertain amount $Y = X|H = XH + \mu H^c$;
2) the same agent evaluates $\mathbb{P}(Y|K) = \nu$, with $Y = X|H$, by accepting then a transaction where, by paying ν, one receives the uncertain amount $Y|K$.
Then, operatively: $\nu = \mathbb{P}(Y|K) = \mathbb{P}[(XH + \mu H^c)|K] = \mathbb{P}[(X|H)|K]$; that is, to evaluate the prevision of $Y|K$ amounts to evaluate the prevision of the iterated c.r.q. $(X|H)|K$. We point out that our notion of iterated conditioning does not concern those situations, typical of Bayesian updating, where a collection of pieces of evidence is synthesized by their conjunction and managed in a coherent way. Clearly, coherence plays a basic role also in our approach; for instance, concerning the discussion above, the agent must check coherence of the assessment (μ, ν) on $\{X|H, (X|H)|K\}$. This aspect will be considered in Section 5.

In the next result we show that $(X|H)|K$ may coincide with $X|H$, or $X|K$.

Proposition 1. Given any r.q. X and any nonimpossible events H, K, we have: (i) $(X|H)|K \neq (X|K)|H$; (ii) $(X|H)|K \neq X|HK$; (iii) if $H \subseteq K$, or $K \subseteq H$, then $(X|K)|H = (X|H)|K = X|HK$.

Proof. (i) The assertion follows by Definition 2.
(ii) Defining $\mathbb{P}(X|H) = \mu, \mathbb{P}(X|HK) = \eta$, in general it holds that $\mu \neq \eta$; thus, by condition (i) of Theorem 5, we have

$$X|HK = (XH + \eta H^c)|K \ \neq \ (XH + \mu H^c)|K = (X|H)|K\,.$$

(iii.a) If $H \subseteq K$, defining $\mathbb{P}(X|H) = \mu, \mathbb{P}(X|K) = z, \mathbb{P}(X|HK) = \eta$, we have $X|HK = X|H$ and $\eta = \mu$; then (by condition (i) of Theorem 5) we obtain $(X|H)|K = (XH + \mu H^c)|K = (XH + \eta H^c)|K = X|HK = X|H$. Moreover, $H \subseteq K$ implies $XK^c|H = zK^c|H = 0$; hence $XK|H = X(K + K^c)|H = X|H$. Then $(X|K)|H = (XK + zK^c)|H = XK|H = X|H = X|HK$.
(iii.b) If $K \subseteq H$, the assertion follows by a symmetric reasoning .

Remark 4. Note that, by condition (iii) in Proposition 1, $X|H = (X|H)|(H \vee K)$; then $\mathbb{P}(X|H) = \mathbb{P}[(X|H)|(H \vee K)]$. Indeed, defining $\mathbb{P}(X|H) = \mu$, we have $\mathbb{P}[(X|H)|(H \vee K)] = \mathbb{P}[(XH + \mu H^c)|(H \vee K)] = \mathbb{P}(XH|H \vee K) + \mathbb{P}(\mu H^c|H \vee K) = = \mathbb{P}(X|H)P(H|H \vee K) + \mu P(H^c|H \vee K) = \mu$.

The next result shows that the sum $X|H + Y|K$ of two c.r.q.'s, with *different conditioning events* H, K, can be represented as a suitable c.r.q. $Z|(H \vee K)$.

Proposition 2. Given a coherent prevision assessment (μ, η) on $\{X|H, Y|K\}$, it holds that: $X|H + Y|K = Z|(H \vee K)$, where $Z = XH + \mu H^c + YK + \eta K^c$ and $\mathbb{P}[Z|(H \vee K)] = \mu + \eta$.

Proof. We observe that $H \subseteq (H \vee K), K \subseteq (H \vee K)$; then, from condition (iii) in Proposition 1, we have $X|H = (X|H)|(H \vee K) = (XH + \mu H^c)|(H \vee K)$, $Y|K = (Y|K)|(H \vee K) = (YK + \eta K^c)|(H \vee K)$. Then, by Theorem 2, we obtain $X|H + Y|K = (XH + \mu H^c + YK + \eta K^c)|(H \vee K) = Z|(H \vee K)$. Moreover, by Theorem 3 (see also Remark 4), $\mathbb{P}[Z|(H \vee K)] = \mu + \eta$.

We observe that, given any events A, H, K, if $H \subseteq K$, or $K \subseteq H$, then $(A|K)|H = (A|H)|K = A|HK$, and the Import-Export Principle ([33]) would be valid. But, in general we have $(A|H)|K \neq (A|K)|H$, $(A|H)|K \neq A|HK$, $(A|K)|H \neq A|HK$; that is, in agreement with other authors (see, e.g., [1, 30]), the Import-Export Principle does not hold, as illustrated by the example below.

Example 1. Given any events A, H, K, with $HK = \emptyset$, we denote by p the probability of $A|H$, $P(A|H)$, and by α the prevision of $(A|H)|K$, $\mathbb{P}[(A|H)|K]$. By Definition 2, $(A|H)|K = (AH + pH^c)|K = AHK + pH^cK + \alpha K^c = pK + \alpha K^c$; moreover, *conditionally on K being true* the r.q. $AH + pH^c$ is constant and equal to p; then, by Remark 1, $\alpha = \mathbb{P}[(AH + pH^c)|K] = p$. Therefore, from $HK = \emptyset$ it follows: $(A|H)|K = p$ (more in general, given any r.q. X, with $\mathbb{P}(X|H) = \mu$, if $HK = \emptyset$, then $(X|H)|K = \mu$). If the Import-Export Principle were valid, we would have $(A|H)|K = A|HK = A|\emptyset$, which makes no sense; indeed, in Bayesian updating it is absurd to consider two logically incompatible evidences H, K.

In the framework of Bayesian inference, given any uncertain hypothesis H and any evidences $E_1, E_2, \ldots, E_n$, we iteratively compute $P(H|E_1)$, $P(H|E_1E_2)$, $\cdots$, $P(H|E_1 \cdots E_n)$; this amounts to synthesizing the sequence $E_1, \ldots, E_n$ by the conjunction $E_1 \cdots E_n$. If iterated conditioning were defined in agreement with the Import-Export Principle, it would be $H|E_1E_2 = (H|E_1)|E_2$, and so on; then

$$P(H|E_1E_2) = P[(H|E_1)|E_2]\,,\ P(H|E_1E_2E_3) = P[((H|E_1)|E_2)|E_3]\,,\ \ldots\,.$$

But, in our approach we have $(H|E_1)|E_2 \neq H|E_1E_2$, and so on; thus, Bayesian updating cannot be formalized by our iterated conditioning. For instance, we cannot look at the prevision $\mathbb{P}[(H|E_1)|E_2]$ as the probability $P(H|E_1E_2)$. Indeed, defining $P(H|E_1) = p_1, P(H|E_1E_2) = p_2$, by condition (i) of Theorem 5 we have $H|E_1E_2 = (HE_1 + p_2E_1^c)|E_2 \neq (HE_1 + p_1E_1^c)|E_2 = (H|E_1)|E_2$.
As discussed in Remark 3, our notion of iterated conditioning is useful for applications different from Bayesian updating.

5 Coherent Prevision Assessments for Iterated Conditional Random Quantities

In this section we introduce the notion of coherent prevision assessments on iterated c.r.q.'s, like

$$\mathbb{P}[(X_1|H_1)|K_1] = \nu_1\,,\ \mathbb{P}[(X_2|H_2)|K_2] = \nu_2\,,\ \cdots\,,\ \mathbb{P}[(X_n|H_n)|K_n] = \nu_n\,;$$

then, we will discuss a simple example. We observe that the iterated conditional random quantities $(X_1|H_1)|K_1, \cdots, (X_n|H_n)|K_n$ involve the assessment $(\mu_1, \ldots, \mu_n)$ on $\{X_1|H_1, \ldots, X_n|H_n\}$; then, in the definition of coherence we must consider the global assessment $(\mu_1, \ldots, \mu_n, \nu_1, \ldots, \nu_n)$. We have

Definition 3. Given any random quantities $X_1, \ldots, X_n$ and any events $H_1, \ldots, H_n, K_1, \ldots, K_n$, with $H_i \neq \emptyset, K_i \neq \emptyset$, $i = 1, \ldots, n$, the prevision assessment $(\mu_1, \ldots, \mu_n, \nu_1, \ldots, \nu_n)$ on $\mathcal{F} = \{X_1|H_1, \ldots, X_n|H_n, Y_1|K_1, \cdots, Y_n|K_n\}$, where $Y_1 = X_1|H_1, \ldots, Y_n = X_n|H_n$, is coherent if and only if, for every subfamily $\mathcal{S} \subseteq \mathcal{F}$, defining $\mathcal{H} = \bigvee_{i:X_i|H_i \in \mathcal{S}} H_i$, $\mathcal{K} = \bigvee_{i:Y_i|K_i \in \mathcal{S}} K_i$, and denoting by $G_{\mathcal{H}\vee\mathcal{K}}$ the set of possible values of the random gain

$$\mathcal{G} = \textstyle\sum_{i:X_i|H_i \in \mathcal{S}} s_iH_i(X_i - \mu_i) + \sum_{i:Y_i|K_i \in \mathcal{S}} \tau_iK_i(X_iH_i + \mu_iH_i^c - \nu_i)$$

restricted to $\mathcal{H} \vee \mathcal{K}$, with s_i, τ_i arbitrary real numbers for every i, it holds that $\inf G_{\mathcal{H}\vee\mathcal{K}} \leq 0 \leq \sup G_{\mathcal{H}\vee\mathcal{K}}$.

We observe that Definition 3 is nothing but Definition 1 applied to the family $\{X_i|H_i, Y_i|K_i, i = 1, \ldots, n\}$, where $Y_i = X_i|H_i = X_iH_i + \mu_iH_i^c$, $\forall\, i$; hence the value $g_0 = 0$ of the random gain $\mathcal{G}$, associated with the atom $H_1^c \cdots H_n^cK_1^c \cdots K_n^c$ (*all the bets on* $X_i|H_i, (X_i|H_i)|K_i$, $i = 1, \ldots, n$, *called off*), is discarded when defining coherence of the prevision assessment $(\mu_1, \ldots, \mu_n, \nu_1, \ldots, \nu_n)$ on the family $\{X_1|H_1, \ldots, X_n|H_n, (X_1|H_1)|K_1, \cdots, (X_n|H_n)|K_n\}$. Moreover, the checking for coherence can be made by the usual methods already existing in literature (see, e.g., [7, 12, 13]). Based on the geometrical approach related to Theorem 1, we illustrate Definition 3 by the example below.

Example 2. Given a r.q. $X \in \{1, 2, \ldots, 10\}$, we set $K = (X \in \{2, 4, \ldots, 10\})$, $H = (X \leq 6)$, $\mathbb{P}(X|H) = \mu$, $\mathbb{P}[(X|H)|K] = \nu$, $\mathcal{M}_1 = (\mu)$, $\mathcal{S}_1 = \{X|H\}$, $\mathcal{M}_2 = (\nu)$, $\mathcal{S}_2 = \{Y|K\}$, $\mathcal{M}_3 = (\mu, \nu)$, $\mathcal{S}_3 = \{X|H, Y|K\}$, where $Y = X|H = XH + \mu H^c$.

As shown below, the set Π of coherent assessments (μ, ν) on $\{X|H, (X|H)|K\}$ is the (non convex) polygon whose boundary is the closed polygonal with vertices the points $(1, 1), (2, 2), (5, 2), (5, 5), (6, 6), (1, 6)$. We observe that Π is the union of the triangle T_1, with vertices the points $(1, 1), (6, 6), (1, 6)$, and the triangle T_2, with vertices the points $(2, 2), (5, 2), (5, 5)$.

We denote by $\mathcal{I}_j$ the convex hull associated with the pair $(\mathcal{S}_j, \mathcal{M}_j)$, $j = 1, 2, 3$. From a geometrical point of view, the coherence of (μ, ν) amounts to conditions $\mathcal{M}_j \in \mathcal{I}_j$, $j = 1, 2, 3$. Of course, $\mathcal{M}_1 \in \mathcal{I}_1$ if and only if $1 \leq \mu \leq 6$. If $2 \leq \mu \leq 6$, then $\mathcal{M}_2 \in \mathcal{I}_2$ is satisfied for every $\nu \in [2, 6]$; if $\mu < 2$, then $\mathcal{M}_2 \in \mathcal{I}_2$ for every $\nu \in [\mu, 6]$. To check if $\mathcal{M}_3 \in \mathcal{I}_3$, we determine the set of constituents contained in $H \vee K$, i.e. different from $H^c K^c$, which are obtained by expanding the expression $(HK \vee HK^c \vee H^c K) \wedge (A_1 \vee \cdots \vee A_{10})$, where $A_i = (X = i), i = 1, \ldots, 10$. These constituents are A_2, A_4, A_6, A_1, A_3, A_5, A_8, A_{10}; the associated points Q_h's, for the pair $(\mathcal{S}_3, \mathcal{M}_3)$, are $(2, 2)$, $(4, 4)$, $(6, 6)$, $(1, \nu)$, $(3, \nu)$, $(5, \nu)$, (μ, μ), where with A_8 and A_{10} it is associated the same point (μ, μ).

We distinguish two cases: (i) $\mu \geq 2$; (ii) $\mu < 2$.

Case (i). For the convex hull it is enough to consider the points $(2, 2), (6, 6), (1, \nu)$, $(5, \nu)$. If $\mu \leq 5$ then (μ, ν) belongs to the segment with vertices $(1, \nu)$, $(5, \nu)$, so that the condition $\mathcal{M}_3 \in \mathcal{I}_3$ is satisfied and we have to continue by considering the condition $\mathcal{M}_2 \in \mathcal{I}_2$. If K is true, then $Y|K \in \{2, 4, 6, \mu\}$; hence, it must be $\nu \in [2, 6]$. If $\mu > 5$, then (μ, ν) belongs to the convex hull if and only if $\mu \leq \nu \leq 6$. In fact in this case (μ, ν) belongs to the triangle with vertices the points $(2, 2)$, $(6, 6)$, $(1, \nu)$. Of course, the condition $\nu \in [2, 6]$ is satisfied too.

Case (ii). For the convex hull it is enough to consider the points (μ, μ), $(6, 6), (1, \nu)$, $(5, \nu)$. Condition $\mathcal{M}_3 \in \mathcal{I}_3$ is satisfied because (μ, ν) belongs to the segment with vertices $(1, \nu), (5, \nu)$. Condition $\mathcal{M}_1 \in \mathcal{I}_1$ is satisfied because $1 \leq \mu < 2$; finally, the condition $\mathcal{M}_2 \in \mathcal{I}_2$ is satisfied for every $\nu \in [\mu, 6]$.

6 Conclusions

Based on betting scheme of de Finetti, we represented a c.r.q. as a suitable unconditional r.q., for which the assessed conditional prevision is one of the possible values. We obtained some results on basic operations among c.r.q.'s, by examining in particular a condition for the equality of two c.r.q.'s $X|H$ and $Y|K$. Then, we represented a c.r.q. $X|HK$ as a suitable c.r.q. $Y|K$ and we considered an application to Bayesian updating, by also deepening some aspects of Bayes' formula. We introduced a notion of iterated c.r.q. $(X|H)|K$, defined as a suitable c.r.q. $Y|K$, and we analyzed the relationship between $(X|H)|K$ and $X|HK$. Even if Bayesian updating cannot be formalized in our approach, we showed that our notion of iterated conditioning has an economic rationale. We discussed Bayesian updating in terms of iterated conditioning under the

Import-Export Principle. But, such a principle is not valid in general and does not work in applications where our iterated conditioning does. Finally, we defined the notion of coherence for prevision assessments on iterated c.r.q.'s, by also giving an example. Future work should concern the extension of our results to the case of imprecise conditional prevision assessments.

Acknowledgments. The authors are grateful to the anonymous referees for they valuable criticisms and suggestions.

References

1. Adams, E.W.: The Logic of Conditionals. Reidel, Dordrecht (1975)
2. Biazzo, V., Gilio, A.: A generalization of the fundamental theorem of de Finetti for imprecise conditional probability assessments. Int. J. Approx. Reason. 24(2-3), 251–272 (2000)
3. Biazzo, V., Gilio, A.: On the linear structure of betting criterion and the checking of coherence. Ann. Math. Artif. Intell. 35(1-4), 83–106 (2002)
4. Biazzo, V., Gilio, A., Lukasiewicz, T., Sanfilippo, G.: Probabilistic logic under coherence: complexity and algorithms. Ann. Math. Artif. Intell. 45(1-2), 35–81 (2005)
5. Biazzo, V., Gilio, A., Sanfilippo, G.: Coherence checking and propagation of lower probability bounds. Soft Computing 7(5), 310–320 (2003)
6. Biazzo, V., Gilio, A., Sanfilippo, G.: On the Checking of G-Coherence of Conditional Probability Bounds. Internat. J. Uncertain. Fuzziness Knowledge-Based Systems 11(suppl. 2), 75–104 (2003)
7. Biazzo, V., Gilio, A., Sanfilippo, G.: Generalized coherence and connection property of imprecise conditional previsions. In: Proc. IPMU 2008, Malaga, Spain, June 22-27, pp. 907–914 (2008)
8. Biazzo, V., Gilio, A., Sanfilippo, G.: Coherent Conditional Previsions and Proper Scoring Rules. In: Greco, S., Bouchon-Meunier, B., Coletti, G., Fedrizzi, M., Matarazzo, B., Yager, R.R. (eds.) IPMU 2012, Part IV. CCIS, vol. 300, pp. 146–156. Springer, Heidelberg (2012)
9. Brozzi, A., Capotorti, A., Vantaggi, B.: Incoherence correction strategies in statistical matching. Int. J. Approx. Reason. 53(8), 1124–1136 (2012)
10. de Campos, C.P., Cozman, F.G.: The inferential complexity of bayesian and credal networks. In: Kaelbling, L.P., Saffiotti, A. (eds.) IJCAI, pp. 1313–1318. Professional Book Center (2005)
11. Capotorti, A., Lad, F., Sanfilippo, G.: Reassessing Accuracy Rates of Median Decisions. The American Statistician 61(2), 132–138 (2007)
12. Capotorti, A., Vantaggi, B.: Locally Strong Coherence in Inference Processes. Ann. Math. Artif. Intell. 35(1-4), 125–149 (2002)
13. Coletti, G., Scozzafava, R.: Probabilistic logic in a coherent setting. Trends in logics, vol. 15. Kluwer, Dordrecht (2002)
14. Coletti, G., Scozzafava, R., Vantaggi, B.: Inferential processes leading to possibility and necessity. Information Sciences (2012), doi:10.1016/j.ins.2012.10.034
15. Dubois, D., Gilio, A., Kern-Isberner, G.: Probabilistic abduction without priors. Int. J. Approx. Reason. 47(3), 333–351 (2008)
16. de Finetti, B.: La Logique de la Probabilité. In: Actes du Congrès International de Philosophie Scientifique, Paris, 1935, pp. IV-1– IV-9, Hermann et C.ie (1936)
17. de Finetti, B.: Teoria delle probabilità, vols. 2. Ed. Einaudi, Torino (1970)
18. Gilio, A.: Probabilistic logic under coherence, conditional interpretations, and default reasoning. Synthese 146(1-2), 139–152 (2005)

19. Gilio, A.: Criterio di penalizzazione e condizioni di coerenza nella valutazione soggettiva della probabilità. Boll. Un. Mat. Ital. 4-B(3, Serie 7), 645–660 (1990)
20. Gilio, A.: Probabilistic Reasoning Under Coherence in System P. Ann. Math. Artif. Intell. 34(1-3), 5–34 (2002)
21. Gilio, A.: Generalizing inference rules in a coherence-based probabilistic default reasoning. Int. J. Approx. Reasoning 53(3), 413–434 (2012)
22. Gilio, A., Over, D.: The psychology of inferring conditionals from disjunctions: A probabilistic study. Journal of Mathematical Psychology 56(2), 118–131 (2012)
23. Gilio, A., Sanfilippo, G.: Conditional Random Quantities and Compounds of Conditionals, http://arxiv.org/abs/1304.4990
24. Gilio, A., Sanfilippo, G.: Quasi Conjunction and p-entailment in Nonmonotonic Reasoning. In: Borgelt, C., González-Rodríguez, G., Trutschnig, W., Lubiano, M.A., Gil, M.Á., Grzegorzewski, P., Hryniewicz, O. (eds.) Combining Soft Computing and Statistical Methods in Data Analysis. AISC, vol. 77, pp. 321–328. Springer, Heidelberg (2010)
25. Gilio, A., Sanfilippo, G.: Quasi conjunction and inclusion relation in probabilistic default reasoning. In: Liu, W. (ed.) ECSQARU 2011. LNCS, vol. 6717, pp. 497–508. Springer, Heidelberg (2011)
26. Gilio, A., Sanfilippo, G.: Conjunction, Disjunction and Iterated Conditioning of Conditional Events. In: Kruse, R., Berthold, M., Moewes, C., Gil, M.A., Grzegorzewski, P., Hryniewicz, O. (eds.) Synergies of Soft Computing and Statistics. AISC, vol. 190, pp. 399–407. Springer, Heidelberg (2013)
27. Gilio, A., Sanfilippo, G.: Probabilistic entailment in the setting of coherence: The role of quasi conjunction and inclusion relation. Int. J. Approx. Reason. 54(4), 513–525 (2013)
28. Gilio, A., Sanfilippo, G.: Quasi conjunction, quasi disjunction, t-norms and t-conorms: Probabilistic aspects. Information Sciences (2013), doi:10.1016/j.ins.2013.03.019
29. Jaumard, B., Hansen, P., Poggi de Aragão, M.: Column generation methods for probabilistic logic. ORSA Journal on Computing 3(2), 135–148 (1991)
30. Kaufmann, S.: Conditionals Right and Left: Probabilities for the Whole Family. Journal of Philosophical Logic 38(1), 1–53 (2009)
31. Lad, F.: Operational Subjective Statistical Methods: a mathematical, philosophical, and historical introduction. John Wiley, New York (1996)
32. Lad, F., Sanfilippo, G., Agró, G.: Completing the logarithmic scoring rule for assessing probability distributions. In: AIP Conf. Proceedings, vol. 1490(1), pp. 13–30 (2012)
33. McGee, V.: Conditional Probabilities and Compounds of Conditionals. Philosophical Review 98(4), 485–541 (1989)
34. Miranda, E., Zaffalon, M., de Cooman, G.: Conglomerable natural extension. Int. J. Approx. Reason. 53(8), 1200–1227 (2012)
35. Nilsson, N.J.: Probabilistic logic. Artificial Intelligence 28(1), 71–87 (1986)
36. Pfeifer, N., Kleiter, G.D.: Inference in conditional probability logic. Kybernetika 42, 391–404 (2006)
37. Walley, P.: Statistical Reasoning with Imprecise Probabilities. Chapman and Hall, London (1991)
38. Walley, P., Pelessoni, R., Vicig, P.: Direct algorithms for checking consistency and making inferences from conditional probability assessments. Journal of Statistical Planning and Inference 126(1), 119–151 (2004)
39. Wallmann, C., Kleiter, G.D.: Beware of too much information. In: Kroupa, T., Vejnarova, J. (eds.) Proceedings of WUPES 2012, Prague, pp. 214–225 (2012)

Distance-Based Measures of Inconsistency

John Grant[1] and Anthony Hunter[2]

[1] Department of Computer Science, University of Maryland
College Park, MD 20742, USA
[2] Department of Computer Science, University College London, Gower Street,
London WC1E 6BT, UK

Abstract. There have been a number of proposals for measuring inconsistency in a knowledgebase (i.e. a set of logical formulae). These include measures that consider the minimally inconsistent subsets of the knowledgebase, and measures that consider the paraconsistent models (3 or 4 valued models) of the knowledgebase. In this paper, we present a new approach that considers the amount each formula has to be weakened in order for the knowledgebase to be consistent. This approach is based on ideas of knowledge merging by Konienczny and Pino-Perez. We show that this approach gives us measures that are different from existing measures, that have desirable properties, and that can take the significance of inconsistencies into account. The latter is useful when we want to differentiate between inconsistencies that have minor significance from inconsistencies that have major significance. We also show how our measures are potentially useful in applications such as evaluating violations of integrity constraints in databases.

1 Introduction

Understanding the nature of inconsistency is an important topic if we are to develop autonomous systems that are able to behave intelligently with conflicting information. Although the early work of Grant in [1] showed more than 30 years ago that it is possible to compare inconsistent sets of formulae, the great amount of research on measuring inconsistency occurred in the past decade. It turns out that there are different reasonable ways of measuring the inconsistency of a knowledgebase; these measures tend to be incompatible with one another in the sense that one measure assigns a larger inconsistency value to knowledgebase Δ than to Δ' while another does not.

The purpose of this paper is to introduce several inconsistency measures based on model distance. We work in propositional logic and assume that a knowledgebase contains only consistent formulae. This is a reasonable assumption as portions of conflicting information are typically consistent. However, we note that every inconsistent formula (other than the special case $\bot$) requires a conjunction; such a formula can always be split into consistent fragments. Every consistent formula has at least one model. We think of each model as a point in Euclidean space. The models of a knowledgebase are exactly the intersection

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 230–241, 2013.

of the set of models for each formula. When the knowledgebase is inconsistent, this intersection is empty.

In our method we use distance measures to measure the distances between models (points in space). The idea of our method is to dilate the points representing the models to regions of space in a minimal way so that the intersection of these regions is no longer empty. Our various proposals count different aspects of these dilations to come up with measures of inconsistency. Furthermore, this approach lends itself to assigning weights to atoms thereby capturing better the significance of inconsistencies and provides new insight into the nature of inconsistency. For applications, it offers a better account for distances in the significance of parts of the knowledge that may be inconsistent. We illustrate how the new measures are potentially valuable tools for applications by considering violations of integrity constraints in databases.

2 Preliminaries

We assume a propositional language $\mathcal{L}$ of formulae composed from a set of atoms $\mathcal{A}$ and the logical connectives $\wedge$, $\vee$, $\neg$. We use ϕ and ψ for arbitrary formulae and α and β for atoms. All formulae are assumed to be in conjunctive normal form. Hence every formula ϕ has the form $\psi_1 \wedge \ldots \wedge \psi_n$, where each ψ_i, $1 \leq i \leq n$, has the form $\beta_{i1} \vee \ldots \vee \beta_{im}$, where each β_{ik}, $1 \leq k \leq m$ is a literal (an atom or negated atom). A knowledgebase Δ is a finite set of formulae. We let $\vdash$ denote the classical consequence relation. Logical equivalence is defined in the usual way: $\Delta \equiv \Delta'$ iff $\Delta \vdash \Delta'$ and $\Delta' \vdash \Delta$. We find it useful to define also a stronger notion of equivalence we call b(ijection)-equivalence as follows. Knowledgebase Δ is b(ijection)-equivalent to knowledgebase Δ', denoted $\Delta \equiv_b \Delta'$ iff there is a bijection $f : \Delta \rightarrow \Delta'$ such that for all $\phi \in \Delta$, ϕ is logically equivalent to $f(\phi)$. For example, $\{a, b\}$ is logically equivalent but not b(ijection)-equivalent to $\{a \wedge b\}$. We write $\mathcal{R}^{\geq 0}$ for the set of nonnegative real numbers and $\mathcal{K}$ for the set of all knowledgebases (where $\mathcal{K} = \{\Delta \mid \Delta \subseteq \mathcal{L}\}$).

Given a language $\mathcal{L}$, the set of models (i.e. interpretations) of the language is denoted $\mathcal{M}_\mathcal{L}$. Each **model** in $\mathcal{L}$ is an assignment of true or false to the atoms of the language from which an assignment is generated for all formulae of the language defined in the usual way for classical logic. For $\phi \in \mathcal{L}$, $\mathsf{Models}(\phi)$ denotes the set of models of ϕ (i.e. $\mathsf{Models}(\phi) = \{m \in \mathcal{M}_\mathcal{L} \mid m \models \phi\}$), and for $\Delta \subseteq \mathcal{L}$, $\mathsf{Models}(\Delta)$ denotes the set of models of Δ (i.e. if $\Delta = \{\phi_1, ..., \phi_n\}$, then $\mathsf{Models}(\Delta) = \mathsf{Models}(\phi_1) \cap .. \cap \mathsf{Models}(\phi_n)$).

To represent models $\mathcal{M}_\mathcal{L}$ of the language $\mathcal{L}$, we declare a **signature**, denoted $\mathcal{S}_\mathcal{L}$, which is the atoms of the language $\mathcal{L}$ given in a sequence $(a_1, ..., a_n)$, and then each model is given as a binary number $b_1, ..., b_n$ where for each digit b_i, if $b_i = 1$, then a_i is true in the model, otherwise $b_i = 0$ and a_i is false in the model.

Example 1. Let the atoms of $\mathcal{L}$ be $\{a, b, c\}$, and so $\mathcal{L}$ contains the usual propositional formulae that can be formed from these three atoms. Let the signature $\mathcal{S}^\mathcal{L}$ be (a, b, c), and so the models $\mathcal{M}^\mathcal{L}$ are $\{111, 110, 101, 100, 011, 010, 001, 000\}$.

Consider $m = 101$ which means that a is true, b is false, and c is true. This can equivalently be represented by the formula $a \wedge \neg b \wedge c$.

We introduce a couple of subsidiary functions to analyse models. For a model m, let $\mathsf{Digit}_i(m)$ return the ith digit of the model m (e.g. for the model 1010, $\mathsf{Digit}_2(1010) = 0$), and let $\mathsf{Atom}_i(m)$ return the atom corresponding to the ith digit of the model m (e.g. for the signature $\mathcal{S}_\mathcal{L} = (a,b,c,d)$, $\mathsf{Atom}_2(1010) = b$).

Next, we define the concept of an inconsistency measure for knowledgebases. We use the terminology that for a knowledgebase Δ, $\mathsf{Free}(\Delta)$ is the set of formulae not in any minimal inconsistent subset of Δ.

Definition 1. *An inconsistency measure I assigns a nonnegative real value to every knowledgebase Δ. We assume three requirements for inconsistency measures as proposed in [2] where (1) is called* consistency*, (2) is called* monotony*, and (3) is called* free formula independence*.*

1. $I(\Delta) = 0$ *iff* Δ *is consistent.*
2. *If* $\Delta \subseteq \Delta'$*, then* $I(\Delta) \leq I(\Delta')$*.*
3. *For all* $\alpha \in \mathsf{Free}(\Delta), (I(\Delta) = I(\Delta \backslash \{\alpha\})$*.*

The constraints 1 to 3 ensure that all and only consistent knowledgebases get measure 0, the measure is monotonic for subsets, and the removal of a formula that does not participate in an inconsistency leaves the measure unchanged.

3 Distance Measures

Given a set of models for a language $\mathcal{M}_\mathcal{L}$, a distance measure, as defined next, is an assignment of a real number to each pair of models in $\mathcal{M}_\mathcal{L}$. This is a very general notion that we will constrain in various ways in this paper.

Definition 2. *For a set of models $\mathcal{M}_\mathcal{L}$, a* **distance measure***, denoted d, is a function $d : \mathcal{M}_\mathcal{L} \times \mathcal{M}_\mathcal{L} \rightarrow \mathbb{R}^+$ satisfiying the following conditions.*

1. $d(m, m') = 0$ *iff* $m = m'$
2. $d(m, m') = d(m', m)$
3. $d(m, m') + d(m', m'') \geq d(m, m'')$

For example, the function that assigns distance 1 between any two distinct models is a distance measure.

Definition 3. *For a set of models $\mathcal{M}_\mathcal{L}$, a distance measure d is a* **drastic measure** *iff $d(m, m') = 1$ if $m \neq m'$ and $d(m, m') = 0$ if $m = m'$.*

We introduce the contrary function to define the Dalal (Hamming) measure.

Definition 4. *The* **contrary function***, denoted* $\mathsf{Contrary} : \{0,1\} \times \{0,1\} \rightarrow \{0,1\}$*, is defined as follows:* $\mathsf{Contrary}(1,1) = 0$*;* $\mathsf{Contrary}(1,0) = 1$*;* $\mathsf{Contrary}(0,1) = 1$*; and* $\mathsf{Contrary}(0,0) = 0$*.*

Definition 5. *Let $\mathcal{L}$ be composed from n atoms, and so $\mathcal{M}_\mathcal{L}$ contains models with n digits. A distance measure d is a* **Dalal measure** *iff*

$$d(m, m') = \sum_{i=1}^{n} \mathsf{Contrary}(\mathsf{Digit}_i(m), \mathsf{Digit}_i(m'))$$

A distance measure d is a Dalal measure [3] when $d(m, m')$ is the number of digits that differ between m and m'. For a fixed n the Dalal measure is unique.

Example 2. Consider the following measure which is a Dalal measure

$$\begin{array}{llll} d(11,11)=0 & d(11,10)=1 & d(11,01)=1 & d(11,00)=2 \\ d(10,11)=1 & d(10,10)=0 & d(10,01)=2 & d(10,00)=1 \\ d(01,11)=1 & d(01,10)=2 & d(01,01)=0 & d(01,00)=1 \\ d(00,11)=2 & d(00,10)=1 & d(00,01)=1 & d(00,00)=0 \end{array}$$

We use the following notion of a weighting function to assign a weight to each atom in a model. We write $w(i)$ for the weight of the ith atom. The idea is that the weight represents the significance of the atom.

Definition 6. *Given an n digit model, a* **weighting function** *is function $w : \{1, ..., n\} \to \mathbb{R}^+$. Special cases of weighting function $w : \{1, ..., n\} \to \mathbb{R}^+$ include:*

- *w is* **uniform** *iff for all $i \in \{1, ..., n\}$, $w(i) = r$ for some $r \in \mathbb{R}^+$*
- *w is* **positive** *iff for all $i \in \{1, ..., n\}$, $w(i) > 0$*
- *w is* **discounting** *iff there exists $i \in \{1, ..., n\}$, $w(i) < 1$*
- *w is* **binary** *iff for all $i \in \{1, ..., n\}$, $w(i) = 1$ or $w(i) = 0$*

Example 3. Let $\mathcal{M}_\mathcal{L} = \{11, 10, 01, 00\}$. So $w(1) = 0.5$ and $w(2) = 3$ is a positive weighting function.

Definition 7. *A distance measure is a* **weighted measure** *when there is a weighting function that weights each atom in the model.*

Next we will define two types of weighted measures: Manhattan measure and Euclidean measure.

Definition 8. *Let $\mathcal{L}$ be composed from n atoms, so that $\mathcal{M}_\mathcal{L}$ contains models with n digits. A distance measure d is a* **Manhattan measure** *iff there is a weighting function w such that*

$$d(m, m') = \sum_{i=1}^{n} w(i) \times \mathsf{Contrary}(\mathsf{Digit}_i(m), \mathsf{Digit}_i(m'))$$

Example 4. Consider the following measure which is a Manhattan measure with the positive weighting function w where $w(1) = 3$ and $w(2) = 2$.

$$\begin{array}{llll} d(11,11)=0 & d(11,10)=2 & d(11,01)=3 & d(11,00)=5 \\ d(10,11)=2 & d(10,10)=0 & d(10,01)=5 & d(10,00)=3 \\ d(01,11)=3 & d(01,10)=5 & d(01,01)=0 & d(01,00)=2 \\ d(00,11)=5 & d(00,10)=3 & d(00,01)=2 & d(00,00)=0 \end{array}$$

So a Dalal measure is a Manhattan measure with a uniform weighting function w where $w(i) = 1$ for each i. Another type of distance measure is the Euclidean distance, which treats space geometrically, as follows.

Definition 9. *Let $\mathcal{L}$ be composed from n atoms, and so $\mathcal{M}_\mathcal{L}$ contains models with n digits. A distance measure d is a* **Euclidean measure** *iff there is a weighting function w such that*

$$d(m, m') = \sqrt{\sum_{i=1}^{n} [w(i) \times \mathsf{Contrary}(\mathsf{Digit}_i(m), \mathsf{Digit}_i(m'))]^2}$$

Example 5. Consider the following Euclidean measure where $w(1) = 3$ and $w(2) = 2$.

$$\begin{array}{cccc} d(11,11) = 0.0 & d(11,10) = 2.0 & d(11,01) = 3.0 & d(11,00) = \sqrt{13} \\ d(10,11) = 2.0 & d(10,10) = 0.0 & d(10,01) = \sqrt{13} & d(10,00) = 3.0 \\ d(01,11) = 3.0 & d(01,10) = \sqrt{13} & d(01,01) = 0.0 & d(01,00) = 2.0 \\ d(00,11) = \sqrt{13} & d(00,10) = 3.0 & d(00,01) = 2.0 & d(00,00) = 0.0 \end{array}$$

Suppose we represent our n-digit models as points in n-dimensional space, then we can see that the Manhattan distance (which involves following the edges of the hypercube) gives a greater distance between two points than the Euclidean distance (which takes the direct line between the two points). The Manhattan distance treats each side of the hypercube equally and adds the traversal of all of them. This means that each atom of the model has to be taken additively. In contrast, the Euclidean distance discounts the distance with each further atom under consideration. Consider the models 11 and 10. The Manhattan distance and Euclidean distance is the same. Now consider the models 11 and 00. The Euclidean distance in effect "discounts" the effect of the second digit being different between the models. In other words, let d_d be the Manhattan distance (i.e. the Dalal distance), and let d_e be the Euclidean distance, then

$$d_d(11,11) = d_e(11,11) < d_d(11,10) = d_e(11,10) < d_e(11,00) < d_d(11,00)$$

We note that the Manhattan distance and the Euclidean distance are compatible with one another in the sense that $d_d(m_1, m_2) < d_d(m_3, m_4)$ iff $d_e(m_1, m_2) < d_e(m_3, m_4)$ and $d_d(m_1, m_2) = d_d(m_3, m_4)$ iff $d_e(m_1, m_2) = d_e(m_3, m_4)$.

4 Dilation of a Formula

In order to define our new class of inconsistency measures we turn to the notion of dilation. Bloch and Lang, in [4], explore how some operations from mathematical morphology translate into a logical framework. One of the most basic operations is the dilation of a set, which translates into the dilation of a formula (or its set of models). Essentially, for a formula ϕ, and a distance measure d, a dilation returns the models (or equivalently the formula specified by those models) that

are at most a certain distance from ϕ. The Dalal measure is a simple choice of distance measure to illustrate the idea. Suppose that ϕ is $a \wedge b$, and so the set of models is $\{11\}$. Using the Dalal distance, the set of dilations of distance 1 would be $\{11, 01, 01\}$, and so the resulting formula would be $a \vee b$. Then, the set of dilations of distance 2 would be $\{11, 01, 01, 00\}$, and so the resulting formula would be $\top$. Note how each dilation possibly weakens the previous formula in the sense that if ϕ is dilated to ϕ' then $\phi \vdash \phi'$.

Definition 10. *Let $\phi \in \mathcal{L}$ be a propositional formula, let $k \in \mathbb{R}$, and let d be a distance measure. The set of* **k-dilations** *of ϕ with respect to d is $M_d^k(\phi)$ as follows: $M_d^k(\phi) = \{m \in \mathcal{M}_\mathcal{L} \mid \exists m' \in M(\phi)$ such that $d(m', m) \leq k\}$.*

Hence, $M_d^k(\phi)$ is the set of models whose distance (using d) is not more than k from some model of ϕ. Next, we extend Definition 10 to apply to sets of formulae. For this purpose it will be convenient to assume an arbitrary ordering, called the **standard ordering**, over the formulae in $\mathcal{L}$. This could be, for instance, alphabetical ordering, but the ordering has no significance. It just gives a standard way to put formulae into a sequence. For any $\Delta \subseteq \mathcal{L}$, we can then represent Δ as a tuple $(\phi_1, \ldots, \phi_n)$, which we call the **standard form** of Δ, where $\Delta = \{\phi_1, \ldots, \phi_n\}$ and $<$ is the standard ordering, and for each i, if $1 \leq i < n$, then $\phi_i < \phi_{i+1}$.

Definition 11. *Let $(\phi_1, \ldots, \phi_n)$ be the standard form of Δ, where each $\phi_i \in \Delta$ is consistent, and let d be a distance measure. The set of* **k-dilation profiles** *with respect to d is $P_d(\Delta) = \{(k_1, ..., k_n) \mid M_d^{k_1}(\phi_1) \cap ... \cap M_d^{k_n}(\phi_n) \neq \emptyset\}$.*

Here is what happens. We start with the sequence $(\phi_1, \ldots, \phi_n)$ of formulae, or equivalently, the sequence of their sets of models. $P_d(\Delta)$ is a sequence of numbers $(k_1, \ldots, k_n)$ such that the k_i-dilations of all the ϕ_i for $1 \leq i \leq n$ have a nonempty intersection. If we think of each k_i-dilation as the formula represented by the models, say ψ_i, then the nonempty intersection means that $\{\psi_1, \ldots, \psi_n\}$ is consistent. We minimize $P_d(\Delta)$ and use it to measure inconsistency.

Example 6. For $\Delta = \{a \wedge b, \neg a \wedge b\}$, and using the Dalal measure d,

$$P_d(\Delta) = \{(x, y) \mid x + y \geq 1\}$$

Proposition 1. *Let $\Delta = \{\phi_1, ..., \phi_n\} \subseteq \mathcal{L}$ be a set of propositional formulae where each $\phi_i \in \Delta$ is consistent, and $(\phi_1, \ldots, \phi_n)$ is the standard form of Δ. Let d be a weighted measure with weighting w.*

(a) *If w is positive, then $(0, ..., 0) \in P_d(\Delta)$ iff Δ is consistent.*
(b) *If $\Delta' = \{\phi'_1, ..., \phi'_n\}$, and $(\phi'_1, \ldots, \phi'_n)$ is the standard form of Δ', and $\phi_1 \equiv \phi'_1$, and ... and $\phi_n \equiv \phi'_n$, then $P_d(\Delta) = P_d(\Delta')$*

The following result shows that the drastic measure is not sufficiently discriminating for our purposes since just a dilation of 1 will return all the models.

Proposition 2. *Let $\phi \in \mathcal{L}$ be a consistent propositional formula and let d be the drastic measure. For $k \geq 1$, $M_d^k(\phi) = \mathcal{M}_\mathcal{L}$.*

In the next section, we will see examples of using dilation with the weighted measure. We will use minimal dilations defined next.

Definition 12. *A k-dilation* $(k_1, \ldots, k_n) \in P_d(\Delta)$ *is called* **minimal** *if and only if there is no k-dilation* $(k'_1, \ldots, k'_n) \in P_d(\Delta)$ *such that* $(k_1, \ldots, k_n) \neq (k'_1, \ldots, k'_n)$ *and* $k'_i \leq k_i$ *for all* i, $1 \leq i \leq n$. *We write* $P_d^{min}(\Delta)$ *for the set of minimal dilations.*

So in Example 6, $P_d^{min}(\Delta) = \{(0,1),(1,0)\}$.

5 Using Dilation to Measure Inconsistency

Now we can use the set of k-dilation profiles of a knowledgebase to assign it a measure of inconsistency. We define three measures. The first one sums a minimal sequence; the second picks the maximum value of a minimal sequence; while the third counts the number of nonzero values in a minimal sequence.

Definition 13. *Let* $\Delta \subseteq \mathcal{L}$ *be a set of propositional formulae where each* $\phi_i \in \Delta$ *is consistent, and let* d *be a distance measure. The* **d-sum inconsistency measure** *is* $I_d^{sum}(\Delta) = \mathsf{Min}\{x \mid (k_1, ..., k_n) \in P_d(\Delta)$ *and* $k_1 + ... + k_n = x\}$.

Definition 14. *Let* $\Delta \subseteq \mathcal{L}$ *be a set of propositional formulae where each* $\phi_i \in \Delta$ *is consistent, and let* d *be a distance measure. The* **d-max inconsistency measure** *is* $I_d^{max}(\Delta) = \mathsf{Min}\{x \mid (k_1, ..., k_n) \in P_d(\Delta)$ *and* $\mathsf{Max}\{k_1, ..., k_n\} = x\}$.

It is clear from the definitions that for all Δ, $I_d^{max}(\Delta) \leq I_d^{sum}(\Delta)$.

The third measure is somewhat different from the first two as it takes into account the number of formulae that need to be dilated (hit) in order to make the set consistent. Intuitively, the more hits, the more inconsistency there is in the set of formulae. Note, for this definition, the only information used for the calculation is whether the distance measure is zero or greater than zero. Hence, the magnitude of the distance measure is not taken into account.

Definition 15. *Let* $\Delta \subseteq \mathcal{L}$ *be a set of propositional formulae where each* $\phi_i \in \Delta$ *is consistent, and let* d *be a distance measure. The* **d-hit inconsistency measure** *is defined as follows.*

$$I_d^{hit}(\Delta) = \mathsf{Min}\{x \mid (k_1, ..., k_n) \in P_d(\Delta) \text{ and } \mathsf{Hit}(k_1, \ldots, k_n) = x\}$$

where $\mathsf{Hit}(k_1, \ldots, k_n) = \Sigma_{i=1}^{n} z(k_i)$ *where* $z(k_i) = 1$ *if* $k_i > 0$ *and* $z(k_i) = 0$ *if* $k_i = 0$.

Before showing that these three definitions really define inconsistency measures, we give four examples. In these examples we use the Dalal measure.

Example 7. Let $\Delta_1 = \{a \wedge b, \neg a \wedge \neg b\}$. $P_d(\Delta_1)$ includes $(1,1)$, $(2,0)$, and $(0,2)$. Hence, $I_d^{sum}(\Delta_1) = 2$, $I_d^{max}(\Delta_1) = 1$, and $I_d^{hit}(\Delta_1) = 1$.

k $a \wedge b$	$\neg a \wedge \neg b$
0 { 11 }	{ 00 }
1 { 11,10,01 }	{ 10,01,00 }
2 { 11,10,01,00 }	{ 11,10,01,00 }

Example 8. Let $\Delta_2 = \{a, \neg a \vee \neg b, b\}$. $P_d(\Delta_2)$ includes $(1,0,0)$, $(0,1,0)$, and $(0,0,1)$. Hence, $I_d^{sum}(\Delta_2) = 1$, $I_d^{max}(\Delta_2) = 1$, and $I_d^{hit}(\Delta_2) = 1$.

k a	$\neg a \vee \neg b$	b
0 { 11,10 }	{ 01,10,00 }	{ 11,01 }
1 { 11,10,01,00 }	{ 11,10,01,00 }	{ 11,10,01,00 }

Example 9. Let $\Delta_3 = \{a \wedge b \wedge c, \neg a \wedge \neg b \wedge \neg c\}$. $P_d(\Delta_3)$ includes $(1,2)$, $(2,1)$, $(3,0)$, and $(0,3)$. Hence, $I_d^{sum}(\Delta_3) = 3$, $I_d^{max}(\Delta_3) = 2$, and $I_d^{hit}(\Delta_3) = 1$.

k $a \wedge b \wedge c$	$\neg a \wedge \neg b \wedge \neg c$
0 { 111 }	{ 000 }
1 { 111,110,101,011 }	{ 010,001,100, 000 }
2 { 111,110,101,011,100,010,001 }	{ 110,101,011,010,001,100, 000 }
3 { 111,110,101,011,100,010,001,000 }	{ 111,110,101,011,010,001,100, 000 }

Example 10. Let $\Delta_4 = \{a, b, c, \neg a, \neg b, \neg c\}$. $P_d(\Delta)$ contains profiles including $(1,1,1,0,0,0)$, $(1,1,0,0,0,1)$, $(1,0,0,0,1,1)$, etc. Hence, $I_d^{sum}(\Delta) = 3$, $I_d^{max}(\Delta) = 1$, and $I_d^{hit}(\Delta) = 3$. We omit the table here because the second of the two rows is too long to include.

Next, we show that the three inconsistency measures defined above satisfy the consistency, monotony, and free formula independence properties.

Proposition 3. *The* **d-sum inconsistency measure**, *the* **d-max inconsistency measure**, *and the* **d-hit inconsistency measure**, *each satisfy conditions 1 to 3 of Definition 1, and therefore all three are inconsistency measures.*

The d-sum inconsistency measure and the d-max inconsistency measure have been influenced by the definition for model-based merging operators by Konieczny and Pino Perez [5], and the dilation-based reformalization of them [6].

Next we show that a useful property for inconsistency measures, called dominance, holds for all of these measures.

Proposition 4. *If* $\{\alpha\} \vdash \beta$, *and* α *is consistent, then*

1. $I_d^{sum}(\Delta \cup \{\alpha\}) \geq I_d^{sum}(\Delta \cup \{\beta\})$
2. $I_d^{max}(\Delta \cup \{\alpha\}) \geq I_d^{max}(\Delta \cup \{\beta\})$
3. $I_d^{hit}(\Delta \cup \{\alpha\}) \geq I_d^{hit}(\Delta \cup \{\beta\})$

In order to compare two inconsistency measures, we define I_x and I_y to be *order-compatible* if for all knowledgebases Δ_1 and Δ_2, $I_x(\Delta_1) < I_x(\Delta_2)$ iff $I_y(\Delta_1) < I_y(\Delta_2)$ and *order-incompatible* otherwise.

Proposition 5. *The* **d-sum inconsistency measure***, the* **d-max inconsistency measure***, and the* **d-hit inconsistency measure** *are pairwise order-incompatible.*

In [7], we reviewed the main proposals in the literature for measuring inconsistency, such as measures based on 3 or 4 valued models and measures based on minimal inconsistent subsets of knowledge, and we showed that they were pairwise order-incomparable. We can also show that these three new measures are pairwise incomparable with the existing proposals. This means we cannot use existing measures to substitute for these new proposals. Hence, these new measures offer new tools for analysing inconsistency.

We can use a geometric interpretation of dilation using Euclidean distance in n-dimensional space. So take the case with n atoms and weighting function w. For model $b_1...b_n$ assign the point $(b_1 \cdot w(1), ...b_n \cdot w(n))$. For example, let $n = 3$ and weight function $w(1) = 2$, $w(2) = 5$, $w(3) = 4$. Then the model 101 is mapped to the point (2,0,4) and the model 110 is mapped to the point (2,5,0) (all points are in 3-dimensional space). For the distance between points (the models) we are using the Manhattan distance of moving along the edges of a hypercube, whereas the Euclidean distance is the "straight line" distance between the points. Looking at the models this way as points in n-dimensional space using Euclidean distance, the k-dilation of a model is the set of points that represent models in a hypersphere of radius k with center at that point. As the k-dilation of a formula is the k-dilations of its models, geometrically, the k-dilation of a formula becomes the set of points that represent models in a union of hyperspheres. For the Manhattan distance substitute "hypercube" for "hypersphere". It is possible for two such hypersheres or hypercubes to have a nonempty intersection that does not contain any models. Suppose that in the given example $(1, 4, 2)$ is a point in the intersection. Such a point does not represent a model for the given weights. However, if we were using fractional truth values, the point would represent a model, namely with fractional truth values .5, .8, and .5 respectively for the atoms. We do not pursue this matter further in this paper.

6 Significance

There are two reasons for presenting the distance-based measures of inconsistency in this paper. The first is to extend our understanding of the nature of inconsistency and how it can be measured. The second is to develop techniques for taking the significance of inconsistency into account.

A simple way of taking significance into account is to assume a weighting function, and use a distance measure that can take this weight into account such as the Manhattan distance or the Euclidean distance, as illustrated next.

Example 11. Consider the atoms a = "rain in my city" and b = "rain in a city 100Km from my city". Consider the set of 2-digit models with the signature (a, b) (i.e. the first digit refers to a, the second digit to b). Let $w(1) = 1$ and $w(2) = 0.1$ be the weighting function, and let d be the Manhattan distance.

Δ	$\{a \wedge b, \neg a \wedge \neg b\}$	$\{a \wedge b, \neg a \wedge b\}$	$\{a \wedge b, a \wedge \neg b\}$	$\{\neg a \wedge b, \neg a \wedge \neg b\}$
$I_d^{sum}(\Delta)$	1.1	1	0.1	0.1
$I_d^{max}(\Delta)$	1	1	0.1	0.1
$I_d^{hit}(\Delta)$	1	1	1	1

Using weights allows us to reduce inconsistency by applying a resolution function (see [7]) that has maximal impact. For example, if $\Delta = \{a, \neg a, b, \neg b\}$ and w(1)=1, w(2) = 10, then deleting b or $\neg b$ reduces the inconsistency far better than deleting a or $\neg a$.

Whilst Example 11 shows how we can have different degrees of inconsistency based on significance, it does not take the context of the inconsistency into account. To illustrate what we mean by this, consider the following example where the measure is not a weighted measure.

Example 12. Consider the atoms a = "earthquake" and b = "electricity fails". In this situation, some assumptions we may have about the significance of inconsistency is as follows.

- if we have an inconsistency about whether or not there is an earthquake, then we have a very significant inconsistency.
- if we have an inconsistency about whether or not the electricity fails, then we have a moderate inconsistency.
- however, if we know that there is an earthquake, and there is an inconsistency about the electricity failing, then the significance of the inconsistency is low.

Consider the set of 2-digit models with the signature (a, b) (i.e. the first digit refers to a, the second digit to b). We can capture this significance using the following distance measure.

$$\begin{array}{llll} d(11,11) = 0 & d(11,10) = 1 & d(11,01) = 9 & d(11,00) = 9 \\ d(10,11) = 1 & d(10,10) = 0 & d(10,01) = 9 & d(10,00) = 9 \\ d(01,11) = 9 & d(01,10) = 9 & d(01,01) = 0 & d(01,00) = 2 \\ d(00,11) = 9 & d(00,10) = 9 & d(00,01) = 2 & d(00,00) = 0 \end{array}$$

We illustrate the use of this distance measure with the following examples of knowledgebases.

Δ	$\{a \wedge b, \neg a \wedge \neg b\}$	$\{a \wedge b, \neg a \wedge b\}$	$\{a \wedge b, a \wedge \neg b\}$	$\{\neg a \wedge b, \neg a \wedge \neg b\}$
$I_d^{sum}(\Delta)$	9	9	1	2
$I_d^{max}(\Delta)$	9	9	1	2
$I_d^{hit}(\Delta)$	1	1	1	1

The difference between a weighted measure and a non-weighted measure is that for a weighted measure the atoms are independent of one another. That is not the case for non-weighted measures. So in Example 12 we can think of the 4 models as being in 2 groups: the group $\{11, 10\}$ and the group $\{00, 01\}$. Models within a group are close to one another but models in different groups have a larger distance. In that example the first atom is more important than the second atom; however the second atom does not have a unique weight: its weight depends on

the truth value of the first atom. However, if the groups are $\{11, 00\}$ and $\{01, 10\}$ then they are based on the sameness of the truth values of the atoms. With more atoms more groups can be formed.

7 Measuring Violations of Integrity Constraints

In this section we consider measuring violations of integrity constraints in knowledgebases. As integrity constraints must be satisfied, we slightly revise our definitions so that only the data is dilated and not the integrity constraints. We assume that relational data is represented by a set of ground predicates Δ, and a set of integrity constraints Γ. We treat both Δ and Γ as propositional formulae.

Definition 16. *Let $\Delta \subseteq \mathcal{L}$ be a set of ground predicates (atomic formulae), and $(\phi_1, \ldots, \phi_n)$ be the standard form of Δ. Let $\Gamma \subseteq \mathcal{L}$ be a consistent set of ground formulae, and let d be a distance measure. The set of* **k-dilation profiles** *with respect to d is $P_d(\Delta)$ as follows.*

$$P_d(\Delta, \Gamma) = \{(k_1, ..., k_n) \mid M_d^{k_1}(\phi_1) \cap ... \cap M_d^{k_n}(\phi_n) \cap M(\Gamma) \neq \emptyset\}$$

The weights could be chosen so that the significance of the inconsistency rises as the difference in the values taken by the data deviate. In order to assign the weights, we may choose to use an equation, as we illustrate in the following example where we consider weight to be a linear function of the difference between the value and the median value.

Example 13. Let Δ be the six literals in the following table and Γ the integrity constraints obtained from the axiom scheme $salary(bob, X_1) \rightarrow \neg salary(bob, X_2)$, where $X_1 \neq X_2$. Here we assume that the weight is dependent on the range of values for the salary for Bob. So the most extreme values for the salary (i.e. 1000 and 2000) have highest significance, whereas the least extreme value (i.e. 1400 and 1600) have the lowest significance. We capture this by the following equation where X^* is the mid-point between the minimum and maximum value for the salary.

$$w(salary(bob, X)) = \frac{\mid X - X^* \mid}{100} + 1$$

Using this equation, we get the following weight for the example.

	w		w		w
salary(bob,1000)	6	salary(bob,1400)	2	salary(bob,1900)	5
salary(bob,1100)	5	salary(bob,1600)	2	salary(bob,2000)	6

Here the inconsistency measures are $I_d^{sum}(\Delta) = 20$, $I_d^{max}(\Delta) = 6$, and $I_d^{hit}(\Delta) = 5$ using the Manhattan distance with the above weights.

Taking significance into account using these measures means that we consider how "incorrect" or how extreme the literals are. Smaller ranges of values in the data have lower weights than wider ranges of values in the data. So we can define

these weights in the form of any kind of equation that is appropriate for the application. Furthermore, it is straightforward to define equations for obtaining the weights that consider multiple dimensions of inconsistency in the data. For instance, the tuple salary(bob,1000,45) might be inconsistent with regard to any combination of name, or salary, or age.

8 Discussion

In future work, we plan to further develop the application features of this framework in context-sensitive approaches to dealing with inconsistency (e.g. [8]). We also plan to address some of the shortcomings of using the Hamming distance, as discussed by Lafage and Lang [9], by using distances based on Choquet integrals. These can avoid the assumption of independence between propositional variables, and ameliorate problems of syntax sensitivity. Finally, we plan to establish connections with measures of inconsistency for probabilistic knowledge [10] and fuzzy knowledge [11].

References

1. Grant, J.: Classifications for inconsistent theories. Notre Dame Journal of Formal Logic 19, 435–444 (1978)
2. Hunter, A., Konieczny, S.: On the measure of conflicts: Shapley inconsistency values. Artificial Intelligence 174, 1007–1026 (2010)
3. Dalal, M.: Investigations into a theory of knowledge base revision. In: Proceedings of the Seventh National Conference on Artificial Intelligence, AAAI 1988, vol. 2, pp. 475–479 (1988)
4. Bloch, I., Lang, J.: Towards Mathematical Morpho-Logics. In: Technologies for Constructing Intelligent Systems, vol. 2, pp. 367–380. Springer (2002)
5. Konieczny, S., Pérez, R.P.: On the logic of merging. In: Sixth International Conference on Principles of Knowledge Representation and Reasoning, KR 1998, pp. 488–498 (1998)
6. Gorogiannis, N., Hunter, A.: Implementing semantic merging operators using binary decision diagrams. International Journal of Approximate Reasoning 49(1), 234–251 (2008)
7. Grant, J., Hunter, A.: Measuring consistency gain and information loss in stepwise inconsistency resolution. In: Liu, W. (ed.) ECSQARU 2011. LNCS, vol. 6717, pp. 362–373. Springer, Heidelberg (2011)
8. Subrahmanian, V.S., Amgoud, L.: A general framework for reasoning about inconsistency. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), pp. 599–504 (2007)
9. Lafage, C., Lang, J.: Propositional distances and preference representation. In: Benferhat, S., Besnard, P. (eds.) ECSQARU 2001. LNCS (LNAI), vol. 2143, pp. 48–59. Springer, Heidelberg (2001)
10. Thimm, M.: Inconsistency measures for probabilistic logics. Artificial Intelligence 197, 1–24 (2013)
11. Muiño, D.: Measuring and repairing inconsistency in knowledge bases with graded truth. Fuzzy Sets and Systems 197, 108–122 (2011)

Safe Probability: Restricted Conditioning and Extended Marginalization

Peter Grünwald

CWI, Amsterdam and Leiden University, The Netherlands
pdg@cwi.nl

Abstract. Updating probabilities by conditioning can lead to bad predictions, unless one explicitly takes into account the mechanisms that determine (1) what is observed and (2) what has to be predicted. Analogous to the *observation*-CAR (coarsening at random) condition, used in existing analyses of (1), we propose a new *prediction task*-CAR condition to analyze (2). We redefine conditioning so that it remains valid if the mechanisms (1) and (2) are unknown. This will often update a singleton distribution to an imprecise set of probabilities, leading to dilation, but we show how to mitigate this problem by marginalization. We illustrate our notions using the Monty Hall Puzzle.

1 Introduction

Let P be a probability distribution on some space $\mathcal{Y}$. Suppose we are given information in the form of an event $B \subset \mathcal{Y}$. We are then asked to give the probability of another event $A \subset \mathcal{Y}$, given information B. Many people would be inclined to say "this probability is equal to $P(A|B)$, defined as $P(A,B)/P(B)$; this is just the standard definition of conditional probability". In this paper, we boldly propose a little extension of probability theory, in which we always have to make an additional calculation, to check whether predicting with $P(A|B)$ is *valid*, or at least *safe*. If it is unsafe, we should not use $P(A|B)$; we then risk getting answers that are wrong under any reasonable operational interpretation of probability. We explain this in Section 2, right after Example 3, and give formal definitions of safety and validity in Section 4.1, Definition 1; for now, we just note that unsafety implies there are other ways of updating P based on B that provably lead to better predictions. Indeed, we identify realistic situations in which updating by a "predictive distribution" $\tilde{P}$ *different* from standard conditioning is "safe", whereas standard conditioning is "unsafe".

All this may sound worrisome, especially to Bayesian readers: isn't there a plethora of evidence (by e.g. Savage (1954) and many, many others), axiomatic and otherwise, implying that conditioning is the *only* reasonable way to update probabilities? The answer is: yes, there is, and all our 'safe' updates are in fact compatible with conditioning if we were to work in a larger sample space $\mathcal{Z}$ that takes explicitly into account the *observation selection mechanism* (OSM) and the *task selection mechanism* (TSM). Here a 'task' can be any decision, prediction, or

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 242–253, 2013.

inference problem. Earlier work on OSM has been done within the CAR (coarsening at random) literature (Heitjan and Rubin, 1991, Grünwald and Halpern, 2003, De Cooman and Zaffalon, 2004). We generalize CAR-based OSM's and connect them to TSM's, which, to the best of our knowledge, have not been studied before. In practice such selection mechanisms, while relevant, may often be unknown, so we do not know the appropriate distribution P^* on $\mathcal{Z}$. We only know that P^* must be a member of some set of distributions $\mathcal{P}^*$, consisting of all distributions on $\mathcal{Z}$ that satisfy some known constraints. The 'safe' predictive distributions $\tilde{P}$ that we advocate typically coincide with a marginal distribution corresponding to some specific, special distribution in the set $\mathcal{P}^*$.

In the remainder of this introduction, we describe two well-known probability puzzles that motivate our research. Section 2 summarizes relevant insights from the CAR literature. Our original contributions are in Section 3 and beyond, in which we develop two notions of safety: the strong *guaranteed-validity* notion and a weaker notion which we just call *safety*. In the final section we return to the two puzzles to see what safe probability implies for them. Mathematical proofs and further discussion will be provided in the full paper of which this submission is an extended abstract.

Example 1. [**Monty Hall Puzzle**] (vos Savant, 1994, Gill, 2011) Suppose that you're on a game show and given a choice of three doors, named a, b and c. Behind one is a car; behind the others are goats. You pick door a. Before opening door a, Monty Hall, the quiz master (who knows what is behind each door) opens one of the other doors (say, door c), which has a goat. He then asks you if you still want to take what's behind door a, or to take what's behind the closed door (door b, in our case) instead. Should you switch? You may assume that, initially, the car was equally likely to be behind each of the doors, so it seems natural to define a sample space $\mathcal{Y} = \{a, b, c\}$ where $Y = y$ indicates that the car is behind door a, and $P(a) = P(b) = P(c) = 1/3$. You observe that the car is not behind door c, i.e. the remaining possibilities are $\{a, b\}$, Conditioning now gives that $P(b \mid \{a, b\}) = (1/3)/(2/3) = 1/2$, which suggests that the car is now equally likely to be behind door a and door b. Thus, there seems no reason to switch.

Now, 23 years after this problem was popularized, almost everybody agrees that this simple answer is wrong: as vos Savant pointed out, it is strongly in your interest to switch. However, initially, most people who heard about the puzzle, including some professors of probability theory (see (vos Savant, 1994)), were very hard to convince of this. It is here that safe probability can be useful: from the definition of safety in Section 3, one *immediately* sees that conditioning as we did above is 'unsafe', implying it will lead to suboptimal decisions. Briefly, for general spaces $\mathcal{Y}$, if the set of events $\mathcal{X}$ on which you can condition is not a partition of $\mathcal{Y}$, then conditioning on any of these events is unsafe. In the present case, the set of events is $\mathcal{X} = \{\{a, b\}, \{a, c\}\}$ (the latter would be observed if the quiz master had opened door b). The two events overlap (a is a member of both), hence do not form a partition, and hence you should not update by conditioning. This part is only of limited novelty — it has been argued before by many authors (perhaps most notably Shafer (1985)) that updating by conditioning only makes

sense if a protocol is specified (corresponding to what we call an 'observation selection mechanism' below). Shafer (1996) formalizes this in terms of event trees which implicitly require conditioning events to form a partition. The only novelty here is our insight that, in practical cases in which the 'correct' event tree may be hard to construct, checking for overlap provides a *very simple sanity check* which *immediately* indicates that a problem is represented in a space in which conditioning makes no sense.

The real novelty of safe probability relates to the question whether, if the quiz master opens door c, the probability that the car is behind a *remains* $1/3$. If one assumes that, in those cases in which the car is actually behind door a, the quiz master tosses a fair coin to decide whether to open door b or c, then the answer that $P(a)$ remains $1/3$ is valid. However, it is unclear whether in the game as it was actually played on TV, Monty Hall really tossed a fair coin; for all we know he might have followed a very different rule, for example, open door c whenever you can. Previous analyses such as by Grünwald and Halpern (2003) that take into account that Monty's protocol is unknown, conclude that after the quiz master opened door b or c, a precise probability of the car being behind door a cannot be given any more: it can be anything between 0 and $1/2$. In other words, the probability has *dilated* (Seidenfeld and Wasserman, 1993): it seems that, by observing additional information, one knows less than before (this will be explained in Example 2). But this does not seem satisfactory either: many people would reason that, since the quiz master in fact *has to* open door b or c, he gives no information about a, so the probability should remain $1/3$. Using safe probability we can partially vindicate this intuition: we show that $1/3$ does have a special status, even if the quiz master's protocol is unknown — the reasoning is, to some extent, correct after all, if our goal is just to asses whether the car is behind door a: let $Y' = 1$ iff a obtains, $Y' = 0$ otherwise. In Section 4 we show that $\tilde{P}$ defined by $\tilde{P}(Y' = 1 \mid \{a, b\}) = \tilde{P}(Y' = 1 \mid \{a, c\}) = 1/3$ is a sort of *marginal* distribution, and we show that predictions based on $\tilde{P}$ will behave exactly as they would if $\tilde{P}$ were actually the correct conditional distribution. Hence it is *safe* to act as if the probability remains $1/3$.

2 The Problem with Overlapping Sets

Notation. All sets we introduce below are finite. All probability distributions mentioned below are defined on $\mathcal{Z}$, our generic symbol for the sample space. A random variable (RV) is any function from $\mathcal{Z}$ to some arbitrary finite set. For a given RV we denote its range in calligraphic script. For example, a RV X maps $z \in \mathcal{Z}$ to $\mathcal{X}$. For RV Y with range $\{y_1, \ldots, y_m\}$, when we write $P(Y)$ we really mean the vector $(P(y_1), \ldots, P(y_m))$, where $P(y)$ abbreviates $P(Y = y)$.

Example 2. **[Dice]** This example is really just Monty Hall, without any misleading aspects. Suppose you and me play the following game: I roll a die, which we both know to be fair, i.e. $\mathcal{Z} = \mathcal{Y} = \{1, \ldots, 6\}$. I get to see the outcome, but you don't. I only tell you whether the outcome is below 3 or not, i.e. whether $Y \in \{1, 2\}$ or $Y \in \{3, 4, 5, 6\}$. Given this information, you are asked to give

the probability that $Y = 3$. We agreed beforehand that, after throwing the die, I will tell you exactly one of the two statements, and that I won't lie. If I tell you $\{1, 2\}$, you would probably answer 'the probability of 3 is now 0', and if I tell you $\{3, 4, 5, 6\}$, you would say 'the probability of 3 is now 1/4'. This is the answer you get by conditioning: $P(Y = 3 \mid Y \in \{1, 2\}) = 0$ and $P(Y = 3 \mid Y \in \{3, 4, 5, 6\}) = 1/4$, and here it is obviously valid.

But now let's slightly change the game: we now agree beforehand that, after throwing the die, I will tell you either "$Y \in \{1, 2, 3, 4\}$" or "$Y \in \{3, 4, 5, 6\}$". Suppose that, when we actually play, I tell you $Y \in \{3, 4, 5, 6\}$. Given this observation, what is now the probability of 3? Many people would still say 1/4 but this answer is wrong. To see this, note that if, after throwing the die, I observe outcome 3 or 4, then I have a *choice* in what to tell you, and you do not know how I choose. For example, I may decide to always say $\{1, 2, 3, 4\}$ whenever I observe 3 or 4. In that case, if I say $Y \in \{3, 4, 5, 6\}$, the actual probability that $Y = 3$ is 0 rather than 1/4! (for if I had observed 3, I had certainly told you $\{1, 2, 3, 4\}$). Even if I am 'fair', i.e., when I observe 3 or 4, I flip a fair coin to decide whether to tell you $\{1, 2, 3, 4\}$ or $\{3, 4, 5, 6\}$, the answer 1/4 is still wrong: as we calculate below in (2), the probability of $Y = 3$ given $\{3, 4, 5, 6\}$ then becomes 1/6. Note that when we write '1/4 is invalid' we do not refer to the mathematical definition of conditional probability (the statement $P(Y = 6 \mid \{3, 4, 5, 6\}) = 1/4$ is after all a correct application of the definition of conditional probability). We explain what 'invalid' means here right after Example 3.

As explained by e.g. Grünwald and Halpern (2003) (GH from now on), to formalize problems such as this correctly, we need to move to a larger sample space in which we can explicitly represent the fact that I sometimes have a choice in what to tell you. This can be done by representing the problem in the space $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$, where $\mathcal{Y}$ is the *outcome space* as before, and $\mathcal{X}$ is the *observation* space, with associated RVs Y and X, respectively. $\mathcal{Z}$ was called the "sophisticated space" by GH. We assume (uncontroversially, see e.g. (Heitjan and Rubin, 1991)) that in this larger space, conditioning is the valid thing to do. In our case, $\mathcal{Y} = \{1, \ldots, 6\}$ as before, and $\mathcal{X} = \{\{1, 2, 3, 4\}, \{3, 4, 5, 6\}\}$. We know that the distribution P on $\mathcal{Z}$ must be compatible with the distribution on $\mathcal{Y}$, and we also agreed that I don't lie, so in our case this means that $P(Y = y) = 1/6$ for all $y \in \mathcal{Y}$, and $P(Y \in x \mid X = x) = 1$ for both $x \in \mathcal{X}$. This is not sufficient to specify $P((x, y))$ for all $(x, y) \in \mathcal{Z}$. For this, we would need two more probabilities p and q in $[0, 1]$, defined by setting

$$P(X = \{3, 4, 5, 6\} \mid Y = 3) = p, \;\; P(X = \{3, 4, 5, 6\} \mid Y = 4) = q. \qquad (1)$$

Once we specify p and q, we can determine $P(x, y)$, and, more importantly for us, $P(y \mid x)$, for each $(x, y) \in \mathcal{Z}$.The interpretation is that when e.g. $Y = 3$, I flip a coin with bias p. If it lands heads I say $\{3, 4, 5, 6\}$, otherwise I say $\{1, 2, 3, 4\}$.

Example 3. We can now calculate the actual probability that $Y = 6$ given that I say $\{3, 4, 5, 6\}$ as

$$\begin{aligned} P(Y=6 \mid X=\{3,4,5,6\}) &= \frac{P(Y=6, X=\{3,4,5,6\})}{P(X=\{3,4,5,6\})} \\ &= \frac{P(6)}{P(3,X=\{3,4,5,6\})+P(4,X=\{3,4,5,6\})+P(5)+P(6)} \\ = \frac{P(6)}{P(3)P(X=3..6|3)+P(4)P(X=3..6|4)+P(5)+P(6)} &= \frac{\frac{1}{6}}{\frac{1}{6}\cdot(p+q+2)} = \frac{1}{p+q+2}, \end{aligned} \tag{2}$$

where in the third line we abbreviated all occcurrences of $Y = y$ to y, for $y \in \{3, \ldots, 6\}$. Suppose that we make no assumptions on p and q. This includes the deterministic cases (if $p = q = 1$ or $p = q = 0$) in which, when I have a choice, I'll *always* say the same thing. By varying p and q in (2), we find that $P(Y = 6 \mid X = \{3, 4, 5, 6\})$ can take on any value between $1/4$ and $1/2$, depending on the value of p and q.

All this shows that conditioning cannot always be valid. The meaning of 'valid' can be understood in three ways: (I) frequentist: conditioning is not *calibrated* , i.e. if we were to repeat the game of Example 2 independently many times, each time casting the die anew, then conditional relative frequencies will not converge to the corresponding conditional probabilities. For example, if I follow the strategy with $p = q = 0$, and we play, say, 6000 times, then each time I say $\{3, 4, 5, 6\}$, you will say that the probability of 3 is now $1/4$; but of all the (approximately 2000) times that I will say $\{3, 4, 5, 6\}$, the actual outcome will be 5 or 6, so the conditional frequency of 3 is 0 rather than $1/4$. (II) (perhaps more appealing to a Bayesian): decision-theoretic: not surprisingly, in the light of (I), using the conditional distributions to make predictions about Y can be suboptimal; we will see several examples of this in the next sections. (III) As we just saw, even if we do assume that conditioning is valid if the problem is modelled in the large space $\mathcal{X} \times \mathcal{Y}$, which takes into account the protocol, then, even if the protocol is 'fair', conditioning in the small space, omitting the protocol, can be invalid.

The original space $\mathcal{Y}$ was called the 'naive space' by GH. We may now ask when conditioning in the naive space is valid. The answer is given by the *coarsening at random (CAR) condition* (Heitjan and Rubin, 1991). For us, only a partial characterization is important: let $\mathcal{X}$ be a collection of subsets of $\mathcal{Y}$ and P^* be a distribution on $\mathcal{Y}$, and suppose that there is 'no lying'. If $\mathcal{X}$ partitions $\mathcal{Y}$, then naive conditioning is valid, i.e. conditioning in the naive space must coincide with conditioning in the sophisticated space. If $\mathcal{X}$ does not partition $\mathcal{Y}$, then naive condititioning can always be invalid: there exist distributions P on $\mathcal{X} \times \mathcal{Y}$ with marginal on Y equal to P^* and $x \in \mathcal{X}$ such that $P^*(X = x) > 0$, $P^*(Y \mid Y \in x) \neq P(Y \mid X = x)$; see Prop. 4.1 and Theorem 4.4(b) of GH.

3 Towards Safe Probability

First Attempt to restrict Conditioning The partition result above suggests a very simple definition of 'validity': we say conditional probability $P(A \mid B)$ is undefined unless a set $\mathcal{B}$ with $B \in \mathcal{B}$ is specified; we then write $P(A \mid B)$ as

$P_{\mathcal{B}}(A \mid B)$. $\mathcal{B}$ is the set of alternative events that might have been observed instead of B. We could then simply define conditioning $P_{\mathcal{B}}(A \mid B)$ to be 'valid' iff $\mathcal{B}$ is a partition, and restrict conditioning to valid cases: if $\mathcal{B}$ is not a partition, then it is undefined. This would already take care of the sanity check for the Monty Hall problem (Example 1), but it falls short of dealing with the second issue in Example 1 (how to assess the probability $\tilde{P}(Y' \mid \{a, b\})$), as well as the more general type of prediction task selection problems we will encounter below. We found these issues a lot more amenable to a random-variable based treatment, so that is the direction we take below.

Random Variables and Partitions. The main advantage of a RV treatment is that the problem of invalid conditioning goes away — to some extent — automatically, since for every arbitrary random variables X, there is a partition Π such that conditioning on the value of X is equivalent to conditioning on the element of the partition that obtains (trivial proof provided in full paper). Thus, by our preliminary definition of validity based on event-conditioning as above, conditioning on a fixed RV X must *always* be valid. Thus, we could *define* conditional probability as $P(Y \mid X)$ for fixed RVs Y and X, and leave probabilities of the sort P(event A | event B) undefined. One might argue that under such a definition of conditional probability, our problem of invalid conditioning goes away automatically. But it is more complicated than that: the problem goes away automatically *only* if it is implicitly understood that the distribution P for which $P(Y \mid X)$ is specified, will *only* be used to make predictions or decisions about Y given the value of X, irrespective of the value of X that is actually observed. Thus, for example, if $Y = (Y_1, Y_2)$, it is not valid in general to make a prediction about just Y_1 if $X = a$ is observed, and a prediction about just Y_2 if $X = b$ is observed (see Example 5 below). Yet if the prediction problem at hand satisfies the implicit *fixed X, fixed Y*—requirement, then conditioning on a fixed RV is indeed valid. This requirement often holds in signal processing, information-theoretic and machine learning applications such as classification and regression with i.i.d. random design.

Beyond Fixed RVs. However, in many other standard applications of probability, we routinely make predictions about *various* RVs $Y_1, Y_2, \ldots$ conditioned on *various* RVs $X_1, X_2, \ldots$, and it is not precisely specified on what grounds a specific X_s or Y_t is chosen. For example, the Monty Hall and dice example can be interpreted in this way, as we show in Example 7. As a practically more relevant example, Bayes nets are often used to compute, e.g., how the probability that a patient has a certain disease would change counterfactually if (a) we were to observe that $X_1 = x_1$, or (b) we were to observe that $X_2 = x_2$; the result is then used to determine whether we should, in fact, observe RV X_1 or RV X_2 — both X_1 and X_2 may correspond to costly medical tests, and we may want to avoid doing two tests rather than one.

We only get away with such applications of probability if particular additional independence assumptions hold, which are usually left implicit. Rather than relying on such tacitly made assumptions to hold, as is usually done, it seems

safer to use probability in a way which forces us to explicitly represent the *task selection mechanism* TSM (which determines what Y_j is observed) and the *observation selection mechanism* OSM (which determines what X_i is observed), so that we cannot violate our assumptions by mistake (which happens in the invalid $P(b \mid \{a, b\})$ answer to the Monty Hall problem, Ex. 1) or unnecessarily dilate a distribution (as happens in the Monty Hall problem when the probability $P(Y' = 1 \mid \{a, b\})$ is merely assessed to be in $[0, 1/2]$ instead of $1/3$). We now develop such an explicit representation of OSMs and TSMs.

4 Observation and Task Selection Mechanisms

We start with two examples that motivate the general definitions further below. Example 4 concerns OSMs: we show that event-based conditioning with overlap in the conditioning events (as in our three examples) can be rephrased as conditioning on a RV X_S selected from a set of RV $\{X_s \mid s \in \mathcal{S}\}$, where S is itself random. S then represents the OSM. Example 5 then concerns TSMs that determine what random variable Y_T has to be predicted.

Additional Notation in This Section. For an event $\mathcal{E} \subset \mathcal{Z}$, we define the *indicator random variable* $I_{\mathcal{E}}$ to be 1 if $\mathcal{E}$ holds and 0 otherwise. For distribution P on $\mathcal{Z}$ and RV U we define $\text{SUPPORT}_P(U) = \{u \in \mathcal{U} : P(U = u) > 0\}$. For a set of distributions $\mathcal{P}^*$ on $\mathcal{Z}$ and RVs U, V, W on $\mathcal{Z}$ we write $U \perp_{\mathcal{P}^*} V \mid W$ iff U and V are *conditionally independent* given W, that is, if for all $P \in \mathcal{P}^*$, for all $(u, v, w) \in \text{SUPPORT}_P(U, V, W)$, it holds $P(U = u \mid V = v, W = w) = P(U = u \mid W = w)$. We say that $P, P' \in \mathcal{P}^*$ *agree* on an event $\mathcal{E}$ if $P(\mathcal{E}) = P'(\mathcal{E})$. We write $\mathcal{P}^*(\mathcal{E})$ to denote the set $\{P(\mathcal{E}) : P \in \mathcal{P}^*\}$. If all $P \in \mathcal{P}^*$ agree on $\mathcal{E}$, the probability of $\mathcal{E}$ is known relative to $\mathcal{P}^*$ and we write $P^*(\mathcal{E})$ rather than $\mathcal{P}^*(\mathcal{E})$. For two RVs U, V on $\mathcal{Z}$, we write $U \rightsquigarrow V$ ("U determines V" or "U is a *coarsening* of V") if there is a function f such that for all $z \in \mathcal{Z}$, $V(z) = f(U(z))$.

Example 4. **[Observation Selection]** We define $\mathcal{S} = \{a, b\}$ and set RV $X_a := \{1, 2, 3, 4\}$ if $Y \in \{1, 2, 3, 4\}$ and RV $X_a = \{5, 6\}$ otherwise. We set $X_b := \{3, 4, 5, 6\}$ if $Y \in \{3, 4, 5, 6\}$ and $X_b = \{1, 2\}$ otherwise. Example 2 is equivalent to a scenario in which you observe X_S, where S is set to a if $Y \in \{1, 2\}$; S is set to b if $Y \in \{5, 6\}$, and if $Y \in \{3, 4\}$, then whether you observe X_a or X_b depends on my protocol. To this end, we define the extended sample space $\mathcal{Z} = \mathcal{Y} \times \mathcal{S}$. We then set $P^*(S = a \mid Y = 1) = P^*(S = a \mid Y = 2) = P^*(S = b \mid Y = 5) = P^*(S = b \mid Y = 6) = 1$, and $P^*(S = b \mid Y = 3) = p, P^*(S = b \mid Y = 4) = q$. S — which in this case is just my protocol — is an example of what we call an observation-selection mechanism. We set $\mathcal{P}^*$ to be the set of all distributions on $\mathcal{Z}$ of the form above. The fact that we now have a set, rather than a single distribution reflects our ignorance of the precise protocol. The resulting setting is equivalent to Example 3: for example, if $Y = 3$, we will, with probability $1 - p$, observe $\{1, 2, 3, 4\}$. Note that $S = a$ iff RV X in Example 3 is equal to $\{1, 2, 3, 4\}$, and $S = b$ iff $X = \{3, 4, 5, 6\}$. Thus, observing S is equivalent to observing X and we see that $\mathcal{Z} = \mathcal{Y} \times \mathcal{S}$ as here has equivalent representative power as $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ as defined above Example 3.

Example 5. [**Task Selection**] Let $X \in \{0,1\}$ and $Y \in \{a,b,c\}$. Imagine your goal is to predict aspects of Y given X, where an 'aspect' is a function that is determined by Y — for example, you might want to either predict $Y_1 = I_{Y=a}$ or $Y_2 = I_{Y=b}$ — and 'predicting Y_j' means coming up with a distribution $\tilde{P}$ for Y_j ($\tilde{P}$ may then be used as the basis for making decisions about Y under various loss function in the standard way, i.e. you choose the act that minimizes expected loss under $\tilde{P}$). Some external process determines whether Y_1 or Y_2 should be predicted. This process is modelled by an additional RV $T \in \mathcal{T} = \{1,2\}$. T is what we call a (prediction) task selection mechanism. The idea is that, in any realization of the system, Y_T (rather than the full Y) has to be predicted. Suppose you represent your uncertainty on (X,Y,T) by a set of distributions $\mathcal{P}^*$, all of which agree on (X,Y); hence the marginal distribution $P^*(X,Y)$ is known but the dependencies between (X,Y) and T may not be known. For concreteness, let's take some $P^*(X,Y)$ such that $P^*(Y = a \mid X = 1) = 0.8, P^*(Y = b \mid X = 0) = 0.9$, $P^*(Y = a) = 0.6$. If you think that T is determined independently of Y (for example, I ask you to predict either Y_1 or Y_2, and you know that I make my choice on external grounds, without knowing X or Y or $P^*(X,Y)$ myself), then $\mathcal{P}^*$ would be the set of all distributions P on $\mathcal{Z}$ with $P(X,Y) = P^*(X,Y)$ and with $(X,Y) \perp_P T$. Yet, if you don't know how I determine what RV I ask you to predict, you may want to choose for $\mathcal{P}^*$ the set of *all* distributions P on (X,Y,T) with $P(X,Y) = P^*(X,Y)$.

In standard uses of probability, the process T is rarely modelled explicitly, and upon observing $X = x$ and being asked to predict Y_t, you may be tempted to predict Y_t with the conditional distribution $P^*(Y_t \mid X = x)$. But in fact, this standard procedure is only valid if you are indeed in the situation with $Y \perp_{\mathcal{P}^*} T$, i.e. the process determining what you are asked is independent of Y itself. For otherwise, it would, for example, be possible to ask you about Y_1 whenever $Y = a$ and to ask about Y_2 whenever $Y \neq a$. Then, when observing $X = 1$, you will predict Y_1 with distribution $P^*(Y_1 = 1 \mid X = 1) = 0.8$, whereas the probability of $Y_1 = 1$ given that you are asked about it is really 1. Clearly standard conditioning is once again invalid, unless some independences involving (T,X,Y) hold. It seems we implicitly must assume, when we condition, that 'something like' $T \perp_{\mathcal{P}^*} Y \mid X$ is the casee (this includes the case that T is constant, fixed in advance); see Definition 3 below for a sharper formulation. Indeed, consider a scenario B in which I always ask you to predict Y_1 whenever $X = 1$ and Y_2 whenever $X = 0$, i.e. $T = f(X)$ with $f(1) = 1$ and $f(0) = 2$. Then T still depends on (X,Y) but now $T \perp Y \mid X$ and indeed T can now be safely ignored: the answers $P(Y_1 = 1 \mid X = 1) = 0.8$ and $P(Y_2 = 1 \mid X = 0) = 0.9$ are now valid.

But now, suppose that X is hidden from you yet you are still asked to predict Y_t; I still play scenario B but you don't know this. It is then standard practice for you to use the *marginal* distribution of Y_t, $P^*(Y_t) := \sum_x P^*(Y_t, X = x)$. In this case, when you predict Y_1, you will say that $P(Y_1 = 1) = 0.6$ (the marginal) whereas in fact, because I asked you for Y_1, it is 0.8 in this case. The problem is that, since you don't condition on X, Y still depends on T and hence you cannot ignore T when predicting Y. Thus, standard marginalization can be

invalid when T is not independent of Y — just as we saw that conditioning on X could be invalid when T is not conditionally independent of Y given X. Now a sufficient (not necessary, see Def. 3 below) condition for valid prediction is that $T \perp_{\mathcal{P}^*} Y$ (since we marginalize, there is no conditioning on X any more). *Whenever we marginalize a probability in a practical application, we implicitly make an assumption like this!*

4.1 Main Definitions and Main Result

As in the examples, in our definition of a *predictive system* below, we represent a situation by a set $\mathcal{P}^*$ rather than a single P^* to reflect our ignorance: we believe that one $P^* \in \mathcal{P}^*$ is true (in a more Bayesian interpretation, it is the appropriate representation of our uncertainty), but we do not know which one.

Definition 1. *Let $\mathcal{P}^*$ be a set of distributions on $\mathcal{Z}$, and let X, Y be RVs on $\mathcal{Z}$ with ranges $\mathcal{X}, \mathcal{Y}$ such that $(*)$ all $P^* \in \mathcal{P}^*$ agree on (X, Y). Let $\mathcal{S}, \mathcal{T}$ be finite sets, let $\{X_s \mid s \in \mathcal{S}\}$ be a collection of RVs on $\mathcal{Z}$ such that for all $s \in \mathcal{S}$, $X_s \rightsquigarrow X$; let $\{Y_t \mid t \in \mathcal{T}\}$ be a collection of RVs on $\mathcal{Z}$ such that for all $t \in \mathcal{T}$, $Y_t \rightsquigarrow Y$. We call the collection* $\mathbf{PS} = (\mathcal{P}^*, \mathcal{Z}, \mathcal{S}, \mathcal{T}, \{X_s \mid s \in \mathcal{S}\}, \{Y_t \mid t \in \mathcal{T}\})$ *a* predictive system. *We call any RV (typically denoted S) that maps $\mathcal{Z}$ to $\mathcal{S}$ an OSM for* **PS** *; and any RV (denoted T) that maps $\mathcal{Z}$ to $\mathcal{T}$ a TSM for* **PS** *.*

Thus, we consider a setting in which, by $(*)$, the distribution $P^*(X, Y)$ is known to the DM (decision-maker). Since (X, Y) determine all variables X_s that we may observe and all variables Y_t to be predicted, the distribution of these RVs is known as well. The DM observes $X_S = x$, i.e. $X_s = x$ is observed for some X_s; but the X_s whose value is presented, is itself determined, perhaps randomly, by OSM S. Given this observation, DM has to predict RV Y_T, i.e. specify a distribution on Y_t for a t which itself determined, perhaps randomly, by TSM T. *Since we specifically do not require that all $P^* \in \mathcal{P}^*$ agree on* (S, T), DM may be ignorant on the actual details of the distribution of (S, T). Our goal is to find out whether it makes sense for DM to predict Y_T given X_S, S, T based on distributions that ignore S and/or T — this is what actual DMs (people) usually do and we want to see when they can get away with it. In many cases S and/or T are not observed, so DM cannot even condition on them; also their distribution may be unknown (not all $P^* \in \mathcal{P}^*$ may agree on them), so in such cases DM cannot even marginalize them out; he can just ignore them by acting as if the randomly determined (S, T) are actually not random but fixed in advance. The standard predictive distribution used by such a DM upon observing x is thus given by

$$\tilde{P}_{\text{standard}}(y \mid x, s, t) := P^*(Y_t = y \mid X_s = x), \tag{3}$$

the conditional distribution of Y_T that would arise if T were fixed in advance to t and S were fixed in advance to s. Yet the 'correct' conditional distribution is a member of the set $\{P(Y_t = y \mid X_s = x, S = s, T = t) : P \in \mathcal{P}^*\}$, and $\tilde{P}_{\text{standard}}(y \mid x, s, t)$ may not be equal to it. We want to find out when it can be safely used any way — this is determined in Definition 2 and Theorem 1 below.

Note that $\tilde{P}_{\text{standard}}$ can be calculated without knowing the *distribution* of T or S and in some cases even without knowing the realized *value* s (CAR settings, Example 6 below). Note also that $\tilde{P}_{\text{standard}}$ does not 'marginalize out' S or T; it just pretends they are not random at all.

As seen in Example 5, a DM sometimes likes to predict Y or Y_T based on the *marginal* distribution of Y, with X marginalized out. In our setting, if X is marginalized out and S and T are ignored, this marginal distribution becomes

$$\tilde{P}_{\text{marginal}}(y \mid x, s, t) := P^*(Y_t = y) = \mathbf{E}_{X_s \sim P^*}[P^*(Y_t = y \mid X_s)] \tag{4}$$

Note that this distribution can be calculated without knowing either x or s or the distribution of S or T, but the treatment is asymmetrical: S and T are just ignored, X is marginalized out. Below we will see that it is sometimes smart to use $\tilde{P}_{\text{marginal}}$ rather than $\tilde{P}_{\text{standard}}$ even in situations in which x is observable.

The following definition can be applied to more general predictive distributions $\tilde{P}_\phi$ defined as $\tilde{P}_\phi(y \mid x, s, t) := P^*(Y_t = y \mid \phi(X_s) = \phi(x))$, for some function $\phi : \bigcup_{s \in \mathcal{S}} \mathcal{X}_s \to \Phi$ (the special case with $S \equiv T \equiv 0$ and $X_0 \equiv X$, $Y_0 \equiv Y$, so that the TSM and OSM play no role, corresponds to the notion of "$\mathcal{C}$-conditioning" from Grünwald and Halpern (2011) with $\phi(x) = \mathcal{C}(x)$). $\tilde{P}_{\text{marginal}}$ and $\tilde{P}_{\text{standard}}$ are the special cases that use $\phi(x) \equiv 1$ and $\phi(x) = x$ respectively; for overall notational consistency we always include argument x in $\tilde{P}_\phi(y \mid x, s, t)$, even for $\tilde{P}_{\text{marginal}}$ which doesn't really depend on x.

Definition 2. *We say that a predictive distribution $\tilde{P}$ is* guaranteed-to-be-valid *(GTBV) for $Y_T \mid X_S$ relative to a predictive system* **PS** *if for all $P \in \mathcal{P}^*$, all $s, t, x, y \in \text{SUPPORT}_P(S, T, X_s, Y)$,*

$$\tilde{P}(y \mid x, s, t) = P(Y_t = y \mid X_s = x, S = s, T = t). \tag{5}$$

We say that $\tilde{P}_\phi$ is safe *for $Y_T \mid X_S$ if for all $(s, t) \in \text{SUPPORT}_{\mathcal{P}^*}(S, T)$, for all $P \in \mathcal{P}^*$, all x, y with $(s, t, x, y) \in \text{SUPPORT}_P(S, T, X_s, Y)$,*

$$\tilde{P}_\phi(y \mid x, s, t) = P(Y_t = y \mid \phi(X_s) = \phi(x), S = s, T = t). \tag{6}$$

In the full paper we extend the definition of safety to general $\tilde{P}$, but below we only use it for $\tilde{P}$ equal to $\tilde{P}_\phi$ for some ϕ as above. Intuitively, when observing X_S and having to predict Y_T, we would ideally like to use a $\tilde{P}$ that is GTBV. However, when the distributions of S and/or T are unknown, we cannot always determine this $\tilde{P}$. In some cases, we may still have that $\tilde{P}_{\text{standard}}$ is GTBV; but this will in general only be the case if the OSM S and TSM T play no crucial role, as formalized in Theorem 1 below. If S cannot be ignored, then we cannot determine a GTBV $\tilde{P}$ any more; but, as also shown in Theorem 1, if S cannot but T *can* be ignored, we can resort to predicting by $\tilde{P}_{\text{marginal}}$ and our predictions will still be 'safe'. 'Safety' is the condition that we always implicitly have to assume any way whenever we want to use a marginal distribution. In a frequentist view, if we use a 'safe' $\tilde{P}_\phi$ to repeatedly predict Y_T given X_S, where the (X_S, Y_T) pairs are sampled i.i.d. from some $P^* \in \mathcal{P}^*$ (hence $\mathcal{P}^*$ is 'true'),

then the data *will behave exactly as if* $\tilde{P}_\phi$ *were the true conditional distribution.* The only way to find out whether data behave differently than predicted by $\tilde{P}_\phi$ would be to test $\tilde{P}_\phi$ (use it to make predictions) in situations in which Y_t depends on (S,T) given $\phi(X_s)$, yet does not depend on (S,T) given X_s (as in the final example in Ex. 5, where $\phi(X_s) \equiv 0$). Yet, since for $\tilde{P}_\phi$ the left-hand-side in (6) is equal to $P^*(Y_t = y \mid \phi(X_s) = \phi(x))$, the definition of 'safe' rules out exactly such T. In a Bayesian interpretation, no Dutch book can be made against $\tilde{P}_\phi$ by an adversary, unless that adversary has information about X that gets lost under the coarsening $\phi(X)$.

Definition 3. *We say that* T *represents an* ignorable TSM *for* $Y_T \mid X_S$ *if for all* $t \in \text{SUPPORT}_{\mathcal{P}^*}(T)$, $Y_t \perp_{\mathcal{P}^*} I_{T=t} \mid X_S$. *We say that* S *represents an* ignorable OSM *for* $Y_T \mid X_S$ *if for all* $s \in \text{SUPPORT}_{\mathcal{P}^*}(S)$, $Y_T \perp_{\mathcal{P}^*} I_{S=s} \mid X_s$.

We encountered ignorable TSMs in Example 5, where we had $X_S \equiv X$ (so the OSM plays no role), and we suggested the simpler but unnecessarily strong condition $Y \perp_{\mathcal{P}^*} T \mid X$, which implies that for all supported t, $P^*(Y_t \mid T = t, X) = P^*(Y_t \mid X)$, which coincides with the form in Definition 3. The analogously defined ignorable OSMs are related to CAR (Example 6). In normal, day-to-day probability uses, if X is not observed, we would like to use the marginal distribution $\tilde{P}_{\text{marginal}} = \tilde{P}_\phi$ for $\phi \equiv 0$, but if the TSM is not ignorable for Y_T, i.e. for $Y_T \mid \phi(X)$, then the resulting predictions can be disastrous, as shown at the end of Ex. 5; marginalization if X is unobserved can only be justified if T is ignorable for $Y_T \mid \phi(X)$. Now we turn the argument on its head: if T is ignorable for $Y_T \mid \phi(X)$ and X *is* observed, but the set of conditional distributions $\mathcal{P}^*(Y \mid X)$ may lead to bad predictions because it is too widely dilated, then it is preferable to use $\tilde{P}_{\text{marginal}}$ — since it is safe and the testing process is ignorable, data will behave exactly as if $\tilde{P}_{\text{marginal}}$ were fully valid, as shown in Theorem 1, part 2:

Theorem 1. *Let* **PS** *be a predictive system. (1) Suppose that* T *is an ignorable TSM for* $Y_T \mid X_S$ *and that* S *is an ignorable OSM for* $Y \mid X_S$. *Then* $\tilde{P}_{\text{standard}}$ *is safe for* $Y_T \mid X_S$. *(2) Suppose that* T *is ignorable for* $Y_T \mid \phi(X_S)$ *for some function* ϕ. *Then (even if* S *is not ignorable for* $Y \mid X_S$*),* $\tilde{P}_\phi$ *is safe for* $Y_T \mid X_S$.

Example 6. [**Embedding Event-Based Conditioning**] We can extend the idea of Example 4 to represent general overlapping event-based conditioning scenarios to our predictive systems. Given any collection $\mathcal{X}$ of nonempty subsets of $\mathcal{Y}$, we may simply set $\mathcal{S} = \mathcal{X}$, set $\mathcal{Z} = \mathcal{Y} \times \mathcal{S}$ and define, for each $s \in \mathcal{S}$, the RV X_s by $X_s((y,s')) = s$ if $y \in s$ and $X_s((y,s')) = \mathcal{Y} \setminus s$ otherwise — thus $X_s = s$ iff $Y \in s$. Assuming a trivial task selection mechanism ($\mathcal{T} = \{1\}$, $Y_1 = Y$, only the fixed RV Y has to be predicted), this re-represents event-based conditioning in terms of RVs. Observing set y translates to observing $X_S = y$; $\tilde{P}_{\text{standard}}$ corresponds to naive conditioning, since now $\tilde{P}_{\text{standard}}(y \mid x, s, t) = P^*(Y_1 = y \mid X_s = x) = P^*(Y = y \mid X_s = s) = P^*(Y = y \mid y \in s)$. The CAR condition (end of Section 2) expresses under what conditions on $\mathcal{P}^*$ naive conditioning is valid, i.e., in our new language, when $\tilde{P}_{\text{standard}}$ coincides with the true conditional distribution $P^*(Y = y \mid X_S = x, S = s)$ (we assumed $T \equiv 1$ so

T can be ignored). By Definition 2 and Theorem 1 we see that CAR is implied if S is an ignorable OSM. With a little more work one shows that, for every event-based conditioning problem, one can construct an S as above, leading to the conclusion that 'ignorable S' generalizes standard CAR; similarly, we can think of 'ignorable T' as a kind of general 'prediction-task CAR'.

Example 7. **[Conclusion: Monty Hall, revisited]** We can model Monty Hall as a predictive system as in Definition 1 in complete analogy to the dice example: $Y \in \{a, b, c\}$; we observe X_S, with $S \in \{1, 2\}$, $X_1 = \{a, b\}$ and $S = 1$ if door c is opene; and $X_2 = \{a, c\}$ and $S = 2$ otherwise. We set $T \equiv 1$, i.e. the prediction task is independent of (X, Y). We want to find the distribution of $Y_1 = I_{Y=a}$. Checking Def. 2 we find that $\tilde{P}_{\text{standard}}$ (naive conditioning, see above) is *unsafe* for $Y_1 \mid X_S$. Yet, by Theorem 1, $\tilde{P}_{\text{marginal}}$ is *safe* for $Y_1 \mid X_S$. Hence, $\tilde{P}_{\text{standard}}$ should be avoided; yet if the goal is to predict $Y_1 = I_{Y=a}$, we advocate the use of $\tilde{P}_{\text{marginal}}$: if all uncertainty can be represented by a single distribution P^* and X is observable, then it is always preferable to predict with $\tilde{P}_\phi$ with $\phi(X) \equiv X$ and not marginalize, since our predictions will be sharper. Yet if uncertainty is represented by a *set* $\mathcal{P}^*$, as here, then the set of true conditional distributions given X_S may be dilated; and then, as long as it is safe, updating by $\tilde{P}_\phi$ for coarser ϕ may be preferable. This is the case here, where $\mathcal{P}^*(Y_1 = 1 \mid \{a, b\}) = [0, 1/2]$ whereas $\tilde{P}_{\text{marginal}}(Y_1 = 1 \mid \{a, b\}) = 1/3$ is precise, undilated and safe — so let's use it!

But now let $Y_2 = I_{Y=b}$. Can we also say that $\tilde{P}(Y_2 \mid \{a, b\}) = 2/3$? It turns out that this is still 'safe', but in a weaker sense than before; this will be treated in the full paper.

Acknowledgements. for the anonymous referees who gave insightful comments.

References

De Cooman, G., Zaffalon, M.: Updating beliefs with incomplete observations. Artificial Intelligence 159(1), 75–125 (2004)

Gill, R.: The three door problem...-s. Invited Contribution to Springer's International Encyclopeadia of Statistical Science (2011)

Grünwald, P.D., Halpern, J.Y.: Updating probabilities. Journal of Artificial Intelligence Research (JAIR) 19, 243–278 (2003)

Grünwald, P.D., Halpern, J.Y.: Making decisions using sets of probabilities: Updating, time consistency, and calibration. Journal of Artificial Intelligence Research (JAIR) 42, 393–426 (2011)

Heitjan, D.F., Rubin, D.B.: Ignorability and coarse data. Annals of Statistics 19, 2244–2253 (1991)

Savage, L.J.: The Foundations of Statistics. Dover Publications (1954)

Seidenfeld, T., Wasserman, L.: Dilation for convex sets of probabilities. The Annals of Statistics 21, 1139–1154 (1993)

Shafer, G.: Conditional probability. International Statistical Review 53(3), 261–277 (1985)

Shafer, G.: The art of causal conjecture. The MIT Press (1996)

vos Savant, M.: Ask Marilyn. St. Martins Mass Market Paperback (1994)

Maximin Safety: When Failing to Lose Is Preferable to Trying to Win*

Brad Gulko and Samantha Leung

Department of Computer Science, Cornell University
{bgulko,samlyy}@cs.cornell.edu

Abstract. We present a new decision rule, *maximin safety*, that seeks to maintain a large margin from the worst outcome, in much the same way minimax regret seeks to minimize distance from the best. We argue that maximin safety is valuable both descriptively and normatively. Descriptively, maximin safety explains the well-known *decoy effect*, in which the introduction of a dominated option changes preferences among the other options. Normatively, we provide an axiomatization that characterizes preferences induced by maximin safety, and show that maximin safety shares much of the same behavioral basis with minimax regret.

1 Introduction

Representing uncertainty using a probability distribution, and making decisions by maximizing expected utility, is widely accepted, founded on formal mathematical principles, and satisfies intuitive notions of rationality such as independence of irrelevant alternatives and the sure thing principle [20]. However, enforcing seemingly appealing concepts of rationality can ultimately lead to decisions inconsistent with what real humans consider reasonable. For example, observed behavior under unquantified (Knightian [14]/ strict [15]) uncertainty, such as that in the Ellsberg paradox [8], demonstrates how appealing concepts of rationality can lead to inconsistency with human choices. Alternative decision rules, such as maximin utility [25] and minimax regret [20,17] provide rationally plausible decisions in ambiguous situations and can be used to resolve such paradoxes, but still fail to explain some human behavioral patterns. A particularly illustrative example of such behavior is called the *decoy effect* [13], in which the introduction of a *dominated* option changes the preference among the *undominated* ones. While the decoy effect has been investigated in the psychology [6,26] and economics literature [3,22], we are unaware of any axiomatic treatment of it. To address this, we introduce a criterion called *safety* as the basis for a *maximin safety* decision rule. [1] Safety serves as a dual to regret that quantifies

* We thank Joseph Y. Halpern for useful discussions and anonymous reviewers for useful comments. Work partly supported by NSF grants IIS-0812045 and CCF-1214844, and ARO grant W911NF-09-1-0281.

[1] This decision rule has been mentioned in passing, inside a proof by Hayashi [11], where it was referred to as 'maximin joy'. We use the term 'safety' rather than 'joy' to avoid confusion with the concept called 'joy of winning' in [11].

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 254–265, 2013.

distance from a worst outcome, much as regret quantifies proximity to a best outcome. Maximin safety also satisfies familiar properties common to maximin utility and minimax regret, and hence also resolves the Ellsberg paradox. Moreover, maximin safety accommodates observed preferences that are incompatible with minimax regret and maximin utility. We demonstrate how safety-seeking behavior can produce the decoy effect, and show how maximin safety can explain it. We also extend Stoye's [24] axiomatizations of standard decision rules to include maximin safety, thus allowing a comparison between maximin safety and state-of-the-art decision rules.

1.1 Relative Preferences and Regret

It is not hard to imagine situations in which performance *relative* to other possible outcomes is more important than *absolute* performance. Consider, for example, a group of duck hunters surprised by a hungry bear [5,4]. The hunters all attempt to escape by running in the same direction while the slowest one despairs: "this is hopeless, we can never outrun the bear." The hunter in front of him snickers, "I don't need to outrun the bear, I just need to outrun *you*." Whether the prospect is being picked from a group of peers for a date [2], winning a gold medal, or obtaining an 'A' in a class, success is often measured by relative performance, rather than by an absolute standard. One such preference for relative performance is embodied in the well-known decision theoretical concept of *regret* [20,17]. While psychological literature on regret focuses on the bad feelings that occur *after* a choice leads to an inferior outcome, some also considers that anticipation of such negative emotions may influence the choice itself [22,16,19].

In this paper, we assume that uncertainty is captured by a set of possible worlds, one of which is the true state of the world. Regret is a measure of distance between the value of a considered outcome and the value of the best possible outcome, under a given state. This leads to an important property that is always true for regret – the introduction of a *dominated* option does not change the regrets of the existing options. We will refer to this property as *independence of dominated alternatives* (IDA). Those who believe in regret avoidance may think that this property is perfectly reasonable. For example, suppose you have a $10 bill and you can either buy a $10 lottery ticket, or two $5 lottery tickets. Most would agree that your choice should not be affected by a dominated third option, "burning the $10 bill". Other standard decision rules, such as expected utility maximization, have even stronger independence guarantees. The ranking of two choices under expected utility maximization is *menu-independent*, i.e., completely independent of the set of feasible choices (the *menu*). Menu-independence implies IDA. In contrast, regret-based preferences are menu-dependent, but since they conform to IDA, they are not compatible with observed biases sensitive to dominated options [2]. While IDA seems intuitively appealing, there is a great deal of empirical evidence that human preferences are indeed affected by dominated options in measurable and sometimes profound ways.

1.2 The Decoy Effect in Decision Theory

Suppose you are offered $6 in cash, and the option of trading it for a Cross pen. The pen is nice, but you have plenty of pens, so decide to keep the cash. Right before you walk away, you are offered an alternative pen in exchange for the $6. You see the new pen and find it hideous. A smile comes to your face as you turn around and say, "you know, I'll take that original Cross pen after all."

This story dramatizes an actual experiment [13]. When the first choice was offered to 106 people, 64% took the cash, 36% took the pen. When the second pen was added to the offer to 115 other subjects, 52% took the cash, 46% took the Cross pen, and 2% took the decoy. Generally, a *decoy* is an option that is designed to be inferior to another option in every way (i.e., it is a dominated option). Despite the intuitive appeal of IDA, the presence of a dominated option drove selection of the Cross pen from 36% to 46%. In this paper, we focus on a particular class of *decoy effect*, called *asymmetric dominance*, which occurs when the decoy is dominated by one existing alternative, but not by another. Empirical studies show that the decoy is rarely chosen, but its addition to a set of choices consistently drives decision makers toward the *dominating choice*.

Numerous empirical studies have also shown decoy effects in class action settlements [27], recreational land management [3], choice of healthcare plans and political candidates [12], purchase of consumer goods such as cameras and personal computers [22], restaurant choices [13], and even romantic attraction [2]. Surprisingly, a decoy effect can occur even if the decoy is not actually an option, but merely a recent memory of an option (a phantom decoy [9,6]). Furthermore, the decoy effect is not limited to humans, but is also observed in honeybees and grey jays [21].

In an attempt to explain the decoy effect, experts in the behavioral sciences have offered a variety of domain-specific analyses, including "perceptual framing" [13], "value-shift" [26], "extremeness aversion" [22], and "contrast bias" [22,27]. All of these explanations focus on valuing the discrepancy between the decoy and the dominating alternative. Intuitively, this provides a compelling example of preferring the margin of safety from the worst outcome. As we are not aware of any formalization in decision theory that is consistent with the intuitive preference for "margin of safety", we offer one here.

To illustrate our new decision rule, recall the example of the unfortunate duck hunters. As they run from the bear, they approach a blind curve and have no idea what is around it: it could be wet or dry. If it is dry they will cover the most ground if they try to run faster, however if it is wet (thus slippery) they will be better off if they slow down and maintain balance. The options and the distance traveled under each circumstance are summarized in Table 1. In general, exerting excessive effort on a wet road leads to slipping and less distance covered; exerting effort on a dry road leads to more distance covered.

Table 1. Hunters running from a bear

	Wet Road	Dry Road
Sprint	1	9
Hustle	3	6
Jog	2	2

If the probability of the road conditions is unknown, and only the first two options are available (sprint and hustle), there is no intuitively preferred choice

and we may assume there are enough hunters such that at least one will pick each option. However, if we add a new option, jog, something interesting happens. As jog is dominated by hustle, IDA requires that its availability should not change the preferences among the other options. However, regardless of whether the road is wet or dry, hustle is never the worst alternative: if the road is wet, hustle (3) is faster than sprint (1), and if the road is dry, hustle (6) is faster than jog (2). In either case, selecting hustle prevents the hunter from being the slowest and getting caught by the bear.

While it may be callous, it seems perfectly reasonable for a hunter to decide to run just fast enough to make sure there is someone behind him. In other words, the most sensible decision might be to run just fast enough to guarantee the maximum possible margin between himself and the slowest runner, in the worst scenario. This margin between the hunter and his slowest compatriot can be considered a measure of *safety*, which is at the heart of our paper.

The rest of the paper proceeds as follows. Section 2 provides a formalization of the decoy paradox along with basic decision-theoretical notation. Section 3 describes the relationship between minimax regret and *maximin safety* and shows how maximin safety resolves the decoy paradox. Section 4 provide an axiomatic characterization of maximin safety. Section 5 suggests a unification of utility, regret, and safety using *anchoring functions*, and also considers a generalization to qualitative relative preferences.

2 The Formal Framework

Given a set S of *states* and a set X of *outcomes*, an *act* a (over S and X) is a function mapping S to X. The set of all acts is thus X^S, which we will denote by A. For simplicity in this paper, we take S to be finite. Associated with each outcome $x \in X$ is a *utility*: $U(x)$ is the utility of outcome x. For convenience, we will omit the explicit representation of the outcome, and denote $U(a(s))$ by $U(a, s)$ for each state $s \in S$. We call a tuple (S, X, U) a (non-probabilistic) decision problem. To define regret and safety, we need to assume that we are also given a set $M \subseteq A$ of feasible acts, called the *menu*. The reason for the menu is that, as we have shown, regret and safety can depend on the menu. We will only consider finite menus, from which randomized strategies can be chosen.

Consider the problem of a decision maker (DM) contemplating a camera purchase, summarized in Table 2. The DM has a choice between buying a rugged travel camera (a_1) that takes decent pictures in a wide variety of circumstances, and buying a delicate sports camera with higher speed and image quality (a_2). Each state characterizes the possible situations that a purchaser may experience during the useful life of the camera (Will the DM experience harsh conditions? Or win tickets

Table 2. Utilities in the camera purchase example

	s_1 Safari	s_2 World Cup
a_1: Travel	4	4
a_2: Sports	2	6
a_3: Decoy	3	3

Table 3. Standard decision rules and most valued acts in the camera example

Decision Rule	Value of an act a	Decision rule description	Best
maximax utility	$V(a) = \max_{s \in S} U(a, s)$	Optimize the best-case outcome.	a_2
maximin utility	$V(a) = \min_{s \in S} U(a, s)$	Optimize the worst-case outcome.	a_1
minimax regret	$V(a, M) = -reg_M(a)$	Pick an act to minimize the worst-case distance from the best outcome.	a_1, a_2

to the World Cup?) The utility $U(a, s)$ of act a under state s represents an abstract net value to the DM if the true world is state s.

If the DM ends up going on a safari (s_1), then act a_1 results in moderate quality pictures of exciting wildlife ($U(a_1, s_1) = 4$), but act a_2 results in a few exquisite shots and many missed opportunities ($U(a_2, s_1) = 2$). On the other hand, if the DM goes to the World Cup (s_2), then act a_2 results in many great pictures in a safe environment ($U(a_2, s_2) = 6$), while act a_1 provides only moderate quality pictures ($U(a_1, s_2) = 4$).

If the DM can assign probabilities $P(s_1)$ and $P(s_2)$ to the states, she can calculate an expected utility $E[U(a_i)] = \sum_{s \in S} P(s)U(a_i, s)$, and simply select the act that maximizes expected utility. However, if the state probabilities are unavailable, we have unquantified uncertainty. In such cases, the DM must find another method for aggregating the utility of each act across states in order to assign a *value* to each camera. Here we will focus on the methods of maximax utility, minimax utility, and minimax regret. To understand minimax regret, we need to define the notion of regret. For a menu M and act $a \in M$, the regret of a with respect to M and decision problem (S, X, U) is

$$\max_{s \in S}(\max_{a' \in M} U(a', s) - U(a, s)).$$

We denote this as $reg_M^{(S,X,U)}(a)$, and usually omit the superscript (S, X, U).

When comparing decision rules, it is often convenient to define a *value function* that assigns a numeric value to each act, for the purpose of ranking the acts. Formally, for a decision problem (S, X, U), a value function is a function

$$V^{(S,X,U)}(a, M) : X^S \times 2^A \to \mathbb{R}.$$

We will usually omit the superscript (S, X, U) and just write $V(a, M)$, or $V(a)$ if the value function is menu-independent.

We say that the value function V *represents* the family of preference relations $\succ_{V,M}$, if for all menus M and all $a, a' \in M$,

$$a \succ_{V,M} a' \Leftrightarrow V(a, M) > V(a', M).$$

In other words, act a is (strictly) preferred to act a' with respect to menu M if and only if $V(a, M) > V(a', M)$. The value functions and preferences of several standard decision rules are given in Table 3.

Now, perhaps the camera vendor would like to sell more travel cameras, so the vender puts an obsolete travel camera a_3 next to a_1 as a *decoy*. Camera

a_3 has the same price as a_1, but fewer features and lower picture quality. The vendor hopes to make a_1 more appealing by contrast with a_3. Table 2 illustrates the decision problem when a_3 is added to the menu. The ranking between a_1 and a_2 according to each of the decision rules in Table 3 is unaffected by the introduction of a_3 to the menu. The addition of a_3 also illustrates the concept of dominance. We say that an act a *dominates* a', if for all $s \in S$, $U(a, s) > U(a', s)$.

3 Maximin Safety

While minimax regret seeks to minimize separation from best outcomes, *maximin safety* is a conceptual dual that seeks to maximize separation from the worst outcomes. For a menu M and act $a \in M$, the safety of a in state s is defined as:

$$safety_M^{(S,X,U)}(a, s) = U(a, s) - \min_{a' \in M}(U(a', s)),$$

and in keeping with the convention for regret, the safety of an act is defined as:

$$safety_M^{(S,X,U)}(a) = \min_{s \in S}(safety_M^{(S,X,U)}(a, s)).$$

We will often omit the superscript (S, X, U).

The family of maximin safety preferences $\succ_{saf,M}$ represented by the *safety* value function satisfies, for all M and $a, a' \in M$,

$$a \succ_{saf,M} a' \Leftrightarrow safety_M(a) > safety_M(a').$$

Table 4. Camera purchase with and without decoy

	Utility		Safety (no decoy)		Safety (w. decoy)	
	s_1	s_2	s_1	s_2	s_1	s_2
a_1: travel	4	4	2	0	2	1
a_2: sports	2	6	0	2	0	3
a_3: decoy	3	3			1	0

Table 5. Different decision rules select different acts for the same problem

	Utility		Regret		Safety		Optimal for
	s_1	s_2	s_1	s_2	s_1	s_2	
a_1	1	9	3	0	0	5	maximax utility
a_2	3	6	1	3	2	2	maximin safety
a_3	2	7	2	2	1	3	minimax regret
a_4	4	4	0	5	3	0	maximin utility

Now we reconsider the camera example using safety (Table 4). Without the decoy, both acts have the same safety of 0, since each act has the lowest utility in some state; so there is no clear safety preference. However, when the decoy is present, the act a_1 never has the lowest utility at any state, and thus it has a strictly positive safety. In this case, $safety_{\{a_1,a_2,a_3\}}(a_1)$ is the unique maximum among the acts $\{a_1, a_2, a_3\}$, and therefore a_1 is the preferred choice. The relative increase in the safety of an act due to the addition of the dominated act is an essential element in solving the decoy paradox. Intuitively, this may correspond to a sense that even if a particular act gets low utility in the realized state, the DM may think that "I'm better off than the fools who bought the worse camera",

or in a more positive light, "I must be getting a steal with this better camera for the same price". In competitive survival games (such as the reality game show Survivor), the notion of maximizing safety may also embody a preference to maintain a maximal distance from the lowest performer, which reduces the chance of elimination. Table 5 compactly demonstrates how choices based on maximin safety differs from the other standard decision rules.

In the camera example, the addition of a decoy created a strict preference between two acts that were initially tied. The introduction of a dominated act can actually *reverse* preferences between acts. Table 6 shows a menu of three acts: $M = \{a_1, a_2, a_3\}$. Act a_2 has a minimum safety of 1, while both a_1 and a_3 have the lowest utility for some state, so each has minimum safety of 0. Consequently, a_2 is the most preferred choice under the safety preference. When a new choice a_4 is added, act a_4 is dominated by a_3, but it has higher utility than the other acts in some states. This situation is known as *asymmetric dominance*, which is typically associated with decoy effects. In this example, asymmetric dominance guarantees that a_3 is never one of the worst choices, and thus has a strictly positive safety value. In other words, the addition of a_4 to the menu M does not affect the safety of a_1 or a_2, but increases the safety of a_3 to make $a_3 \succ_{saf, M \cup \{a_4\}} a_2$.

Table 6. Without a_4, $M = \{a_1, a_2, a_3\}$, and $a_2 \succ_{saf,M} a_3$. Adding a_4 (dominated by a_3) reverses the maximin safety preference between a_2 and a_3.

	Utility			Safety(no decoy)			Safety (w. decoy $a4$)		
	s_1	s_2	s_3	s_1	s_2	s_3	s_1	s_2	s_3
a_1	9	2	6	5	0	0	8	0	0
a_2	5	3	7	1	1	1	4	1	1
a_3	4	8	8	0	6	2	3	6	2
a_4	1	5	6				0	3	0

4 Axiomatic Analysis

To provide an axiomatic characterization of maximin safety, we employ the standard *Anscombe-Aumann* (AA) framework [1], where outcomes are restricted to lotteries. Maximin safety is characterized by modifying one of the axioms in an existing characterization of minimax regret provided by Stoye [24].

Given a set Y of *prizes*, a *lottery* over Y is just a probability with finite support on Y. As in the AA framework, we let the set of *outcomes* be $\Delta(Y)$, the set of all lotteries over Y. Thus, *acts* are functions from S to $\Delta(Y)$. We can think of a lottery as modeling objective, quantified uncertainty, while the states model unquantified uncertainty. The technical advantage of considering such a set of outcomes is that we can consider convex combinations of acts. If f and g are acts, define the act $\alpha f + (1-\alpha)g$ to be the act that maps a state s to the lottery $\alpha f(s) + (1-\alpha)g(s)$. For simplicity, we follow Stoye [24] and restrict to menus that are the convex hull of a finite number of acts, so that if f and g are acts in M, then so is $pf + (1-p)g$ for all $p \in [0, 1]$.

In this setting, we assume that there is a utility function U on prizes in Y, and that there are at least two prizes y_1 and y_2 in Y, with different utilities. Note that

$l(y)$ is the probability of getting prize y if lottery l is played. We will use l^* to denote a constant act that maps all states to l. The utility of a lottery l is just the expected utility of the prizes obtained, that is, $u(l) = \sum_{\{y \in Y\,:\, l(y)>0\}} l(y)U(y)$. The expected utility of an act f with respect to a probability Pr is then just $u(f) = \sum_{s \in S} \Pr(s)u(f(s))$, as usual. Given a set Y of prizes, a utility U on the prizes, and a state space S, we have a family $\succeq_{saf,M}^{S,\Delta(Y),u}$ of preference orders on acts determined by maximin safety, where u is the utility function on lotteries as determined by U.[2] For convenience, from here on we will write $\succeq_M^{S,Y,U}$ rather than $\succeq_{saf,M}^{S,\Delta(Y),u}$. We will state the axioms in a way such that they can be compared to standard axioms and those for minimax regret in [24]. The axioms are universally quantified over acts f, g, and h, menus M and M', and $p \in (0,1)$. Whenever we write $f \succeq_M g$ we assume that $f, g \in M$.

Axiom 1. *(Monotonicity)* $f \succeq_M g$ *if* $(f(s))^* \succeq_{\{(f(s))^*,(g(s))^*\}} (g(s))^*, \forall s \in S$.

Axiom 2. *(Completeness)* $f \succeq_M g$ *or* $g \succeq_M f$.

Axiom 3. *(Nontriviality)* $f \succ_M g$ *for some acts* f *and* g *and menu* M.

Axiom 4. *(Mixture Continuity) If* $f \succ_M g \succ_M h$, *then there exists* $q, r \in (0,1)$ *such that* $qf + (1-q)h \succ_M g \succ_M rf + (1-r)h$.

Axiom 5. *(Transitivity)* $f \succeq_M g \succeq_M h \Rightarrow f \succeq_M h$.

Menu-independent versions of Axioms 1 to 5 are standard in other axiomatizations, and in particular hold for maximin utility. Axiom 3 is used in the standard axiomatizations to get a nonconstant utility function in the representation. While maximin safety does not satisfy menu-independence, it does satisfy menu-independence when restricted to menus consisting of only constant acts. This property is captured by the following axiom.

Axiom 6. *(Menu independence for constant acts) If* l^* *and* $(l')^*$ *are constant acts, then* $l^* \succeq_M (l')^*$ *iff* $l^* \succeq_{M'} (l')^*$.

We also have a menu-dependent version of the von Neumann-Morgenstern (VNM) Independence axiom. Like the VNM Independence axiom, Axiom 7 says that ranking between two acts does not change when both acts are mixed with a third act; but unlike VNM Independence, the menu used to compare the original acts in Axiom 7 is different from that used to compare the mixtures. Axiom 7 holds for minimax regret and maximin safety, but not for maximin utility.

Axiom 7. *(Independence)* $f \succeq_M g \Leftrightarrow pf + (1-p)h \succeq_{pM+(1-p)h} pg + (1-p)h$.

Axiom 8. *(Symmetry) For a menu* M, *suppose that* $E, F \in 2^S \setminus \{\emptyset\}$ *are disjoint events such that for all* $f \in M$, f *is constant on* E *and on* F. *Define* f' *by*

$$f'(s) = \begin{cases} f(s') \text{ for some } s' \in E, \text{ if } s \in F \\ f(s') \text{ for some } s' \in F, \text{ if } s \in E \\ f(s) \text{ otherwise} \end{cases}$$

[2] We let $f \succeq_{saf,M}^{S,\Delta(Y),u} g$ iff $g \not\succ_{saf,M}^{S,\Delta(Y),u} f$, and $f \sim_M g$ iff $f \succeq_M g$ and $g \succeq_M f$.

Let M' be the menu generated by replacing every act $f \in M$ with f'. Then

$$f \succeq_M g \Leftrightarrow f' \succeq_{M'} g'$$

Symmetry, which is one of the characterizing axioms for minimax regret in [24], captures the intuition that no state can be considered more or less likely than another. Therefore Symmetry helps distinguish the probability-free decision rules maximin utility, minimax regret, and maximin safety, from their probabilistic counterparts [10,23].

Axiom 9. *(Ambiguity Aversion) $f \sim_M g \Rightarrow pf + (1-p)g \succeq_M g$.*

Axiom 9 says that the decision maker weakly prefers to hedge her bets. Axioms 1-9 are all part of the characterization in [24] of minimax regret (which consists of Axioms 1-9 and Symmetry). Axioms 1-5 and 9 are also sound for the maximin decision rule [24].

In [24], one of the axioms characterizing minimax regret is Independence of Never Strictly Optimal alternatives (INA), which states that adding or removing acts that are not strictly potentially optimal in the menu does not affect the ordering of acts. [3] By varying this INA axiom, we obtain a characterization for maximin safety. We say that an act a is *never strictly worst relative to M* if, for all states $s \in S$, there is some $a' \in M$ such that $a(s) \succeq a'(s)$.

Axiom 10. *(Independence of Never Strictly Worst Alternatives (INWA)) If an act a is never strictly worst relative to M, then $f \succeq_M g$* iff *$f \succeq_{M \cup \{a\}} g$.*

Although adding acts to the menu, in general, can affect minimax regret preferences, INA implies the Independence of Dominated Alternatives property that we used earlier when discussing the decoy effect. Thus, INA guarantees that minimax regret can never be compatible with the decoy effect.

Theorem 1. *For all Y, U, S, the family of maximin safety preference orders $\succeq_{saf,M}^{S,Y,U}$ induced by a decision problem $(S, \Delta(Y), u)$ satisfies Axioms 1–10. Conversely, if the family of preference orders $\succeq_M$ on the acts in $\Delta(Y)^S$ satisfies Axioms 1–10, then there exists a utility function U on Y that determines a utility u on $\Delta(Y)$ such that $\succeq_M = \succeq_{saf,M}^{S,Y,u}$. Moreover, U is unique up to affine transformations.*

Proof. The soundness of the axioms are straightforwardly verified, so we show only the completeness of the axioms. We will use the same general sequence of arguments that Stoye uses in [24]. First, we establish a nonconstant utility function U, where constant acts are ranked by their expected utilities. Since we have the standard axioms $(1-5)$, we get U from standard arguments, and it is unique up to affine transformations. Next, we observe the following lemma:

Lemma 1. *Suppose the family $\succeq_M$ satisfies Axioms 1-10, and $\succeq_{M^+}$ is representable by maximin safety, where M^+ is the menu of all acts with nonnegative utilities. Then the family $\succeq_M$ is representable by maximin safety.*

[3] An act h is *never strictly optimal relative to M* if, for all states $s \in S$, there is some $f \in M$ such that $(f(s))^* \succeq (h(s))^*$.

Lemma 1 follows from an argument analogous to that for regret in [24]. The next step is to establish that the axioms on $\succeq_M$ restrict $\succeq_{M^+}$ to satisfy the axioms of ambiguity aversion, monotonicity, completeness, transitivity, non-triviality, and symmetry. It is a straightforward verification that will not be reproduced here. Theorem 1 (iii) of [24] then implies that $\succeq_{M^+}$ is the maximin utility ordering. Next, let g_M be an act such that $u \circ g_M(s) = -\min_{h \in M} u(h, s)$, so that we have

$$\begin{aligned} f \succeq_M g &\Leftrightarrow \tfrac{1}{2}f + \tfrac{1}{2}g_M \succeq_{M^+} \tfrac{1}{2}g + \tfrac{1}{2}g_M \\ &\Leftrightarrow \min_{s \in S} u(\tfrac{1}{2}f + \tfrac{1}{2}g_M, s) \geq \min_{s \in S} u(\tfrac{1}{2}g + \tfrac{1}{2}g_M, s) \\ &\Leftrightarrow \min_{s \in S}(\tfrac{1}{2}(u(f,s) - \min_{h \in M} u(h,s))) \geq \min_{s \in S}(\tfrac{1}{2}(u(g,s) - \min_{h \in M} u(h,s))). \end{aligned}$$

□

The characterizing axioms serve as a justification for maximin safety in the sense that behaving as a safety maximizer is equivalent to accepting the axioms. Axioms 1-7 are standard and broadly accepted to be reasonable, while symmetry and ambiguity aversion are implied by both maximin utility and minimax regret. Whether the INA axiom (for regret) or the INWA axiom (for safety) is more reasonable would depend on the individual and the nature of the decision problem. Thus, we believe that the reasonableness of the maximin safety decision rule is comparable to that of minimax regret.

Individual necessity of the axioms can be established, as is commonly done [11,24], by giving examples of preferences that satisfy all the axioms except for the one whose necessity is being shown. For the axioms shared with minimax regret, the same examples found in [24] shows their individual necessity. For the INWA axiom, the required example is minimax regret. Indeed, a decision rule equivalent to maximin safety was used by Hayashi [11] as an example to justify minimax regret's entailment of INA.

Clearly, just as minimax regret is readily generalized to *minimax expected regret* when uncertainty is represented by a set of probability distributions over the state space, maximin safety can be readily extended to *maximin expected safety* in the same manner. As one would expect, given an axiomatization of minimax expected regret [23], the modification of the INA axiom to INWA results in an axiomatization for maximin *expected* safety.

5 Discussion, Generalizations, and Future Work

Both minimax regret and maximin safety embody preferences based on *relative*, rather than *absolute* utility. In Table 5, the act preferred by safety has a lower minimum utility than the act preferred by maximin utility, just as the act picked by minimax regret neglects a higher maximum utility in order to minimize the *margin* to each state's maximum utility. The shared preference for relative over absolute performance is reflected in a striking similarity in the structure of the value functions for regret and safety. In comparison, minimax regret can be expressed for all acts a, b as:

$$a \succ_{reg,M} b \text{ iff } \min_{s \in S}(U(a,s) - \max_{a' \in M} U(a',s)) > \min_{s \in S}(U(b,s) - \max_{a' \in M} U(a',s)).$$

Similarly, maximin safety is represented for all acts a, b as

$$a \succ_{saf,M} b \text{ iff } \min_{s\in S}(U(a,s) - \min_{a'\in M} U(a',s)) > \min_{s\in S}(U(b,s) - \min_{a'\in M} U(a',s)).$$

The structural resemblance suggests a common form for the value function. By defining a menu-dependent *anchoring function* $t : S \times 2^A \to \mathbb{R}$, we can represent several previously discussed value functions as:

$$V_t(a, M) = \min_{s\in S} U'(a, s, M, t),$$

where $U'(a, s, M, t) = U(a, s) - t(s, M)$ can be viewed as an *anchored effective utility*. One can see that V_t represents maximin utility if $t(s, M) = 0$; minimax regret if $t(s, M) = \max_{a'\in M} U(a', s)$; and maximin safety if $t(s, M) = \min_{a'\in M} U(a', s)$. Note that by varying just the anchoring function t, we can obtain all the mentioned decision rules, and more. While we focus only on maximin safety in this paper, other forms for $t(s, M)$ maximize the positive margin from a state-dependent average, median, or some other characteristic of interest to a DM. For example, college students might seek to conservatively maximize their margin above a desired quantile, in order to achieve a particular grade.

The present work is motivated by behavioral observations of the decoy effect that are typically described in empirical quantities such as distance, price and volume, and thus is most intuitive in a *quantitative* framework. However, the key observation is that safety, like regret, is a notion of relative performance with respect to a set of outcomes, rather than absolute performance. As absolute quantitative utility $U : X \to \mathbb{R}$ can be generalized to a qualitative framework by replacing the U with a mapping $X \to L$ for some ordered set L, relative utility may be made qualitative by considering the mapping with $2^X \times X \to L$. In the case of safety and regret, the particular element of 2^X is the set of all possible outcomes in a state, given a menu of acts. Aggregation of N state-specific orderings into an ordering over acts can be accomplished by an aggregation function $M : L^N \to L$ [18]. This generalization can be readily applied to various characterizations of uncertainty, including probability, plausibility, and the strict uncertainty used in this paper [15]. While the authors expect that the present quantitative axiomatization can be adapted to a qualitative framework (see, e.g. [7]), it is beyond the scope of the current paper.

References

1. Anscombe, F., Aumann, R.: A definition of subjective probability. Annals of Mathematical Statistics 34, 199–205 (1963)
2. Ariely, D.: Predictably Irrational: The Hidden Forces That Shape Our Decisions, 1st edn. HarperCollins (February 2008)
3. Bateman, I.J., Munro, A., Poe, G.L.: Decoy effects in choice experiments and contingent valuation: Asymmetric dominance. Land Economics 84(1) (2008)
4. Chao, L., Hanley, K.A., Burch, C.L., Dahlberg, C., Turner, P.E.: Kin selection and parasite evolution: Higher and lower virulence with hard and soft selection. The Quarterly Review of Biology 75(3), 261–275 (2000)

5. Crawley, M.: Some sociological misconceptions, center for population biology. In: Pan-European Conference on the Potential Long-Term Ecological Impact of Genetically Modified Organisms: Proceedings, 276 Pages (November 1993)
6. Doyle, J.R., O'Connor, D.J., Reynolds, G.M., Bottomley, P.A.: The Robustness of the Asymmetrically Dominated Effect: Buying Frames, Phantom Alternatives, and In-Store Purchases. Psychology and Marketing 16(3) (1999)
7. Dubois, D., Fargier, H., Perny, P.: Qualitative decision theory with preference relations and comparative uncertainty: An axiomatic approach. Artificial Intelligence 148(12), 219–260 (2003)
8. Ellsberg, D.: Risk, ambiguity, and the savage axioms. QJE 75(4) (1961)
9. Farquhar, P.H., Pratkanis, A.R.: Decision structuring with phantom alternatives. Management Science 39(10), 1214–1226 (1993)
10. Gilboa, I., Schmeidler, D.: Maxmin expected utility with non-unique prior. Journal of Mathematical Economics 18(2), 141–153 (1989)
11. Hayashi, T.: Regret aversion and opportunity dependence. JET 139(1) (2008)
12. Hedgcock, W., Rao, A.R., Chen, H.A.: Could Ralph Nader's Entrance and Exit Have Helped Al Gore? The Impact of Decoy Dynamics on Consumer Choice. Journal of Marketing Research 46(3) (1999)
13. Huber, J., Payne, J.W., Puto, C.: Adding asymmetrically dominated alternatives: Violations of regularity and the similarity hypothesis. JCR 9(1) (1982)
14. Knight, F.H.: Risk, Uncertainty and Profit. Houghton Mifflin (1921)
15. Larbi, R.B., Konieczny, S., Marquis, P.: A characterization of optimality criteria for decision making under complete ignorance. In: KR (2010)
16. Larrick, R.P., Boles, T.L.: Avoiding regret in decisions with feedback: A negotiation example. OBHDP 63(1), 87–97 (1995)
17. Luce, R.D., Raiffa, H.: Games and decisions: introduction and critical survey. Wiley (1957)
18. Marichal, J.-L.: An axiomatic approach of the discrete Sugeno integral as a tool to aggregate interacting criteria in a qualitative framework. IEEE Transactions on Fuzzy Systems 9(1), 164–172 (2001)
19. Ritov, I.: Probability of regret: Anticipation of uncertainty resolution in choice. Organizational Behavior and Human Decision Processes 66(2), 228–236 (1996)
20. Savage, L.J.: The Foundations of Statistics. John Wiley (1954)
21. Shafir, S., Waite, T.A., Smith, B.H.: Context-Dependent violations of rational choice in honeybees (apis mellifera) and gray jays (perisoreus canadensis). Behavioral Ecology and Sociobiology 51(2), 180–187 (2002)
22. Simonson, I., Tversky, A.: Choice in context: Tradeoff contrast and extremeness aversion. Journal of Marketing Research 29(3), 281–295 (1992)
23. Stoye, J.: Axioms for minimax regret choice correspondences. JET 146(6) (2011)
24. Stoye, J.: Statistical decisions under ambiguity. Theory and Decision 70(2), 129–148 (2011)
25. Wald, A.: Statistical Decision Functions. John Wiley, NY (1950)
26. Wedell, D.H.: Distinguishing among models of contextually induced preference reversals. Journal of Experimental Psychology 17(4), 767–778 (1991)
27. Zimmerman, A.S.: Funding Irrationality. Duke Law Journal 59(1105) (2010)

Weighted Regret-Based Likelihood: A New Approach to Describing Uncertainty

Joseph Y. Halpern*

Cornell University
Ithaca, NY 14853, USA
halpern@cs.cornell.edu

Abstract. Recently, Halpern and Leung [8] suggested representing uncertainty by a *weighted* set of probability measures, and suggested a way of making decisions based on this representation of uncertainty: *maximizing weighted regret*. Their paper does not answer an apparently simpler question: what it means, according to this representation of uncertainty, for an event E to be more likely than an event E'. In this paper, a notion of comparative likelihood when uncertainty is represented by a weighted set of probability measures is defined. It generalizes the ordering defined by probability (and by lower probability) in a natural way; a generalization of upper probability can also be defined. A complete axiomatic characterization of this notion of regret-based likelihood is given.

1 Introduction

Recently, Samantha Leung and I [8] suggested representing uncertainty by a *weighted* set of probability measures, and suggested a way of making decisions based on this representation of uncertainty: *maximizing weighted regret*. However, we did not answer an apparently simpler question: given this representation of uncertainty, what does it mean for an event E to be more likely than an event E'? This is what I do in this paper. To explain the issues, I start by reviewing the Halpern-Leung approach.

It has frequently been observed that there are many situations where an agent's uncertainty is not adequately described by a single probability measure. For example, there seems to be a big difference between a coin known to be fair and a coin whose bias an agent does not know, yet if the agent were to use a single measure to represent her uncertainty, in both of these cases it would seem that the measure that assigns heads probability 1/2 would be used.

One approach for representing ignorance is to use a set $\mathcal{P}$ of probability measures [7]. That approach has the benefit of representing uncertainty in general, not by a single number, but by a range of numbers. This allows us to distinguish the certainty that a coin is fair (in which case the uncertainty of heads is

* Supported in part by NSF grants IIS-0812045, IIS-0911036, and CCF-1214844, by AFOSR grants FA9550-08-1-0438, FA9550-09-1-0266, and FA9550-12-1-0040, and by ARO grant W911NF-09-1-0281.

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 266–277, 2013.

represented by a single number, 1/2) from knowing only that the probability of heads could be anywhere between, say, 1/3 and 2/3.

But this approach also has its problems. For example, consider an agent who believes that a coin may have a slight bias. Thus, although it is unlikely to be completely fair, it is close to fair. How should we represent this with a set of probability measures? Suppose that the agent is quite sure that the bias is between 1/3 and 2/3. We could, of course, take $\mathcal{P}$ to consist of all the measures that give heads probability between 1/3 and 2/3. But how does the agent know that the possible biases are *exactly* between 1/3 and 2/3. Does she not consider $2/3 + \epsilon$ possible for some small ϵ? And even if she is confident that the bias is between 1/3 and 2/3, this representation cannot take into account the possibility that she views biases closer to 1/2 as more likely than biases further from 1/2.

There is also a second well-known concern: learning. Suppose that the agent initially considers possible all the measures that gives heads probability between 1/3 and 2/3. She then starts tossing the coin, and sees that, of the first 20 tosses, 12 are heads. It seems that the agent should then consider a bias of greater than 1/2 more likely than a bias of less than 1/2. But if we use the standard approach to updating with sets of probability measures (see [7]), and condition each of the measures on the observation, since the coin tosses are viewed as independent, the agent will continue to believe that the probability of the next coin toss is between 1/3 and 2/3. The observation has no impact as far as learning to predict better. The set $\mathcal{P}$ stays the same, no matter what observation is made.

There is a well-known solution to these problems: using a second-order measure on these measures to express how likely the agent considers each of them to be. (See [6] for a discussion of this approach and further references.) For example, an agent can express the fact that the bias of a coin is more likely to be close to 1/2 than far from 1/2. In addition, the problem of learning can be dealt with by straightforward conditioning. But this approach leads to other problems. Essentially, it seems that the ambiguity that an agent might feel about the outcome of the coin toss seems to have disappeared. For example, suppose that the agent has no idea what the bias is. The obvious second-order probability to use is the uniform probability on possible biases. While we cannot talk about the probability that the coin is heads, the *expected* probability of heads is 1/2. Why should an agent that has no idea of the bias of the coin know or believe that the expected probability of heads is 1/2? Moreover, if our interest is in making decisions, then maximizing the expected utility using the expected probability again does not take the agent's ignorance into account. Kyburg [12] and Pearl [16] have even argued that there is no need for a second-order probability on probabilities; whatever can be done with a second-order probability can already be done with a basic probability.

Nevertheless, when it comes to decision-making, it does seem useful to use an approach that represents ambiguity, while still maintaining some of the features of having a second-order probability on probabilities. One suggestion, made by Walley [18], is to put a second-order possibility measure on probability measures; see also [2,3]. Leung and I similarly suggested putting weights on each probability

measure in $\mathcal{P}$. Since we assumed that the weights are normalized so that the supremum of the weights is 1, these weights can also be viewed as a possibility measure. If the set $\mathcal{P}$ is finite, we can also normalize so as to view the weights as being second-order probabilities. As with second-order probabilities, the weights can vary over time, as more information is acquired, and can be used to represent the fact that some probabilities in the set $\mathcal{P}$ are more likely than others.

What makes this approach different from just using a second-order probability on $\mathcal{P}$ lies in how decisions are made. Leung and I used *regret*, a standard approach to decision-making that goes back to Niehans [15] and Savage [17]. If uncertainty is represented by a set $\mathcal{P}$ of probability measures, then regret works as follows: for each act a and each measure $\Pr \in \mathcal{P}$, we can compute the expected regret of a with respect to Pr; this is the difference between the expected utility of a and the expected utility of the act that gives the highest expected utility with respect to Pr. We can then associate with an act a its worst-case expected regret of a, over all measures $\Pr \in \mathcal{P}$, and compare acts with respect to their worst-case expected regret. With weights in the picture, we modify the procedure by multiplying the expected regret associated with measure Pr by the weight of Pr, and compare acts according to their worst-case *weighted* expected regret. This approach to making decisions is very different from that suggested by Walley [18]. Moreover, using the weights in the way means that we cannot simply replace a set of weighted probability measures by a single probability measure; the objections of Kyburg [12] and Pearl [16] do not apply.

Leung and I [8] show that this approach seems to do reasonable things in a number of examples of interest, and provide an elegant axiomatization of decision-making. So how can we represent relative likelihood using this approach? This is something not considered in earlier papers using sets of weighted probabilities. If uncertainty is represented by a single probability measure, the answer is immediate: E is more likely than E' exactly if the probability of E is greater than the probability of E'. When using sets of probability measures, various approaches have been considered in the literature. The most common takes E to be more likely than E' if the *lower probability* of E is greater than the lower probability of E', where the lower probability of E is its worst-case probability, taken over the measures in $\mathcal{P}$ (see Section 3). We could also compare E and E' with respect to their upper probabilities (the best-case probability with respect to the measures in $\mathcal{P}$). Another possibility is to take E to be more likely than E' if $\Pr(E) \geq \Pr(E')$ for all measures $\Pr \in \mathcal{P}$; this gives a partial order on likelihood.

In this paper, I define a notion of relative likelihood when uncertainty is represented by a weighted set of probability measures that generalizes the ordering defined by lower probability in a natural way; I also define a generalization of upper probability. We can then associate with an event E two numbers that are analogues of lower and upper probability. If uncertainty is represented by a single measure, then these two numbers coincide; in general, they do not. The interval can be thought of as representing the degree of ambiguity in the likelihood of E. Indeed, in the special case when all the weights are 1, the numbers

are essentially just the lower and upper probability (technically, they are 1 minus the lower and upper probability, respectively). Interestingly, the approach to assigning likelihood is based on the approach to decision-making. Essentially, what I am doing is the analogue of defining probability in terms of expected utility, rather than the other way around. The approach can be viewed as generalizing both probability and lower probability.

Why we should be interested in such a representation. If all that we ever did with probability was to use it to make decisions, then arguably this wouldn't be of much interest; Halpern and Leung's work already shows how weighted sets of probabilities can be used in decision-making. The results of this paper add nothing further to that question. However, we often talk about the likelihood of events quite independent of their use in decision-making (think of the use of probability in physics, to take just one example). Thus, having an analogue of probability seems important and useful in its own right.

2 Weighted Expected Regret: A Review

Consider the standard setup in decision theory. We have a state space S and an outcome space O. An *act* is a function from S to O; it describes an outcome for each state. Suppose that we have a utility function u on outcomes and a set $\mathcal{P}^+$ of *weighted probability measures*. That is, $\mathcal{P}^+$ consists of pairs $(\Pr, \alpha_{\Pr})$, where $\alpha_{\Pr}$ is a weight in $[0,1]$ and Pr is a probability on S. Let $\mathcal{P} = \{\Pr : \exists\alpha((\Pr,\alpha) \in \mathcal{P}^+)\}$. For each $\Pr \in \mathcal{P}$ there is assumed to be exactly one α, denoted $\alpha_{\Pr}$, such that $(\Pr,\alpha) \in \mathcal{P}^+$. It is further assumed that weights have been normalized so that there is at least one measure $\Pr \in \mathcal{P}$ such that $\alpha_{\Pr} = 1$.[1] Finally, $\mathcal{P}^+$ is assumed to be *weakly closed*, so that if $(\Pr_n, \alpha_n) \in \Pr^+$ for $n = 1, 2, 3, \ldots$, $(\Pr_n, \alpha_n) \to (\Pr, \alpha_{\Pr})$, and $\alpha_{\Pr} > 0$, then $(\Pr, \alpha_{\Pr}) \in \mathcal{P}^+$. (I discuss below why I require $\mathcal{P}^+$ to be just weakly closed, rather than closed.)

Where are the weights in $\mathcal{P}^+$ coming from? In general, they can be viewed them as subjective, just like the probability measures. However, as observed in [8], there is an important special case where the weights can be given a natural interpretation. Suppose that, as in the case of the biased coin in the Introduction, we make observations in a situation where the probability of making a given observation is determined by some objective source. Then we can start by giving all probability measures a weight of 1. Given an observation *ob* (e.g., sequence of coin tosses in the example in the Introduction), we can compute

[1] The assumption that at least one probability measure has a weight of 1 is convenient for comparison to other approaches; see below. However, making this assumption has no impact on the results of this paper; as long as we restrict to sets where the weight is bounded, all the results hold without change. Note that the assumption that the weights are probabilities runs into difficulties if we have an infinite number of measures in $\mathcal{P}$; for example, if $\mathcal{P}$ includes all measures on heads from 1/3 to 2/3, as discussed in the Introduction, using a uniform probability, we would be forced to assign each individual probability measure a weight of 0, which would not work well for our later definitions.

$\Pr(ob)$ for each measure $\Pr \in \mathcal{P}$; we can then update the weight of $\Pr$ to be $\Pr(ob)/\sup_{\Pr' \in \mathcal{P}} \Pr'(ob)$. Thus, the more likely the observation is according to $\Pr$, the higher the updated weight of $\Pr$.[2] (The denominator is just a normalization to ensure that some measure has weight 1.) With this approach to updating, if there is a true underlying measure generating the data, then as an agent makes more observations, almost surely, the weight of the true measure approaches 1, while the weight of all other measures approaches 0.[3]

I now review the definition of weighted regret, and introduce the notion of *absolute* (weighted) regret. I start with regret. The regret of an act a in a state $s \in S$ is the difference between the utility of the best act at state s and the utility of a at s. Typically, the act a is not compared to all acts, but to the acts in a set M, called a *menu*. Thus, the regret of a in state s relative to menu M, denoted $reg^M(a, s)$, is $\sup_{a' \in M} u(a'(s)) - u(a(s))$. There are typically some constraints put on M to ensure that $\sup_{a' \in M} u(a'(s))$ is finite—this is certainly the case if M is finite, or the convex closure of a finite set of acts, or if there is a best possible outcome in the outcome space O. The latter assumption holds in this paper, so I assume throughout that $\sup_{a' \in M} u(a'(s))$ is finite.

For simplicity, I assume that the state space S is finite. Given a probability measure $\Pr$ on S, the expected regret of an act a with respect to $\Pr$ relative to menu M is just $reg^M_{\Pr}(a) = \sum_{s \in S} reg^M(a, s) \Pr(s)$. The *(expected) regret* of a with respect to $\mathcal{P}$ and a menu M is just the worst-case regret, that is,

$$reg^M_{\mathcal{P}}(a) = \sup_{\Pr \in \mathcal{P}} reg^M_{\Pr}(a).$$

Similarly, the *weighted (expected) regret* of a with respect to $\mathcal{P}^+$ and a menu M is just the worst-case weighted regret, that is,

$$wr^M_{\mathcal{P}^+}(a) = \sup_{\Pr \in \mathcal{P}} \alpha_{\Pr} reg^M_{\Pr}(a).$$

Thus, regret is just a special case of weighted regret, where all weights are 1.

Note that, as far weighted regret goes, it does not hurt to augment a set $\mathcal{P}^+$ of weighted probability measures by adding pairs of the form $(\Pr, 0)$ for $\Pr \notin \mathcal{P}$. But if we start with an unweighted set $\mathcal{P}$ of probability measures, the weighted set $\mathcal{P}^+ = \{(\Pr, 1) : \Pr \in \mathcal{P}\} \cup \{(\Pr, 0) : \Pr \notin \mathcal{P}\}$ is not closed in general, although it is weakly closed. There may well be a sequence $\Pr_n \to \Pr$, where $\Pr_n \notin \mathcal{P}$ for all n, but $\Pr \in \mathcal{P}$. But then we would have have $(\Pr_n, 0) \in \mathcal{P}^+$ converging to $(\Pr, 0) \notin \mathcal{P}^+$. This is exactly why I required only weak closedness. Note for

[2] The idea of putting a possibility on probabilities in $\mathcal{P}$ that is determined by likelihood also appears in the work of Moral [14], although he does not consider a general approach to dealing with sets of weighted probability measures.

[3] The "almost surely" is due to the fact that, with probability approaching 0, as more and more observations are made, it is possible that an agent will make a misleading observations, that are not representative of the true measure. This also depends on the set of possible observations being rich enough to allow the agent to ultimately discover the true measure generating the observations. Since learning is not a focus of this paper, I do not make this notion of "rich enough" precise here.

future reference that, since $\mathcal{P}^+$ is assumed to be weakly closed, if $wr^M_{\mathcal{P}^+}(a) > 0$, then there is some element $(\Pr, \alpha_{\Pr}) \in \mathcal{P}^+$ such that $wr^M_{\mathcal{P}^+}(a) = \alpha_{\Pr} reg^M_{\Pr}(a)$.

Weighted regret induces an obvious preference order on acts: act a is at least as good as a' with respect to $\mathcal{P}^+$ and M, written $a \succeq^{reg}_{\mathcal{P}^+,M} a'$, if $wr^M_{\mathcal{P}^+}(a) \leq wr^M_{\mathcal{P}^+}(a')$. As usual, I write $a \succ^{reg}_{\mathcal{P}^+,M} a'$ if $a \succeq^{reg}_{\mathcal{P}^+,M} a'$ but it is not the case that $a' \succeq^{reg}_{\mathcal{P}^+,M} a$. The standard notion of regret is the special case of weighted regret where all weights are 1. I sometimes write $a \succeq^{reg}_{\mathcal{P},M} a'$ to denote the unweighted case (i.e., where all the weights in $\mathcal{P}^+$ are 1).

In this setting, using weighted regret gives an approach that allows an agent to transition smoothly from regret to expected utility. It is well known that regret generalizes expected utility in the sense that if $\mathcal{P}$ is a singleton $\{\Pr\}$, then $wr^M_{\mathcal{P}}(a) \leq wr^M_{\mathcal{P}}(a')$ iff $\mathrm{EU}_{\Pr}(a) \geq \mathrm{EU}_{\Pr}(a')$ (where $\mathrm{EU}_{\Pr}(a)$ denotes the expected utility of act a with respect to probability Pr).[4] (In particular, this means that if $\mathcal{P}$ is a singleton, regret is menu independent.) If we start with all the weights being 1, then, as observed above, the weighted regret is just the standard notion of regret. As the agent makes observations, if there is a measure Pr generating the uncertainty, the weights will get closer and closer to a situation where Pr gets weight 1, with the weights of all other measures dropping off quickly to 0, so the ordering of acts will converge to the ordering given by expected utility with respect to Pr.

There is another approach with some similar properties, that again starts with uncertainty being represented by a set $\mathcal{P}$ of (unweighted) probability measures. Define $wc_{\mathcal{P}}(a) = \inf_{\Pr \in \mathcal{P}} \mathrm{EU}_{\Pr}(a))$. Thus $wc_{\mathcal{P}}(a)$ is the worst-case expected utility of a, taken over all $\Pr \in \mathcal{P}$. Then define $a \succeq^{mm}_{\mathcal{P}} a'$ if $wc_{\mathcal{P}}(a) \geq wc_{\mathcal{P}}(a')$. This is the maxmin expected utility rule, quite often used in economics [5]. There are difficulties in getting a weighted version of maxmin expected utility [8] (see Section 3); however, Epstein and Schneider [4] propose another approach that can be combined with maxmin expected utility. They fix a parameter $\alpha \in (0, 1)$, and update $\mathcal{P}$ after an observation ob by retaining only those measures Pr such that $\Pr(ob) \geq \alpha$. For any choice of $\alpha < 1$, we again end up converging almost surely to a single measure, so again this approach converges almost surely to expected utility.

I conclude this section with a discussion of menu dependence. Maxmin expected utility is not menu dependent; the preference ordering on acts induced by regret can be, as the following example illustrates.

Example 1. Take the outcome space to be $\{0, 1\}$, and the utility function to be the identity, so that $u(1) = 1$ and $u(0) = 0$. As usual, if $E \subseteq S$, 1_E denotes the *indicator function* on E, where, for each state $s \in S$, we have $1_E(s) = 1$ if $s \in E$, and $1_E(s) = 0$ if $s \notin E$. Let $S = \{s_1, s_2, s_3, s_4\}$, $E_1 = \{s_1\}$, $E_2 = \{s_2\}$, $E_3 = \{s_2, s_3\}$, $M_1 = \{1_{E_1}, 1_{E_2}\}$, $M_2 = \{1_{E_1}, 1_{E_2}, 1_{E_3}\}$, and $\mathcal{P} = \{\Pr_1, \Pr_2\}$, where $\Pr_1(s_1) = \Pr_1(s_3) = \Pr_1(s_4) = 1/3$, $\Pr_2(s_2) = 1/4$, and $\Pr_2(s_3) = 3/4$. A straightforward calculation shows that $reg^{M_1}_{\Pr_1}(1_{E_1}) = 0$,

[4] This follows from the observation that, given a menu M, there is a constant c_M such that, for all acts $a \in M$, $wr^M_{\{\Pr\}}(a) = c_M - \mathrm{EU}_{\Pr}(a)$.

$reg^{M_1}_{\Pr_1}(1_{E_2}) = 1/3$, $reg^{M_1}_{\Pr_2}(1_{E_1}) = 1/4$, $reg^{M_1}_{\Pr_2}(1_{E_2}) = 0$, $reg^{M_2}_{\Pr_1}(1_{E_1}) = 1/3$, $reg^{M_2}_{\Pr_1}(1_{E_2}) = 2/3$, $reg^{M_2}_{\Pr_2}(1_{E_1}) = 1$, and $reg^{M_2}_{\Pr_2}(1_{E_2}) = 3/4$. Thus, $1/4 = reg^{M_1}_{\mathcal{P}}(1_{E_1}) < reg^{M_1}_{\mathcal{P}}(1_{E_2}) = 1/3$, while $1 = reg^{M_2}_{\mathcal{P}}(1_{E_1}) > reg^{M_2}_{\mathcal{P}}(1_{E_2}) = 3/4$. The preference on 1_{E_1} and 1_{E_2} depends on whether we consider the menu M_1 or the menu M_2. □

Suppose that there is an outcome $o^* \in O$ that gives the maximum utility; that is, $u(o^*) \geq u(o)$ for all $o \in O$. If $\bar{o}^*$ is the constant act that gives outcomes o^* in all states, then $\bar{o}^*$ is clearly the best act in all states. If there is such a best act, an "absolute", menu-independent notion of weighted expected regret can be defined by always comparing to $\bar{o}^*$. That is, define

$$\begin{aligned}
&reg(s,a) = u(o^*) - u(a(s));\\
&reg_{\Pr}(a) = \textstyle\sum_{s\in S}(u(o^*) - u(a(s)) \Pr(s) = u(o^*) - \mathrm{EU}_{\Pr}(a);\\
&reg_{\mathcal{P}}(a) = \sup_{\Pr\in\mathcal{P}} \textstyle\sum_{s\in S}(u(o^*) - u(a(s)) \Pr(s)\\
&\qquad\quad = u(o^*) - \inf_{\Pr\in\mathcal{P}}(\mathrm{EU}_{\Pr}(a);\\
&wr_{\mathcal{P}^+}(a) = \sup_{\Pr\in\mathcal{P}} \alpha_{\Pr} \textstyle\sum_{s\in S}(u(o^*) - u(a(s)) \Pr(s)\\
&\qquad\quad = \sup_{\Pr\in\mathcal{P}} \alpha_{\Pr}(u(o^*) - \mathrm{EU}_{\Pr}(a)).
\end{aligned}$$

If there is a best act, then I write $a \succeq_{\mathcal{P}^+} a'$ if $wr_{\mathcal{P}^+}(a) \leq wr_{\mathcal{P}^+}(a')$; similarly in the unweighted case, I write $a \succeq_{\mathcal{P}} a'$ if $wr_{\mathcal{P}}(a) \leq wr_{\mathcal{P}}(a')$.

Conceptually, we can think of the agent as always being aware of the best outcome o^*, and comparing his actual utility with a to $u(o^*)$. Equivalently, the absolute notion of regret is equivalent to a menu-based notion with respect to a menu M that includes $\bar{o}^*$ (since if the menu includes $\bar{o}^*$, it is the best act in every state). As we shall see, in our setting, we can always reduce menu-dependent regret to this absolute, menu-independent notion, since there is in fact a best act: 1_S.

3 Relative Ordering of Events Using Weighted Regret

In this section, I consider how a notion of comparative likelihood can be defined using sets of weighted probability measures.

As in Example 1, take the outcome space to be $\{0,1\}$, the utility function to be the identity, and consider indicator functions. It is easy to see that $\mathrm{EU}_{\Pr}(1_E) = \Pr(E)$, so that with this setup, we can recover probability from expected utility. Thus, if uncertainty is represented by a single probability measure Pr and we make decisions by preferring those acts that maximize expected utility, then we have $1_E \succeq 1_{E'}$ iff $\Pr(E) \geq \Pr(E')$.

Consider what happens if we apply this approach to maxmin expected utility. Now we have that $1_E \succeq^{mm}_{\mathcal{P}} 1_{E'}$ iff $\inf_{\Pr\in\mathcal{P}} \Pr(E) \geq \inf_{\Pr\in\mathcal{P}} \Pr(E')$. In the literature, $\inf_{\Pr\in\mathcal{P}} \Pr(E)$, denoted $\mathcal{P}_*(E)$, is called the *lower probability* of E, and is a standard approach to describing likelihood. The dual *upper probability*, $\sup_{\Pr\in\mathcal{P}} \Pr(E)$, is denoted $\mathcal{P}^*(E)$. An easy calculation shows that $\mathcal{P}^*(E) = 1 - \mathcal{P}_*(\overline{E})$ (where, as usual, $\overline{E}$ denotes the complement of E). The interval

$[\mathcal{P}_*(E), \mathcal{P}^*(E)]$ can be thought of as describing the uncertainty of E; the larger the interval, the greater the ambiguity.

What happens if we apply this approach to regret? First consider unweighted regret. If we restrict to acts of the form 1_E, then the best act is clearly 1_S, which is just the constant function 1. Thus, we can (and do) use the absolute notion of regret here, and for the remainder of this paper. We then get that $1_E \succeq^{reg}_{\mathcal{P}} 1_{E'}$ iff $\sup_{\Pr\in\mathcal{P}}(1 - \Pr(E)) \leq \sup_{\Pr\in\mathcal{P}}(1 - \Pr(E'))$ iff $\sup_{\Pr\in\mathcal{P}} \Pr(\overline{E}) \leq \sup_{\Pr\in\mathcal{P}} \Pr(\overline{E}')$; that is, $\mathcal{P}^*(\overline{E}) \leq \mathcal{P}^*(\overline{E}')$. Moreover, easy manipulation shows that $\sup_{\Pr\in\mathcal{P}}(1 - \Pr(E)) = 1 - \inf_{\Pr\in\mathcal{P}} \Pr(E) = 1 - \mathcal{P}_*(E)$. It follows that $1_E \succeq^{reg}_{\mathcal{P}} 1_{E'}$ iff $(1 - \mathcal{P}_*(E)) \leq (1 - \mathcal{P}_*(E'))$ iff $\mathcal{P}_*(E) \geq \mathcal{P}_*(E')$ iff $1_E \succeq^{mm}_{\mathcal{P}} 1_{E'}$; both regret and maxmin expected utility put the same ordering on events.

The extension to weighted regret is immediate. Let $\mathcal{P}^+_{reg}(E)$, the *(weighted) regret-based likelihood of* E, be defined as $\sup_{\Pr\in\mathcal{P}} \alpha_{\Pr} \Pr(\overline{E})$. If $\mathcal{P}^+$ is unweighted, so that all the weights are 1, I write $\mathcal{P}_{reg}(E)$ to denote $\sup_{\Pr\in\mathcal{P}} \Pr(\overline{E})$. Note that $\mathcal{P}_{reg}(E) = 1 - \mathcal{P}_*(E)$, so $\mathcal{P}_{reg}(E) \leq \mathcal{P}_{reg}(E')$ iff $\mathcal{P}_*(E) \geq \mathcal{P}_*(E')$. That is, the ordering induced by $\mathcal{P}_{reg}$ is the opposite of that induced by $\mathcal{P}_*$. So, for example, $\mathcal{P}_{reg}(\emptyset) = 1$ and $\mathcal{P}_{reg}(S) = 0$; smaller sets have a larger regret-based likelihood.[5]

Regret-based likelihood provides a way of associating a number with each event, just as probability and lower probability do. Moreover, just as lower probability gives a lower bound on uncertainty, we can think of $\mathcal{P}^+_{reg}(E)$ as giving an upper bound on the uncertainty. (It is an upper bound rather than a lower bound because larger regret means less likely, just as smaller lower probability does.) The naive corresponding lower bound is given by $\inf_{\Pr\in\mathcal{P}} \alpha_{\Pr} \Pr(\overline{E})$. This lower bound is not terribly interesting; if there are probability measures $\Pr' \in \mathcal{P}$ such that $\alpha_{\Pr'}$ is close to 0, then this lower bound will be close to 0, independent of the agent's actual feeling about the likelihood of E. A more reasonable lower bound is given by the expression $\underline{\mathcal{P}}^+_{reg}(E) = 1 - \mathcal{P}^+_{reg}(\overline{E})$ (recall that the analogous expression relates upper probability and lower probability). The intuition for this choice is the following. If nature were conspiring against us, she would try to prove us wrong by making $\alpha_{\Pr} \Pr(\overline{E})$ as large as possible—that is, make the weighted probability of being wrong as large as possible. On the other hand, if nature were conspiring with us, she would try to make $\alpha_{\Pr} \Pr(E)$ as large as possible, or, equivalently, make $1 - \alpha_{\Pr} \Pr(E)$ as small as possible. Note that this is different from making $\alpha_{\Pr} \Pr(\overline{E})$ as large as possible, unless $\alpha_{\Pr} = 1$ for all $\Pr \in \mathcal{P}$. An easy calculation shows that

$$\begin{aligned} 1 - \mathcal{P}^+_{reg}(\overline{E}) &= 1 - \sup_{\Pr\in\mathcal{P}} \alpha_{\Pr} \Pr(E) \\ &= \inf_{\Pr\in\mathcal{P}}(1 - \alpha_{\Pr} \Pr(E)). \end{aligned}$$

This motivates the definition of $\underline{\mathcal{P}}^+_{reg}$.

The following lemma clarifies the relationship between these expressions, and shows that $[\underline{\mathcal{P}}^+_{reg}(E), \mathcal{P}^+_{reg}(E)]$ really does give an interval of ambiguity.

[5] Since an act with smaller regret is viewed as better, the ordering on acts of the form 1_E induced by regret is the same as that induced by maxmin expected utility.

Lemma 1. $\inf_{\Pr \in \mathcal{P}} \alpha_{\Pr} \Pr(\overline{E}) \leq 1 - \mathcal{P}^+_{reg}(\overline{E}) \leq \mathcal{P}^+_{reg}(E)$.[6]

In general, equality does not hold in Lemma 1, as shown by the following example. The example also illustrates how the "ambiguity interval" can decrease with weighted regret, if the weights are updated as suggested in [8].

Example 2. Suppose that the state space consists of $\{h, t\}$ (for heads and tails); let $\Pr_\beta$ be the measure that puts probability β on h. Let $\mathcal{P}^+_0 = \{(\Pr_\beta, 1) : 1/3 \leq \beta \leq 2/3\}$. That is, we initially consider all the measures that put probability between 1/3 and 2/3 on heads. We toss the coin and observe it land heads. Intuitively, we should now consider it more likely that the probability of heads is greater than 1/2. Indeed, applying likelihood updating, we get the set $\mathcal{P}^+_1 = \{(\Pr_\beta, 3\beta/2) : 1/3 \leq \beta \leq 2/3\}$;[7] the probability measures that give h higher probability get higher weight. In particular, the weight of $\Pr_{2/3}$ is still 1, but the weight of $\Pr_{1/3}$ is only 1/2. If the coin is tossed again and this time tails is observed, we update further to get $\mathcal{P}^+_2 = \{(\Pr_\beta, 4\beta(1-\beta)) : 1/3 \leq \beta \leq 2/3\}$. An easy calculation shows that $[\underline{\mathcal{P}}^+_{0,reg}(h), \mathcal{P}^+_{0,reg}(h)] = [1/3, 2/3]$, $[\underline{\mathcal{P}}^+_{1,regret}(h), \mathcal{P}^+_{1,reg}(h)] = [1/3, 3/8]$, and $[\underline{\mathcal{P}}^+_{2,reg}(h), \mathcal{P}^+_{2,reg}(h)] = [11/27, 16/27]$.

It is also easy to see that $\inf_{\Pr} 4\beta(1-\beta)\Pr_\beta(t) = 8/27$, so $\inf_{\Pr \in \mathcal{P}_2} 4\beta(1-\beta)\Pr_\beta(t) < 1 - \mathcal{P}^+_{2,reg}(t) < \mathcal{P}^+_{2,reg}(h)$. Thus, for $\mathcal{P}^+_2$, we get strict inequalities for the expressions in Lemma 1. □

The width of the interval $[\underline{\mathcal{P}}^+_{reg}(E), \mathcal{P}^+_{reg}(E)]$ can be viewed as a measure of the ambiguity the agent feels about E, just as the interval $[\mathcal{P}_*(E), \mathcal{P}^*(E)]$. Indeed, if all the weights are 1, the two intervals have the same width, since $\mathcal{P}_*(E) = 1 - \mathcal{P}^+_{reg}(E)$ and $\mathcal{P}^*(E) = 1 - \underline{\mathcal{P}}^+_{reg}(E)$ in this case.

However, weighted regret has a significant advantage over upper and lower probability here. If the true bias of the coin is, say 5/8, then if the set $\mathcal{P}^+_k$ represents the uncertainty after k steps, as k increases, almost surely, $[\underline{\mathcal{P}}^+_{k,reg}(h), \mathcal{P}^+_{k,reg}(h)]$ will be a smaller and smaller interval containing $1 - 5/8 = 3/8$. More generally, using likelihood updated combined with weighted regret provides a natural way to model the reduction of ambiguity via learning.

One concern with the use of regret has been the dependence of regret on the menu. It is also worth noting that, in this context, there is a sense in which we can work with the absolute notion of weighted regret without loss of generality: if we restrict to indicator functions, then a preference relative to a menu can always be reduced to an absolute preference. Given a menu M consisting of indicator functions, let $E_M = \cup\{E : 1_E \in M\}$. that is, E_M is the union of the events for which the corresponding indicator function is in M.

Proposition 1. *If M is a menu consisting of indicator functions, and $1_{E_1}, 1_{E_2} \in M$, then $1_{E_1} \succeq^{reg}_{\mathcal{P}^+, M} 1_{E_2}$ iff $1_{E_1} + 1_{\overline{E}_M} \succeq^{reg}_{\mathcal{P}^+} 1_{E_2} + 1_{\overline{E}_M}$.*

[6] The proof of this result and all others can be found in the full paper, available at `http://www.cs.cornell.edu/home/halpern/papers/wregret.pdf`.

[7] The weight of $\Pr_\beta$ is the likelihood of observing heads according to $\Pr_\beta$, which is just β, normalized by the likelihood of observing heads according to the measure that gives heads the highest probability, namely 2/3.

4 Characterizing Weighted Regret Likelihood

The goal of this section is to characterize weighted regret likelihood axiomatically. In order to do so, it is helpful to review the characterizations of probability and lower probability.

A probability measure on a finite set S maps subsets of S to $[0,1]$ in a way that satisfies the following three properties:[8]

Pr1. $\Pr(S) = 1$.
Pr2. $\Pr(\emptyset) = 0$.[9]
Pr3. $\Pr(E \cup E') = \Pr(E) + \Pr(E')$ if $E \cap E' = \emptyset$.

These three properties characterize probability in the sense that any function $f : 2^S \to [0,1]$ that satisfies these properties is a probability measure.

Lower probabilities satisfy analogues of these properties:

LP1. $\mathcal{P}_*(S) = 1$.
LP2. $\mathcal{P}_*(\emptyset) = 0$.
LP3′. $\mathcal{P}_*(E \cup E') \geq \mathcal{P}_*(E) + \mathcal{P}_*(E')$ if $E \cap E' = \emptyset$.

However, these properties do not characterize lower probability. There are functions that satisfy LP1, LP2, and LP3′ that are not the lower probability corresponding to some set of probability measures. (See [9, Proposition 2.2] for an example showing that analogous properties do not characterize $\mathcal{P}^*$; the same example also shows that they do not characterize $\mathcal{P}_*$.)

Various characterizations of $\mathcal{P}_*$ (and $\mathcal{P}^*$) have been proposed in the literature [1,10,11,13,19,20], all similar in spirit. I discuss one due to Anger and Lembcke [1] here, since it makes the contrast between lower probability and regret particularly clear. The characterization is based on the notion of *set cover*: a set E is said to be covered n times by a multiset M of sets if every element of E appears at least n times in M. It is important to note here that M is a multiset, not a set; its elements are not necessarily distinct. (Of course, a set is a special case of a multiset.) Let $\sqcup$ denote multiset union; thus, if M_1 and M_2 are multisets, then $M_1 \sqcup M_2$ consists of all the elements in M_1 or M_2, which appear with multiplicity that is the sum of the multiplicities in M_1 and M_2. For example, using the $\{\{\ldots\}\}$ notation to denote a multiset, then $\{\{1,1,2\}\} \sqcup \{\{1,2,3\}\} = \{\{1,1,1,2,2,3\}\}$.

If $E \subseteq S$, then an (n,k)-*cover of* (E,S) is a multiset M that covers S k times and covers E $n+k$ times. Multiset M is an n-*cover* of E if M covers E n times. For example, if $S = \{1,2,3\}$, then $\{\{1,1,1,2,2,3\}\}$ is a $(2,1)$-cover of $(\{1\},S)$, a $(1,1)$-cover of $(\{1,2\},S)$, and a 3-cover of $\{1\}$. Consider the following property:

LP3. For all integers m, n, k and all subsets $E_1, \ldots, E_m$ of S, if $E_1 \sqcup \ldots \sqcup E_m$ is an (n,k)-cover of (E,S), then $k + n\mathcal{P}_*(E) \geq \sum_{i=1}^{m} \mathcal{P}_*(E_i)$.[10]

[8] Since I assume that S is finite here, I assume that all probability measures have domain 2^S, and ignore measurability issues.

[9] This property actually follows from the other two, using the observation that $\Pr(S \cup \emptyset) = \Pr(S) + \Pr(\emptyset)$; I include it here to ease the comparison to other approaches.

[10] Note that LP3 implies LP2, using the fact that $\emptyset \sqcup \emptyset$ is a (1,0)-cover of $(\emptyset, S)$.

There is an analogous property for upper probability, where $\geq$ is replaced by $\leq$. It is easy to see that LP3 implies LP3′ (since $E \sqcup E'$ is a $(1,0)$ cover of $E \cup F$).

Theorem 1. [1] *If $f : 2^S \to [0,1]$, then there exists a set $\mathcal{P}$ of probability measures with $f = \mathcal{P}_*$ if and only if f satisfies LP1, LP2, and LP3.*

Moving to regret-based likelihood, clearly we have

REG1. $\mathcal{P}^+_{reg}(S) = 0$.
REG2. $\mathcal{P}^+_{reg}(\emptyset) = 1$.

The whole space S has the least regret; the empty set has the greatest regret. In the unweighted case, since $\mathcal{P}_{reg}(E) = \mathcal{P}^*(\overline{E})$, REG1, REG2, and the following analogue of LP3 (appropriately modified for $\mathcal{P}^*$) clearly characterize $\mathcal{P}_{reg}$:

REG3′. For all integers m, n, k and all subsets $E_1, \ldots, E_m$ of S, if $\overline{E}_1 \sqcup \ldots \sqcup \overline{E}_m$ is an (n,k)-cover of $(\overline{E}, S)$, then $k + n\mathcal{P}_{reg}(E) \leq \sum_{i=1}^m \mathcal{P}_{reg}(E_i)$.

Note that complements of sets $(\overline{E}_1, \ldots, \overline{E}_m, \overline{E})$ are used here, since regret is minimized if the probability of the complement is maximized. This need to work with the complement makes the statement of the properties (and the proofs of the theorems) slightly less elegant, but seems necessary.

It is not hard to see that REG3′ does not hold for weighted regret-based likelihood. For example, suppose that $S = \{a, b, c\}$ and $\mathcal{P}^+ = ((\Pr_1, 2/3), (\Pr_2, 2/3), (\Pr_3, 1))$, where, identifying the probability Pr with the tuple $(\Pr(a), \Pr(b), \Pr(c))$, we have $\Pr_1 = (2/3, 0, 1/3)$, $\Pr_2 = (1/3, 0, 2/3)$, and $\Pr_3 = (1/3, 1/3, 1/3)$. Then $\mathcal{P}^+_{reg}(\{a,b\}) = \mathcal{P}^+_{reg}(\{b,c\}) = 4/9$, while $\mathcal{P}^+_{reg}(\{b\}) = 2/3$. Since $\{a,b\} \sqcup \{b,c\}$ is a (1,1)-cover of $(\{b\}, \{a,b,c\})$, REG3′ would require that

$$\mathcal{P}^+_{reg}(\{a,b\}) + \mathcal{P}^+_{reg}(\{b,c\}) \geq 1 + \mathcal{P}^+_{reg}(\{b\}),$$

which is clearly not the case.

We must thus weaken REG3′ to capture weighted regret-based likelihood. It turns out that the appropriate weakening is the following:

REG3. For all integers m, n and all subsets $E_1, \ldots, E_m$ of S, if $\overline{E}_1 \sqcup \ldots \sqcup \overline{E}_m$ is an n-cover of $\overline{E}$, then $n\mathcal{P}^+_{reg}(E) \leq \sum_{i=1}^m \mathcal{P}^+_{reg}(E_i)$.

Although REG3 is weaker than REG3′, it still has some nontrivial consequences. For example, it follows from REG3 that $\mathcal{P}^+_{reg}$ is anti-monotonic. If $E \subseteq E'$, then $\overline{E}$ is a 1-cover of $\overline{E}'$, so by REG3, we must have $\mathcal{P}^+_{reg}(E) \geq \mathcal{P}^+_{reg}(E')$. Since $E \sqcup E'$ is trivially a 1-cover of $E \cup E'$, it also follows that $\mathcal{P}^+_{reg}(\overline{E}) + \mathcal{P}^+_{reg}(\overline{E}') \geq \mathcal{P}^+_{reg}(\overline{E \cup E'})$. REG3 also implies REG1, since $\emptyset$ $(= \overline{S})$ is an n-cover of itself for all n.

I can now state the representation theorem. It says that a representation of uncertainty satisfies REG1, REG2, and REG3 iff it is the weighted regret likelihood determined by some set $\mathcal{P}^+$. The set $\mathcal{P}^+$ is not unique, but it can be taken to be *maximal*, in the sense that if weighted likelihood regret with respect to some other set $(\mathcal{P}')^+$ gives the same representation, then for all pairs $(\Pr, \alpha') \in (\mathcal{P}')^+$, there exists $\alpha \geq \alpha'$ such that $(\Pr, \alpha) \in \mathcal{P}^+$. This (unique) maximal set $\mathcal{P}^+$ can be viewed as the canonical representation of uncertainty.

Theorem 2. *If $f : 2^S \to [0,1]$, then there exists a weakly closed set $\mathcal{P}^+$ of weighted probability measures with $f = \mathcal{P}^+_{reg}$ if and only if f satisfies REG1, REG2, and REG3; moreover, $\mathcal{P}^+$ can be taken to be maximal.*

Acknowledgments. I thank Samantha Leung and the reviewers of ECSQARU for many useful comments on the paper.

References

1. Anger, B., Lembcke, J.: Infinitely subadditive capacities as upper envelopes of measures. Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete 68, 403–414 (1985)
2. Chateauneuf, A., Faro, J.: Ambiguity through confidence functions. Journal of Mathematical Economics 45, 535–558 (2009)
3. de Cooman, G.: A behavioral model for vague probability assessments. Fuzzy Sets and Systems 154(3), 305–358 (2005)
4. Epstein, L., Schneider, M.: Learning under ambiguity. Review of Economic Studies 74(4), 1275–1303 (2007)
5. Gilboa, I., Schmeidler, D.: Maxmin expected utility with a non-unique prior. Journal of Mathematical Economics 18, 141–153 (1989)
6. Good, I.J.: Some history of the hierarchical Bayesian methodology, pp. 489–504 (1980)
7. Halpern, J.Y.: Reasoning About Uncertainty. MIT Press, Cambridge (2003)
8. Halpern, J.Y., Leung, S.: Weighted sets of probabilities and minimax weighted expected regret: new approaches for representing uncertainty and making decisions. In: Proc. 29th Conf. on Uncertainty in Artificial Intelligence, pp. 336–345 (2012)
9. Halpern, J.Y., Pucella, R.: A logic for reasoning about upper probabilities. Journal of A.I. Research 17, 57–81 (2002)
10. Huber, P.J.: Kapazitäten statt Wahrscheinlichkeiten? Gedanken zur Grundlegung der Statistik. Jber. Deutsch. Math.-Verein 78, 81–92 (1976)
11. Huber, P.J.: Robust Statistics. Wiley, New York (1981)
12. Kyburg Jr., H.E.: Higher order probabilities and intervals. International Journal of Approximate Reasoning 2, 195–209 (1988)
13. Lorentz, G.G.: Multiply subadditive functions. Canadian Journal of Mathematics 4(4), 455–462 (1952)
14. Moral, S.: Calculating uncertainty intervals from conditional convex sets of probabilities. In: Proc. 8th Conf. on Uncertainty in Artificial Intelligence, pp. 199–206 (1992)
15. Niehans, J.: Zur preisbildung bei ungewissen erwartungen. Schweizerische Zeitschrift für Volkswirtschaft und Statistik 84(5), 433–456 (1948)
16. Pearl, J.: Do we need higher-order probabilities and, if so, what do they mean? In: Proc. 3rd Workshop on Uncertainty in Artificial Intelligence, pp. 47–60 (1987)
17. Savage, L.J.: The theory of statistical decision. Journal of the American Statistical Association 46, 55–67 (1951)
18. Walley, P.: Statistical inferences based on a second-ordr possibility distribution. International Journal of General Systems 26(4), 337–383 (1997)
19. Williams, P.M.: Indeterminate probabilities. In: Przelecki, M., Szaniawski, K., Wojcicki, R. (eds.) Formal Methods in the Methodology of Empirical Sciences, pp. 229–246, Reidel, Dordrecht (1976)
20. Wolf, G.: Obere und untere Wahrscheinlichkeiten. Ph.D. thesis, ETH, Zurich (1977)

Structural Properties for Deductive Argument Systems

Anthony Hunter[1] and Stefan Woltran[2]

[1] Department of Computer Science, University College London, Gower Street, London, WC1E 6BT, UK
[2] Institute of Information Systems 184/2, Technische Universität Wien, Favoritenstrasse 9-11, 1040 Vienna, Austria

Abstract. There have been a number of proposals for using deductive arguments for instantiating abstract argumentation. These take a set of formulae as a knowledgebase, and generate a graph where each node is a logical argument and each arc is a logical attack. This then raises the question of whether for a specific logical argument system S, and for any graph G, there is a knowledgebase such that S generates G. If it holds, then it can be described as a kind of "structural" property of the system. If it fails then, it means that there are situations that cannot be captured by the system. In this paper, we explore some features, and the significance, of such structural properties.

1 Introduction

Abstract argumentation provides a clear and precise approach to formalizing aspects of argumentation. However, in the approach, arguments are treated as atomic. If we want to understand individual arguments, we need to provide content for them. This leads to the idea of "instantiating" abstract argumentation with logical arguments, such as proposed by Cayrol [1].

There are various ways that logical arguments can be defined. A simple kind is a **deductive argument** which is a tuple $\langle \Phi, \alpha \rangle$ where Φ is a set of premises, and α is a claim such that for a consequence relation $\vdash_i$, $\Phi \vdash_i \alpha$ holds. Further constraints include consistency (i.e. $\Phi \nvdash_i \bot$) and minimality (there is no $\Psi \subset \Phi$ s.t. $\Psi \vdash_i \alpha$). Pollock was perhaps the first proponent of deductive arguments [2]. Subsequently, deductive arguments for classical logic [1, 3–5], description logic [6], and defeasible logic [7, 8] have been proposed.

In this paper, we consider how deductive argument systems generate constellations of arguments and counterarguments, and in particular, we are interested in the class of argument graphs they can induce. As we will see, some deductive argument systems only generate certain subclasses of graph. This is, however, not necessarily bad news. In fact, it is known that the computational complexity of evaluating argumentation frameworks (when considered as abstract frameworks following Dung [9]) can be decreased if the class of graphs is restricted, for instance to acyclic, bipartite or symmetric graphs or to graphs which have certain parameters (like treewidth) fixed (see e.g. [10–13]).

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 278–289, 2013.

2 Preliminaries

In this section we review some established definitions for graph theory, for logical languages and inference, and for logical argumentation.

A graph G is a tuple of the form (N, E) where N is a set of nodes and E is a set of edges. If a graph has no nodes and no edges, then it is the **empty graph**, denoted $G_\emptyset$. We consider various graph types including the following: A graph (N, E) is **weakly connected** iff for all nodes $n, n' \in N$ there is an undirected path (i.e. ignoring the direction of the arrows) in E from n to n'; A graph (N', E') is a **component** of a graph (N, E) iff (N', E') is the maximum subgraph of (N, E) that is weakly connected; A graph (N, E) is a **self-loop graph** iff there is a node $n \in N$ such that there is an edge $(n, n) \in E$; A graph (N, E) is a **rooted graph** iff (N, E) is acyclic and there is a node n in N, called the **root**, such that for all other nodes m in N, there is a path from m to n; A graph (N, E) is a **tree** iff (N, E) is a rooted graph and for each non-root node n in N, there is a unique path from n to the root; A graph (N, E) is **complete bipartite** iff there is a partition N_1 and N_2 of N such that $E = \{(n_1, n_2), (n_2, n_2) \mid n_1 \in N_1 \text{ and } n_2 \in N_2\}$. And a graph (N, E) is a **rational graph** iff (N, E) is a component and (N, E) is not a self-loop graph. The set of all graphs is denoted `Graphs`, with various subsets including the following: `Components` which is the set of all connected graphs; `Trees` which is the set of all trees; `AcyclicGraphs` which is the set of all acyclic graphs; `Bipartites` which is the set of all complete bipartite graphs; `RootedGraphs` which is the set of rooted graphs; and `RationalGraphs` which is the set of rational graphs.

In general, we use `Formulae` to denote the set of formulae of a language. In this paper, we focus on two languages. The **language of defeasible formulae**, denoted `DefFormulae`, is the set of literals and the set of rules of the form $\alpha_1 \wedge \cdots \wedge \alpha_n \to \beta$ where $\alpha_1, \ldots, \alpha_n$ are literals, and β is a literal. The **language of propositional formulae**, denoted `PropFormulae`, is the usual language for classical propositional logic that can formed from the logical connectives of $\vee$, $\wedge$, $\neg$ and $\to$. We consider the **classical consequence relation**, denoted $\vdash$, which is defined as usual, and the **defeasible consequence relation**, denoted $\vdash_d$, which is defined as follows: For $\Delta \subseteq$ `DefFormulae`, if $\alpha_1 \wedge \cdots \wedge \alpha_n \to \beta \in \Delta$, and for each $\alpha_i \in \{\alpha_1, \ldots, \alpha_n\}$, either $\alpha_i \in \Delta$ or $\Delta \vdash_d \alpha_i$, then $\Delta \vdash_d \beta$.

We consider two types of deductive argument. For $\Phi \subseteq$ `DefFormulae`, and a literal $\alpha \in$ `DefFormulae`, $\langle \Phi, \alpha \rangle$ is a **defeasible argument** iff $\Phi \vdash_d \alpha$ and there is no proper subset Φ' of Φ such that $\Phi' \vdash_d \alpha$. For $\Phi \subseteq$ `PropFormulae`, and a formula $\alpha \in$ `PropFormulae`, $\langle \Phi, \alpha \rangle$ is a **classical argument** iff $\Phi \vdash \alpha$, $\Phi \nvdash \bot$ and there is no proper subset Φ' of Φ such that $\Phi' \vdash \alpha$. For an argument $A = \langle \Phi, \alpha \rangle$, the function $\mathsf{Support}(A)$ returns Φ and the function $\mathsf{Claim}(A)$ returns α.

For defeasible arguments A and B, we consider the following type of **defeasible attack**: A is a **defeasible undercut** of B if (1) there is a rule $\alpha_1 \wedge \cdots \wedge \alpha_n \to \beta$ in $\mathsf{Support}(B)$ and $\mathsf{Claim}(A)$ is the complement of β (i.e. if $\mathsf{Claim}(A)$ is an atom ψ, then β is $\neg\psi$, and if β is an atom ψ, then $\mathsf{Claim}(A)$ is $\neg\psi$); Or (2) $\mathsf{Claim}(A)$ is the complement of a literal in $\mathsf{Support}(B)$. We have a wider range of options for defining attack for classical logic, such as rebuttals [2, 14], direct undercuts

[1, 15, 16], and canonical undercuts [3]. For classical arguments A and B, we consider the following type of **classical attack** in this paper: A is a **classical direct undercut** of B if $\exists \phi \in \mathsf{Support}(B)$ s.t. $\mathsf{Claim}(A) \equiv \neg\phi$; A is a **classical canonical undercut** of B if $\mathsf{Claim}(A) \equiv \neg \bigwedge \mathsf{Support}(B)$; And A is a **classical rebuttal** of B if $\mathsf{Claim}(A) \equiv \neg\mathsf{Claim}(B)$. We give some examples of logical attack in the following.

$\langle\{e, e \rightarrow \neg b\}, \neg b\rangle$ is a defeasible undercut of $\langle\{c, d, c \rightarrow b, b \wedge d \rightarrow a\}, a\rangle$
$\langle\{\neg a \wedge \neg b\}, \neg a\rangle$ is a classical direct undercut of $\langle\{a, b, c\}, a \wedge b \wedge c\rangle$
$\langle\{\neg a \wedge \neg b\}, \neg(a \wedge b \wedge c)\rangle$ is a classical canonical undercut of $\langle\{a, b, c\}, a \wedge b \wedge c\rangle$
$\langle\{a, a \rightarrow b\}, b \vee c\rangle$ is a classical rebuttal of $\langle\{\neg b, \neg c\}, \neg(b \vee c)\rangle$

3 Logical Argument Systems

In this paper, we consider a variety of logical argument systems based on deductive arguments using either defeasible logic or classical logic.

Definition 1. *A* **logical argument system** *is a tuple* $\langle \mathtt{Kbs}, \mathtt{Arg}, \mathtt{Att}, \mathtt{Con}\rangle$*, denoted* $\mathtt{Sys}$*, where for some language of formulae* $\mathtt{Formulae}$*, we have* $\mathtt{Kbs} = \wp(\mathtt{Formulae})$*,* $\mathtt{Arguments} = \{\langle \Phi, \psi\rangle \mid \Phi \in \mathtt{Kbs}$ *and* $\psi \in \mathtt{Formulae}\}$*, and* $\mathtt{Attacks} = \mathtt{Arguments} \times \mathtt{Arguments}$ *and*

$$\begin{aligned} &\mathtt{Arg} : \wp(\mathtt{Formulae}) \rightarrow \wp(\mathtt{Arguments})\\ &\mathtt{Att} : \wp(\mathtt{Formulae}) \rightarrow \wp(\mathtt{Attacks})\\ &\mathtt{Con} : \wp(\mathtt{Formulae}) \times \mathtt{Arguments} \rightarrow \mathtt{Graphs}\end{aligned}$$

This is a general definition that can be instantiated by a wide variety of logical argument systems. We give examples in Section 3.1. The sets $\mathtt{Arguments}$ and $\mathtt{Attacks}$ are given as types for the functions $\mathtt{Arg}$, $\mathtt{Att}$, and $\mathtt{Con}$. We explain the parameters $\langle \mathtt{Kbs}, \mathtt{Arg}, \mathtt{Att}, \mathtt{Con}\rangle$ of the definition as follows. The $\mathtt{Kbs}$ set is the set of knowledgebases that can be used by the system. In this paper, we focus on the knowledgebases given by the languages of defeasible formulae and classical formulae. The $\mathtt{Arg}$ function gives the set of arguments that can be generated from a knowledgebase. In this paper, we focus on defeasible arguments and classical arguments. The $\mathtt{Att}$ function gives the set of attacks that can be generated from a knowledgebase, and so $(A, B) \in \mathtt{Att}(\Delta)$ means that A attacks B (e.g. defeasible undercut or classical rebuttal). The $\mathtt{Con}$ function (called the **constructor function**) that, given a knowledgebase Δ and a specified argument A, called the **focal argument**, returns a graph s.t. if $A \in \mathtt{Arg}(\Delta)$, then the construction starts with A as a node in the graph and then builds the graph using a subset of the attacks relation (i.e. a subset of $\mathtt{Att}(\Delta)$) as the edges, and if $A \notin \mathtt{Arg}(\Delta)$, then the graph is the empty graph $G_\emptyset$.

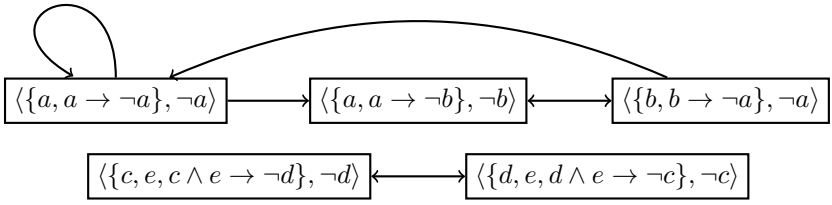
The constructor function encodes the method by which we generate an argument graph from a set of logical arguments and attacks between those arguments. In this paper, we consider the four constructor functions that we define and illustrate in the rest of this subsection.

Definition 2. *Let* $\langle \texttt{Kbs}, \texttt{Arg}, \texttt{Att}, \texttt{Con} \rangle$ *be a logical argument system.*

- $\texttt{Con}$ *is a* **trivial constructor** *iff for any knowledgebase* Δ*, and for any argument* $A \in \texttt{Arg}(\Delta)$*,* $\texttt{Con}(\Delta, A)$ *is the graph* $(\texttt{Arg}(\Delta), \texttt{Att}(\Delta))$*.*
- $\texttt{Con}$ *is a* **simple constructor** *iff for any knowledgebase* Δ*, and for any argument* $A \in \texttt{Arg}(\Delta)$*,* $\texttt{Con}(\Delta, A)$ *is the component in the graph* $(\texttt{Arg}(\Delta), \texttt{Att}(\Delta))$ *containing* A*.*

We can regard the simple constructor as starting with the focal argument A, and adding all the arguments that attack the argument (by adding the node and arc(s) for each of these counterarguments). Then it repeats this step iteratively until no more arguments can be added.

Example 1. For $\Delta = \{a, b, c, d, e, a \to \neg a, b \to \neg a, a \to \neg b, d \wedge e \to \neg c, c \wedge e \to \neg d\}$, let $\texttt{Arg}(\Delta) = \{ A_1, A_2, A_3, A_4, A_5 \}$, where A_1 is $\langle \{a, a \to \neg a\}, \neg a \rangle$, A_2 is $\langle \{a, a \to \neg b\}, \neg b \rangle$, A_3 is $\langle \{b, b \to \neg a\}, \neg a \rangle$, A_4 is $\langle \{c, e, c \wedge e \to \neg d\}, \neg d \rangle$, and A_5 is $\langle \{d, e, d \wedge e \to \neg c\}, \neg c \rangle$, and $\texttt{Att}(\Delta) = \{ (A_1, A_1), (A_1, A_2), (A_2, A_3), (A_3, A_1), (A_3, A_2), (A_4, A_5), (A_5, A_4) \}$. For this, the trivial constructor $\texttt{Con}(\Delta, A)$ returns the following graph, where any of A_1 to A_5 is the focal argument A. Furthermore, the simple constructor $\texttt{Con}(\Delta, A)$ returns the component below containing A_1 to A_3, where any of A_1 to A_3 is the focal argument A. Likewise, if the focal argument A would be A_4 or A_5, the simple constructor would return the other component.



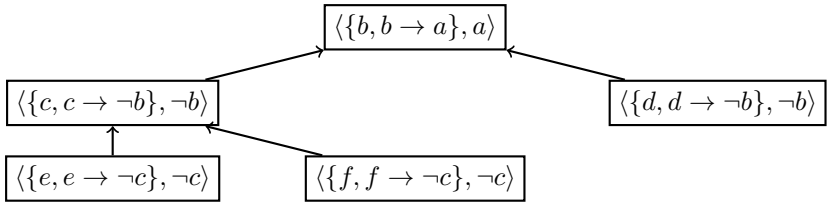
The recursive constructor (defined next) is related to proposals for constructing trees in classical logic [3] and in defeasible logic programming [7]. For the following definition, the constructor starts with the focal argument as the root, and all attackers are added as children. Then by recursion, for each argument in the graph, all the attackers of the argument are added as children. The only restriction to this is the so called "no recycle" condition, which says that when adding an attacker to the graph, it should contain at least one formula in the support that has not been used as a premise in any ancestor argument (i.e. an argument that is on the branch to the root). Consequently, the recursive constructor always yields a (directed) acyclic graph.

Definition 3. *Let* $\langle \texttt{Kbs}, \texttt{Arg}, \texttt{Att}, \texttt{Con} \rangle$ *be a logical argument system. Let* Δ *be a knowledgebase, and let* A *be an argument.* $\texttt{Con}$ *is a* **recursive constructor** *iff for any knowledgebase* Δ*, and for any argument* $A \in \texttt{Arg}(\Delta)$*,* $\texttt{Con}(\Delta, A)$ *is the directed graph* G *constructed by adding exactly the arguments as follows:*

1. *A is the root of G*

2. *if* $(B, A) \in \texttt{Att}$, *then* B *is a child of* A *in* G
3. *by recursion, for each node* C *in* G, *if* $(D, C) \in \texttt{Att}$, *and the support of* D *contains at least one premise that does not appear in the support of any argument on the branch from* C *to* A, *then* D *is a child of* C *in* G.

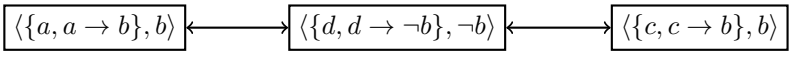
Example 2. For $\Delta = \{b, b \rightarrow a, c, c \rightarrow \neg b, d, d \rightarrow \neg b, e, e \rightarrow \neg c, f, f \rightarrow \neg c\}$, let $(\texttt{Arg}(\Delta) = \{A_1, A_2, A_3, A_4, A_5\}$, where A_1 is $\langle\{b, b \rightarrow a\}, a\rangle$, A_2 is $\langle\{c, c \rightarrow \neg b\}, \neg b\rangle$, A_3 is $\langle\{d, d \rightarrow \neg b\}, \neg b\rangle$, A_4 is $\langle\{e, e \rightarrow \neg c\}, \neg c\rangle$, A_5 is $\langle\{f, f \rightarrow \neg c\}, \neg c\rangle$, A_6 is $\langle\{b, c \rightarrow \neg b\}, \neg c\rangle$, A_7 is $\langle\{c, e \rightarrow \neg c\}, \neg e\rangle$, A_8 is $\langle\{c, f \rightarrow \neg c\}, \neg f\rangle$, and A_9 is $\langle\{b, d \rightarrow \neg b\}, \neg d\rangle$, and $\texttt{Att}(\Delta)) = \{$ (A_2, A_1), (A_2, A_6), (A_3, A_1), (A_3, A_9), (A_4, A_2), (A_4, A_7), (A_4, A_8), (A_5, A_2), (A_5, A_8), (A_6, A_2), (A_6, A_7), (A_6, A_8), (A_7, A_4), (A_8, A_5), (A_9, A_3) $\}$. For this, the recursive constructor returns the following graph containing arguments $A_1 \ldots A_5$, where A_1 is the focal argument. Note that arguments $A_6 \ldots A_9$ do not appear in the graph, due to the third condition of Definition 3.



The rebuttal constructor (defined next) is similar to proposals for reasoning with pros and cons. Given the focal argument, all arguments with a logically equivalent claim or with a contradictory claim are included. The rebuttal constructor always yields a complete bipartite graph.

Definition 4. *Let* $\langle \texttt{Kbs}, \texttt{Arg}, \texttt{Att}, \texttt{Con}\rangle$ *be a logical argument system. Let* Δ *be a knowledgebase, and let* A *be an argument.* $\texttt{Con}$ *is a* **rebuttal constructor** *iff for any knowledgebase* Δ, *and for any argument* $A \in \texttt{Arg}(\Delta)$, $\texttt{Con}(\Delta, A)$ *is the graph obtained by taking the nodes to be the arguments that either have a claim that is logically equivalent to* $\mathsf{Claim}(A)$ *or a claim that is logically equivalent to* $\neg\mathsf{Claim}(A)$ *and by taking the edges to be the rebuttals between these nodes.*

Example 3. For $\Delta = \{a, a \rightarrow b, c, c \rightarrow b, d, d \rightarrow \neg b, d, d \rightarrow \neg c, c, c \rightarrow \neg d\}$, let $(\texttt{Arg}(\Delta) = \{$ A_1, A_2, A_3, A_4, A_5 $\}$, where A_1 is $\langle\{a, a \rightarrow b\}, b\rangle$, A_2 is $\langle\{c, c \rightarrow b\}, b\rangle$, A_3 is $\langle\{d, d \rightarrow \neg b\}, \neg b\rangle$, A_4 is $\langle\{d, d \rightarrow \neg c\}, \neg c\rangle$, and A_5 is $\langle\{c, c \rightarrow \neg d\}, \neg d\rangle$, and $\texttt{Att}(\Delta)) = \{$ (A_1, A_3), (A_2, A_3), (A_3, A_1), (A_3, A_2) $\}$. For this, the rebuttal constructor returns the following graph, where any of A_1 to A_3 is the focal argument.



In general, we do not impose constraints on a logical argument system. In this paper, we only consider instances of $\texttt{Arg}$, $\texttt{Att}$, and $\texttt{Con}$ that are monotonic. However, it would be reasonable to consider non-monotonic versions of the functions, but we leave that to future work.

3.1 Examples of Logical Argument Systems

To illustrate the idea of logical argument systems, we present some instances, denoted System 1 to System 5, next, and then in the following section, we will consider properties of these systems.

System 1. *The tuple* $\langle \texttt{Kbs}, \texttt{Arg}, \texttt{Att}, \texttt{Con} \rangle$ *is a system based on defeasible logic where* Kbs *is* $\wp(\texttt{DefFormulae})$, $\texttt{Arg}(\Delta)$ *is the set of defeasible arguments from* Δ *such that if* $B \in \texttt{Arg}(\Delta)$, *then* $\texttt{Support}(B) \subseteq \Delta$, $\texttt{Att}(\Delta)$ *is* $\{(B, C) \mid B, C \in \texttt{Arg}(\Delta)$ *and* B *is a defeasible undercut of* $C\}$, *and* $\texttt{Con}(\Delta, A)$ *is the simple constructor.*

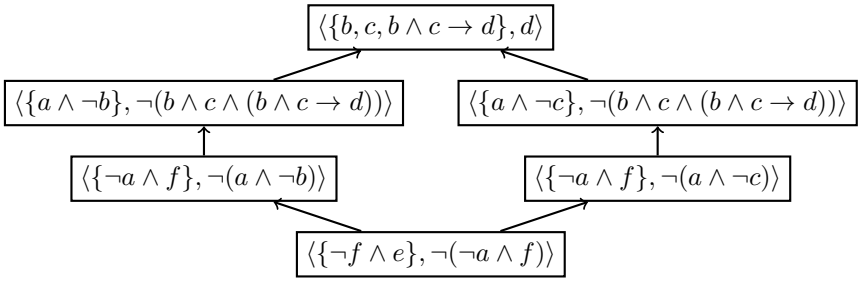
Example 4. Consider System 1 with $\Delta = \{a, b, a \rightarrow \neg a, b \rightarrow \neg a, a \rightarrow \neg b, d \rightarrow \neg c, c \rightarrow \neg d\}$. For the focal argument $\langle \{a, a \rightarrow \neg a\}, \neg a \rangle$, $\texttt{Con}(\Delta, A)$ gives the constructed graph that is the component with three arguments in Example 1.

From the directed graph obtained by the next system, it is simple to obtain the argument tree of Besnard and Hunter[1] [3].

System 2. *The tuple* $\langle \texttt{Kbs}, \texttt{Arg}, \texttt{Att}, \texttt{Con} \rangle$ *is a system based on classical logic where* Kbs *is* $\wp(\texttt{PropFormulae})$, $\texttt{Arg}(\Delta)$ *is the set of classical arguments such that if* $B \in \texttt{Arg}(\Delta)$, *then* $\texttt{Support}(B) \subseteq \Delta$, $\texttt{Att}(\Delta)$ *is* $\{(B, C) \mid B, C \in \texttt{Arg}(\Delta)$ *and* B *is a canonical undercut of* $C\}$, *and* $\texttt{Con}(\Delta, A)$ *is the recursive constructor.*

So the constructor function returns the smallest graph obtained by starting with A, adding all the canonical undercuts to A, and by recursion adding all the canonical undercuts A_n to each of the canonical undercuts A_{n-1} subject to the condition that each canonical undercut A_n has a premise that does not appear in any support on the path of arguments A_{n-1}, ..., A_1 where A_1 is A.

Example 5. Consider System 2 with $\Delta = \{b, c, a \wedge \neg b, a \wedge \neg c, \neg a \wedge f, \neg f \wedge e, b \wedge c \rightarrow d\}$. For the focal argument $\langle \{b, c, b \wedge c \rightarrow d\}, d \rangle$, $\texttt{Con}(\Delta, A)$ gives the following constructed graph which is an example of a rooted graph.



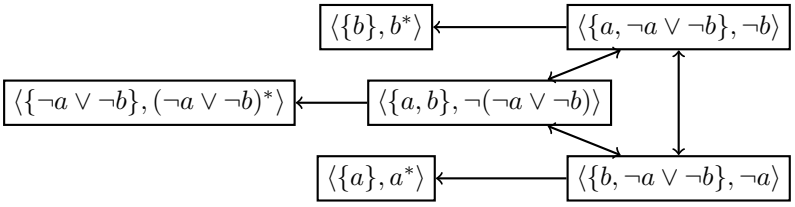
[1] A rooted graph is translated to an argument tree of Besnard and Hunter as follows: Start from the bottom of the graph working upwards. For each node with multiple parents, a copy is made of the node and its offspring for each of its parent, so that each copy has exactly one parent. For Example 5, the bottom node is copied so the argument occurs in two leaf nodes

The next system is based on the idea of exhaustively generating all arguments from a knowledgebase and all attacks (according to a particular definition of attack) and using the resulting graph without restriction (as first proposed in [1], and further explored in [5]).

System 3. *The tuple* $\langle \mathtt{Kbs}, \mathtt{Arg}, \mathtt{Att}, \mathtt{Con} \rangle$ *is a system based on classical logic where* $\mathtt{Kbs}$ *is* $\wp(\mathtt{PropFormulae})$, $\mathtt{Arg}(\Delta)$ *is the set of classical arguments such that if* $B \in \mathtt{Arg}(\Delta)$, *then* $\mathsf{Support}(B) \subseteq \Delta$, $\mathtt{Att}(\Delta)$ *is* $\{(B, C) \mid B, C \in \mathtt{Arg}(\Delta)$ *and* B *is a direct undercut of* $C\}$, *and* $\mathtt{Con}(\Delta, A)$ *is the simple constructor.*

Note, here we use the classical direct undercut. But, we could use the classical defeater, classical direct defeater, classical undercut, classical canonical undercut, or classical literal undercut, as an alternative (see definitions in [5]). Hence, we have a range of systems based on the choice of attack.

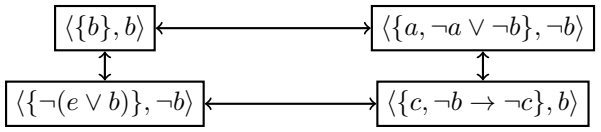
Example 6. Consider System 3 with $\Delta = \{a, \neg a \vee \neg b, b\}$. For the focal argument $A = \langle \{b\}, b \rangle$, $\mathtt{Con}(\Delta, A)$ gives the following constructed graph. For this, each argument with a claim with an asterisk, i.e. a claim of the form α^*, denotes any argument with the same premises and a claim that it is implied by α.



The following system is the same as the previous system but restricts consideration to classical rebuttals rather than direct undercuts [5]. As an alternative we could consider classical direct defeating rebuttals (see definition in [3]).

System 4. *The tuple* $\langle \mathtt{Kbs}, \mathtt{Arg}, \mathtt{Att}, \mathtt{Con} \rangle$ *is a system based on classical logic where* $\mathtt{Kbs}$ *is* $\wp(\mathtt{PropFormulae})$, $\mathtt{Arg}(\Delta)$ *is the set of classical arguments such that if* $A \in \mathtt{Arg}(\Delta)$, *then* $\mathsf{Support}(B) \subseteq \Delta$, $\mathtt{Att}(\Delta)$ *is* $\{(B, C) \mid B, C \in \mathtt{Arg}(\Delta)$ *and* B *is a rebuttal of* $C\}$, *and* $\mathtt{Con}(\Delta, A)$ *is the rebuttal constructor.*

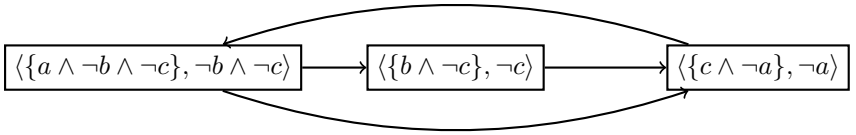
Example 7. Consider System 4 with $\Delta = \{a, \neg a \vee \neg b, b, c, \neg b \rightarrow \neg c, \neg(e \vee b)\}$. For argument $A = \langle \{b\}, b \rangle$, $\mathtt{Con}(\Delta, A)$ gives the following constructed graph.



Finally, we give an example of a new logical argument system that is very constrained with respect to the kinds of arguments that are allowed. Essentially, this system allows us to avoid the symmetrical relationships that usually hold for attack for a classical argument system.

System 5. *The tuple* $\langle \texttt{Kbs}, \texttt{Arg}, \texttt{Att}, \texttt{Con} \rangle$ *is a system based on classical logic where* $\texttt{ConForm} = \{ \alpha \wedge \beta_1 \wedge ... \wedge \beta_n \mid \alpha$ *is a positive literal and* $\beta_1, ..., \beta_n$ *are negative literals* $\}$, $\texttt{Kbs}$ *is* $\wp(\texttt{ConForm})$, $\texttt{Arg}(\Delta)$ *is the set of tuples of the form* $\langle \{\phi\}, \psi \rangle$ *where* $\phi \in \Delta$ *and* $\{\phi\} \vdash \psi$ *and* ψ *is the conjunction of negative literals occurring in* ϕ, $\texttt{Att}(\Delta)$ *is* $\{(B, C) \mid B, C \in \texttt{Arg}(\Delta)$ *and* B *is a classical undercut of* $C\}$, *and* $\texttt{Con}(\Delta, A)$ *is the simple constructor.*

Example 8. Consider System 5 with $\Delta = \{a \wedge \neg b \wedge \neg c, b \wedge \neg c, c \wedge \neg a\}$. For the focal argument $A = \langle \{a \wedge \neg b\}, \neg b \rangle$, $\texttt{Con}(\Delta, A)$ gives the following graph.



In this section, we have presented a non-exhaustive range of logical argument systems. Most are based on well-known approaches. The last system is a new proposal for studying structural properties rather than being useful in its own right.

4 Induced Graphs

The following definition captures the relationships that we will consider between a logical argument system and a class of graphs. The more general the class of graphs that a logical argument system can cover, the wider the range of argumentation situations the logical argument systems can capture.

Definition 5. *Let* $\texttt{Sys} = \langle \texttt{Kbs}, \texttt{Arg}, \texttt{Att}, \texttt{Con} \rangle$ *be a logical argument system and let* X *be a graph type.*

- $\texttt{Sys}$ **constructively covers** X *iff for all* $G \in X$, *there is a* $\Delta \in \texttt{Kbs}$, *and there is an* $A \in \texttt{Arg}(\Delta)$, *such that* $\texttt{Con}(\Delta, A) = G$.
- $\texttt{Sys}$ *is* **constructively covered** *by* X *iff for all* $\Delta \in \texttt{Kbs}$, *and for all* $A \in \texttt{Arguments}$, *if* $\texttt{Con}(\Delta, A) = G$, *then* $G \in X$.
- $\texttt{Sys}$ *is* **constructively complete** *for* X *iff* $\texttt{Sys}$ *constructively covers* X *and* $\texttt{Sys}$ *is constructively covered by* X.

Since the constructor function returns a graph, by definition, any logical argument system is constructively covered by $\texttt{Graphs}$. We now consider in more detail the systems from the previous section starting with System 1. Note, if we use the trivial constructor instead of the simple constructor, it is straightforward to show it is constructively complete for $\texttt{Graphs}$.

Proposition 1. *System 1 is constructively complete for* $\texttt{Components}$.

Turning to System 2, Example 5 illustrates that it is not constructively complete for $\texttt{Trees}$.

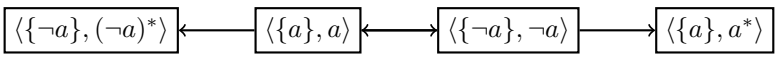
Proposition 2. *System 2 is constructively complete for* `RootedGraphs`.

System 3 does not correspond to any of the classes of graphs presented earlier. In particular, it does not constructively cover `RootedGraphs`, `AcyclicGraphs`, `RationalGraphs`, `Bipartites`, or `Components`. Furthermore, it is not covered by `RootedGraphs`, `AcyclicGraphs`, and `Bipartites`. However, we do show next that as System 3 does not allow inconsistent premises, it excludes self-cycles, and so it is covered by `RationalGraphs`.

Proposition 3. *System 3 is constructively covered by* `RationalGraphs`.

To illustrate the difficulty in identifying a tighter bound on the set of graphs that System 3 is covered by, we consider the problem of constructing a component with two arguments attacking each other. We indicate by the following example that this is not possible. Note, this is not a pathological example as there are many simple graphs that cannot be generated by System 3.

Example 9. For System 3, let $\Delta = \{a, \neg a\}$. Hence, there are two classical arguments $\langle\{a\}, a\rangle$ and $\langle\{\neg a\}, \neg a\rangle$ that are direct undercuts of each other. Plus, there are two further kinds of argument, $\langle\{a\}, a^*\rangle$ and $\langle\{\neg a\}, (\neg a)^*\rangle$, where a^* is strictly weaker than a (i.e. $\{a\} \vdash a^*$ and $\{a^*\} \nvdash a$). and $(\neg a)^*$ is strictly weaker than $\neg a$.



For System 4, the definition of the rebuttal constructor renders it straightforward to show that the system is constructively complete for `Bipartites`.

Proposition 4. *System 4 is constructively complete for* `Bipartites`.

The restrictions on the form of the arguments arising in System 5 allow us to show that even with classical logic, we can get almost the same completeness results as with defeasible logic. What are missing are the self-loop components.

Proposition 5. *System 5 is constructively complete for* `RationalGraphs`.

In this section, we have shown how restricted systems such as those based on defeasible logic (e.g. System 1), or those based on very restricted arguments (e.g. System 5) are constructively complete for rational graphs, or even components, whereas unrestricted use of classical logic means these properties do not hold.

5 Local and Global Constructors

To get completeness results for components, graphs, or rational graphs, the logical argument system needs to be restricted in some way. For example, the proof theory of System 1, for generating arguments is weak (it is modus ponens) and for System 5, the arguments are restricted to having a single premise and the

claim being a conjunction of negative literals. From the systems we have considered so far, we see a trade-off with regard to how restricted the system is and the completeness results that hold for it. We investigate this issue in this section by classifying constructors. For this we need the subsidiary notion of a characteristic function Test which is a function from sets of attacks to {"*yes*", "*no*"}.

Definition 6. *A constructor function* Con *is* **local** *iff there is a characteristic function* Test *s.t. for all* Δ, A, *if* $\texttt{Con}(\Delta, A) = (N, E)$, *and* (N, E) *is a component, and* $B_i \in N$, *and* $(B_j, B_i) \in \texttt{Att}(\Delta)$, *and* $\texttt{Test}((B_j, B_i)) =$ "*yes*", *then* $(B_j, B_i) \in E$ *and* $B_j \in N$. *A constructor function* Con *is* **global** *iff* Con *is not local.*

A local constructor function thus constructs a component by adding nodes and arcs incrementally starting with A. The local constructor makes a local decision on whether to add a node or arc based on the nature of the attack. It does not take into account any other aspect of the graph. In other words, no global view is taken into account when constructing the graph.

Proposition 6. *The trivial constructor function, simple constructor function and the rebuttal constructor function are local, whereas the recursive constructor function is global.*

The following results show that unless a system is highly restricted, it is not possible to generate every graph with a local constructor function. In order to directly compare defeasible and classical logics, we have used a restricted version of the defeasible logic system considered earlier.

Theorem 1. *Let* $\texttt{Sys} = \langle \texttt{Kbs}, \texttt{Arg}, \texttt{Att}, \texttt{Con} \rangle$ *be a logical argument system such that* Kbs *is the set of defeasible knowledgebases,* Arg *is the set of non-self attacking defeasible arguments from* Δ *(i.e. for each argument* $A \in \texttt{Arg}(\Delta)$, A *does not attack* A*), and* Att *is defeasible undercut. There is a constructor* Con *such that* Con *is local and* Sys *is constructively complete for* RationalGraphs.

Theorem 2. *Let* $\texttt{Sys} = \langle \texttt{Kbs}, \texttt{Arg}, \texttt{Att}, \texttt{Con} \rangle$ *be a logical argument system such that* Kbs *is the set of classical knowledgebases,* Arg *is the set of classical arguments from* Δ, *and* Att *is classical defeater, classical direct defeater, classical undercut, classical canonical undercut, or classical direct undercut. If* Sys *is constructively complete for* RationalGraphs, *then* Con *is global.*

The main ramification of the above result is that if we want to use richer logics such as classical logic, then we need to use global constructors. In other words, to reflect any abstract argument graph in a logical argument system based on a richer logic, we need to be selective in the choice of arguments taken from $\texttt{Arg}(\Delta)$ and the choice of attacks taken from $\texttt{Att}(\Delta)$ for any given Δ and A. Therefore, these results in a sense justify the need to better understand the notion of global constructors.

Furthermore, this is not just for theoretical interest. Practical argumentation often seems to use richer logics such as classical logic, and often the presentation of arguments and counterarguments is not exhaustive. Therefore, we

need to better understand how the arguments presented are selected. For example, suppose agent 1 posits $A_1 = \langle \{b, b \rightarrow a\}, a \rangle$, and agent 2 then posits $A_2 = \langle \{c, c \rightarrow \neg b\}, \neg b \rangle$. It would be reasonable for this dialogue to stop at this point even though there are further arguments that can be constructed from the public knowledge such as $A_3 = \langle \{b, c \rightarrow \neg b\}, \neg c \rangle$. So in terms of constructing the constellation of arguments and counterarguments from the knowledge, we need to know what the underlying principle is for ascertaining that just the two arguments are sufficient given the public knowledge, and that this means we need to know more about the global constructor function. It may also mean that we need to better understand how meta-knowledge (about the premises and/or about the participants) is used to select arguments and counterarguments.

6 Discussion

In this paper we have provided: (1) A general framework for describing diverse logical argument systems; (2) A classification scheme for logical argument systems in terms of the class of graphs that they induce; (3) An analysis of local and global methods of constructing argument graphs from a knowledgebase which has ramifications for using richer logics in argumentation.

There are further options that we may consider for logical argument systems by for instance changing the definition of attack or changing the choice of base logic: (i) defeasible logic with annotations for truth values (such as for Belnap's four-valued logic) [17] and for possibility theory [18], (ii) temporal reasoning calculi [19, 20], (iii) minimal logic [21], and (iv) modal logic [22]. Indeed, any logic could be potentially used as a base logic [8].

Whilst, the focus of the paper has been on deductive arguments, the issues raised may also have ramifications for further argumentation systems such as ASPIC+ [23] and ABA [24]. We leave investigation of this to future work.

Acknowledgments. This research has partially been supported by the Austrian Science Fund (FWF) through project I1102.

References

1. Cayrol, C.: On the relation between argumentation and non-monotonic coherence-based entailment. In: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI 1995), pp. 1443–1448 (1995)
2. Pollock, J.: Defeasible reasoning. Cognitive Science 11(4), 481–518 (1987)
3. Besnard, P., Hunter, A.: A logic-based theory of deductive arguments. Artif. Intell. 128, 203–235 (2001)
4. Amgoud, L., Cayrol, C.: A reasoning model based on the production of acceptable arguments. Annals of Mathematics and Artificial Intelligence 34, 197–215 (2002)
5. Gorogiannis, N., Hunter, A.: Instantiating abstract argumentation with classical logic arguments: Postulates and properties. Artif. Intell. 175(9-10), 1479–1497 (2011)

6. Black, E., Hunter, A., Pan, J.Z.: An argument-based approach to using multiple ontologies. In: Godo, L., Pugliese, A. (eds.) SUM 2009. LNCS, vol. 5785, pp. 68–79. Springer, Heidelberg (2009)
7. García, A., Simari, G.: Defeasible logic programming: An argumentative approach. Theory and Practice of Logic Programming 4, 95–138 (2004)
8. Hunter, A.: Base logics in argumentation. In: Proceedings of the 3rd Conference on Computational Models of Argument (COMMA 2010). Frontiers in Artificial Intelligence and Applications, vol. 216, pp. 275–286. IOS Press (2010)
9. Dung, P.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and n-person games. Artif. Intell. 77, 321–357 (1995)
10. Dunne, P.: Computational properties of argument systems satisfying graph-theoretic constraints. Artif. Intell. 171(10-15), 701–729 (2007)
11. Coste-Marquis, S., Devred, C., Marquis, P.: Symmetric argumentation frameworks. In: Godo, L. (ed.) ECSQARU 2005. LNCS (LNAI), vol. 3571, pp. 317–328. Springer, Heidelberg (2005)
12. Dvořák, W., Szeider, S., Woltran, S.: Abstract argumentation via monadic second order logic. In: Hüllermeier, E., Link, S., Fober, T., Seeger, B. (eds.) SUM 2012. LNCS, vol. 7520, pp. 85–98. Springer, Heidelberg (2012)
13. Dvorák, W., Pichler, R., Woltran, S.: Towards fixed-parameter tractable algorithms for abstract argumentation. Artif. Intell. 186, 1–37 (2012)
14. Pollock, J.: How to reason defeasibly. Artif. Intell. 57(1), 1–42 (1992)
15. Elvang-Gøransson, M., Krause, P., Fox, J.: Acceptability of arguments as 'logical uncertainty'. In: Moral, S., Kruse, R., Clarke, E. (eds.) ECSQARU 1993. LNCS, vol. 747, pp. 85–90. Springer, Heidelberg (1993)
16. Elvang-Gøransson, M., Hunter, A.: Argumentative logics: Reasoning with classically inconsistent information. Data & Knowledge Engineering 16(2), 125–145 (1995)
17. Takahashi, T., Sawamura, H.: A logic of multiple-valued argumentation. In: Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004), pp. 800–807. IEEE Computer Society (2004)
18. Alsinet, T., Chesñevar, C., Godo, L., Simari, G.: A logic programming framework for possibilistic argumentation: Formalization and logical properties. Fuzzy Sets and Systems 159(10), 1208–1228 (2008)
19. Augusto, J., Simari, G.: Temporal defeasible reasoning. Knowledge and Information Systems 3(3), 287–318 (2001)
20. Mann, N., Hunter, A.: Argumentation using temporal knowledge. In: Proceedings of the 2nd Conference on Computational Models of Argument (COMMA 2008). Frontiers in Artificial Intelligence and Applications, vol. 172, pp. 204–215. IOS Press (2008)
21. Krause, P., Ambler, S., Elvang-Gøransson, M., Fox, J.: A logic of argumentation for reasoning under uncertainty. Computational Intelligence 11, 113–131 (1995)
22. Fox, J., Das, S.: Safe and Sound: Artificial Intelligence in Hazardous Applications. MIT Press (2000)
23. Prakken, H.: An abstract framework for argumentation with structured arguments. Argument and Computation 1, 93–124 (2010)
24. Dung, P., Kowalski, R., Toni, F.: Dialectical proof procedures for assumption-based admissible argumentation. Artificial Intelligence 170, 114–159 (2006)

Measuring Inconsistency through Minimal Proofs

Said Jabbour and Badran Raddaoui

CRIL - Univ. Artois
CNRS UMR 8188, 62307 Lens Cedex, France
{jabbour,raddaoui}@cril.fr

Abstract. Measuring the degree of inconsistency of a knowledge base provides important context information for making easier inconsistency handling. In this paper, we propose a new fine-grained measure to quantify the degree of inconsistency of propositional formulae. Our inconsistency measure uses in an original way the minimal proofs to characterize the responsibility of each formula in the global inconsistency. We give an extension of such measure to quantify the inconsistency of the whole base. Furthermore, we show that our measure satisfies the important properties characterizing an intuitive inconsistency measure. Finally, we address the problem of restoring consistency using an inconsistency measure.

Introduction

Inconsistencies are well-known and essential concept in several research areas. They occur when working with logic knowledge bases; for example, when revising or merging several bases due to not fully reliable sources. Measuring such inconsistencies have received a growing interest recently because it has been shown to be very helpful in various fields including e-commerce protocols [1], software specifications [2], belief merging [3], news reports [4], requirements engineering [5], integrity constraints [6], databases [7], ontologies [8], semantic web [8], and network intrusion detection [9]. In Artificial Intelligence, and particularly in knowledge representation and reasoning, there have been a growing interest on how to design relevant measures to quantify the inconsistency of a given knowledge base. Such analysis have been shown to be very helpful for deciding on how to act on such inconsistency [4]. In other words, such analysis might help to decide if such inconsistency needs to be resolved or simply ignored.

In order to analyze and to measure the amount of inconsistency of a knowledge, several logic-based approaches have been proposed. Among them, there are the maximal η-consistency measures based on variables [10] or via multi-valued models [11–13, 4, 14–17], the η-consistency and η-probability measures [18], the measures based on minimal inconsistent subsets [19–22], and the Shapley inconsistency value proposed in [5]. A comparative study of all these measures is clearly a challenging task. However, they can be roughly divided into two main categories. The first one computes the proportion of the formulae affected by

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 290–301, 2013.

the inconsistency. In this category, most of the inconsistency measures are often based on some paraconsistent semantics. The second category involves syntactic measures based on the minimal inconsistent subsets of formulae. It considers minimal inconsistent subsets of formulae as the best and relevant reason of inconsistency. However, to derive relevant inconsistency handling methods, one needs to first identify the inconsistency or the inconsistent parts of the knowledge base before restoring the consistency. Limiting the first step to a simple consistency checking is not sufficient, and doesn't tell us much on the different inconsistencies and their interactions.

In this paper, we deal with two kind of measures. The first one commonly called "degree of inconsistency" aims to evaluate the contribution of each formula in the inconsistency of the knowledge base. The second measure, designed by "inconsistency measure", allows to evaluate the inconsistency of the whole base. We propose a new fine-grained measure to quantify the inconsistency of propositional knowledge base. Our measure is based on the so called "minimal proofs". We show its relationship with the minimal inconsistent subsets. Our first characterization allows to give the contribution/responsibility of each formula of the knowledge base in the inconsistency and we extend it to compute the inconsistency of the whole base. Finally, we address the second important step of inconsistency handling, and then we show how inconsistency measures can be used to restore the consistency of the knowledge base.

The paper is organized as follows: after stating some preliminary definitions and notations, we briefly review different approaches to measuring the degree of inconsistency in the literature based on minimal inconsistent subsets. In the second section, we study how to use minimal proofs in order to define inconsistency values. In section three, we discuss some logical properties satisfied by our measure. Then, we address the problem of restoring consistency in an inconsistent knowledge base using inconsistency measures. In the last section we conclude and give some perspectives for future works.

1 Preliminaries

1.1 Propositional Logic and Satisfiability

In this paper, we consider the propositional fragment of classical logic. Let $\mathcal{L}$ be a propositional language built from a finite set of propositional symbols P under logical connectives $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$. We will use $a, b, c, \ldots$ to denote propositional variables. A *literal* is a positive (p) or negative ($\neg p$) propositional variable and a *clause* is a disjunction of literals. A formula is defined recursively under the set of propositional symbols and literals. We use greek letters $\alpha, \beta, \gamma, \ldots$ to denote propositional formulae. We denote by $Var(\alpha)$ the set of propositional variables occurring in α and by $Lit(\alpha) = \{x, \neg x \mid x \in Var(\alpha)\}$ the set of literals occurring in α. A *propositional knowledge base* K is a finite set of propositional formulae. We denote by $|K|$ the cardinality of the knowledge base K. A knowledge base K is inconsistent if there is a formula α such that $K \vdash \alpha$ and $K \vdash \neg\alpha$,

where $\vdash$ is the consequence relation in propositional logic. We write $K \vdash \bot$ to denote that K is inconsistent.

Let us firstly introduce the notion of minimal inconsistent subsets of a given knowledge base K. These subsets can be considered as the most relevant way to characterize syntactically the inconsistency.

Definition 1 (MUS). *Let K be a knowledge base. Then, $\mathcal{M}$ is a minimal unsatisfiable (inconsistent) subset (MUS) of K iff:*

1. $\mathcal{M} \subseteq K$
2. $\mathcal{M} \vdash \bot$
3. $\forall \mathcal{M}' \subset \mathcal{M}, \mathcal{M}' \nvdash \bot$

If a knowledge base is inconsistent, then it contains at least one MUS. We denote by $MUSes(K)$ the set of all minimal inconsistent subsets of K, i.e. $MUSes(K) = \{\mathcal{M} \mid \mathcal{M}\ is\ a\ MUS\ of\ K\}$.

We call a formula α of K a *free formula* of K if α doesn't belongs to any minimal inconsistent subset of K. That is, α is not concerned with the minimal inconsistent subsets of K.

1.2 Inconsistency Measures

In this section, we describe some inconsistency measures based on minimal inconsistent subsets and the properties usually used for their characterization. We limit our presentation to the most important and related measures to the one proposed in this paper.

Several important inconsistency measures have been defined through minimal inconsistent subsets theories. In [23], the authors introduce a scoring function allowing to measure the contribution of each subset of a knowledge base to the inconsistency. For each subset K' of the knowledge base K, the scoring function is defined as the decrease of the number of minimal inconsistent subsets while K' is removed ($|MUSes(K)| - |MUSes(K - K')|$). The higher the variation is, the better the scoring assigned to K' gets. By extending the scoring function, the authors introduce an inconsistency measure I_{MI} of the whole base [24]. $I_{MI}(K)$ is defined as the number of minimal inconsistent subsets of K, i.e. $I_{MI}(K) = |MUSes(K)|$. In [19], the authors present another family "MinInc inconsistency values MIV" based on minimal inconsistent subsets. For instance, $MIV_D(K, \alpha)$ is a simple measure that is worth 1 if α belongs to a minimal inconsistent subset and 0 otherwise. While $MIV_\#$ is defined similarly to the scoring function, i.e. $MIV_\#(K, \alpha) = |\{\mathcal{M} \in MUSes(K) \mid \alpha \in \mathcal{M}\}|$. A third MIV value takes into account the size of each minimal inconsistent subset in addition to the number of minimal inconsistent subsets of K, i.e. $MIV_C(K, \alpha) = \sum_{\alpha \in \mathcal{M} | \mathcal{M} \in MUSes(K)} \frac{1}{|\mathcal{M}|}$.

Unlike the semantic measures, the approaches based on minimal inconsistent subsets have some gaps. Indeed, such syntactic approaches do not make a

distinction between two different knowledge bases with exactly the same size and the same number of minimal inconsistent subsets, what motivated the new approach introduced in [25]. This approach combines both the minimal inconsistent subsets and the maximal consistent subsets in order to give an inconsistency measure of a given knowledge base. Another approach that combine semantic and syntax based approaches have been introduced in [22]. It is based on counting the variables of MUSes and the minimal correction subsets [26].

2 A Minimal Proof-Based Approach for Measuring Inconsistency

The most syntactic measures of the degree of inconsistency does not differentiate rigorously between the formulae of an inconsistent knowledge base. To illustrate, let us consider the following set of clauses $K = \{a \vee b \vee c, \neg a \vee \neg b \vee \neg c, \neg a \vee b, \neg b \vee c, \neg c \vee a\}$. K is a minimal inconsistent base. Now, if we consider the $MIV_{\#}$ or MIV_C measures, then all formulae in K possess the same degree on inconsistency. In particular, $\forall \alpha \in K$, $MIV_{\#}(K, \alpha) = 1$ and $MIV_C(K, \alpha) = 1/5$.

Clearly, this means that the degree of inconsistency is uniformly distributed between all the formulae of K, and their responsibilities in the inconsistency are consequently shared equally. However, by analyzing the proof of inconsistency of K, one can easily notice that the formulae $\alpha_1 = \neg a \vee b$, $\alpha_2 = \neg b \vee c$, and $\alpha_3 = \neg c \vee a$ are used twice in the resolution scheme [27], whereas $a \vee b \vee c$ and $\neg a \vee \neg b \vee \neg c$ are used only once. Therefore, the formulae $\{\alpha_1, \alpha_2, \alpha_3\}$ are more involved in the inconsistency of K. This simple example shows clearly that we need to analyze deeply the knowledge base in order to exhibit the real contribution/responsibility of each formula in the inconsistency. As the formulae of a knowledge base are not necessarily clauses, one needs to take into account the proof system considered to assess the involvement of each formula in the inconsistency of the base.

Otherwise, one can reason differently to evaluate the participation of the formulae in the inconsistency using what so called *minimal proofs* defined as follows.

Definition 2 (Minimal Proof). *Let K be a knowledge base and π a subset of K. π is a minimal proof of x, iff:*

1. $x \in Lit(\pi)$
2. $\pi \vdash x$
3. $\forall \pi' \subset \pi$, $\pi' \nvdash x$

Note that, the item 2 of Definition 2 shows that x is a logical consequence of π. This means that it can be represented differently as $\pi \cup \{\neg x\} \vdash \bot$. The set of all minimal proofs of a given literal x can be obtained using the MUSes of $K \cup \{\neg x\}$ and, also, the MUSes of K containing only one formula.

Proposition 1. *Let K be a knowledge base and π a subset of K. π is a minimal proof of a literal x, iff $x \in Lit(\pi)$ and*

- $\pi \cup \{\neg x\} \in MUSes(K \cup \{\neg x\})$, *or*
- $\pi \vdash \bot$ *such that* $|\pi| = 1$

Proof. The items 1 and 2 traduce exactly the meaning of Definition 2.

In the sequel, we denote by π_x a *minimal proof* of x in K.

Example 1. Let $K_1 = \{a \vee b \vee c, \neg a \vee b, \neg b \vee c, \neg c \vee a, d \wedge \neg d, a \wedge e, \neg a\}$. Then,

$\pi_a^1 = \{a \wedge e\}$	$\pi_{\neg a}^1 = \{\neg a \vee \neg b \vee \neg c, \neg a \vee b, \neg b \vee c\}$
$\pi_a^2 = \{a \vee b \vee c, \neg b \vee c, \neg c \vee a\}$	$\pi_{\neg a}^2 = \{\neg a\}$
$\pi_b = \{a \vee b \vee c, \neg a \vee b, \neg c \vee a\}$	$\pi_{\neg b} = \{\neg a \vee \neg b \vee \neg c, \neg b \vee c, \neg c \vee a\}$
$\pi_c = \{a \vee b \vee c, \neg a \vee b, \neg b \vee c\}$	$\pi_{\neg c} = \{\neg a \vee \neg b \vee \neg c, \neg a \vee b, \neg c \vee a\}$
$\pi_d = \{d \wedge \neg d\}$,	$\pi_{\neg d} = \{d \wedge \neg d\}$
$\pi_e = \{a \wedge e\}$	$\pi_{\neg e} = \emptyset$

Proposition 2. *Let K be a knowledge base. K is inconsistent if and only if there exists two minimal proofs π_x and $\pi_{\neg x}$ such that $x \in Var(K)$ and $\pi_x \cup \pi_{\neg x} \subseteq K$.*

Proof. ($\longrightarrow$) K is inconsistent, then there exists $x \in Var(K)$ such that $K \vdash x$ and $K \vdash \neg x$. As consequently, there exists $\pi_x \subseteq K$ and $\pi_{\neg x} \subseteq K$ where $\pi_x \vdash x$ and $\pi_{\neg x} \vdash \neg x$. Then, $\pi_x \cup \pi_{\neg x} \subseteq K$.
($\longleftarrow$) Let π_x and $\pi_{\neg x}$ be two minimal proofs such that $\pi_x \cup \pi_{\neg x} \subseteq K$. Using Definition 2, one can deduce that $\pi_x \vdash x$ and $\pi_{\neg x} \vdash \neg x$. Then, $\{\pi_x \cup \pi_{\neg x}\} \vdash x \wedge \neg x$. As consequently, $\{\pi_x \cup \pi_{\neg x}\} \vdash \bot$. Thus, $K \vdash \bot$.

Proposition 3. *Let K be a knowledge base. Then,*

$$\{\mathcal{M} \mid \mathcal{M} \in MUSes(K)\} \subseteq \bigcup \{\{\pi_x \cup \pi_{\neg x}\} \mid x \in Var(K)\}.$$

Proof. A direct consequence of Proposition 2.

Example 2. The following knowledge base $K = \{a \wedge \neg a, a\}$ has one MUS $\mathcal{M} = \{a \wedge \neg a\}$, whereas when considering minimal proofs, K has two minimal proofs $\pi_a^1 = \{a \wedge \neg a\}$ and $\pi_a^2 = \{a\}$ for a and one minimal proof $\pi_{\neg a} = \{a \wedge \neg a\}$ for $\neg a$. Thus, $\pi_a^1 \cup \pi_{\neg a} = \{a \wedge \neg a\}$ and $\pi_a^2 \cup \pi_{\neg a} = \{a, a \wedge \neg a\}$.

The Proposotions 2 and 3 show the relationship between MUSes and minimal proofs in general case. While considering a knowledge base as a set of clauses, the following result holds.

Proposition 4. *Let K be a knowledge base and $\mathcal{M}$ a MUS of K. If the formulae of K are restricted to clauses, we have:*

1. *for all $x \in Lit(\mathcal{M})$, there exists a unique minimal proof π_x in $\mathcal{M}$, and $\mathcal{M} = \pi_x \cup \pi_{\neg x}$*
2. $\{\mathcal{M} \mid \mathcal{M} \in MUSes(K)\} = \bigcup \{\{\pi_x \cup \pi_{\neg x}\} \mid x \in Var(K)\}$

Proof. We know that for each literal $x \in Lit(\mathcal{M})$, there exists a clause α in $\mathcal{M}$ such that $\neg x \in Lit(\alpha)$, since $\mathcal{M}$ is a MUS. Moreover, we know that $\mathcal{M} \setminus \{\alpha\}$ is consistent and $\mathcal{M} \setminus \{\alpha\} \vdash \neg\alpha$. Therefore, we have $\mathcal{M} \setminus \{\alpha\} \vdash x$. Now, it suffices to extract a minimal proof of x from $\mathcal{M} \setminus \{\alpha\}$.

Let us now show that there exists exactly one minimal proof of x in $\mathcal{M}$. Suppose that there exist two minimal proofs π_x^1 and π_x^2 of x such that $\pi_x^1 \neq \pi_x^2$. Let $\pi_{\neg x}$ be a minimal proof of $\neg x$. Then, it is easy to find out that $\pi_x^1 \cup \pi_{\neg x} \vdash \bot$ and $\pi_x^2 \cup \pi_{\neg x} \vdash \bot$. As $\mathcal{M}$ is a MUS, then $\pi_x^1 \cup \pi_{\neg x} = \pi_x^2 \cup \pi_{\neg x} = \mathcal{M}$. Subsequently, $\pi_x^1 = \pi_x^2 = \mathcal{M} \setminus \pi_{\neg x}$, which is contradicted to the assumption.

As for item 2 is a direct consequence of item 1.

Proposition 5. *Let K be a set of clauses and $\mathcal{M}$ a MUS of K. Then, there exists exactly $|Lit(\mathcal{M})|$ minimal proofs in $\mathcal{M}$.*

Proof. A direct consequence of Proposition 4.

Example 3. Let $K_2 = \{a, \neg a, a \vee b, \neg b, \neg b \vee c\}$ be a set of clauses. K_2 has two MUSes $\mathcal{M}_1 = \{a, \neg a\}$ and $\mathcal{M}_2 = \{\neg a, a \vee b, \neg b\}$. Then, $\mathcal{M}_1$ contains two minimal proofs $\pi_a = \{a\}$ and $\pi_{\neg a} = \{\neg a\}$ as $|Lit(\mathcal{M}_1)| = 2$. $\mathcal{M}_2$ contains four minimal proofs $\pi_{\neg a} = \{\neg a\}$, $\pi_a = \{a \vee b, \neg b\}$, $\pi_b = \{\neg a, a \vee b\}$ and $\pi_{\neg b} = \{\neg b\}$ as $|Lit(\mathcal{M}_2)| = 4$.

In the sequel, we denote by $\mathcal{P}_m$ the set of all minimal proofs in K, i.e. $\mathcal{P}_m = \{\pi_x \mid x \in Lit(K)\}$. The subset of $\mathcal{P}_m$ restricted to the minimal proofs of a given literal $x \in K$, is defined as $\mathcal{P}_m(x) = \{\pi \in \mathcal{P}_m \mid \pi \vdash x\}$.

Example 4. Let us consider again the knowledge base K_1 of Example 1.

$$\begin{array}{lll} \mathcal{P}_m(a) = \{\pi_a^1, \pi_a^2\} & \mathcal{P}_m(b) = \{\pi_b\} & \mathcal{P}_m(c) = \{\pi_c\} \\ \mathcal{P}_m(\neg a) = \{\pi_{\neg a}^1, \pi_{\neg a}^2\} & \mathcal{P}_m(\neg b) = \{\pi_{\neg b}\} & \mathcal{P}_m(\neg c) = \{\pi_{\neg c}\} \\ \mathcal{P}_m(d) = \{\pi_d\} & \mathcal{P}_m(\neg d) = \{\pi_{\neg d}\} & \\ \mathcal{P}_m(e) = \{\pi_e\} & & \end{array}$$

Note that in general case, a MUS can involve literals that are not concerned with inconsistency e.g. $\{a \wedge b, \neg a\}$. However if the formulae are restricted to clauses, each variable is necessarily involved in the inconsistency. Thus, we have the following result.

Proposition 6. *Let K be a set of clauses and x a literal of K. Then,*

$$|\{\mathcal{M} \in MUSes(K) \mid x \in Lit(\mathcal{M})\}| \;=\; |\mathcal{P}_m(x)| \;=\; |\mathcal{P}_m(\neg x)|.$$

Proof. Using Proposition 4, each MUS $\mathcal{M}$ involves one minimal proofs for each x and $\neg x$ of K. Moreover, each minimal proof is included in a minimal inconsistent subset. Thus, $|\mathcal{P}_m(x)| = |\{\mathcal{M} \in MUSes(K) \mid x \in Lit(\mathcal{M})\}|$ holds.

Note that while the formulae of a knowledge base are not necessarily clauses, the Proposition 6 becomes $|\{\mathcal{M} \in MUSes(K) \mid x \in Lit(\mathcal{M})\}| \leq |\mathcal{P}_m(x)|$. For instance, let us consider the knowledge base $K = \{a \wedge \neg a, a\}$. The literal a has two minimal proofs, while K contains one MUS $\mathcal{M} = \{a \wedge \neg a\}$.

Proposition 7. *Let K be a knowledge base and $\alpha \in K$ such that $\alpha \nvdash \bot$. If $\{x \mid \exists\ \pi_x, \pi_{\neg x},\ \alpha \in \pi_x\ \cap\ \pi_{\neg x}\} = \emptyset$, then α is a free formula in K.*

Proof. Let α be a consistent formula in K. Assume that α is a free formula and there exists a variable x such that $\alpha \in \pi_x\ \cap\ \pi_{\neg x}$, then $\pi_x \neq \emptyset$ and $\pi_{\neg x} \neq \emptyset$ and consequently, $\pi_x \cup \pi_{\neg x} \vdash \bot$. From $\pi_x \cup \pi_{\neg x}$ we can extract a MUS $\mathcal{M}$ that contains α which is contradictory with the assumption.

As explained in the beginning of this section, we plan to define an inconsistency measure that captures better the structure of the knowledge base. Our aim is to quantify the contribution/responsibility of a formula in the different minimal proofs. Indeed, to measure the degree of inconsistency of a subset of formula $K' \subseteq K$, our approach will increase this degree each time K' is involved in the minimal proofs of a literal x and its negation $\neg x$.

Since a given literal can have several minimal proofs, we need to take into account such occurrences to assess the inconsistency measure of a set of formulae in a knowledge base. The following definition introduces two inconsistency values of a set of formulae $K' \subseteq K$ with respect to a given variable x.

Definition 3. *Let K be a knowledge base, K' a subset of K and $x \in Var(K)$. The inconsistency value of K' with respect to x is:*

$$I_{\mathcal{P}_m}(x, K') = |\{(\pi_x, \pi_{\neg x}) \in \mathcal{P}_m(x) \times \mathcal{P}_m(\neg x) \mid \pi_x \cap K' \neq \emptyset, \pi_{\neg x} \cap K' \neq \emptyset\}|.$$

Example 5. Let us consider the knowledge base K_1. Using $I_{\mathcal{P}_m}$ value, we have the following results: $I_{\mathcal{P}_m}(a, \{a \vee b \vee c, \neg a \vee b\}) = 1$, $I_{\mathcal{P}_m}(a, \{a \vee b \vee c, \neg a \vee b, a \wedge e, \neg a\}) = 4$, and $I_{\mathcal{P}_m}(d, \{d \wedge \neg d\}) = 1$.

An alternative definition of $I_{\mathcal{P}_m}(x, K')$ can be introduced using only MUSes of K as follows:

$$I'_{\mathcal{P}_m}(x, K') = |\{\mathcal{M} \mid \mathcal{M} \in Muses(K), \exists\ (\pi_x, \pi_{\neg x}) \in \mathcal{M} \times \mathcal{M}, \pi_x \cap K' \neq \emptyset, \pi_{\neg x} \cap K' \neq \emptyset\}|.$$

Proposition 8. *Let K be a knowledge base and K' a subset of K. If $I_{\mathcal{P}_m}(K') \neq 0$, then there exists $\mathcal{M} \in MUSes(K)$ such that $K' \cap \mathcal{M} \neq \emptyset$.*

Proof. Assume that $I_{\mathcal{P}_m}(x, K') \neq 0$. Then, there exists $x \in Var(K)$ such that $\mathcal{P}_m(x) \neq \emptyset$ and $\mathcal{P}_m(\neg x) \neq \emptyset$. By Definition 3, there exists two minimal proofs $\pi^1_x \in \mathcal{P}_m(x)$ and $\pi^1_{\neg x} \in \mathcal{P}_m(\neg x)$ such that $K' \cap \pi^1_x \neq \emptyset$ and $K' \cap \pi^1_{\neg x} \neq \emptyset$. As $\pi^1_x \vdash x$ and $\pi^1_{\neg x} \vdash \neg x$, $\{\pi^1_x \cup \pi^1_{\neg x}\} \vdash \bot$. Then, there exists a MUS $\mathcal{M}$ such that $\mathcal{M} \subseteq \{\pi^1_x \cup \pi^1_{\neg x}\}$. Thus, $K' \cap \mathcal{M} \neq \emptyset$ holds, since $K' \cap \{\pi^1_x \cup \pi^1_{\neg x}\} \neq \emptyset$.

Now, we define the degree of inconsistency of a subset K' of K as follows:

Definition 4. *Let K be a knowledge base and K' a subset of K. We define $I_{\mathcal{P}_m}(K')$ the degree of inconsistency of K' as:*

$$I_{\mathcal{P}_m}(K') = \sum_{x \in Var(K)} I_{\mathcal{P}_m}(x, K').$$

The definition 4 can be extended to compute the degree of inconsistency of each formula α in K by considering a single formula instead a subset of formulae. Formally, let α be a formula in K, then:

$$I_{\mathcal{P}_m}(\alpha) = \sum_{x \in Var(K)} I_{\mathcal{P}_m}(x, \alpha).$$

Example 6. Let us consider the knowledge base $K_1 = \{a \vee b \vee c, \neg a \vee b, \neg b \vee c, \neg c \vee a, d \wedge \neg d, a \wedge e, \neg a\}$. Using our measure of inconsistency, we have the following results:

$I_{\mathcal{P}_m}(\{a \wedge e\}) = 0$ $\quad$ $I_{\mathcal{P}_m}(\{d \wedge \neg d\}) = 1$
$I_{\mathcal{P}_m}(\{\neg a \vee b, \neg b \vee c, \neg c \vee a\}) = 3$ $\quad$ $I_{\mathcal{P}_m}(\{\neg a\}) = 0$
$I_{\mathcal{P}_m}(\{a \vee b \vee c, \neg a \vee \neg b \vee \neg c\}) = 3$ $\quad$ $I_{\mathcal{P}_m}(\{\neg c\}) = 0$
$I_{\mathcal{P}_m}(\{\neg a \vee b, \neg b \vee c, \neg c \vee a, \neg a, a \wedge e\}) = 6$

Note that for the knowledge base K_1, the maximum degree of inconsistency based on minimal proofs can not exceed the value 6 since each minimal proof of $\mathcal{P}_m(K_1)$ contains at least one formula from $\{\neg a \vee b, \neg b \vee c, \neg c \vee a, \neg a, a \wedge e\}$.

Up to now, we proposed a measure of the degree of inconsistency of a subset K' of K, i.e. $I_{\mathcal{P}_m}(K')$. This measure tried to catch the implication of K' in the overall inconsistency of K. To define the inconsistency measure of the whole base K, it is just to consider $K' = K$.

Definition 5. *Let K be a knowledge base. We define $I_{\mathcal{P}_m}$ the degree of inconsistency of K as:*

$$I_{\mathcal{P}_m}(K) = \sum_{x \in Var(K)} I_{\mathcal{P}_m}(x, K).$$

Proposition 9. $I_{\mathcal{P}_m}(K) = \sum_{x \in Var(K)} |\mathcal{P}_m(x) \times \mathcal{P}_m(\neg x)|$.

Proof. According to Definition 3, $I_{\mathcal{P}_m}(x, K) = |\{(\pi_x, \pi_{\neg x}) \in \mathcal{P}_m(x) \times \mathcal{P}_m(\neg x) \mid \pi_x \cap K' \neq \emptyset, \pi_{\neg x} \cap K' \neq \emptyset\}| = |\{(\pi_x, \pi_{\neg x}) \in \mathcal{P}_m(x) \times \mathcal{P}_m(\neg x)\}| = |\mathcal{P}_m(x) \times \mathcal{P}_m(\neg x)|$. Thus, $I_{\mathcal{P}_m}(K) = \sum_{x \in Var(K)} |\mathcal{P}_m(x) \times \mathcal{P}_m(\neg x)|$.

The measure $I_{\mathcal{P}_m}$ provides a more fine-grained way for measuring inconsistency. It aims at taking into account the structure of the knowledge base in terms of minimal proofs and the occurring variables in each minimal proof. While the formulae of K are clauses, $I_{\mathcal{P}_m}(K)$ can be rewritten as follows:

Proposition 10. *Let K be a knowledge base where each formula of K is a clause. Then,*

$$I_{\mathcal{P}_m}(K) = \sum_{\mathcal{M} \in MUSes(K)} |Var(\mathcal{M})|.$$

Proof. Let K be a set of clauses. Each literal of a given MUS $\mathcal{M}$ in K has a minimal proof and this minimal proof is unique as proven previously. So from each minimal proof π_x and $\pi_{\neg x}$ we can build a MUS $\mathcal{M} = \{\pi_x \cup \pi_{\neg x}\}$. Also, as proved before, $|\{\mathcal{P}_m(x) \times \mathcal{P}_m(\neg x)\}| = |\mathcal{M} \mid \mathcal{M} \in MUSes(K), x \in Var(\mathcal{M})|$. Thus, we conclude that $I_{\mathcal{P}_m}(K) = \sum_{\mathcal{M} \in MUSes(K)} |Var(\mathcal{M})|$.

Example 7. Again, let us consider Example 3. $MUSes(K_2) = \{\{a, \neg a\}, \{\neg a, a \vee b, \neg b\}\}$. Then, we have $I_{\mathcal{P}_m}(K_2) = 2$.

3 Logical Properties

In [19], the authors define some required properties that a basic inconsistency measure should satisfy (see Definition 6). For example, the property (3) states that the set of formulae not involved in any minimal inconsistent subset are not considered in the inconsistency measure. The monotony property (2) shows that the inconsistency value of a knowledge base has to be increased while adding new formulae. Finally the dominance property (4) shows that if we substitute a consistent formula by a logical consequence one, the inconsistency measure can not be increased.

Definition 6 ([19]). *Let K and K' be two knowledge bases and α and β two formulae in $\mathcal{L}$. A basic inconsistency measure I_M is an inconsistency measure satisfying the following properties.*

(1) Consistency: $I_M(K) = 0$ if K is consistent
(2) Monotony: $I_M(K) \leq I_M(K \cup K')$
(3) Free Formula Independence: if α is a free formula in $K \cup \{\alpha\}$, then $I_M(K \cup \{\alpha\}) = I_M(K)$
(4) Dominance: if $\alpha \vdash \beta$ and $\alpha \nvdash \bot$, then $I_M(K \cup \{\beta\}) \leq I_M(K \cup \{\alpha\})$

Note that $I_{\mathcal{P}_m}$ inconsistency measure satisfies the two first properties in Definition 6. In particular, if K is consistent, then $I_{\mathcal{P}_m}(K) = 0$ ($\forall x, \mathcal{P}_m(x) \times \mathcal{P}_m(\neg x) = 0$). Concerning the monotony property, adding new formulae to the knowledge base increases the number of minimal proofs in the base, and consequently $I_{\mathcal{P}_m}(K) \leq I_{\mathcal{P}_m}(K \cup K')$. Nevertheless, adding free formulae to a knowledge base can enlarge the set of minimal proofs in K (e.g. $K = \{a \wedge \neg a\}$, $\alpha = a$), hence the free formula independence is not satisfied.

Recently in [25], the authors have discussed the limitations of some properties like dominance and free formula independence, especially while dealing with syntactic inconsistency measures. For instance, they prove that I_{MI} measure doesn't satisfy the dominance property. As the minimal proofs and minimal unsatisfiable subsets are correlated, $I_{\mathcal{P}_m}$ doesn't satisfy the dominance property.

4 Restoring Consistency through Inconsistency Measures

As mentioned in the introduction, the second important step in the inconsistency handling process concerns how to restore consistency. To this end, inconsistency measures can be used to guide the process of resolving inconsistency. We often encounter several alternative solutions to restore consistency [28]. In the absence of specific and explicit additional information about the origin of each formula, one of the easiest way to restore consistency is based on formula deletion. It based on removing as less information as possible i.e. removing one formula from each MUS of the given knowledge base. The smallest subset to remove in order to establish consistency is called a *minimal hitting set* (see Definition 7) of the hypergraph representing the set of MUSes as defined in [26]. The number of

such hitting sets is known to be exponential in the worst case. Then, selecting a "best" hitting set is a challenging task since we lack enough information about the original knowledge base. For instance, one can select minimal hitting sets involving the smallest subset of variables. Other criterions can be introduced to pick up the best ones as we discuss below.

Definition 7. *H is a hitting set of a set of sets $\mathcal{S}$ iff $\forall H' \in \mathcal{S}$, $H \cap H' \neq \emptyset$. A hitting set H is minimal if there is no other hitting set H'' such that $H'' \subset H$.*

In order to use the inconsistency measure to restore consistency, we select the consistent subsets K' of K that are maximal in terms of the measure of inconsistency I and in terms of size at a time. To define formally the restoration procedure, we denote, at first, by $\mathcal{C}_{\mathcal{S}}(K)$ the set of all consistent subsets of K: $\mathcal{C}_{\mathcal{S}}(K) = \{K' \mid K' \subseteq K \text{ and } K' \not\vdash \bot\}$.

Definition 8. *Let K be a knowledge base and I a degree of inconsistency measure. The knowledge bases resulting from the restoration of consistency of K, are the subsets K' of K which verify the following conditions:*

1. $K' \in \mathcal{C}_{\mathcal{S}}(K)$
2. $\forall K'' \subseteq K \setminus K'$, *if* $I(K', K) < I(K' \cup K'', K)$, *then* $K' \cup K'' \vdash \bot$
3. $\forall K'' \in \mathcal{C}_{\mathcal{S}}(K)$, $|K''| \leq |K'|$

In the rest of the paper, we denote the subset of K verifying items 1, 2, and 3 by $\mathcal{C}^m_{\mathcal{S}}(I, K)$. Let us now give a characterization of $\mathcal{C}^m_{\mathcal{S}}(I, K)$ through some inconsistency measures. For instance, we show that the set $\mathcal{C}^m_{\mathcal{S}}(MIV_{\#}, K)$ is closed with respect to the minimal hitting sets.

Proposition 11. *Let K be a knowledge base and $\{H_1, \ldots, H_n\}$ the set of minimal hitting sets of $MUSes(K)$. Then,*

$$\mathcal{C}^m_{\mathcal{S}}(MIV_{\#}, K) = \{K \setminus H_1, K \setminus H_2, \ldots, K \setminus H_n\}.$$

Proof. Each subset $K \setminus H_i$ corresponds to the so called maximal consistent subset (MCS) of K. In [29], the authors show that a subset C of K is an MCS if and only if C is a minimal hitting set of $MUSes(K)$. Furthermore, $MIV_{\#}(K \setminus H_i) = |MUSes(K)|$ which is the maximal value possible for $MIV_{\#}$ measure. Thus, $\mathcal{C}^m_{\mathcal{S}}(MIV_{\#}, K) = \{K \setminus H_1, K \setminus H_2, \ldots, K \setminus H_n\}$.

In the sequel, we illustrate, through an example, the resulting set $\mathcal{C}^m_{\mathcal{S}}$ using $MIV_{\#}$ and $I_{\mathcal{P}_m}$.

Example 8. Let us consider the knowledge base $K_3 = \{a \vee b, \neg a \vee c, a \vee \neg b, \neg c, d \vee e, \neg d \vee c, d \vee \neg e\}$. The minimal inconsistent subsets of K_3 are: $\mathcal{M}_1 = \{a \vee b, \neg a \vee c, a \vee \neg b, \neg c\}$ and $\mathcal{M}_2 = \{\neg c, d \vee e, \neg d \vee c, d \vee \neg e\}$. These two MUSes share one only common formula $\neg c$. Hence, the only minimal hitting set H_1 is equal to $\{\neg c\}$. According to Definition 8, restoring the consistency of K_3 leads to

$\mathcal{C}^m_{\mathcal{S}}(MIV_{\#}, K_3) = K_3 \setminus \{\neg c\} = \{a \vee b, \neg a \vee c, a \vee \neg b, d \vee e, \neg d \vee c, d \vee \neg e\}$. Using our $I_{\mathcal{P}_m}$ measure, a consistent subset of K_3 satisfying Definition 8 is $\mathcal{C}^m_{\mathcal{S}}(I_{\mathcal{P}_m}, K_3) = \{\neg a \vee c, a \vee \neg b, \neg c, \neg d \vee c, d \vee \neg e\}$. This example shows that restoring consistency using $\mathcal{C}^m_{\mathcal{S}}(MIV_{\#}, K)$ and $\mathcal{C}^m_{\mathcal{S}}(I_{\mathcal{P}_m}, K)$ leads to different sets.

Note that, from Example 8 and using minimal hitting sets, we obtain a new knowledge base where a, c, and d are logical consequences but not b and e; whereas with the proposed measure $I_{\mathcal{P}_m}$, the literals $\neg a$, b, $\neg c$, $\neg d$, and e are logically deduced. As our approach is minimal proofs based, restoring consistency using $I_{\mathcal{P}_m}$ aims to maximize the number of literals that are logical consequences.

5 Conclusion

In this paper, on the one hand, we present a new fine-grained inconsistency measure to quantify the degree of inconsistency of a propositional knowledge base using minimal proofs. It allows us to consider minimal proofs as the purest form of inconsistency instead of minimal inconsistent subsets. The proposed measure overcomes the limitations raised by the minimal inconsistent subsets by taking into account the structure of the knowledge base. Also, we discuss both the satisfied logical properties and those which are not. On the other hand, we introduce some logical properties in order to define a restoring approach based inconsistency measure. Then, we compare the resulting consistent subsets using our measure and an existing one.

In future works, we plan to analyze the computational complexity of using minimal proofs to measure the degree of inconsistency of a knowledge base, develop algorithms and implementations, and undertake case studies of applications of our measure.

References

1. Chen, Q., Zhang, C., Zhang, S.: A verification model for electronic transaction protocols. In: Yu, J.X., Lin, X., Lu, H., Zhang, Y. (eds.) APWeb 2004. LNCS, vol. 3007, pp. 824–833. Springer, Heidelberg (2004)
2. Martinez, A.B.B., Arias, J.J.P., Vilas, A.F.: On measuring levels of inconsistency in multi-perspective requirements specifications. In: PRISE 2004, pp. 21–30 (2004)
3. Qi, G., Liu, W., Bell, D.A.: Measuring conflict and agreement between two prioritized belief bases. In: IJCAI, pp. 552–557 (2005)
4. Hunter, A.: How to act on inconsistent news: Ignore, resolve, or reject. Data Knowl. Eng. 57(3), 221–239 (2006)
5. Hunter, A., Konieczny, S.: Shapley inconsistency values. In: KR, pp. 249–259 (2006)
6. Grant, J., Hunter, A.: Measuring inconsistency in knowledgebases. J. Intell. Inf. Syst. 27(2), 159–184 (2006)
7. Martinez, M.V., Pugliese, A., Simari, G.I., Subrahmanian, V.S., Prade, H.: How dirty is your relational database? An axiomatic approach. In: Mellouli, K. (ed.) ECSQARU 2007. LNCS (LNAI), vol. 4724, pp. 103–114. Springer, Heidelberg (2007)

8. Zhou, L., Huang, H., Qi, G., Ma, Y., Huang, Z., Qu, Y.: Measuring inconsistency in dl-lite ontologies. In: Web Intelligence, pp. 349–356 (2009)
9. McAreavey, K., Liu, W., Miller, P., Mu, K.: Measuring inconsistency in a network intrusion detection rule set based on snort. Int. J. Semantic Computing 5(3) (2011)
10. Knight, K.: Measuring inconsistency. J. Philosophical Logic 31(1), 77–98 (2002)
11. Grant, J.: Classifications for inconsistent theories. Notre Dame Journal of Formal Logic 19(3), 435–444 (1978)
12. Hunter, A.: Measuring inconsistency in knowledge via quasi-classical models. In: AAAI/IAAI, pp. 68–73 (2002)
13. Oller, C.A.: Measuring coherence using lp-models. J. Applied Logic 2(4), 451–455 (2004)
14. Grant, J., Hunter, A.: Analysing inconsistent first-order knowledgebases. Artif. Intell. 172(8-9), 1064–1093 (2008)
15. Ma, Y., Qi, G., Xiao, G., Hitzler, P., Lin, Z.: Computational complexity and anytime algorithm for inconsistency measurement. Int. J. Software and Informatics 4(1), 3–21 (2010)
16. Xiao, G., Lin, Z., Ma, Y., Qi, G.: Computing inconsistency measurements under multi-valued semantics by partial max-sat solvers. In: KR (2010)
17. Ma, Y., Qi, G., Hitzler, P.: Computing inconsistency measure based on paraconsistent semantics. J. Log. Comput. 21(6), 1257–1281 (2011)
18. Doder, D., Raskovic, M., Markovic, Z., Ognjanovic, Z.: Measures of inconsistency and defaults. Int. J. Approx. Reasoning 51(7), 832–845 (2010)
19. Hunter, A., Konieczny, S.: Measuring inconsistency through minimal inconsistent sets. In: KR, pp. 358–366 (2008)
20. Mu, K., Liu, W., Jin, Z.: A general framework for measuring inconsistency through minimal inconsistent sets. Knowl. Inf. Syst. 27(1), 85–114 (2011)
21. Mu, K., Liu, W., Jin, Z.: Measuring the blame of each formula for inconsistent prioritized knowledge bases. J. Log. Comput. 22(3), 481–516 (2012)
22. Xiao, G., Ma, Y.: Inconsistency measurement based on variables in minimal unsatisfiable subsets. In: ECAI, pp. 864–869 (2012)
23. Hunter, A.: Logical comparison of inconsistent perspectives using scoring functions. Knowl. Inf. Syst. 6(5), 528–543 (2004)
24. Hunter, A., Konieczny, S.: On the measure of conflicts: Shapley inconsistency values. Artif. Intell. 174(14), 1007–1026 (2010)
25. Mu, K., Liu, W., Jin, Z., Bell, D.A.: A syntax-based approach to measuring the degree of inconsistency for belief bases. Int. J. Approx. Reasoning 52(7), 978–999 (2011)
26. Reiter, R.: A theory of diagnosis from first principles. Artif. Intell. 32(1), 57–95 (1987)
27. Robinson, A.J.: A machine-oriented logic based on the resolution principle. Journal of the ACM 12(1), 23–41 (1965)
28. Grant, J., Hunter, A.: Measuring the good and the bad in inconsistent information. In: IJCAI, pp. 2632–2637 (2011)
29. Liffiton, M.H., Sakallah, K.A.: Algorithms for computing minimal unsatisfiable subsets of constraints. J. Autom. Reasoning 40(1), 1–33 (2008)

Representing Synergy among Arguments with Choquet Integral

Souhila Kaci[1] and Christophe Labreuche[2]

[1] LIRMM - UMR 5506, Montpellier, France
kaci@lirmm.fr
[2] Thales Research & Technology, Palaiseau, France
christophe.labreuche@thalesgroup.com

Abstract. Preference-based argumentation frameworks are instantiation of Dung's framework in which the defeat relation (in the sense of Dung) is computed from an attack relation and a preference relation over the set of arguments. Value-based argumentation framework is a preference-based argumentation framework where the preference relation over arguments is derived from a preference relation over values they promote. We extend value-based argumentation framework with collective defeats and arguments promoting values with various strengths. In the extended framework, we define a function which computes the strength of a collective defeat. We define desired properties for the proposed function. Surprisingly, we show that this function obeying the corresponding properties is Choquet integral, a well-known aggregation function at work in multiple criteria decision.

1 Introduction

Argumentation is a reasoning framework which consists first in constructing the arguments, then identifying the acceptable ones and finally drawing conclusions. Dung has proposed an abstract argumentation framework that is composed of a set of arguments and a binary relation which is interpreted as a defeat relation between the arguments [8]. Two basic properties are used: conflict-freeness and defense. These two concepts define the output of an argumentation framework which is a set of sets of arguments that can be accepted together.

Dung's argumentation framework is said abstract as arguments and defeat relation are abstract, i.e. their origin is not known. This had the advantage to see this framework instantiated or extended in different ways. For example a noticeable extension consists of combined defeats: Several arguments may interact and entail a stronger defeat than each can do individually [21]. Dung's framework has also been instantiated with preferences. It is commonly acknowledged that preferences play an important role to solve conflicts between arguments. Preference-based argumentation frameworks are instantiation of Dung's framework in which the defeat relation is derived from an attack relation between arguments and a preference relation over the arguments [24, 1–3, 14, 12]. An attack succeeds (thus called a defeat) if the attacked argument is not strictly preferred to the attacking one. Different ways have been proposed to compute a preference relation over the arguments. For example, the latter may promote different values which may be decisions, point of views, actions, etc. From the audience's preference relation over

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 302–314, 2013.

the values, one can derive a preference relation over the arguments. This framework is called value-based argumentation framework [3].

Why should several arguments interact? The basic idea of the paper is that interaction may actually arise from synergy among values supported by arguments. Synergies among values as felt by the audience are easier to elicit than directly interaction among arguments. Hence we extend value-based argumentation framework with collective defeats with varied strengths. Consider the following example.

Example 1 (Humanitarian action in Africa). In a small village, there is no well so that inhabitants have to go quite far away to get water. In order to help inhabitants, a humanitarian association decided to construct a well inside the village. Actually this action has turned the population against the association for the following reasons:

- There was a local economy around the transportation of water from the remote well. The construction of the well has turned this economy into bankruptcy. These people become hostile to the association.
- As water became an easily accessible resource, people started to waste it. Yet in an area that suffers from severe drought, water is a scarce resource and its waste endangers the equilibrium of the whole area.
- The decision from the association has been seen as interference because local authority has not been sufficiently consulted.

There are several values involved here: $\mathcal{V} = \{health, eco, env, pol\}$, where $health$, eco, env and pol respectively stand for health, economy, environment and political stability. We assume that the values in the previous list are ordered from the most preferred one to the least preferred one. The following arguments can be defined:

- a: Construct the well to help the village solve the water problem. It promotes value $health$.
- b: Do not construct the well in order to avoid turning local economy into bankruptcy. It promotes value eco.
- c: Do not construct the well in order to avoid water waste. It promotes value env.
- d: Do not construct the well in order to avoid interference. The fact that the local authority has not been sufficiently consulted might weaken a little bit its stability. But this will by no mean deeply undermine political stability. Hence argument d promotes only partly value pol.

Argument a is in conflict with any argument b, c and d. In this example, argument a is stronger than any other argument b, c, d as it promotes the most important value. Hence the single attacks of b, c and d on a are not sufficient to undermine a, whereas a defeats any of the three arguments b, c, d. However, arguments b, c, d together promote three values that (considered together) may be stronger than value $health$. Hence the combined attack of b, c and d on a may convince the audience. In this paper, we propose an argumentation framework which handles such considerations. The basic ingredient will be the concept of *capacity* to represent the potential interaction among values.

The rest of the paper is structured as follows. In the next two sections we recall Dung's argumentation framework and its main instantiations/extensions. Subsection 3.2 is however novel. It extends collective argumentation framework (in which the defeat

relation is defined between sets of arguments) with a varied strength defeat relation. In Section 4 we extend value-based argumentation framework with collective and varied strength defeat relations. The new framework is based on a function to model interaction among values. Surprisingly, we show that this function obeying some properties is the Choquet integral, a well-known multiple criteria aggregation function. Lastly we conclude.

2 Argumentation Theory

2.1 Dung's Argumentation Framework

Argumentation is a reasoning model based on constructing arguments, determining potential conflicts between arguments and selecting acceptable arguments. In Dung's framework, arguments are supposed to be given. Conflicts between arguments are represented by a binary *defeat* relation.

Definition 1. *[8] An* argumentation framework *(AF) is a tuple* $\langle \mathcal{A}, \rightharpoonup \rangle$ *where* $\mathcal{A}$ *is a finite set of arguments and* $\rightharpoonup \subseteq \mathcal{A} \times \mathcal{A}$ *is a binary defeat relation.*

The outcome of Dung's argumentation framework is sets of arguments, called *extensions*, that are robust against defeats. We say that $A \subseteq \mathcal{A}$ *defends* a if $\forall b \in \mathcal{A}$ s.t. $b \rightharpoonup a$, $\exists c \in A$ such that $c \rightharpoonup b$. We say that $A \subseteq \mathcal{A}$ is *conflict-free* if there are no $a, b \in A$ such that $a \rightharpoonup b$. A subset $A \subseteq \mathcal{A}$ of arguments is an *admissible extension* iff it is conflict-free and it defends all elements in A. Other acceptability semantics exist [8].

2.2 Preference-Based Argumentation Framework

Preference-based argumentation framework is an instantiation of Dung's framework. It is based on a binary attack relation between arguments and a preference relation over the set of arguments.

Definition 2. *[1] A* preference-based argumentation framework *(PAF) is a 3-tuple* $\langle \mathcal{A}, \rightsquigarrow, \succeq \rangle$ *where* $\mathcal{A}$ *is a set of arguments,* $\rightsquigarrow \subseteq \mathcal{A} \times \mathcal{A}$ *is a binary attack relation and* $\succeq$ *is a preorder over* $\mathcal{A}$*.*

$\succeq$ is called a Boolean preference relation. A PAF $\langle \mathcal{A}, \rightsquigarrow, \succeq \rangle$ *represents* $\langle \mathcal{A}, \rightharpoonup \rangle$ iff

$$\forall a, b \in \mathcal{A} \quad : \quad a \rightharpoonup b \quad \text{iff} \quad (a \rightsquigarrow b \text{ and } not(b \succ a)), \tag{1}$$

where $b \succ a$ is true if and only if $b \succeq a$ holds but $a \succeq b$ does not.

The extensions of a PAF are simply the extensions of the AF it represents.

Different ways have been proposed in the literature to compute the preference relation $\succeq$ over $\mathcal{A}$. For example, a weight function $w : \mathcal{A} \to [0, 1]$ can be defined. Then

$$\forall a, b \in \mathcal{A} \quad : \quad a \succeq b \text{ iff } w(a) \geq w(b).$$

In some applications, the arguments need to be compared not on the basis of their internal structure but with respect to the viewpoints or decisions they promote [3]. This may be due to the fact that the internal structure of the arguments is not available or

because the values must be considered. This is particularly true in persuasion dialogs when the preference over values induces the preference over arguments promoting the values [3]. Thus, if two arguments are conflicting then the argument promoting a preferred value is accepted. Bench-Capon developed an argumentation framework which models the above considerations [3]. Like Dung's framework, he considers abstract arguments. Moreover, he considers (i) a set of values promoted by the arguments and (ii) a set of audiences where an audience corresponds to a preference relation over values.

Definition 3. *[3] A value-based argumentation framework is a five-tuple,* $VAF = \langle \mathcal{A}, \rightsquigarrow, \mathcal{V}, val, \Delta \rangle$*, where* $\mathcal{A}$ *is a finite set of arguments,* $\rightsquigarrow$ *is an attack relation over* $\mathcal{A} \times \mathcal{A}$*,* $\mathcal{V}$ *is a nonempty set of values,* $val : \mathcal{A} \rightarrow 2^{\mathcal{V}}$ *returns the set of values promoted by each argument, and* Δ *is the set of possible audiences. An audience specific argumentation framework is a five-tuple,* $VAF_\delta = \langle \mathcal{A}, \rightsquigarrow, \mathcal{V}, val, \succ_\delta \rangle$*, where* $\delta \in \Delta$ *is an audience and* $\succ_\delta$ *is a partial order over* $\mathcal{V}$*.*

In this paper we consider audience specific argumentation framework and denote it $\langle \mathcal{A}, \rightsquigarrow, \mathcal{V}, val, \succ_{\mathcal{V}} \rangle$. We suppose that an argument promotes at least one value. Different ways have been proposed to compute a preference relation over $\mathcal{A}$ given $\succ_{\mathcal{V}}$. We refer the reader to [3, 14]. One may for instance use the following definition:

$$\forall a, b \in \mathcal{A}, \quad a \succ b \quad \text{iff} \quad \exists v \in val(a)\ \forall v' \in val(b) \quad v \succ_{\mathcal{V}} v'. \tag{2}$$

2.3 Argumentation Framework with Varied-Strength Defeats

Strength of defeat relations has been incorporated in argumentation framework in two ways: a qualitative relative way by means of a partial preorder [19, 20] and a quantitative way by means of a numerical function [9]. As far as the present paper is concerned, we follow the second modeling.

Definition 4. *[9] An* argumentation framework with varied-strength defeats (AFV) *is a 3-tuple* $\langle \mathcal{A}, \rightharpoonup, VDef \rangle$ *where* $\langle \mathcal{A}, \rightharpoonup \rangle$ *is a Dung's argumentation framework and* $VDef$ *is a function defined from* $\rightharpoonup$ *to* $(0, 1]$*.*

For simplicity, we consider the interval $(0, 1]$ but any bipolar linearly ordered scale with top, bottom and neutral elements can be used as well. $VDef(a, b)$ is the degree of the statement "a defeats b" being true. Values 0, $\frac{1}{2}$ and 1 for $VDef(a, b)$ mean that the validity of the previous statement is certainly false, unknown and certainly true respectively. We say that a *defeats* b w.r.t. $\langle \mathcal{A}, \rightharpoonup, VDef \rangle$ iff $a \rightharpoonup b$.

Extensions are also defined from the conflict-freeness and defense. Conflict-freeness is defined as for $\langle \mathcal{A}, \rightharpoonup \rangle$. Defense is however extended to the valued case. When $b \rightharpoonup a$ and $c \rightharpoonup b$, the strength of defeats should play a role in the definition of the defense since c is considered as a "serious" defender of a if the defeat of c on b is at least as strong as the defeat of b on a. The set $A \subseteq \mathcal{A}$ *defends* $a \in \mathcal{A}$ w.r.t. $\langle \mathcal{A}, \rightharpoonup, VDef \rangle$ iff for all $b \in \mathcal{A}$ such that $b \rightharpoonup a$, there exists $c \in A$ with [19]:

$$c \rightharpoonup b \text{ and } VDef(c, b) \geq VDef(b, a).$$

Let us now describe an instantiation of this framework where the valued defeat relation is derived from a valued preference relation $P : \mathcal{A} \times \mathcal{A} \rightarrow [0, 1]$. $P(a, b)$ is the degree of the statement *"a is strictly preferred to b"* being true.

Definition 5. *[13] A* valued preference-based argumentation framework (VPAF)[1] *is a 3-tuple* $\langle \mathcal{A}, \rightsquigarrow, P \rangle$ *where* $\mathcal{A}$ *is the set of arguments,* $\rightsquigarrow \subseteq \mathcal{A} \times \mathcal{A}$ *is a binary attack relation and* P *is a function defined from* $\mathcal{A} \times \mathcal{A}$ *to* $[0, 1]$.

A VPAF $\langle \mathcal{A}, \rightsquigarrow, P \rangle$ represents an argumentation framework with varied-strength defeats $\langle \mathcal{A}, \rightharpoonup, VDef \rangle$ iff $a \rightharpoonup b$ if $a \rightsquigarrow b$ and $P(b, a) < 1$, $VDef(a, b) = 1 - P(b, a)$ if $a \rightharpoonup b$. Lastly, $VDef(a, b) = 0$ otherwise. An interesting case is when P is derived from a valuation function w over the arguments. A suitable expression of P is $P(a, b) = w(a) - w(b)$ if $w(a) > w(b)$ and $P(a, b) = 0$ else [13]. This gives

$$a \rightharpoonup b \quad \text{if} \quad a \rightsquigarrow b \text{ and } [w(a) > 0 \text{ or } w(b) < 1], \tag{3}$$

$$VDef(a, b) = \min(1 + w(a) - w(b), 1) \text{ if } a \rightharpoonup b. \tag{4}$$

3 Arguing with Collective Defeat Relations

3.1 Collective Argumentation Framework

Dung's framework has been extended with a defeat relation between sets of arguments.

Definition 6. *[21] A collective argumentation framework is a pair* $\langle \mathcal{A}, \rightrightarrows \rangle$ *where* $\mathcal{A}$ *is a set of arguments and* $\rightrightarrows \subseteq 2^{\mathcal{A}} \times 2^{\mathcal{A}}$ *is the defeat relation, with, for* $A \subseteq \mathcal{A}$ *and* $B \subseteq \mathcal{A}$*, notation* $A \rightrightarrows B$ *means that the arguments in* A *jointly defeat* B.

The authors of [21] argue that there is no need to define defeat of a subset of arguments on another subset of arguments. In fact, they interpret $A \rightrightarrows B$ as $A \rightrightarrows \{b\}$ for every $b \in B$. Hence it is sufficient to see $\rightrightarrows$ as a subset of $2^{\mathcal{A}} \times \mathcal{A}$. This definition implicitly means that if $A \rightrightarrows \{b\}$ for every $b \in B$ then $A \rightrightarrows B$. However this interpretation may not be sufficient in many situations.

Example 2 (Example 1 cont.). a defeats b as the value promoted by a is more important than that promoted by b. Likewise, a defeats arguments c and d. On the other hand, one may conceive that a does not defeat the set of arguments $\{b, c, d\}$ since the values promoted by these arguments are collectively stronger than the value promoted by a.

The previous example indicates that defining defeats among subsets of arguments is important since the fact that a defeats b, c and d considered separately does not necessarily imply that a defeats b, c and d as a whole.

We don't define the meaning of A "jointly" defeats B at this stage. We borrowed collective argumentation framework from [21] as it nicely models our needs. However our interpretation of joint defeat differs from that proposed in [21], as we will see later. In [21] A jointly defeats B is interpreted as "arguments in A do not separately defeat arguments in B but considered together they do".

[1] Valued preference-based argumentation framework must not be confused with value-based argumentation framework [3]. In the latter, arguments promote values which may be point of views, decisions, opinions, etc. Then a preference relation over the set of arguments is derived from a preference relation over the values. In the former, the preference relation over the set of arguments is valued, i.e. it expresses preferences with varied strength, as we will see later.

A set A of arguments is *conflict-free* if there is no subsets $A', A'' \subseteq A$ such that $A' \rightrightarrows A''$. Let $A, B, C \subseteq \mathcal{A}$. We say that $C \subseteq \mathcal{A}$ *defends* $A \subseteq \mathcal{A}$ if $\forall B \subseteq \mathcal{A}$ with $B \rightrightarrows A$ we have $C \rightrightarrows B$. The semantics of acceptability can be defined from the concepts of conflict-freeness and defense as usual.

Relation $\rightrightarrows$ shall satisfy some monotonicity conditions: for all $A, B \subseteq \mathcal{A}$

$$\forall B' \subseteq B, \qquad \text{if } A \rightrightarrows B \text{ then } A \rightrightarrows B'. \tag{5}$$

Therefore we recover the interpretation of $A \rightrightarrows B$ given in [21]. However we do not necessarily have that $A \rightrightarrows B$ if $A \rightrightarrows B', \forall B' \subseteq B, B' \neq B$.

3.2 Arguing with Collective Varied Defeats

In this section we extend the collective argumentation framework defined in the previous subsection with a varied defeat relation.

Definition 7. *A collective argumentation framework with varied defeats is a triplet* $\langle \mathcal{A}, \rightrightarrows, \overline{VDef} \rangle$ *where* $\mathcal{A}$ *is a set of arguments and* $\rightrightarrows \subseteq 2^{\mathcal{A}} \times 2^{\mathcal{A}}$ *is a defeat relation and* $\overline{VDef}$ *is a function from* $\rightrightarrows$ *to* $(0, 1]$.

$\overline{VDef}(A, B)$ is the degree of credibility of statement "A defeats B".

A set A of arguments is *conflict-free* if there is no $A', A'' \subseteq A$ such that $A' \rightrightarrows A''$. We say that $C \subseteq \mathcal{A}$ *defends* $A \subseteq \mathcal{A}$ if for all $B \subseteq \mathcal{A}$ such that $B \rightrightarrows A$, there exists $C' \subseteq C$ such that $C' \rightrightarrows B$ and $\overline{VDef}(C', B) \geq \overline{VDef}(B, A)$. The semantics of acceptability can be defined from the concepts of conflict-freeness and defense as usual.

$\overline{VDef}$ shall satisfy some monotonicity condition. For all $A, B, A', B' \subseteq \mathcal{A}$

$$\text{if } A' \subseteq A,\ B' \supseteq B,\ A \rightrightarrows B \text{ and } A' \rightrightarrows B' \text{ then } \overline{VDef}(A', B') \leq \overline{VDef}(A, B). \tag{6}$$

Indeed the more arguments we add to A the stronger the defeat, and the more arguments we add to B the weaker the defeat.

4 Extended Value-Based Argumentation Framework

In standard value-based argumentation framework arguments fully promote a subset of values in $\mathcal{V}$ [3]. In many applications however, arguments support values with various strengths. In Example 1, argument d promotes only partly value pol. Hence function val is refined in the following way.

Definition 8. *For* $a \in \mathcal{A}$*, we define* $f : \mathcal{A} \times \mathcal{V} \to [0, 1]$ *such that* $f(a, v)$ *is the degree to which argument* $a \in \mathcal{A}$ *supports value* $v \in \mathcal{V}$.

The aim of this section is to extend single defeats (i.e., an argument defeats an argument) to collective defeats (i.e., a set of arguments defeats a set of arguments). These defeats will hold with degrees that will be derived from f.

4.1 Construction

The main question we face to define $\overline{VDef}$ is to what extent different arguments can produce a stronger defeat than each argument can do individually. In some sense, they have some complementarity among themselves. The key idea is that each argument may support a different value and that the audience is much more convinced by a set of relevant values than by only one of them.

Example 3 (Example 1 cont.). The audience may say that value $health$ is more important than any other value eco, env or pol. Hence a defeats b, c, d and none of b, c, d defeats a. On the other hand, arguments b, c, d promote three different values and, one may conceive that b, c, d together defeat a.

In order to define $\overline{VDef}(A, B)$, we need to extend the boolean preference relation $\succ_{\mathcal{V}}$ over $\mathcal{V}$ to a valued preference relation over $2^{\mathcal{V}}$. We represent this preference relation by a numerical function $\mu : 2^{\mathcal{V}} \to \mathbb{R}^+$. For $V \subseteq \mathcal{V}$, $\mu(V)$ is the strength of the preference if all values in V are completely promoted and the remaining values are not promoted at all. This set function, called a capacity, shall satisfy some properties [6].

Definition 9. *A* capacity *on $\mathcal{V}$ is a set function $\mu : 2^{\mathcal{V}} \to \mathbb{R}_+$ satisfying two properties:*

- *(monotonicity) $\mu(V) \leq \mu(V')$ for all $V, V' \subseteq \mathcal{V}$ with $V \subseteq V'$,*
- *(boundary condition) $\mu(\emptyset) = 0$ and $\mu(\mathcal{V}) = 1$.*

The monotonicity condition will serve in the definition of a strength of defeats: the more values a set of arguments supports, the stronger the defeat. The boundary condition essentially says that the audience is not convinced by a set of arguments if they do not support any value[2]. Hence the values represent all possible stakes and points of view the audience may believe in. Normalization condition $\mu(\mathcal{V}) = 1$ comes from the fact that the strength of defeat is bounded by 1.

Note that capacity μ is a refinement of order $\succ_{\mathcal{V}}$:

$$\forall v, v' \in \mathcal{V}, \text{ if } v \succ_{\mathcal{V}} v' \quad \text{then} \quad \forall V \subseteq \mathcal{V} \setminus \{v, v'\} \quad \mu(V \cup \{v\}) > \mu(V \cup \{v'\}).$$

This property is similar to *responsiveness* defined by Roth [23] (see also [4]).

On the basis of the above definitions, we define an extended value-based argumentation framework in the following way:

Definition 10. *An extended value-based argumentation framework is a five-tuple, $\langle \mathcal{A}, \rightsquigarrow, \mathcal{V}, f, \mu \rangle$, where $\mathcal{A}$ is a finite set of arguments, $\rightsquigarrow \subseteq \mathcal{A} \times \mathcal{A}$ is an attack relation, $\mathcal{V}$ is a nonempty set of values, f is a function from $\mathcal{A} \times \mathcal{V}$ to $[0, 1]$, and μ is a capacity over $\mathcal{V}$.*

We are going to derive a collective argumentation framework with varied defeats $\langle \mathcal{A}, \rightrightarrows, \overline{VDef} \rangle$ (see Definition 7). We extend w and relations (3) and (4). To this end, we define a valuation $G : 2^{\mathcal{A}} \to [0, 1]$ of subsets of arguments. More precisely, it evaluates the

[2] Arguments in $\mathcal{A}$ are supposed to promote at least one value, but one may imagine other arguments promoting no value in $\mathcal{V}$.

strength of a set of arguments. This strength will be used to evaluate the degree of defeat between two subsets of arguments. $G(A)$ (with $A \subseteq \mathcal{A}$) depends on the values promoted by arguments in A (see function f) and the strength of these values (see function μ). First of all, extending (3), the defeat relation $\rightrightarrows$ is defined as follows:

$$A \rightrightarrows B \quad \text{if} \quad (G(A) > 0 \text{ or } G(B) < 1) \text{ and } [\forall b \in B \; \exists a \in A \quad a \rightsquigarrow b] \text{ and } [\forall a \in A \; \exists b \in B \quad a \rightsquigarrow b] \tag{7}$$

where Following (4), the intensity of the defeat is given by

$$\overline{VDef}(A, B) = \min(1 + G(A) - G(B), 1) \quad \text{if } A \rightrightarrows B \tag{8}$$

According to these definitions, if $A \rightrightarrows B$ then $\overline{VDef}(A, B) > 0$, as required by Definition 7. The next two subsections are devoted to the definition of the function G.

4.2 Computing $G(A)$: Case When A Is a Singleton

Consider in this section the case where $A = \{a\}$. $G(\{a\})$ depends only on the degree to which values are supported by a (i.e. on $\{f(a, v) | v \in \mathcal{V}\}$) as well as on the strength μ of values. Hence there exists a function denoted by $F_\mu : \mathbb{R}_+^{\mathcal{V}} \to \mathbb{R}_+$ (to be determined) such that:

$$G(\{a\}) = F_\mu(\{f(a, v) | v \in \mathcal{V}\}). \tag{9}$$

We will use an axiomatic approach to get F_μ from a set of wished properties on F_μ.

– Properties of the Function F_μ We already justified monotonicity condition on the capacity (see Definition 9). This condition can be extended to F_μ. If the degree to which an argument supports a value increases, F_μ shall not decrease.

Increasingness (In): $\forall x, y \in \mathbb{R}^{\mathcal{V}}$, if $x_v \leq y_v \; \forall v \in \mathcal{V}$ then $F_\mu(x) \leq F_\mu(y)$.

Element x_v (resp. y_v) represents the degree to which an argument a (resp. another argument b) promote values v in $\mathcal{V}$. As argument b promotes every value at least as well as a ($x_v \leq y_v$ for every $v \in \mathcal{V}$), the valuation of b should not be lower. $F_\mu(x) \leq F_\mu(y)$ derives from $G(\{a\}) \leq G(\{b\})$.

In the previous subsection, we have interpreted $\mu(V)$ as the strength of preference if all values in V are completely promoted and the remaining ones are not. Formally, we write:

Properly Weighted (PW): $F_\mu(\underbrace{1, \cdots, 1}_{v \in V}, \underbrace{0, \cdots, 0}_{v \notin V}) = \mu(V), \forall V \subseteq \mathcal{V}$.

From **(PW)**, if μ is multiplied by a number then the resulting strength is also multiplied by the same factor: $F_{\gamma\mu}(x) = \gamma F_\mu(x)$ for any $\gamma \in \mathbb{R}$. As a capacity μ may be provided by an expert, if another expert provides μ' then one may combine μ and μ' with a linear transformation $\gamma\mu + \delta\mu'$ ($\gamma, \delta \in \mathbb{R}$). Then it is reasonable that the overall

aggregation function equals the same linear transformation of the aggregation for the two decision makers:

Linearity w.r.t. the Measure (LM): For all $x \in \mathbb{R}^{\mathcal{V}}$ and $\gamma, \delta \in \mathbb{R}$,

$$F_{\gamma\mu+\delta\mu'}(x) = \gamma F_\mu(x) + \delta F_{\mu'}(x) \ . \tag{10}$$

The numerical values of $f(a,.)$ correspond to an interval scale in the sense of measurement theory [16]. An interval scale is given up to an affine transformation. Hence F_μ shall be invariant under any affine transformation. However, as all degrees $f(a,v)$ for all v correspond to the same scale, the same transformation shall be applied to all values in $\mathcal{V}$. Starting from **(PW)**, we impose this invariance property only on situation where each value is either completely supported or not supported at all.

Stability for the admissible Positive Linear transformations (weak SPL): For all $V \subset \mathcal{V}$, $\alpha > 0$, and $\beta \in \mathbb{R}$,

$$F_\mu(\underbrace{(\alpha+\beta),\cdots,(\alpha+\beta)}_{v\in V},\underbrace{\beta,\cdots,\beta}_{v\notin V}) = \alpha F_\mu(\underbrace{1,\cdots,1}_{v\in V},\underbrace{0,\cdots,0}_{v\notin V}) + \beta.$$

This axiom is a weak version of the axiom **(SPL)** introduced by Marichal [18] : For all $x \in \mathbb{R}^{\mathcal{V}}$, $\alpha > 0$, and $\beta \in \mathbb{R}$, $F_\mu(\alpha x + \beta) = \alpha F_\mu(x) + \beta$.

Example 4 (Example 1 cont.). The following values of μ are supposed given:

$\mu(\emptyset) = 0$	$\mu(\{health\}) = 0.6$	$\mu(\{eco\}) = 0.2$
$\mu(\{env\}) = 0.1$	$\mu(\{pol\}) = 0.05$	$\mu(\{eco, env\}) = 0.5$
$\mu(\{eco, pol\}) = 0.3$	$\mu(\{env, pol\}) = 0.2$	$\mu(\{eco, env, pol\}) = 0.9$
$\mu(\{health, eco, env, pol\}) = 1$		

We note that there is a strong positive synergy among values eco, env, pol as

$$\mu(\{eco, env, pol\}) > \mu(\{eco\}) + \mu(\{env\}) + \mu(\{pol\}).$$

These three values together are more important than value $health$ alone.

– Function F_μ vs Choquet Integral. Now that we have given properties of F_μ, we show that this function is already at work in multiple criteria decision and known as Choquet integral [6]. The Choquet integral is a generalization of the commonly used weighted sum.

Definition 11. *Let μ be a capacity on $\mathcal{V}$, with $|\mathcal{V}| = n$. Let $x = (x_1, \cdots, x_n) \in \mathbb{R}^{\mathcal{V}}$. The* discrete Choquet integral *of x with respect to μ is defined by*

$$C_\mu(x) = \sum_{i=1}^{n} (x_{(i)} - x_{(i-1)}) \mu(\{(i), \cdots, (n)\}),$$

with $x_{(0)} = 0$, and where $(1), \ldots, (n)$ indicate that the indices have been permuted so that $0 \leq x_{(1)} \leq \cdots \leq x_{(n)}$.

Example 5. Let us illustrate Def. 11 on $x = (0, 0.6, 1, 0.1)$. The worse score of x is its first component (i.e. $(1) = 1$), the second worse score of x is its last component (i.e. $(2) = 4$), the third worse score of x is its second component (i.e. $(3) = 2$) and the best score of x is its third component (i.e. $(4) = 3$). Hence $C_\mu(x) = (x_1 - 0)\mu(\{1,2,3,4\}) + (x_4 - x_1)\mu(\{2,3,4\}) + (x_2 - x_4)\mu(\{2,3\}) + (x_3 - x_2)\mu(\{3\})$.

Besides, the Choquet integral can model typical human behavior such as the veto. This operator is also able to model the importance of values and the interaction between values. Conversely, the Choquet integral can be interpreted in terms of the importance of values, the interaction between values, and veto [10, 11].

Proposition 1 ([17]). *F_μ satisfies* **(LM)**, **(In)**, **(PW)** *and* **(weak SPL)** *if and only if $F_\mu \equiv C_\mu$ in $\mathbb{R}^{\mathcal{V}}$.*

The proof of this proposition can be found in [17]. Proposition 1 shows that if one agrees on properties **(LM)**, **(In)**, **(PW)** and **(weak SPL)**, then he shall use the Choquet integral w.r.t. μ, namely $G(\{a\}) = C_\mu(\{f(a,v)|v \in \mathcal{V}\})$ (see (9)).

4.3 Computing $G(A)$: Case Where A Is Composed of Several Arguments

When A is not reduced to a singleton, we generalize the construction given in Subsection 4.2. Function F_μ can still be used to compute $G(A)$. We denote by $x_{f,A}(v)$ the degree to which all arguments in A promote together value v, with $x_{f,A} \in [0,1]^{\mathcal{V}}$. Hence (9) is generalized as follows:

$$G(A) = F_\mu(x_{f,A}). \tag{11}$$

For $v \in \mathcal{V}$, $x_{f,A}$ is derived from $\{f(a,v)|a \in A\}$. A maximum function could work: $x_{f,A}(v) = \max_{a \in A} f(a,v)$. However, for the same maximal number of the individual $f(a,v)$, this maximal number could be reinforced if $f(a,v')$ is also large for another value v'. A t-conorm[3] denoted by $\oplus$ could be used to express this reinforcement property. Note that $\alpha \oplus 0 = \alpha$ for all $\alpha \in [0,1]$. Hence

$$\forall v \in \mathcal{V} \qquad x_{f,A}(v) = \oplus\{f(a,v)|a \in A\} \tag{12}$$

Example 6 (Example 3 continued). Assume the values of f are: $f(a, health) = 1$, $f(b, eco) = 1$, $f(c, env) = 1$ and $f(d, pol) = .5$. All other values of $f(.,.)$ are equal to 0. Let us compute the value of G for several subsets of arguments

$$G(\{a\}) = C_\mu(1,0,0,0) = \mu(health) = 0.6$$
$$G(\{b\}) = C_\mu(0,1,0,0) = \mu(eco) = 0.2$$
$$G(\{b,c\}) = C_\mu(0,1,1,0) = \mu(\{eco, env\}) = 0.5$$
$$G(\{b,c,d\}) = C_\mu(0,1,1,0.5) = 0.5\,\mu(\{eco, env\}) + 0.5\,\mu(\{eco, env, pol\}) = 0.7.$$

Let us now consider the following subsets of arguments

- Set $\{b,c\}$ is conflict-free but does not defend itself. Indeed for the attack by a, we have $\{a\} \rightrightarrows \{b,c\}$, $\{b,c\} \rightrightarrows \{a\}$ but (see (8))

$$\overline{VDef}(\{b,c\},\{a\}) = 0.9 < \overline{VDef}(\{a\},\{b,c\}) = 1$$

[3] A function $\oplus : [0,1] \times [0,1] \to [0,1]$ is called a *t-conorm* (*triangular conorm*) if it satisfies $\oplus(0,x) = x$ for all $x \in [0,1]$ (neutral element), $\oplus(x,y) = \oplus(y,x)$ for all $x, y \in [0,1]$ (commutativity), $\oplus(x,y) \leq \oplus(u,v)$ for all $0 \leq x \leq u \leq 1$ and $0 \leq y \leq v \leq 1$ (monotonicity), and $\oplus(x, \oplus(y,z)) = \oplus(\oplus(x,y), z)$ for all $x, y, z \in [0,1]$ (associativity) [15].

– Set $\{a\}$ is conflict-free but does not defend itself. Indeed for the attack by b, c, d, we have $\{b, c, d\} \rightrightarrows \{a\}$, $\{a\} \rightrightarrows \{b, c, d\}$ but (see (8))

$$\overline{VDef}(\{a\}, \{b, c, d\}) = 0.9 < \overline{VDef}(\{b, c, d\}, \{a\}) = 1$$

– Set $\{b, c, d\}$ is conflict-free and defends itself. Indeed for the attack by a, we have $\{a\} \rightrightarrows \{b, c, d\}$, $\{b, c, d\} \rightrightarrows \{a\}$ and (see (8))

$$\overline{VDef}(\{b, c, d\}, \{a\}) = 1 > \overline{VDef}(\{a\}, \{b, c, d\}) = 0.9.$$

Hence $\{b, c, d\}$ is the unique set of admissible arguments.

Assume now that argument d is removed from $\mathcal{A}$, so $\mathcal{A} = \{a, b, c\}$. In this case, $\{a\}$ is an extension as it defends itself from the attack of $\{b, c\}$ (since $\{b, c\} \rightrightarrows \{a\}$, $\{a\} \rightrightarrows \{b, c\}$ and $\overline{VDef}(\{a\}, \{b, c\}) = 1 > \overline{VDef}(\{b, c\}, \{a\}) = 0.9$). Thus a becomes acceptable as b, c are not sufficiently strong compared to a.

4.4 Particular Case: No Interaction among Values

Let us see how relation (2) can be satisfied in our framework. Condition (2) considers the case where f takes only values 0 or 1. Hence one can define val from f by: $val(a) = \{v \in \mathcal{V},\ f(a, v) = 1\}$ for every $a \in \mathcal{A}$. Then by **(PW)**, we have for every $a \in \mathcal{A}$, $G(\{a\}) = \mu(val(a))$. Intuitively one feels that relation (2) can be translated in terms of capacity μ in the following way: for all $V, V' \subseteq \mathcal{V}$

$$\mu(V) > \mu(V') \quad \text{iff} \quad \exists v \in V\ \forall v' \in V' \quad \mu(\{v\}) > \mu(\{v'\}). \tag{13}$$

There is no possible cumulative effect (synergy) among the values in this case. This condition is satisfied for instance when $\mu(V) = \max_{v \in V} \mu(\{v\})$, which corresponds to a possibility measure. The next result shows that under (13), collective defeats will not bring added-value to single defeats.

Proposition 2. *Assume that values are either completely promoted or not all by arguments. Assume furthermore that relation (13) holds. Under (8), we have*

– $\forall A, B \subseteq \mathcal{A}$, *if* $A \rightrightarrows B$ *(i.e.* $\overline{VDef}(A, B) > 0$*), then there exists* $a \in A$ *such that* $\{a\} \rightrightarrows B$ *(i.e.* $\overline{VDef}(\{a\}, B) > 0$*);*
– $\forall A \subseteq \mathcal{A}$, $b_1, \ldots, b_p \in \mathcal{A}$, *if* $A \rightrightarrows b_1$ *(i.e.* $\overline{VDef}(A, \{b_1\}) > 0$*),* $\ldots$*,* $A \rightrightarrows b_p$ *(i.e.* $\overline{VDef}(A, \{b_p\}) > 0$*), then* $A \rightrightarrows \{b_1, \ldots, b_p\}$ *(i.e.* $\overline{VDef}(A, \{b_1, \ldots, b_p\}) > 0$*).*

The proof of this proposition is omitted due to the lack of space. This proposition shows that our framework is general and can encompass the standard case of relation (2). More precisely, condition (2) is translated into (13). Proposition 2 shows that under (13), there exists an argument in the attacking set that defeat the attacked set, and if the attacking set defeats each argument in a set, it defeats the set collectively. This corresponds well to the idea behind (2).

5 Conclusion

In some argument-based applications, arguments need to collectively interact. More precisely, defeat relation is defined among sets of arguments. Moreover this relation has varied strengths due to the fact that arguments promote some values (decision, point of view, etc) with varied strengths. In this paper we developed an argumentation framework extending value-based argumentation framework [3] in order to cope with the above considerations. The strength of defeat $\overline{VDef}(A, B)$ of a subset A of arguments over another subset B depends of the values promoted by A and B. The synergy among the values is encoded in a capacity μ defined on the set of values. As arguments may promote only partly the values, the strength of all arguments in A collectively considered is obtained by using an aggregation function depending on capacity μ. We define desired properties for the aggregation function. We show that this function obeying the corresponding properties is Choquet integral.

Our framework may be applicable to model coalition-based problems where sets of arguments correspond to coalitions [5]. For future work we intend to compare our approach with the accrual of arguments proposed in [22]. We also intend to consider other definitions of defense as suggested in [7]. Lastly, qualitative aggregation functions such as the Sugeno integral will be also considered.

References

1. Amgoud, L., Cayrol, C.: Inferring from inconsistency in preference-based argumentation frameworks. International Journal of Approximate Reasoning 29(2), 125–169 (2002)
2. Amgoud, L., Cayrol, C., LeBerre, D.: Comparing arguments using preference orderings for argument-based reasoning. In: ICTAI 1996, pp. 400–403 (1996)
3. Bench-Capon, T.J.M.: Persuasion in practical argument using value-based argumentation frameworks. Journal of Logic and Computation 13(3), 429–448 (2003)
4. Bossert, W.: Preference extension rules for ranking sets of alternatives with a fixed cardinality. Theory and Decision 39, 301–317 (1995)
5. Bulling, N., Dix, J., Chesñevar, C.: Modelling coalitions: Atl + argumentation. In: AAMAS, pp. 681–688 (2008)
6. Choquet, G.: Theory of capacities. Annales de l'Institut Fourier 5, 131–295 (1953)
7. Coste-Marquis, S., Konieczny, S., Marquis, P., Ouali, M.: Weighted attacks in argumentation frameworks. In: KR (2012)
8. Dung, P.M.: On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. Artificial Intelligence 77, 321–357 (1995)
9. Dunne, P.E., Hunter, A., McBurney, P., Parsons, S., Wooldridge, M.: Inconsistency tolerance in weighted argument systems. In: AAMAS, pp. 851–858 (2009)
10. Grabisch, M.: The application of fuzzy integrals in multicriteria decision making. European Journal of Operational Research 89, 445–456 (1996)
11. Grabisch, M., Labreuche, C.: A decade of application of the Choquet and Sugeno integrals in multi-criteria decision aid. Annals of Operation Research 175, 247–286 (2010)
12. Kaci, S.: Refined preference-based argumentation frameworks. In: COMMA, pp. 299–310 (2010)
13. Kaci, S., Labreuche, C.: Arguing with valued preference relations. In: Liu, W. (ed.) ECSQARU 2011. LNCS, vol. 6717, pp. 62–73. Springer, Heidelberg (2011)

14. Kaci, S., van der Torre, L.: Preference-based argumentation: Arguments supporting multiple values. International Journal of Approximate Reasoning 48, 730–751 (2008)
15. Klement, E., Mesiar, R., Pap, E.: Triangular Norms. Kluwer, Dordrecht (2000)
16. Krantz, D.H., Luce, R.D., Suppes, P., Tversky, A.: Foundations of measurement. Additive and Polynomial Representations, vol. 1. Academic Press (1971)
17. Labreuche, C., Grabisch, M.: The Choquet integral for the aggregation of interval scales in multicriteria decision making. Fuzzy Sets & Systems 137, 11–26 (2003)
18. Marichal, J.-L.: An axiomatic approach of the discrete Choquet integral as a tool to aggregate interacting criteria. IEEE Tr. on Fuzzy Systems 8(6), 800–807 (2000)
19. Martínez, D.C., García, A.J., Simari, G.R.: An abstract argumentation framework with varied-strength attacks. In: KR, pp. 135–144 (2008)
20. Martínez, D.C., García, A.J., Simari, G.R.: Strong and weak forms of abstract argument defense. In: COMMA, pp. 216–227 (2008)
21. Nielsen, S.H., Parsons, S.: A generalization of dung's abstract framework for argumentation: Arguing with sets of attacking arguments. In: Maudet, N., Parsons, S., Rahwan, I. (eds.) ArgMAS 2006. LNCS (LNAI), vol. 4766, pp. 54–73. Springer, Heidelberg (2007)
22. Prakken, H.: A study of accrual of arguments, with applications to evidential reasoning. In: ICAIL, pp. 85–94 (2005)
23. Roth, A.: The college admissions problem is not equivalent to the marriage problem. Journal of Economic Theory 36, 277–288 (1985)
24. Simari, G.R., Loui, R.P.: A mathematical treatment of defeasible reasoning and its implementation. Artificial Intelligence 53, 125–157 (1992)

A Reasoning Platform Based on the MI Shapley Inconsistency Value

Sébastien Konieczny and Stéphanie Roussel

CRIL - CNRS, UMR 8188
Université d'Artois
62307 Lens, France

Abstract. In this paper we show how to build a reasoning platform using an inconsistency value. The idea is to use an inconsistency value for evaluating how much each formula of the belief base is responsible of the inconsistency of the base. Then this evaluation allows us to obtain a stratification (total pre-order) of the base, that can be used as the preferential input for different reasoning tasks, such as inference, belief revision, or conciliation. We show that the obtained operators are interesting and have good logical properties. We use as inconsistency value, the MI Shapley inconsistency value, that is known to have good properties, and that can be computed from minimal inconsistent subsets. We developed a java-based platform, that use the Sat4j library for computing the minimal inconsistent subsets, and that allows to have an effective way to compute the MI Shapley inconsistent subsets. We implemented also several inference, revision and conciliation methods, that use this inconsistency value. So this provides a complete reasoning platform, that can be used for instance for academic purposes.

1 Introduction

Belief change and reasoning under inconsistency are two topics that have received considerable attention. There are a lot of theoretical results on these reasoning methods, such as logical characterizations for non-monotonic inference [1,2], belief revision [3,4,5], belief merging [6,7,8], etc. There are also numerous particular methods that have been proposed for belief revision [9], belief merging [8], inference under inconsistency [10], etc.

In contrast, there are very few proposed implemented approaches. Although these implementations can be useful to test the proposed operators, to experiment the different reasoning method, and to disseminate these operators more widely in the AI community. In fact we are aware of only *two*[1] reasoning platforms, that implement several reasoning methods. The first one is the SATEN platform [12,13], developed by Williams and Sims, that allows to perform theory extraction, iterated belief revision, non-monotonic reasoning, possibilistic reasoning and hypothetical reasoning. This platform is written in Java 1.1 and

[1] We can mention also the QUIP project [11], but there is not yet, as far as we know, a corresponding available platform.

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 315–327, 2013.

is based on a theorem prover. It basically uses the Spohnian representation of epistemic states [14]. The second one is the COBA platform [15,16], developed by Delgrande, Liu, Schaub and Thiele, that performs belief revisions and contractions, based on the langage projection approaches developed in [17]. COBA is a Java applet that uses a SAT-solver.

In this work we propose such a reasoning platform. The whole platform is based on the effective computation of a given inconsistency value, namely the MI Shapley inconsistency value $S^{I_{MI}}$ [18], that can be computed easily from the minimal inconsistent subsets of a belief base. In addition to the measure of inconsistency of the formulae of the base and of the whole base given directly by this computation, we use the obtained stratification of the base (a total preorder) as the preferential input for different reasoning tasks, such as inference, belief revision, or conciliation.

The paper is organized as follows. Section 2 presents preliminary definitions and inconsistency measures and values. In sections 3, 4 and 5, we formally study three reasoning operations that are respectively inference, revision and conciliation. Section 6 is dedicated to the platform description. We finally conclude on future works in section 7.

2 Preliminaries - Inconsistency Measures and Values

We consider a propositional language $\mathcal{L}$ built from a finite set of propositional symbols $\mathcal{P}$. A *belief base* K is a finite set of propositional formulae. Let us note $\mathcal{K}_\mathcal{L}$ the set of belief bases definable from formulae of the language $\mathcal{L}$. If a belief base K is not consistent, then one can define the minimal inconsistent subsets [2] of K as: $\mathsf{MI}(K) = \{K' \subseteq K \mid K' \vdash \bot \text{ and } \forall K'' \subset K', K'' \nvdash \bot\}$.

The notion of maximal consistent subset[3] is the dual of that of minimal inconsistent subset. Each maximal consistent subset represents a maximal (regarding set inclusion) subset of the base that is consistent:

$\mathsf{MC}(K) = \{K' \subseteq K \mid K' \nvdash \bot \text{ and } \forall K'' \text{ s. t. } K' \subset K'', K'' \vdash \bot\}$

A profile Ψ is a vector of belief bases $\langle K_1, \ldots, K_n\rangle$. The set of all profiles is denoted $\mathcal{E}$. $\bigwedge \Psi$ denotes the conjunction of the elements of Ψ.

Recently some works have started to study how to measure the inconsistency in a propositional belief base (see e.g. [19]). There are several sensible ways to do that. This is not surprising since it parallels the fact that there are several sensible ways to define non-trivial inference relations from inconsistent bases.

In [18] a distinction has been made between *inconsistency measures*, that measure the inconsistency of a belief base, and *inconsistency values*, that measure the (responsibility for) inconsistency of each formula of a belief base.

Of course the inconsistency values, which work formula-by-formula, can be used to define corresponding inconsistency measures, just by aggregating the obtained inconsistency values.

Let us recall the definition of the Shapley Inconsistency Values (SIV) [18]:

[2] Also called Minimally Unsatisfiable Subsets - MUS.
[3] Also called Maximally Satisfiable Subsets - MSS.

Definition 1 ([18]). *An inconsistency measure I is called a* basic inconsistency measure *if it satisfies the following properties*[4], $\forall K, K' \in \mathcal{K}_{\mathcal{L}}$, $\forall \alpha, \beta \in \mathcal{L}$*:*

- $I(K) = 0$ *iff* K *is consistent* **(Consistency)**
- $I(K \cup K') \geq I(K)$ **(Monotony)**
- *If* α *is a free formula of* K*, then* $I(K) = I(K \setminus \{\alpha\})$ **(Free Formula Independence)**

Now we are able to define the Shapley inconsistency value.

Definition 2 ([18]). *Let I be a basic inconsistency measure. We define the corresponding* Shapley Inconsistency Value *(SIV), noted S^I, as the Shapley value of the coalitional game defined by the function I, i.e. let $\alpha \in K$:*

$$S^I_\alpha(K) = \sum_{C \subseteq K} \frac{(c-1)!(n-c)!}{n!}(I(C) - I(C \setminus \{\alpha\}))$$

where n is the cardinality of K and c is the cardinality of C.

From this value, one can define an inconsistency value for the whole belief base as in the next definition which essentially says that a base is as bad as its worst element.

Definition 3 ([18]). *Let K be a belief base,* $\hat{S}^I(K) = \max_{\alpha \in K} S^I_\alpha(K)$

As examples of simple basic inconsistency measures, one can consider the drastic inconsistency value[5], that is the simplest inconsistency measure one can define, and that is not really interesting by itself. But the corresponding SIV is interesting.

Another example of basic inconsistency measures is the one that counts the conflicts of a base using the number of minimal inconsistent subsets: $I_{MI}(K) = |\mathsf{MI}(K)|$.

The corresponding SIV is interesting, and has been logically characterized [18]. Let us first define these properties on inconsistency values: assume a given basic inconsistency measure I, and the corresponding Shapley inconsistency value S^I :

- $\sum_{\alpha \in K} S^I_\alpha(K) = I(K)$ **(Distribution)**
- If $\alpha, \beta \in K$ are such that for all $K' \subseteq K$ s.t. $\alpha, \beta \notin K'$, $I(K' \cup \{\alpha\}) = I(K' \cup \{\beta\})$, then $S^I_\alpha(K) = S^I_\beta(K)$ **(Symmetry)**
- If α is a free formula of K, then $S^I_\alpha(K) = 0$ **(Minimality)**
- If $|\mathsf{MI}(K_1 \cup \ldots \cup K_n)| = |\mathsf{MI}(K_1)| + \ldots + |\mathsf{MI}(K_n)|$, then $S^I_\alpha(K_1 \cup \ldots \cup K_n) = S^I_\alpha(K_1) + \ldots + S^I_\alpha(K_n)$ **(Decomposability)**
- If $M \in \mathsf{MI}(K)$, then $I(M) = 1$ **(MinInc)**

[4] In [18] an additional **Dominance** property is also asked.
[5] $I_d(K) = 0$ if K is consistent, and $I_d(K) = 1$ otherwise.

Proposition 1 ([18]). *An inconsistency value satisfies Distribution, Symmetry, Minimality, Decomposability and MinInc if and only if it is the MI Shapley Inconsistency Value* $S_\alpha^{I_{MI}}$.

Second, this value is equivalent to the following one:

Definition 4 ([18]). MIV_C *is defined as follows:*

$$MIV_C(K,\alpha) = \sum_{M \in \mathsf{MI}(K) s.t. \alpha \in M} \frac{1}{|M|}$$

Proposition 2 ([18]). $S_\alpha^{I_{MI}}(K) = MIV_C(K,\alpha)$

This alternative definition shows that this value can be computed directly if one knows the minimal inconsistent subsets of a belief base.

Example 1. Consider the base $K = \{\varphi_1, \ldots, \varphi_7\}$ with the following formulae $\varphi_1 = a \wedge b$, $\varphi_2 = a \wedge (c \vee d)$, $\varphi_3 = a \wedge \neg d$, $\varphi_4 = a \wedge \neg c \wedge e$, $\varphi_5 = \neg a \wedge \neg b$, $\varphi_6 = a \wedge (\neg c \rightarrow \neg e)$, $\varphi_7 = a \wedge \neg c \wedge f$. We have $S_{\varphi_1}^{I_{MI}} = \frac{1}{2}$, $S_{\varphi_2}^{I_{MI}} = \frac{7}{6}$, $S_{\varphi_3}^{I_{MI}} = \frac{7}{6}$, $S_{\varphi_4}^{I_{MI}} = \frac{4}{3}$, $S_{\varphi_5}^{I_{MI}} = 3$, $S_{\varphi_6}^{I_{MI}} = 1$, $S_{\varphi_2}^{I_{MI}} = \frac{5}{6}$.

Our reasoning platform is based on this computation of the $S_\alpha^{I_{MI}}$ Shapley value. We use this value for defining new inference relations, revision operators, and conciliation operators.

3 Inference Relations

When one wants to draw non-trivial inferences from an inconsistent propositional belief base then it has either to leave classical logic for choosing a paraconsistent logic, or to reason from the maximal consistent subsets of the base. We will focus on this last class of methods.

Unfortunately, there are not a lot of possibilities when the input is a simple propositional belief base. Let $K = \{\varphi_1, \ldots, \varphi_n\}$ be a belief base, and let $\mathsf{MC}(K) = \{M_1, \ldots M_k\}$ be the set of maximal consistent subsets of K. Then the three main possibilities are [10]:

- Skeptical: $K \vdash_s \varphi$ if $\forall M \in \mathsf{MC}(K)\ M \vdash \varphi$
- Credulous: $K \vdash_s \varphi$ if $\exists M \in \mathsf{MC}(K)\ M \vdash \varphi$
- Argumentative: $K \vdash_a \varphi$ if $\exists M \in \mathsf{MC}(K)\ M \vdash \varphi$ and $\nexists M \in \mathsf{MC}(K)\ M \vdash \neg\varphi$

The credulous inference is not that interesting, in particular it does not guarantee to obtain a consistent inference relation, in the sense that it is possible to obtain both φ and $\neg\varphi$ as result. So this leaves only two different possible inference relations: skeptical and argumentative.

Let us now show how to obtain a whole family of inference relation for each given inconsistency measure. The idea is to use the inconsistency measure to order the base, from the least inconsistent formulae to the most inconsistent

one. This means that we use the inconsistency measure to transform this flat propositional belief base into a stratified one. Then we can use any of the defined inference relations on stratified bases. We recall just the definition of the possibilisitic, linear and preferred inference relations here, see [10] for other ones and explanations. Consider a stratified belief base $\hat{K} = \langle K_1, \ldots, K_m \rangle$, where formulae in the stratum K_i are considered as more important/reliable/prioritary than the formulae in strata K_j with $j > i$.

- **possibilistic.** Define $\pi(\hat{K})$ as $\pi(\hat{K}) = K_1 \cup \ldots \cup K_i$ with $K_1 \cup \ldots \cup K_i$ consistent and $K_1 \cup \ldots \cup K_i \cup K_{i+1}$ inconsistent. $\hat{K} \vdash_\pi \varphi$ if $\pi(\hat{K}) \vdash \varphi$
- **linear.** Define $\lambda(\hat{K})$ inductively as: $\lambda(K_1) = K_1$ if K_1 is consistent, otherwise $\lambda(K_1) = \emptyset$. For i from 2 to m: if $\lambda(\{K_1, \ldots, K_{i-1}\}) \cup K_i$ is consistent then $\lambda(\{K_1, \ldots, K_i\}) = \lambda(\{K_1, \ldots, K_{i-1}\}) \cup K_i$, otherwise $\lambda(\{K_1, \ldots, K_i\}) = \lambda(\{K_1, \ldots, K_{i-1}\})$. $\hat{K} \vdash_l \varphi$ if $\lambda(\hat{K}) \vdash \varphi$
- **preferred.** Define $SMC(\hat{K})$ as the set of sets $A = A_1 \cup \ldots \cup A_m$ where $\forall i \in 1...m\ A_1 \cup \ldots \cup A_i \in \mathsf{MC}(K_1 \cup \ldots \cup K_i)$. $\hat{K} \vdash_p \varphi$ if $\forall X \in SMC(\hat{K})\ X \vdash \varphi$

So, let us now define formally our inference relations. First let us use an inconsistency value to stratify the base:

Definition 5. *Let $K = \{\varphi_1, \ldots, \varphi_n\}$ be a belief base, and V be an inconsistency value, then the stratification of K under V is the set of bases $K^V = \langle K_1, \ldots, K_m \rangle$ where*
- $\bigcup K_i = K$
- $K_i \cap K_j = \emptyset\ \forall i, j$
- $\forall \varphi \in K_i, \varphi' \in K_j$, $V(\varphi) \leq V(\varphi')$ *iff* $i \leq j$

Definition 6. *Let V be an inconsistency value, and $\vdash_A$ be an inference relation on stratified bases. The (V, A)-inference relation $\vdash_A^V$ is defined as $K \vdash_A^V \varphi$ if $K^V \vdash_A \varphi$.*

So, if the stratified inference relation that is used has good logical properties, it straightforwardly gives good properties to our (V, A)-inference relation. So as a consequence of results shown in [10] we know that:

Proposition 3. *Let V be any inconsistency value, then the (V, π)-inference relation, the (V, l)-inference relation, and the (V, p)-inference relation are preferential inference relations [2].*

Example 2. Consider the base of Example 1. The induced stratification of the base is $K^{S^{I_{MI}}} = \langle \{\varphi_1\}, \{\varphi_7\}, \{\varphi_6\}, \{\varphi_2, \varphi_3\}, \{\varphi_4\}, \{\varphi_5\}, \rangle$. And for instance we have $K \vdash_A^V a$ and $K \vdash_A^V \neg c \wedge \neg e$, whereas none of them can be inferred from the skeptical or the argumentative inference.

4 Revision Operators

Belief revision [3,4,9] aims at incorporating a new piece of information into the belief base of an agent. Often, this new piece of information conflicts with some

formulae of the belief base, so some of these formulae have to be removed from the base. One usually uses some preferential information for identifying priorities between formulae that can be preferably preserved. This can be encoded by a pre-order on formulae (such as in epistemic entrenchments [3]), by a pre-order between maximal consistent subsets (such as partial meet contraction functions [4]), by a pre-order between interpretations (such as in faithful assignments [9]), etc.

We propose to define this preferential information from an inconsistency measure. This measure is used to rank the maximal consistent subsets, and to select the best of them.

Definition 7. *Let $K = \{\varphi_1, \ldots, \varphi_n\}$ be a belief base, and φ be a formula. The set $K \perp \varphi$ is the set of sets X such that:*

- $X \subseteq K$
- $X \not\vdash \varphi$
- *There is no X' such that $X \subset X' \subseteq K \cup \{\varphi\}$ and $X' \not\vdash \varphi$*

Let us now define the score of a maximal consistent subset, given by the inconsistency values of its formulae.

Definition 8. *Let $K = \{\varphi_1, \ldots, \varphi_n\}$ be a belief base and φ be a formula. Let I be an inconsistency value. Then define the score of a formula φ_i as its inconsistency value for the base $K \cup \{\varphi\}$: $s_I(\varphi_i) = I_{\varphi_i}(K \cup \{\varphi\})$.*

And the score of a maximal consistent subset $X \in K \perp \varphi$ is the aggregated score of its formula: let g be an aggregation function, $s_{I,g}(X) = g_{\alpha \in X}(s_I(\alpha))$.

Definition 9. *Let $S = K \perp \varphi = \{X_1, \ldots, X_k\}$, a selection function for S is a function γ such that:*

- *If $X \perp \varphi$ is a non-empty set, then $\gamma(S)$ is a non-empty subset of S.*
- *If $X \perp \varphi$ is empty, then $\gamma(X \perp \varphi)$ is empty*

A score-based selection function $\gamma_{I,g,f}$, generated by the inconsistency measure I, the aggregation function g and the selection function f is such that $\gamma_{I,g,f}(S) = argmin_{X_i \in S} f(s_{I,g}(X_i))$.

min should be usually chosen for f, in order to select only the best results, but one could for instance want to obtain not only the MC with best (minimal) scores, but also close-to-the best ones, as for instance the 50% best ones. This is why we define f as an additional parameter.

Let $A = \{A_1, \ldots A_m\}$ be a set, then $A \oplus \alpha$ denotes the set $\{A_1 \cup \{\alpha\}, \ldots A_m \cup \{\alpha\}\}$.

Definition 10. *The* MC *operator $\star_{\mathsf{MC}}$ is defined as $K \star_{\mathsf{MC}} \varphi = \gamma(K \perp \neg\varphi) \oplus \varphi$.*

The score-based MC *operator $\star_{I,g,f}$ is defined as $K \star_{I,g,f} \varphi = \gamma_{I,g,f}(K \perp \neg\varphi) \oplus \varphi$.*

This defines the result of a revision as a set of belief bases. Then one has to choose an inference policy from this set. In the following we will focus on skeptical inference, but other policies can be used:

Definition 11. $K \star_{\mathsf{MC}} \varphi \vdash \alpha$ *if* $\forall B \in \gamma(K\perp\neg\varphi) \oplus \varphi$, $B \vdash \alpha$.

For belief base revision (i.e. when the base is not closed deductively, as opposed to belief sets), Hansson [5] defines the result of the revision as the conjunction of the intersection of all the selected remainder sets with the new piece of information ($\cap\gamma(K\perp\neg\varphi) \cup \varphi$), but this conjunction removes too much information, as illustrated in the next example, so we prefer to keep the full set of possible results as defined above.

Example 3. Consider the base $K = \{a \wedge c, b \wedge c\}$ and the formula $\varphi = \neg a \vee \neg b$. So $K\perp\varphi = \{\{a \wedge c\}, \{b \wedge c\}\}$. Suppose that $\gamma = id$, so $\cap\gamma(K\perp\neg\varphi) = \emptyset$, so it is not possible to infer c from $K \star \varphi$, whereas from the set $K \star_{\mathsf{MC}} \varphi = \{\{a \wedge c, \neg a \vee \neg b\}, \{b \wedge c, \neg a \vee \neg b\}\}$ it is possible to infer c.

So this gives a little more complicated definition, but it allows to obtain more inferences.

We will focus on the $\star_{S^{I_{MI}},max,min}$ operator using the MI Shapley inconsistency value, the max as aggregation function g, and the min as selection function f.

Let us illustrate the behavior of this operator on the following:

Example 4. Consider the base $K = \{a\wedge c, b\wedge c, b\wedge d\}$ and the formula $\varphi = \neg a\vee\neg b$.

So $K\perp\varphi = \{\{a\wedge c\}, \{b\wedge c, b\wedge d\}\}$. As $s_{S^{I_{MI}},max}(\{a\wedge c\}) = 1$, and $s_{S^{I_{MI}},max}(\{b\wedge c, b \wedge d\}) = 0.5$, with $f = min$ only $\{b \wedge c, b \wedge d\}$ is selected, so the result is a singleton set: $K \star_{S^{I_{MI}},max,min} \varphi = \{b \wedge c, b \wedge d, \neg a \vee \neg b\}$.

Let us now translate usual AGM belief revision basic properties [4,3] in this framework:

(K*1) $K * \alpha$ is a theory
(K*2) $K * \alpha \vdash \alpha$
(K*3) $K * \alpha \subseteq K \cup \{\alpha\}$
(K*4) Si $\neg\alpha \notin K$, alors $K \cup \{\alpha\} \subseteq K * \alpha$
(K*5) $K * \alpha = K_\perp$ iff $\vdash \neg\alpha$
(K*6) Si $\vdash \alpha \leftrightarrow \beta$, alors $K * \alpha = K * \beta$

Of course we work in a syntactic (not deductively closed) approach, so (K*1) should not be satisfied. But all other basic revision properties are satisfied:

Proposition 4. *The* MC *operators* $\star_{\mathsf{MC}}$ *satisfy (K*2), (K*3), (K*4), (K*5), (K*6).*

5 Conciliation Operators

Conciliation operators allow to solve the conflicts between a set of belief bases. The idea is to select the most problematic bases, to weaken them, and to iterate this process until there is no conflict left. We first define belief game models [20], that allow to obtain a conflict-free profile. Then the corresponding conciliation operator is just the conjunction of the obtained profile.

Definition 12 ([20]). *A* choice function *is a function $g : \mathcal{E} \to \mathcal{E}$ such that:*

- $g(\Psi) \sqsubseteq \Psi$
- *If* $\bigwedge \Psi \not\equiv \top$, *then* $\exists \varphi \in g(\Psi)$ *s.t.* $\varphi \not\equiv \top$
- *If* $\Psi \equiv \Psi'$, *then* $g(\Psi) \equiv g(\Psi')$

Definition 13 ([20]). *A* weakening function *is a function $\blacktriangledown : \mathcal{L} \to \mathcal{L}$ such that:*

- $\varphi \vdash \blacktriangledown(\varphi)$
- *If* $\varphi \equiv \blacktriangledown(\varphi)$, *then* $\varphi \equiv \top$
- *If* $\varphi \equiv \varphi'$, *then* $\blacktriangledown(\varphi) \equiv \blacktriangledown(\varphi')$

Definition 14 ([20]). *The solution to a belief profile Ψ for a Belief Game Model $\mathcal{N} = \langle g, \blacktriangledown \rangle$ under the integrity constraints μ, is the belief profile $\Psi^{\mu}_{\mathcal{N}}$ defined as:*

- $\Psi_0 = \Psi$
- $\Psi_{i+1} = \blacktriangledown_{g(\Psi_i)}(\Psi_i)$
- $\Psi^{\mu}_{\mathcal{N}}$ *is the first* Ψ_i *that is consistent with* μ

The conciliation operator $\blacktriangle_{\mathcal{N}}$ is defined as $\Psi \blacktriangle_{\mathcal{N}} \mu = \bigwedge \Psi^{\mu}_{\mathcal{N}}$

Definition 15 ([20]). *Let φ be a belief base.*

- *The* drastic weakening function *forgets all the information about that agent, i.e. : $\forall \varphi \ \blacktriangledown_{\top}(\varphi) = \top$.*
- *The* dilation weakening function *is defined as :*

$$mod(\blacktriangledown_{\delta}(\varphi)) = \{\omega \in \mathcal{W} \mid \exists \omega' \models \varphi \ d_H(\omega, \omega') \leq 1\}$$

where d_H is the Hamming distance between interpretations[6]*.*

In this work we use the inconsistency value for defining the most conflicting agents (i.e. the selection function).

Definition 16. *A* Shapley Belief Game Model *is a Belief Game Model $\mathcal{N} = \langle S_I, \blacktriangledown \rangle$, where S_I is a Shapley Inconsistency Value.*

The solution to a belief profile Ψ for a Shapley Belief Game Model $\mathcal{N} = \langle S_I, \blacktriangledown \rangle$ under the integrity constraints μ, is the belief profile $\Psi^{\mu}_{\mathcal{N}}$ defined as:

- $\Psi_0 = \Psi$
- $\Psi_{i+1} = \blacktriangledown_{argmax_{\varphi_j \in \Psi_i}(S_{I_{\varphi_j}}(\Psi_i))}(\Psi_i)$
- $\Psi^{\mu}_{\mathcal{N}}$ *is the first* Ψ_i *that is consistent with* μ

Example 5. Consider the Shapley Belief Game Model $\mathcal{N} = \langle S_{I_{LP_m}}, \blacktriangledown_{\delta} \rangle$. There are seven agents $\Psi = \{\varphi_1, \ldots, \varphi_7\}$ with the following belief bases $\varphi_1 = a \wedge b$, $\varphi_2 = a \wedge (c \vee d)$, $\varphi_3 = a \wedge \neg d$, $\varphi_4 = a \wedge \neg c \wedge e$, $\varphi_5 = \neg a \wedge \neg b$, $\varphi_6 = a \wedge (\neg c \rightarrow \neg e)$, $\varphi_7 = a \wedge \neg c \wedge f$. There are no integrity constraints ($\mu = \top$). We have $S^{I_{MI}}_{\varphi_1} = \frac{1}{2}$,

[6] Let ω and ω' be two interpretations, then $d_H(\omega, \omega') = \#(\{a \in \mathcal{P} \mid \omega(a) \neq \omega'(a)\})$.

$S^{I_{MI}}_{\varphi_2} = \frac{7}{6}$, $S^{I_{MI}}_{\varphi_3} = \frac{7}{6}$, $S^{I_{MI}}_{\varphi_4} = \frac{4}{3}$, $S^{I_{MI}}_{\varphi_5} = 3$, $S^{I_{MI}}_{\varphi_6} = 1$, $S^{I_{MI}}_{\varphi_2} = \frac{5}{6}$. The maximal value is 3, meaning that φ_5 is the agent that brings the most conflicts, and so it is selected by the choice function for weakening. So φ_5 is replaced by $\top$. We have not yet reached a consistent profile, so we must do a further round. Then the new computations of inconsistency values give φ_4 as the most conflictual agent, and it is weakened to $\top$. The profile is still not consistent, so a third round is needed. In this third round φ_2, φ_3 and φ_7 are weakened. The resulting (consistent) profile for the whole process is then: $\Psi^{\top}_{\mathcal{N}} = \{\{a \wedge b\}, \top, \top, \top, \top, a \wedge (\neg c \rightarrow \neg e), \top\}$. So $\Psi \blacktriangle_{\mathcal{N}} \top \equiv a \wedge b \wedge (\neg c \rightarrow \neg e)$.

6 Platform Description

In this section, we describe the PRISM (Platform for Reasoning with Inconsistency Shapley Measure) platform, that we have built in order to test the different operators presented in the previous section. One can use this platform to build a base and perform the different reasoning tasks such as inference, revision and conciliation. PRISM can be downloaded from the following page : `http://www.cril.univ-artois.fr/prism`. This page also contains pieces of information and detailed documentation about the platform. In the following, we present features and details about the implementation.

6.1 Features

The platform is available as a Java application. The user interface is divided into 5 main tabs:

1. *Base* - allows the user to create a base of formulae
2. *Shapley* - computes the Shapley value of each formulae of the base
3. *Inference* - allows to reason given some inference relation
4. *Revision* - allows to reason on the base revised by a new formula
5. *Conciliation* - computes a consistent base from a set of (conflicting) bases

All tabs are structured in the same way. The top-left part of the panel represents the belief base currently used. Depending on the task that has to be performed, the top-right part displays the operators options. The bottom part displays the result of the computations.

Base Tab. The belief base is composed of several formulae. These formulae can either be loaded from a file or be directly written by the user. Accepted formulae have the following syntax :

φ := (φ) ; φ & φ ; φ | φ ; φ -> φ ; φ <-> φ ; *lit*
lit := v ; $\sim v$ *where* v *is a variable name*

Each formula must end with a semicolon ";". Almost all alphanumeric strings are accepted for variable names[7].

[7] The exceptions are the symbols in the following list : $'$, $-$, &, |, $\sim$, (,), <, > and ;. Strings "_nv#i" where i is an integer are also reserved.

Bases can be saved and loaded into the platform. Formulae can be viewed in CNF and can be added, modified or removed from the base.

It is also possible to group formulae. Groups can for instance represent agents to which the formulae belong. In practice, a group is represented by an integer. We distinguish a specific group that is identified by 0 : the constraints group. Formulae belonging to this group are constraints, i.e. formulae that cannot be falsified. These specific formulae can encode background knowledge for inference or revision, and integrity constraints for conciliation. The user can choose to take the groups into account or not.

In the following, we detail tabs of the platform. Each tab includes default operators but it is possible to define its own operator (see table 1).

Shapley Tab. Once a base is available, Shapley values of the formulae composing it can be computed. Details of this computation are available on the Shapley tab. More precisely, the MI are displayed at the bottom left panel. Shapley values are displayed at the bottom right panel. If group classification is taken into account, then the Shapley measures for each group is also displayed. Note that the operator for aggregating an inconsistency measure for a group of formulae can be chosen at the top right panel. By default, two operators are available : *Mean* and *Max*.

Inference Tab. Shapley values can be used to stratify belief bases: the lower the value, the higher in the layers of the base. The belief base stratification is displayed on the inference tab. The user can choose an inference operator from the list and ask whether a formula can be entailed from the base and the chosen operator.
The formula must follow the same syntax as presented previously for the base. The default inference operators are named *Possibilistic* and *Linear* and correspond to the ones presented in section 3.

Revision Tab. This tab allows to revise the belief base by some formula. The formula must be given in the field just under the table representing the base. The syntax must be the same as the one described previously for new formulae of the base. As presented in section 4, revision operator is compounded of different sub-operators : an aggregation function, a selection function and an inference policy. The default aggregation functions are *Max*, *Min* and *Sum*. See table 1 for classes details. The platform proposes three selection functions: *No Selection*, *All Min Selection* and *One of the min selection*. The last one arbitrary takes one MC into account. Finally, the last option is the inference policy. The user can initially choose between *Skeptical* and *Credulous* inferences but, once again, it is possible to implement its own inference policy. All MC are displayed in the bottom left panel, along with their respective score. The table in the bottom right panel shows selected MC. And, as in the revision tab, it is possible to ask whether a formula can be entailed according to the chosen type of inference.

Conciliation Tab. This tab allows to test the conciliation (belief game model) as it has been presented in section 5. This game is based on two operators: a choice operator and a weakening operator. The first choice operator is named *Shapley Choice* and corresponds to $S_{I_{MI}}$. The second operator, *Weak Shapley Choice*, selects randomly one formula amongst the ones selected by the Shapley operator. This allows to reduce the number of weakened formulae. The default weakening operator is named *Drastic* and corresponds to operator $\blacktriangledown_{\top}$ defined in section 5.

6.2 Implementation Details

PRISM is an evolutive platform, i.e. for all the reasoning tasks, one can add its own implementation. Main conditions are to create a class that extends a specific one and to add this class to the classpath[8]. The following table indicates for each operator the abstract class to extend and the main method to implement.

Table 1. Classes and methods to implement in order to add new operators

Tab	Abstract Class	Method
Shapley	ShapleyValueSet	computeShapleyValue(List<Formula> l) : **double**
Inference	InferenceOperator	isAFormulaEntailed(List<Formula> b, Formula f): **boolean**
Revision	MssScoreAggregation	computeMSScore(MSS m) : **double**
	MssSelectionOperator	selectMss(List<MSS> l) : List<MSS>
	InferenceFromMSSOperator	isAFormulaEntailed(List<MSS> l, Formula f) : **boolean**
Conciliation	ChoiceOperator	chooseFormulae(List<Formula> l) : List<Formula>
	WeakOperator	weakFormula(Formula f) : Formula

Assigning a Shapley value to a formula of a base necessitates the computation of all the MI that can be derived from the belief base. In the general case, computing MI is intractable. First, the number of MI can be exponential: a n-clauses SAT instance can exhibit $C_n^{n/2}$ MI in the worst case. Then, checking whether a formula belongs to the set of MI is in Σ_2^p [21]. Our problem is even more difficult since we want to compute all MI and check whether formulae belong to them.

Many approaches have been proposed to extract one MI from a set of clauses ([22,23,24] and many others) or to compute MI covers ([25]). We need here a complete approach, i.e. one that extracts all MI. Candidate tools are thus less numerous: CAMUS [26], HYCAM [27].

We choose here to perform the MI extraction with the Sat4j SAT solver [28]. The main reason for this choice is our willingness to develop a platform independent tool. Using Java based technology is a way to preserve this property, even if we don't get the benefit of the last advances in the extraction af all MI (all dedicated tools cited previously are developed in C or C++). In further versions

[8] Details on how to add a new operator can be found on the online documentation of the platform: `http://www.cril.univ-artois.fr/prism`.

of PRISM, a detection of the running platform will be made in order to allow the use of CAMUS or HYCAM.

This Sat4j solver (2.3.3 release) extracts all MI in a two steps method [29]: all MC of the base are first computed, then MI are obtained through a second pass.

For all the MI extractors cited previously, the input is a CNF formula. Our platform allows the user to populate the belief base with general formulae. This means that we have to transform given formulae into equisatisfiable CNF formulae. To perform this transformation, we use the Tseitin encoding ([30]). This encoding results in a formula with a linear size increase in expense of the addition of new variables[9].

For a given formula, clauses composing its CNF form are grouped when given to the solver. This allows us to maintain equivalence between CNF and formulae.

In order to model general formulae, we have used a domain specific language (dsl) written in Scala. This allows us to have a very efficient parsing and CNF transformation. Moreover, since scala is built on top of the JVM, we preserve the platform independency.

7 Conclusion and Future Works

In this paper, we propose an evolutive platform that uses the MI Shapley inconsistency value to perform different reasoning tasks such as inference, revision and conciliation. Although operators are already implemented for each of these operations, one can write its own implementation and add it dynamically to the platform to test it. The platform is Java-based, which brings us full operating system independency.

In future works, on top of developing more operators, we plan to propose part of this platform as a Java library. Such a library would provide methods to use the MI Shapley inconsistency value and its associated operators for various applications.

References

1. Makinson, D.: General Pattern in nonmonotonic reasoning. In: Handbook of Logic in Artificial Intelligence and Logic Programming, vol. III, pp. 35–110. Clarendon Press, Oxford (1994)
2. Kraus, S., Lehmann, D., Magidor, M.: Nonmonotonic reasoning, preferential models and cumulative logics. Artificial Intelligence 44, 167–207 (1990)
3. Gärdenfors, P.: Knowledge in flux. MIT Press (1988)
4. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. Journal of Symbolic Logic 50, 510–530 (1985)
5. Hansson, S.O.: A Textbook of Belief Dynamics: Theory Change and Database Updating. Kluwer (1999)
6. Revesz, P.Z.: On the semantics of arbitration. International Journal of Algebra and Computation 7, 133–160 (1997)

[9] New variables resulting from the encoding have the reserved names "_nv#i".

7. Konieczny, S., Pino Pérez, R.: Merging information under constraints: a qualitative framework. Journal of Logic and Computation 12, 773–808 (2002)
8. Konieczny, S., Pino Pérez, R.: Logic based merging. Journal of Philosophical Logic 40, 239–270 (2011)
9. Katsuno, H., Mendelzon, A.O.: Propositional knowledge base revision and minimal change. Artificial Intelligence 52, 263–294 (1991)
10. Benferhat, S., Dubois, D., Prade, H.: Some syntactic approaches to the handling of inconsistent knowledge bases: A comparative study. Part ii: The prioritized case. In: Orlowska, E. (ed.) Logic at Work, vol. 24, pp. 473–511. Physica-Verlag, Heidelberg (1998)
11. Egly, U., Eiter, T., Tompits, H., Woltran, S.: Solving advanced reasoning tasks using quantified boolean formulas. In: Proceedings of AAAI 2000, pp. 417–422 (2000)
12. Williams, M.A., Sims, A.: Saten: An object-oriented web-based revision and extraction engine. CoRR cs.AI/0003059 (2000)
13. `http://magic.it.uts.edu.au/systems/saten.html`
14. Spohn, W.: Ordinal conditional functions: a dynamic theory of epistemic states. In: Harper, W.L., Skyrms, B. (eds.) Causation in Decision, Belief Change, and Statistics, vol. 2, pp. 105–134 (1987)
15. Delgrande, J.P., Liu, D.H., Schaub, T., Thiele, S.: COBA 2.0: A consistency-based belief change system. In: Mellouli, K. (ed.) ECSQARU 2007. LNCS (LNAI), vol. 4724, pp. 78–90. Springer, Heidelberg (2007)
16. `http://www.cs.sfu.ca/~cl/software/COBA/coba2.html`
17. Delgrande, J.P., Schaub, T.: A consistency-based approach for belief change. Artificial Intelligence 151, 1–41 (2003)
18. Hunter, A., Konieczny, S.: On the measure of conflicts: Shapley inconsistency values. Artificial Intelligence 174, 1007–1026 (2010)
19. Hunter, A., Konieczny, S.: Approaches to measuring inconsistent information. In: Bertossi, L., Hunter, A., Schaub, T. (eds.) Inconsistency Tolerance. LNCS, vol. 3300, pp. 191–236. Springer, Heidelberg (2005)
20. Konieczny, S.: Belief base merging as a game. Journal of Applied Non-Classical Logics 14, 275–294 (2004)
21. Eiter, T., Gottlob, G.: On the complexity of propositional knowledge base revision, updates, and counterfactuals. Artificial Intelligence 57, 227–270 (1992)
22. Belov, A., Lynce, I., Marques-Silva, J.: Towards efficient mus extraction. AI Commun. 25, 97–116 (2012)
23. Bruni, R.: On exact selection of minimally unsatisfiable subformulae. Annals of Mathematics and Artificial Intelligence 43, 35–50 (2005)
24. Dershowitz, N., Hanna, Z., Nadel, A.: A scalable algorithm for minimal unsatisfiable core extraction. In: Biere, A., Gomes, C.P. (eds.) SAT 2006. LNCS, vol. 4121, pp. 36–41. Springer, Heidelberg (2006)
25. Grégoire, E., Mazure, B., Piette, C.: Tracking muses and strict inconsistent covers. In: Proceedings of FMCAD 2006, San Jose, USA, pp. 39–46 (2006)
26. Liffiton, M., Sakallah, K.: Algorithms for computing minimal unsatisfiable subsets of constraints. Journal of Automated Reasoning 40, 1–33 (2008)
27. Grégoire, E., Mazure, B., Piette, C.: Using local search to find msses and muses. European Journal of Operational Research 199, 640–646 (2009)
28. Le Berre, D., Parrain, A.: The sat4j library, release 2.2. JSAT 7, 56–59 (2010)
29. Castell, T., Cayrol, C., Cayrol, M., Berre, D.L.: Using the davis and putnam procedure for an efficient computation of preferred models. In: Proceedings of ECAI 1996, pp. 350–354 (1996)
30. Tseitin, G.S.: On the complexity of derivations in the propositional calculus. Studies in Mathematics and Mathematical Logic Part II, 115–125 (1968)

Most Inforbable Explanations: Finding Explanations in Bayesian Networks That Are Both Probable *and* Informative

Johan Kwisthout

Radboud University Nijmegen, Donders Institute for Brain, Cognition and Behaviour, Montessorilaan 3, 6525 HR Nijmegen, The Netherlands
j.kwisthout@donders.ru.nl

Abstract. The problems of generating candidate hypotheses and inferring the best hypothesis out of this set are typically seen as two distinct aspects of the more general problem of non-demonstrative inference or abduction. In the context of Bayesian networks the latter problem (computing most probable explanations) is well understood, while the former problem is typically left as an exercise to the modeler. In other words, the candidate hypotheses are pre-selected and hard-coded. In reality, however, non-demonstrative inference is rather an interactive process, switching between hypothesis generation, inference to the best explanation, evidence gathering and deciding which information is relevant. In this paper we will discuss a possible computational formalization of finding an explanation which is both probable and as informative as possible, thereby combining (at least some aspects of) both the 'hypotheses-generating' and 'inference' steps of the abduction process. The computational complexity of this formal problem, denoted Most Inforbable Explanation, is then established and some problem parameters are investigated in order to get a deeper understanding of what makes this problem intractable in general, and under which circumstances the problem becomes tractable.

1 Introduction

Inference to the best explanation is a well-known and well-studied computational problem in Bayesian networks. When "best" is operationalized as "most probable" (as is typically the case in the Bayesian network community, but see, e.g., [11] for alternative notions) it is commonly known as MAP[1]: given a partition of a Bayesian network into an *evidence* set with observed variables, a set of *explanation* variables which together constitute candidate hypotheses, and a set of *intermediate* variables that fall in neither category, compute the most probable joint

[1] Also *Partial* or *Marginal* MAP to distinguish the problem from the more constrained MPE problem, in which the variables of the graph are bi-partitioned in evidence variables and hypothesis variables and no marginalization over other variables is needed.

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 328–339, 2013.

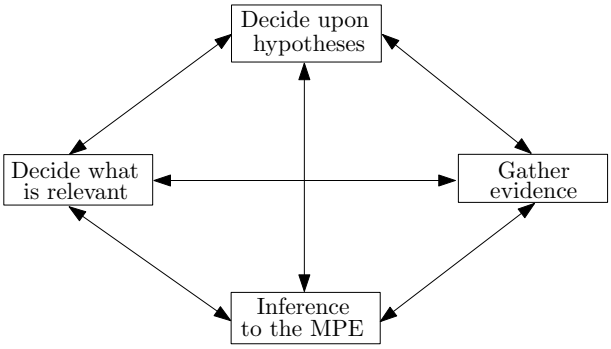


Fig. 1. In everyday problem solving the selection of hypotheses, determining upon relevant information, gathering evidence, and inference to the most probable explanation are concurrent (rather than sequential) and highly connected sub-tasks of the broader *abduction* problem

value assignment to the explanation variables. This computational problem has been studied from an engineering [17] and computational complexity [5,14,22] point of view, and exact and approximate algorithms for MAP are available in abundance [3,6,7,20,21,22,25]. However, the *abduction* or *non-demonstrative inference problem* is broader and more complex than 'merely' solving a MAP problem. It is a heavily intertwined combination of deciding which are the relevant variables, deciding upon candidate hypotheses, evidence gathering, and inference to the most probable explanation (Fig. 1).

Clinical examination (i.e., diagnosing the patient) is an excellent example of such an abduction process, consisting of hypothesis generation, obtaining evidence, evaluating hypotheses, and determining throughout this process what of all the available information is relevant to diagnosing (and preferably curing) the patient; see, e.g., [19] and in particular the highly illustrative case study on page 26-27. Some observations and findings may not be relevant to the diagnosis. The clinician needs to decide which are to be taken into account and which are not. Often, symptoms and signs come in patterns; for example, polyuria, polydipsia, and polyphagia are well known symptoms for diabetes mellitus. Clustering or lumping such observations may benefit hypothesis generation towards a diagnosis. On the other hand, the clinician may miss important aspects in doing so: There is a high probability that orthostatic hypotension is caused by vomiting and diarrhea. Thus, they could be lumped together as cause and effect. In so doing, however, the clinician is at risk of excluding a completely separate and important problem, namely, extracellular volume depletion.

During this process, initial hypotheses are generated and evidence is gathered and judged. Based on the evidence and the posterior probabilities of these initial hypotheses, additional evidence may be gathered and the hypotheses may be further refined, eventually leading to a diagnosis and possibly a treatment procedure. These "real world" aspects of abduction problems, as illustrated in

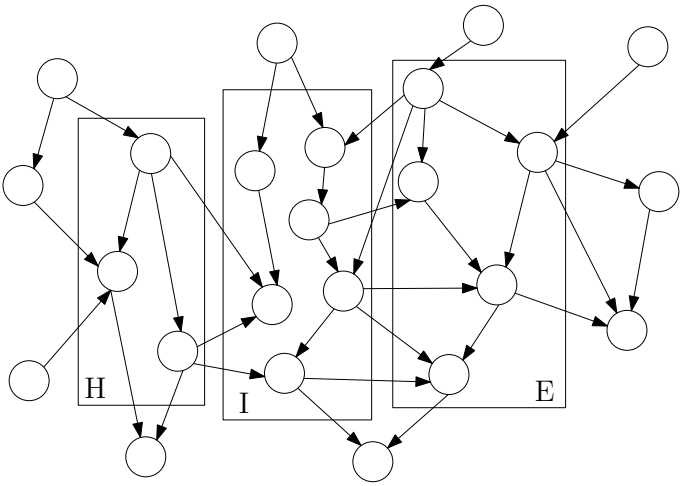


Fig. 2. Partitioning the domain model into hypotheses variables **H**, evidence variables **E**, intermediate variables **I**, and irrelevant or "outside" variables that are not part of the model can be graphically depicted as *establishing boundaries* ("drawing boxes") within a knowledge structure

the above example, are typically not part of the computational problem: they are 'left to the modeler'. Instinctively, this modeling process can be seen as *establishing boundaries* in a knowledge structure such as a Bayesian network (Fig. 2). In this process, numerous decisions need to be made, such as which nodes in the knowledge structure can be dismissed as being irrelevant to the goal or how detailed the explanation should be. These choices are driven by the goal of the abduction process: what counts as a candidate hypotheses or as a relevant variables is determined by what we seek to explain; see for example [8] and [18, Ch. 3, and the references therein].

1.1 Granularity of Explanations

A correct, but hardly informative, explanation of the signs "shortness of breath, coughing with phlegm, and pain while breathing" will be "patient-X is ill". This explanation has (by definition) a higher probability (say 0.95) than the much more informative explanation "patient-X has pneumonia" (say 0.8). The latter explanation of course has more explanatory power at the cost of little probability mass, and thus will, in general, be preferred over the former although this explanation has a higher probability.

This trade off between information and probability is known as the Inverse Relationship Principe [1]: the more *specific* an explanation is, the lower its probability will be. From a mathematical point of view, this may be trivial: surely, $\Pr(A) \leq \Pr(B)$ if $A \subseteq B$. However, in practical situations, there can be many situation-specific circumstances that may determine whether a more specific

explanation is needed. While a general practitioner will need an explanation that is specific enough to successfully describe medication, a project manager needs only a general explanation why one of her team members won't be at his desk for some time. Sometimes it might be costly or impractical to determine more specific explanations. The impact of making the *wrong* decision may be crucial in determining the probability threshold; what risks are we willing to accept?

In this paper, we seek to combine two aspects of the abduction problem into one computational formalism: choosing what to explain (and at which granularity) and inference to the most probable explanation. This computational problem of seeking an explanation which is both *informative* enough for our means and has a high enough *probability* is denoted as the MOST INFORBABLE EXPLANATION problem to emphasize the trade off between informativeness and probability. The remainder of this paper is structured as follows. In the next section we will offer some needed preliminaries on Bayesian networks and computational complexity theory. In Section 3 we formally define MOST INFORBABLE EXPLANATION. We discuss the computational complexity of a decision variant of MOST INFORBABLE EXPLANATION in Section 4. In Section 5 we conclude the paper.

2 Preliminaries

In this section, we give a short overview of a number of concepts from Bayesian networks, graph theory, and complexity theory, in particular definitions of probabilistic networks and treewidth, some background on complexity classes defined by Probabilistic Turing Machines and oracles, and fixed-parameter tractability. For a more thorough discussion of these concepts, the reader is referred to textbooks like [9,10,12,23].

2.1 Bayesian Networks

A Bayesian or probabilistic network $\mathcal{B} = (\mathbf{G}_\mathcal{B}, \Pr)$ is a graphical structure that models a set of stochastic variables, the conditional independences among these variables, and a joint probability distribution over these variables. $\mathcal{B}$ includes a directed acyclic graph $\mathbf{G}_\mathcal{B} = (\mathbf{V}, \mathbf{A})$, modeling the variables and conditional independences in the network, and a set of parameter probabilities Pr in the form of conditional probability tables (CPTs), capturing the strengths of the relationships between the variables. The network models a joint probability distribution $\Pr(\mathbf{V}) = \prod_{i=1}^{n} \Pr(V_i \mid \pi(V_i))$ over its variables, where $\pi(V_i)$ denotes the parents of V_i in $\mathbf{G}_\mathcal{B}$. We will use upper case letters to denote individual nodes in the network, upper case bold letters to denote sets of nodes, lower case letters to denote value assignments to nodes, and lower case bold letters to denote joint value assignments to sets of nodes.

One of the key computational problems in Bayesian networks is the problem to find the most probable explanation for a set of observations, i.e., the joint value assignment to a designated set of variables that has highest posterior probability

given the observed variables in the network. If the network is bi-partitioned into explanation variables and evidence variables this problem is known as MOST PROBABLE EXPLANATION, however, in practice there will often be variables that are neither observed nor to be explained; for example, variables that influence the posterior probability distribution but whose value is impractical or even impossible to observe. In that case, the problem is denoted (PARTIAL) MAP (or MARGINAL MAP, to emphasize that we need to marginalize over the unobserved variables); the decision variant of this problem is defined as follows:

MAP
Instance: A probabilistic network $\mathcal{B} = (\mathbf{G}_\mathcal{B}, \Pr)$, where $\mathbf{V}$ is partitioned into a set of evidence nodes $\mathbf{E}$ with a joint value assignment $\mathbf{e}$, a set of intermediate nodes $\mathbf{I}$, and an explanation set $\mathbf{H}$; a rational number $0 \leq q < 1$.
Question: Is there a joint value assignment $\mathbf{h}$ to $\mathbf{H}$ such that $\Pr(\mathbf{h}, \mathbf{e}) > q$?

MAP is NP-hard under a wide range of constraints, both to compute exact and to approximate [22,14,5,15].

An important structural property of a probabilistic network is its *treewidth*. Treewidth is a graph-theoretical concept, which can be loosely described as a measure on the *'localness'* of the dependencies in the network: when the variables tend to be clustered in small groups with few connections between groups, treewidth is typically low, whereas treewidth tends to be high if the connections between variables are scattered all over the network. Formally, the treewidth of a Bayesian network $\mathcal{B}$ is defined as the minimum width over all tree-decompositions of triangulations of the moralization $\mathbf{G}^{\mathbf{M}}_\mathcal{B}$ of the network [24]. Treewidth plays an important role in the complexity analysis of Bayesian networks, as many otherwise intractable computational problems become tractable when the treewidth of the network is bounded.

2.2 Computational Complexity Theory

In the remainder, we assume that the reader is familiar with basic concepts of computational complexity theory, such as Turing Machines, the complexity classes P and NP, and NP-completeness proofs. In addition to these basic concepts, to describe the complexity of various problems we will use the *probabilistic* class PP, oracles, and some aspects from parameterized complexity theory.

The class PP contains languages L accepted in polynomial time by a *Probabilistic Turing Machine*. Such a machine augments the more traditional nondeterministic Turing Machine with a probability distribution associated with each state transition. Acceptance of an input x is defined as follows: the probability of arriving in an *accept state* is strictly larger than $\frac{1}{2}$ if and only if $x \in L$. This probability of acceptance, however, is not fixed and may (exponentially) depend on the input, e.g., a problem in PP may accept 'yes'-instances with size $|x|$ with probability $\frac{1}{2} + \frac{1}{2^{|x|}}$. PP-complete problems are considered to be intractable. The canonical PP-complete problem is MAJSAT: given a Boolean formula ϕ, does the majority of the truth assignments satisfy ϕ? In Bayesian networks, the canonical

problem of determining whether the probability $\Pr(\mathbf{H} = \mathbf{h} \mid \mathbf{E} = \mathbf{e}) > q$ for a given rational q (known as the INFERENCE problem) is PP-complete [4,13].

A Turing Machine $\mathcal{M}$ has *oracle access* to languages in the class C, denoted as $\mathcal{M}^{\mathsf{C}}$, if it can "query the oracle" in one state transition, i.e., in $\mathcal{O}(1)$. We can regard the oracle as a 'black box' that can answer membership queries in constant time. For example, $\mathsf{NP}^{\mathsf{PP}}$ is defined as the class of languages which are decidable in polynomial time on a non-deterministic Turing Machine with access to an oracle deciding problems in PP.

Sometimes problems are intractable (i.e., NP-hard) in general, but become tractable if some *parameters* of the problem can be assumed to be small. Informally, a problem is called fixed-parameter tractable for a parameter k (or a set $\{k_1, \ldots, k_n\}$ of parameters) if it can be solved in time, exponential *only* in k and polynomial in the input size $|x|$, i.e., in time $\mathcal{O}(f(k) \cdot |x|^c)$ for a constant c and an arbitrary function f. In practice, this means that problem instances can be solved efficiently, even when the problem is NP-hard in general, if k is known to be small.

3 Most Inforbable Explanations

In the MAP problem, one seeks to find the joint value assignment to a set of variables that has maximum posterior probability. Here the candidate solutions consist of joint value assignments to exactly that set of variables, i.e., a conjunction of value assignments $\{(H_1 = h_1) \wedge \ldots \wedge (H_n = h_n)\}$ to the individual variables of the explanation set. This assumes that both the candidate hypotheses and the granularity of the explanation are set beforehand.

In real life, however, candidate hypotheses are formed and considered during the inference process, and the granularity of the explanation varies. Let us assume there is evidence that a patient suffers from a lung disease. On examination, when further evidence becomes available, the diagnosis may be refined to an obstructive lung disease, and later on, even further refined to the more specfic COPD and finally chronic bronchitis (Fig. 3). Preferably, we would like to find an explanation that has high probability and is specific, like $\{(\text{CB} = \text{TRUE}) \wedge (\text{EM} = \text{FALSE}) \wedge \ldots \wedge (\text{LP} = \text{FALSE})\}$, denoting that the patient has chronic bronchitis and no other lung disease is present. But what if there is not enough evidence to clearly distuinguish between chronic bronchitis and emphysema? Would it be wise to ignore the possibility of other lung diseases being present (maybe altering the advised medication) if the probability of their presense is maybe not convincing, but still non-neglectable?

Let us consider the three cases as presented in Table 1. In case a), the explanation is as specific as possible and has a high probability: the patient suffers from chronic bronchitis and no other lung disease is present. Case b) reflects that no clear distinction between chronic bronchitis and emphysema could be made. Note, however, that the probability of the three joint value assignments that correspond with $\{((\text{CB} = \text{TRUE}) \vee (\text{EM} = \text{TRUE})) \wedge \ldots \wedge (\text{LP} = \text{FALSE})\}$ is high. Here, it seems best to restrict the diagnosis to "COPD", rather than to refine it

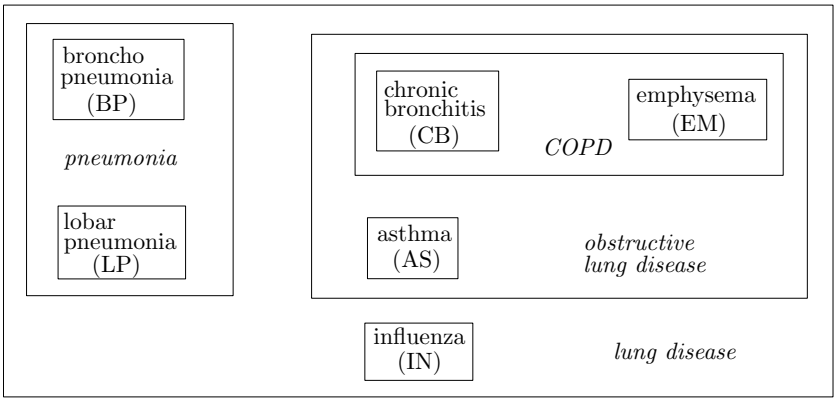


Fig. 3. Example of part of a classification of lung diseases

further. In case c) the patient definately suffers from chronic bronchitis, but in addition, some form of pneumonia may be present. Here, it would be wise (if no further evidence can be gathered) to settle for the diagnosis "chronic bronchitis, and maybe also pneumonia" and describe medication that covers both.

Table 1. Joint value assignments and their probabilities in the *lung disease* example

case	BP	LP	CB	EM	AS	IN	prob.
a	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	0.87
	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	0.04
	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	0.02
	other						0.07
b	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	0.48
	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	0.37
	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	0.10
	other						0.05
c	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	0.48
	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	0.21
	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	0.17
	TRUE	TRUE	TRUE	FALSE	FALSE	FALSE	0.08
	other						0.06

What we did in case a) corresponds to 'plain' MAP. In case b) and c), however, we choose as explanation a *set of joint value assignments* rather than a singleton joint value assignment, namely the set that corresponds to the (informal) diagnoses "COPD" (case b), respectively "chronic bronchitis, and maybe also pneumonia" (case c). Or to put it more formally, the sets of joint

value assignments that correspond to the sentences $\{((\text{CB} = \text{TRUE}) \vee (\text{EM} = \text{TRUE})) \wedge (\text{AS} = \text{FALSE}) \wedge (\text{IN} = \text{FALSE}) \wedge (\text{BP} = \text{FALSE}) \wedge (\text{LP} = \text{FALSE})\}$, respectively $\{(\text{CB} = \text{TRUE}) \wedge (\text{EM} = \text{FALSE}) \wedge (\text{AS} = \text{FALSE}) \wedge (\text{IN} = \text{FALSE})\}$. Thus, we extended MAP to deal with sets of joint value assignments, each consisting of a conjunction of value assignments to the variables in the explanation set[2].

We can also use a possible world semantics to describe these explanations. In case a) the explanation corresponds to the world where CB is TRUE and all other variables are FALSE. In case b) the explanation corresponds to the worlds where either CB or EM or both are set to TRUE, and all other variables are FALSE. In case c) the set of possible worlds are those where CB is TRUE, BP and LP are either TRUE or FALSE, and the other variables are FALSE. If we count these worlds in the three cases, we see that there is a single world in case a) with probability 0.87, there are three worlds in case b) whose probabilities add up to 0.95, and there are four worlds in case c) with total probability 0.94. Thus, in order to gain probabibility mass, in case b) and c) we needed to trade off informativeness, where we define explanation H to be more informative than H' if H corresponds to fewer possible worlds than H'.

3.1 Succinct Encodings

We saw that the formal definition of "chronic bronchitis, and maybe also pneumonia", which corresponds to four possible worlds in the *lung disease* example, can be quite succinctly described as $\{(\text{CB} = \text{TRUE}) \wedge (\text{EM} = \text{FALSE}) \wedge (\text{AS} = \text{FALSE}) \wedge (\text{IN} = \text{FALSE})\}$ because the values of BP and LP are "don't cares". Surely, not every combination of four possible worlds can be described so easily, and we may need to resort to a full enumeration of four joint value assignments to describe that explanation.

That feels quite unnatural and unsatisfactory, in the sense that such an explanation (that consists of an arbitrarily complex sentence over the values of the variables) does not appear to be very informative at first sight. The sentence $(\text{AS} = \text{TRUE})$ corresponds to 32 possible worlds in which the patient has asthma (without committing to a particular value of the other variables), yet this is far more comprehensible and informative than a plain enumeration of, say, 11 possible worlds, so despite being "less informative" given the possible worlds semantics, we would like to enforce some reasonable encoding that makes the explanation easy to understand and to reason with. But there are also complexity-theoretic reason to constrain how the explanation should be encoded: if we allow the explanation to be encoded as an arbitrary set of w possible worlds, where w is given as a binary number, we may need an exponential (in w) number of bits to describe that explanation. Therefore, apart from high probability and a low number of possible worlds, we also require that the sentence describing

[2] Observe that we assume binary variables here for ease of exposition, but we might also include variables with a higher cardinality, like TEMP with values $\{\text{low}, \text{normal}, \text{high}\}$, stating, e.g., $\{((\text{TEMP} = \text{low}) \vee (\text{TEMP} = \text{normal})) \wedge \ldots\}$.

these possible worlds is short, i.e., we also demand succinct encodings. To be precise, we require that the explanation can be encoded by the addition of at most $\hat{w} = \mathcal{O}(\lceil \log_2(w+1) \rceil)$ *partial* joint value assignments to subsets of the explanation set.

We finish this section with an informal problem definition of Most Inforbable Explanation, combining these three requirements:

Most Inforbable Explanation (informal)
Instance: A Bayesian network, partitioned into evidence nodes, explanation nodes, and intermediate variables.
Output: An explanation that has high probability, corresponds to few possible worlds, and is succinctly encodable.

4 Computational Complexity

To investigate the computational complexity of Most Inforbable Explanation, we will formally define a decision variant of this problem as follows.

Most Inforbable Explanation
Instance: A Bayesian network $\mathcal{B} = (\mathbf{G}, \Pr)$, where $\mathbf{V}$ is partitioned into a set of evidence nodes $\mathbf{E}$ with a joint value assignment $\mathbf{e}$, an explanation set $\mathbf{H}$, and intermediate variables $\mathbf{I}$; a rational number $0 \leq q < 1$ and a natural number w.
Question: Is there a set $\{\mathbf{h_1}, \ldots, \mathbf{h_w}\}$ of w distinct joint value assignments $\mathbf{h_1}, \ldots, \mathbf{h_w}$ to $\mathbf{H}$, encodable by the addition of at most $\hat{w} = \mathcal{O}(\lceil \log_2(w+1) \rceil)$ joint value assignments $\mathbf{h'}$ to subsets of $\mathbf{H}$, such that $\sum_{i=1}^{w} \Pr(\mathbf{h_i}, \mathbf{e}) = \sum_{j=1}^{\hat{w}} \Pr(\mathbf{h'_j}, \mathbf{e}) > q$?

Theorem 1. Most Inforbable Explanation *is* $\mathsf{NP}^{\mathsf{PP}}$*-complete.*

Proof. We prove membership in $\mathsf{NP}^{\#\mathsf{P}}$, membership in $\mathsf{NP}^{\mathsf{PP}}$ follows as $\mathsf{P}^{\#\mathsf{P}} = \mathsf{P}^{\mathsf{PP}}$. Membership can be shown by non-deterministically guessing a certificate, consisting of a set of at most $\hat{w}$ joint value assignments $\mathbf{h'}$ to subsets of $\mathbf{H}$; checking that this certificate yields at most w distinct joint value assignments to $\mathbf{H}$; computing, using the $\#\mathsf{P}$ oracle, $\sum_{j=1}^{\hat{w}} \Pr(\mathbf{h'_j}, \mathbf{e})$ (note that $\#\mathsf{P}$ is closed under addition), and finally deciding whether $\sum_{j=1}^{\hat{w}} \Pr(\mathbf{h'_j}, \mathbf{e}) > q$. Note that the number of joint value assignments w may grow exponentially in the input size, as w is encoded in binary notation, but that all three steps of the verification algorithm can be done in polynomial time given the constraint that $\{\mathbf{h_1}, \ldots, \mathbf{h_w}\}$ must be succinctly (i.e., logarithmically in w) encodable. Note that $\mathsf{NP}^{\mathsf{PP}}$-hardness follows since Most Inforbable Explanation has MAP as a special case: take $w = 1$. □

If $w = 0$ then Most Inforbable Explanation degenerates to Inference. If $w = 1$ then Most Inforbable Explanation degenerates to MAP. Furthermore,

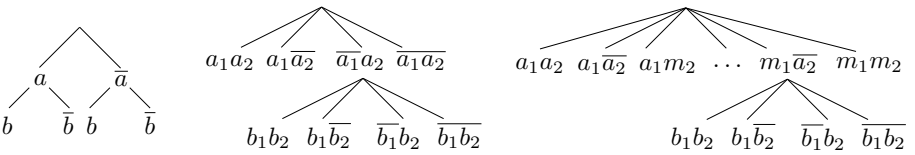


Fig. 4. The fp-tractable MAP algorithm branches on each value assignment to the variables in $\mathbf{H}$, computing marginal distributions over the variables not in $\mathbf{H}$; in the left subfigure the example $\mathbf{H} = \{A, B\}$, $c = 2$ is illustrated. The size of the branching tree is bounded by the probability of the most probable joint value assignment. Here, we extend this algorithm by branching over all possible value assignments in all possible worlds: part of the branching tree for $w = 2$ is drawn in the middle subtree. In the right subtree part of the branching tree for $\hat{w} = 2$ is drawn. Here, for each choice of $\hat{w}$, a variable can take any of its values, or it can take no value at all, denoted with m to illustrate that we marginalize over that variable rather than assign it a value

---

Most Inforbable Explanation inherits the inapproximability results of MAP [22]. MAP is fixed parameter tractable (fp-tractable) for $\{c, 1-p, \text{tw}\}$, i.e., MAP can be solved fast when the treewidth tw of the restricted junction tree and cardinality c of the variables are small *and* the most probable explanation has a high probability[3] ($1-p$ is low) [2,14]. However, this may not hold for Most Inforbable Explanation since we need to choose w joint value assignments out of maximally $c^{|\mathbf{H}|}$ which by itself is a source of complexity. However, the $\{c, 1-p, \text{tw}\}$-fixed-parameter tractable algorithm[4] for MAP can be adjusted by branching on each of the (at most) c^w combinations of values for each variable, rather than on each of the (at most) c values (see Fig. 4). Therefore, Most Inforbable Explanation is fp-tractable for $\{c, 1-p, \text{tw}, w\}$. Since Most Inforbable Explanation is a generalization of both MAP (for $w = 1$) and Inference (for $w = 0$) it follows that Most Inforbable Explanation remains intractable for the set of parameters $\{c, 1-p, w\}$ and $\{c, \text{tw}, w\}$ [16,5].

It can be shown that Most Inforbable Explanation is also fp-tractable for $\{c, 1-p, \text{tw}, \hat{w}\}$, i.e., instead of bounding the number of possible worlds, we bound the size of the encoding. This can be done by further augmenting the above-mentioned algorithm, allowing it to branch on the (at most) $c^{\hat{w}} + c^{\hat{w}-1} + \ldots + 1$ combinations of values and non-assigned variables (that are marginalized over) — see again Fig. 4. Thus, from a computational point of view, Most Inforbable Explanation is not harder than 'plain' MAP, as both are $\mathsf{NP}^{\mathsf{PP}}$-complete. However, to render Most Inforbable Explanation fixed-parameter tractable, an additional constraint needs to be imposed on wither the number of possible worlds w or the number of (partial) joint value assignments $\hat{w}$ encoding these worlds.

[3] Technically speaking, $1-p$ is not a parameter as it is not a natural number; however, it can be mapped one-to-one to a suitable natural parameter [14].

[4] See [2] for the original algorithm for Most Probable Explanation, and [14] for the augmented algorithm for MAP.

5 Conclusion

In this paper, we introduced Most Inforbable Explanation as an extention to MAP, in order to combine both inference to the best explanation and (some aspects of) selecting candidate hypotheses and determining the granularity of the explanations. In human reasoning, the sets h_i are not likely to be arbitrarily chosen, but may correspond to common phrases as "either A or B, or both", "maybe A, but definitely not B", or "likely A, and possibly also B"; simple heuristics may exist that favor such phrases in practice and penalizing more complex structures, thus enforcing the formal logarithmic bound introduced in the formal definition and the fpt-result for $\hat{w}$. A succint encoding of "Asthma, but also at least one other disease" (spanning 31 possible worlds in the example) may be $\{(\text{AS} = \textsc{true})\} \setminus \{(\text{BP} = \textsc{false}) \wedge (\text{LP} = \textsc{false}) \wedge (\text{CB} = \textsc{false}) \wedge (\text{EM} = \textsc{false}) \wedge (\text{IN} = \textsc{false})\}$. We did not include such encodings (allowing for substraction, as well as addition, of partial joint value assignments) as it is not obvious that the above mentioned algorithm is fp-tractable in this case.

A particularly interesting aspect of informativeness of explanations lies in the often *contrasting* nature of explanations: often, we do not simply want to explain: *'Why this?'*, but *'Why this, rather than that?'* [18]. For example, to explain why Alice got tenure, referring to her quality teaching is unsufficient when Bob is an excellent teacher as well, but happend to be denied tenure: a better explanation would (also) refer to her many high-rated publications that Bob lacked. We leave a formal study of how such aspects may be implemented in a computational problem for future work.

Acknowledgments. The author wishes to thank Iris van Rooij, Linda van der Gaag, and Pim Haselager for helpful discussions and literature suggestions.

References

1. Barwise, J.: Information and impossibilities. The Notre Dame Journal of Formal Logic 38(4), 488–515 (1997)
2. Bodlaender, H.L., van den Eijkhof, F., van der Gaag, L.C.: On the complexity of the MPA problem in probabilistic networks. In: Proceedings of the 15th European Conference on Artificial Intelligence, pp. 675–679 (2002)
3. Charniak, E., Shimony, S.E.: Cost-based abduction and MAP explanation. Artificial Intelligence 66(2), 345–374 (1994)
4. Darwiche, A.: Modeling and Reasoning with Bayesian Networks. Cambridge University Press (2009)
5. De Campos, C.P.: New complexity results for MAP in Bayesian networks. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, pp. 2100–2106 (2011)
6. de Campos, L., Gamez, J., Moral, S.: Partial abductive inference in Bayesian belief networks using a genetic algorithm. Pattern Recognition Letters 20(11-13), 1211–1217 (1999)
7. de Campos, L., Gamez, J., Moral, S.: Partial abductive inference in Bayesian belief networks by simulated annealing. International Journal of Approximate Reasoning 27(3), 263–283 (2001)

8. de Campos, L., Gámez, J., Moral, S.: Simplifying explanations in Bayesian belief networks. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 9(4), 461–489 (2001)
9. Downey, R.G., Fellows, M.R.: Parameterized complexity. Springer, Berlin (1999)
10. Garey, M.R., Johnson, D.S.: Computers and Intractability. In: A Guide to the Theory of NP-Completeness, W. H. Freeman and Co., San Francisco (1979)
11. Glass, D.H.: Inference to the best explanation: a comparison of approaches. In: Second Symposium on Computing and Philosophy (2009)
12. Jensen, F.V., Nielsen, T.D.: Bayesian Networks and Decision Graphs, 2nd edn. Springer, New York (2007)
13. Kwisthout, J.: The Computational Complexity of Probabilistic Networks. PhD thesis, Faculty of Science, Utrecht University, The Netherlands (2009)
14. Kwisthout, J.: Most probable explanations in Bayesian networks: Complexity and tractability. International Journal of Approximate Reasoning 52(9), 1452–1469 (2011)
15. Kwisthout, J.: Structure approximation of most probable explanations in Bayesian networks. In: Proceedings of the 24th Benelux Conference on Artificial Intelligence, BNAIC 2012 (2012)
16. Kwisthout, J.: The computational complexity of probabilistic inference. Technical Report ICIS–R11003, Radboud University Nijmegen (2011)
17. Lacave, C., Díez, F.J.: A review of explanation methods for Bayesian networks. The Knowledge Engineering Review 17(2), 107–127 (2002)
18. Lipton, P.: Inference to the best explanation. Routledge (2004)
19. Nardone, D.A.: Collecting and analyzing data: Doing and thinking. In: Walker, H.K., Hall, W.D., Hurst, J.W. (eds.) Clinical Methods: The History, Physical, and Laboratory Examinations, ch. 2, 3rd edn., Butterworths, Boston (1990)
20. Park, J.D., Darwiche, A.: Approximating MAP using local search. In: Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence, pp. 403–410. Morgan Kaufmann Publishers, San Francisco (2001)
21. Park, J.D., Darwiche, A.: Solving MAP exactly using systematic search. In: Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence (UAI 2003), pp. 459–468. Morgan Kaufmann (2003)
22. Park, J.D., Darwiche, A.: Complexity results and approximation settings for MAP explanations. Journal of Artificial Intelligence Research 21, 101–133 (2004)
23. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, Palo Alto (1988)
24. Robertson, N., Seymour, P.D.: Graph minors II: Algorithmic aspects of tree-width. Journal of Algorithms 7, 309–322 (1986)
25. Yuan, C., Lu, T., Druzdzel, M.J.: Annealed MAP. In: Proceedings of the Twentieth Conference in Uncertainty in Artificial Intelligence, pp. 628–635. AUA (2004)

Structure Approximation of Most Probable Explanations in Bayesian Networks

Johan Kwisthout

Radboud University Nijmegen, Donders Institute for Brain, Cognition and Behaviour, Montessorilaan 3, 6525 HR Nijmegen, The Netherlands
j.kwisthout@donders.ru.nl

Abstract. Typically, when one discusses approximation algorithms for (NP-hard) problems (like Traveling Salesperson, Vertex Cover, Knapsack), one refers to algorithms that return a solution whose *value* is (at least ideally) close to optimal; e.g., a tour with almost minimal length, a vertex cover of size just above minimal, or a collection of objects that has close to maximal value. In contrast, one might also be interested in approximation algorithms that return solutions that *resemble* the optimal solutions, i.e., whose *structure* is akin to the optimal solution, like a tour that is almost similar to the optimal tour, a vertex cover that differs in only a few vertices from the optimal cover, or a collection that is similar to the optimal collection. In this paper, we discuss structure-approximation of the problem of finding the most probable explanation of observations in Bayesian networks, i.e., finding a joint value assignment that *looks like* the most probable one, rather than has an *almost as high value*. We show that it is NP-hard to obtain the value of *just a single variable* of the most probable explanation. However, when partial orders on the values of the variables are available, we can improve on these results.

1 Introduction

A key computational problem in Bayesian networks [17] is the computation of the *most probable explanation* (MPE) of a set of observed phenomena; i.e., given a Bayesian network whose variables are partitioned into an *evidence* set $\mathbf{E}$ with observed joint value assignment $\mathbf{e}$ and an *explanation* set $\mathbf{M}$, determine the joint value assignment $\mathbf{m}$ to the explanation set $\mathbf{M}$ such that $\Pr(\mathbf{M} = \mathbf{m}, \mathbf{E} = \mathbf{e})$ is maximal. This problem, also called Bayesian abduction, is a key component in many decision support systems like [15,21], in many Bayesian models of cognition, for example intention recognition [2] or recipient design [22], as well as in various models of sociological [19] or economical [8] processes.

Unfortunately, computing the MPE is in general NP-hard [12,3,18] and remains NP-hard when the most probable explanation is to be *approximated* rather than exactly computed. In particular, it is NP-hard to find a joint value assignment whose probability is within a fixed ratio of the most probable joint value assignment [1] and it is even NP-hard to find a joint value assignment that has a

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 340–351, 2013.

non-zero probability [12]. However, these formal notions of approximation focus on the *value* of the explanation, i.e., the goal is to find an explanation whose *probability* is 'close' to the probability of the most probable explanation. Sometimes we may not be primarily interested in finding explanations with an almost-as-high probability, but rather in explanations that are *quite similar* to the most probable explanation, that is, they *look like* the most probable explanation. For example, in cognitive science, one's goal is to describe, model, and predict human cognition. In such applications it is conceivable that we are most interested in approximating structure, rather than value [16]; we will refer to this notion of approximation as *structure approximation* (note that the term 'structure' does *not* refer to the graphical structure (i.e., the arcs) of the network, but to the structure of the joint value assignments).

Preferably, of course, in many domains we would like to have an approximation that both resembles the optimal solution *and* have an almost-as-high probability [4]. While it may well be the case that 'good' value approximations sometimes have a similar structure as the optimal solution, this need not be the case, as we will show in Subsection 2.3.

Structure approximation has its roots in computational complexity theory [11,6]. The relevance of structure approximation, in particular in the context of the so-called Coherence Problem, was first suggested by Millgram [16] and extensively studied in Hamilton et al. [9] and Van Rooij et al. [23]. In this paper we further build on this work and discuss structure approximations of MPE. In the remainder of this paper, we will discuss some relevant preliminaries and definitions in Bayesian networks and structure approximation in Section 2. In Section 3 we focus on structure-approximating MPE. We discuss the computational complexity of structure approximation of MPE in general in Subsection 3.1, and the effect of having an *ordering* of the variables in Subsection 3.2. In Section 4 we conclude this paper.

2 Preliminaries

In this section we introduce Bayesian networks and, more in particular, the problem of finding the most probable explanation (MPE) for a subset of variables in the network, given observations for the other variables. For more background, the reader is referred to textbooks as [17,10] and overview papers as [14,12]. Furthermore, we introduce a formal definition of structure approximation, as presented in [9]. We assume that the reader is familiar with basic notions in complexity theory, such as the classes P and NP and NP-hardness proofs; for more background, we refer to [7].

2.1 Bayesian Networks and the MPE Problem

A Bayesian or probabilistic network $\mathcal{B}$ is a graphical structure that models a set of stochastic variables, the conditional independencies among these variables, and a joint probability distribution over these variables. $\mathcal{B}$ includes a directed

acyclic graph $\mathbf{G}_\mathcal{B} = (\mathbf{V}, \mathbf{A})$, modeling the variables and conditional independencies in the network, and a set of parameter probabilities Pr in the form of conditional probability tables (CPTs), capturing the strengths of the relationships between the variables. The network models a joint probability distribution $\Pr(\mathbf{V}) = \prod_{i=1}^{n} \Pr(V_i \mid \pi(V_i))$ over its variables, where $\pi(V_i)$ denotes the parents of V_i in $\mathbf{G}_\mathcal{B}$. We will use upper case letters to denote individual nodes in the network, upper case bold letters to denote sets of nodes, lower case letters to denote value assignments to nodes, and lower case bold letters to denote joint value assignments to sets of nodes. We will use $\mathbf{E}$ to denote a set of evidence nodes, i.e., a set of nodes for which a particular joint value assignment $\mathbf{e}$ is observed; likewise, we will use $\mathbf{M}$ to denote a set of nodes for which the explanation is sought. We will sometimes write $\Pr(\mathbf{x})$ as a shorthand for $\Pr(\mathbf{X} = \mathbf{x})$ if no ambiguity can occur. We denote with $\Omega(X)$ the set of all values that X can take; $\Omega(\mathbf{X})$ is defined analogously for sets of variables.

Among other computational problems defined on Bayesian networks, one particularly interesting problem for many applications is the problem of determining the *most probable explanation* for some observations, i.e., the most probable joint value assignment to a subset of variables in the network, given evidence for the other variables[1]. This problem is formally defined as follows [12].

MPE
Instance: A probabilistic network $\mathcal{B} = (\mathbf{G}_\mathcal{B}, \Pr)$, where $\mathbf{V}$ is partitioned into a set of evidence nodes $\mathbf{E}$ with a joint value assignment $\mathbf{e}$, and an explanation set $\mathbf{M}$.
Output: $\text{argmax}_\mathbf{m} \Pr(\mathbf{m}, \mathbf{e})$, i.e., the most probable joint value assignment $\mathbf{m}$ to the nodes in $\mathbf{M}$ and evidence $\mathbf{e}$, or the designated symbol $\perp$ if $\Pr(\mathbf{m}, \mathbf{e}) = 0$ for every joint value assignment $\mathbf{m}$ to $\mathbf{M}$.

MPE is intractable in general; to be precise, the problem is $\mathsf{FP}^{\mathsf{NP}}$-complete and has an NP-complete decision variant [12,18].

2.2 Structure Approximation

The notion of a structure approximation is typically captured using a *solution distance function*, a metric associated with each optimization problem relating candidate solutions with the optimal solution [9]. Let Π be a optimization problem with instance x, let $cansol(x)$ denote a function returning candidate solutions to x, with $optsol(x)$ denoting a function returning the *optimal* solution[2] to x. For any $y, y' \in cansol(x)$, let $d(y, y')$ be the distance between y and y' as defined by d. As d is a metric, the following properties hold for all $a, b, c \in cansol(x)$:

[1] If we have only partial evidence, i.e., the network is partitioned into variables for which the explanation is sought, evidence variables, and other variables that constitute neither evidence nor explanation, then the problem generalized to a Partial (or Marginal) MAP problem. The intractability results presented here generalize also to Partial MAP.

[2] Or, in case of a draw, one of the optimal solutions.

1. $d(a, a) = 0$
2. if $a \neq b$, $d(a, b) > 0$
3. $d(a, b) = d(b, a)$
4. $d(a, b) + d(b, c) \geq d(a, c)$

Typically, for many problems Π, d might correspond to the *Hamming distance* or *edit distance* between two candidate solutions: the number of elements in which the candidate solutions differ, or the number of operations needed to transform one candidate solution into another. We define a h/d-structure approximation of Π as follows:

Definition 1 ([9]). *Given an optimization problem Π, a solution-distance function d, and a non-decreasing function $h : \mathbb{N} \to \mathbb{N}$, an algorithm A is a polynomial-time h/d-structure approximation algorithm if for every instance x of Π, $d(A(x), optsol(x)) \leq h(|x|)$, and A runs in time polynomial in $|x|$.*

Similarly, we define an *expected* h/d-structure approximation of Π as follows:

Definition 2. *Given an optimization problem Π, a solution-distance function d, and a non-decreasing function $h : \mathbb{N} \to \mathbb{N}$, an algorithm A is a polynomial-time expected h/d-structure approximation algorithm if, for a random instance x of Π, the expected distance $E(d(A(x), optsol(x))) \leq h(|x|)$, and A runs in time polynomial in $|x|$.*

2.3 Value versus Structure Approximation

Possibly counter to intuition, a "good" value approximation is not necessarily a "good" structure approximation and vice versa. As an example, consider the Bayesian network in Figure 1 with binary variables $V, X_1, \ldots, X_n$, a uniform probability distribution for the variables X_1 to X_n, and the following conditional probability distribution for V:

$$\Pr(V = \text{TRUE} \mid X_1, \ldots, X_n) = \begin{cases} 1 & \text{if } \forall_i X_i = \text{TRUE} \\ 1 - \epsilon & \text{if } \forall_i X_i = \text{FALSE} \\ 0 & \text{otherwise} \end{cases}$$

Note that the most probable explanation for the observation $V = \text{TRUE}$ would be the explanation where all variables X_i are set to TRUE, and the second most probable explanation where all variables X_i are set to FALSE. Any non-zero value approximation thus would yield an explanation with a completely different structure than the most probable explanation. On the other hand, any explanation that has a similar structure (i.e., differ in only few variables) would have a probability of zero.

3 Structure Approximation of MPE

Let $cansol(\mathcal{B}, \mathbf{e})$ denote the set of explanations (i.e., joint value assignments to $\mathbf{M}$) of a Bayesian network $\mathcal{B}$ with observed evidence $\mathbf{e}$, with $optsol(\mathcal{B}, \mathbf{e})$ as the

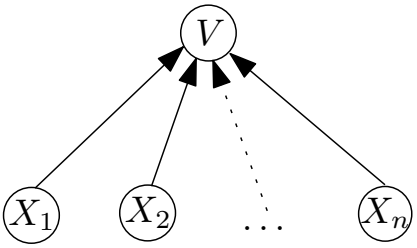


Fig. 1. Example network with distinct structure and value approximations

most probable explanation, i.e., the joint value assignment to $\mathbf{M}$ with the highest joint probability. We define the structure distance function $d_H(\mathbf{m}, optsol(\mathcal{B}, \mathbf{e}))$ as the Hamming distance between explanation $\mathbf{m} \in cansol(\mathcal{B}, \mathbf{e})$ and the most probable explanation.

In the remainder of this paper, we consider h to be a function taking an MPE instance $x = \{\mathcal{B}, \mathbf{e}\}$ and returning a distance. With $h(x)/d_H$-structure-approximate-MPE, we define the problem of finding a structure approximation that differs in *at most* $h(x)$ variables from the most probable explanation $optsol(\mathcal{B}, \mathbf{e})$. With $E(h(x))/d_H$-structure-approximate-MPE we define the problem of finding a joint value assignment that has an *expected* Hamming distance $h(x)$ to $optsol(\mathcal{B}, \mathbf{e})$, i.e., a structure approximation is sought that differs *on average* in at most $h(x)$ variables from the MPE.

3.1 Computational Complexity

In this section we will discuss the computational complexity of structure approximations of MPE. Note that a *random guess* of the values of variables would return a value assignment which gives an *expected* Hamming distance $h(x) = |\mathbf{M}| - \frac{|\mathbf{M}|}{c}$, with c as the cardinality of the (unobserved) variables. In particular, when all unobserved variables are binary, we can expect to guess half of them correctly.

Corollary 1. *MPE is* $E(h(x))/d_H$*-structure approximable for* $h(x) = |\mathbf{M}| - \frac{|\mathbf{M}|}{c}$.

We cannot expect to do better than chance: given that it is NP-hard to $\frac{n}{2} - \epsilon/d_H$-structure approximate 3SAT [6] and we can reduce 3SAT to MPE in polynomial time while preserving the structure of the certificates (by a simple variant of the proof used in [12, p.1457], which is omitted here for reasons of space), any polynomial-time $|\mathbf{M}| - \frac{|\mathbf{M}|}{c} - \epsilon/d_H$-structure approximation algorithm for MPE could be used to find a $\frac{n}{2} - \epsilon/d_H$-structure approximation of any 3SAT instance in polynomial time.

Lemma 1. *MPE is* $h(x)/d_H$*-structure inapproximable for* $h(x) = |\mathbf{M}| - \frac{|\mathbf{M}|}{c} - \epsilon$, *unless* $\mathsf{P} = \mathsf{NP}$.

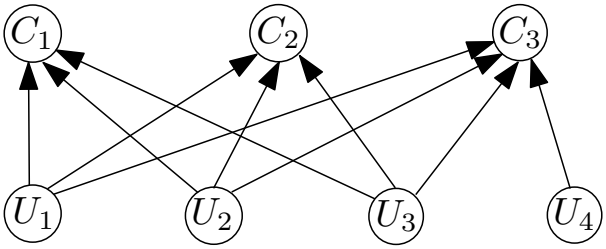


Fig. 2. Construction of $\mathcal{B}_{\phi_{\text{ex}}}$ from ϕ_{ex}

This result holds for binary variables with indegree at most three[3]. Here, we allow the approximation algorithm to select the $h(x)$ variables. If we are allowed to *designate* the variables for which the value is sought, then it is easy to see that we cannot have a polynomial-time structure approximation algorithm A for MPE, even for a single variable, unless $\mathsf{P} = \mathsf{NP}$, as we could use A consecutively for all $|\mathbf{M}|$ unobserved variables of $\mathcal{B}$ and thus obtain a polynomial-time exact algorithm for MPE; as MPE is NP-hard, the result follows as a corollary. However, we can prove a much stronger result for networks with three values per variable and indegree at most six: There cannot exist an algorithm that tells[4] us the value of an *arbitrary* single variable, unless $\mathsf{P} = \mathsf{NP}$:

Theorem 1. *No algorithm can calculate the value of one of the variables in the most probable explanation in polynomial time, unless* $\mathsf{P} = \mathsf{NP}$.

We will prove Theorem 1 with a reduction from 3SAT, defined as follows.

3-CNF SATISFIABILITY (3SAT)
Instance: A Boolean formula $\phi = (U, C)$ in 3-CNF form, with variables $U = u_1, \ldots, u_n$ and literals $C = c_1, \ldots, c_m$.
Question: Does there exist a truth assignment to the variables U such that all clauses C are satisfied?

As a running example, we will construct a network for the following (satisfiable) 3SAT instance [5]:

Example 1. $\phi_{\text{ex}} = (U, C)$, where $U = \{u_1, u_2, u_3, u_4\}$, and $C = \{(u_1 \vee u_2 \vee u_3), (\neg u_1 \vee \neg u_2 \vee u_3), (u_2 \vee \neg u_3 \vee u_4)\}$.

We construct a Bayesian network $\mathcal{B}_\phi$ from a 3SAT instance $\phi = (U, C)$ as follows. For each variable u_i in ϕ we add a *ternary* stochastic variable U_i in $\mathcal{B}_\phi$ with values $\{\text{TRUE}, \text{FALSE}, \#\}$ and uniform prior probability; the set of all U_i

[3] As each clause has three variables, the corresponding MPE instance has indegree at most three.

[4] Note that here we require that the algorithm not only returns a joint value assignment $cansol(x)$, but also tells us *which subset* of $cansol(x)$ matches $optsol(x)$.

is denoted $\mathbf{U}$. For each clause c_j in ϕ we add a binary stochastic variable C_j in $\mathcal{B}_\phi$ with values TRUE and FALSE; the set of all C_j is denoted $\mathbf{C}$. C_j is to be conditioned on the variables $\mathbf{U_j} = \{U_j^1, U_j^2, U_j^3\}$ that correspond to the variables that occur in c_j, *and* (for $j > 1$) on the variables $\mathbf{U_{j-1}} = \{U_{j-1}^1, U_{j-1}^2, U_{j-1}^3\}$ that correspond to the variables that occur in c_{j-1}. To improve readability, we define the following shorthands for joint value assignments to $\mathbf{U_j}$ and $\mathbf{U_{j-1}}$: let $\mathbf{u}_\#$ denote a joint value assignment where *all* variables have the value #, and let $\mathbf{u_{TF}}$ denote a joint value assignment where *none* of the variables have the value #, i.e., all are TRUE or FALSE. For $C_j (j > 1)$ the following conditional probability distribution is defined.

$$\Pr(C_j = \text{TRUE} \mid \mathbf{U_j}, \mathbf{U_{j-1}}) = \begin{cases} 1 \text{ if } \mathbf{U_j} = \mathbf{u}, \text{ where } \mathbf{u} \text{ makes clause } C_j \text{ true,} \\ \quad \text{and } \mathbf{U_{j-1}} = \mathbf{u_{TF}} \\ \epsilon \text{ if } \mathbf{U_j} = \mathbf{u}_\# \text{ and } \mathbf{U_{j-1}} = \mathbf{u}_\# \\ 0 \text{ otherwise} \end{cases}$$

Here, ϵ is defined to be a sufficiently small (i.e., $\epsilon < \frac{1}{2^n}$), yet polynomial-time computable, value. Likewise, C_1 is defined as follows.

$$\Pr(C_1 = \text{TRUE} \mid \mathbf{U_1}) = \begin{cases} 1 \text{ if } \mathbf{U_1} = \mathbf{u}, \text{ where } \mathbf{u} \text{ makes clause } C_1 \text{ true} \\ \epsilon \text{ if } \mathbf{U_1} = \mathbf{u}_\# \\ 0 \text{ otherwise} \end{cases}$$

As an example of this construction, Figure 2 shows the network as constructed from ϕ_{ex}. We set the evidence variables $\mathbf{E} = \mathbf{C}$ with $\mathbf{e} = \wedge_{j=1}^{m} C_j = \text{TRUE}$. We claim that ϕ is satisfiable if and only if *none* of the variables in the most probable joint value assignment $\mathbf{u}$ to $\mathbf{U}$ has the value #, and unsatisfiable if and only if *all* of the variables in $\mathbf{u}$ have the value #. Thus, if an approximation algorithm tells us the value of *any* variable of the most probable explanation of $\mathcal{B}$, we can use that algorithm to solve the corresponding 3SAT instance in polynomial-time.

Proof (of Theorem 1). Assume there exists a polynomial-time structure approximation algorithm A that, when given an MPE instance, returns for one of the variables in the explanation set M a value that corresponds to the value of that variable in the most probable explanation. We will show that A can be used to decide 3SAT in polynomial time; hence, from the existence of such an algorithm it would follow that $\mathsf{P} = \mathsf{NP}$. Let ϕ be an arbitrary instance of 3SAT and let $(\mathcal{B}_\phi, \mathbf{E}, \mathbf{e})$ be the MPE instance as constructed above. Note that we can construct $\mathcal{B}_\phi$ from ϕ in polynomial time, as every literal and clause in ϕ corresponds to a single variable in $\mathcal{B}_\phi$ and the size of the conditional probability tables of each variable is bounded by a constant.

Let $\mathbf{u}$ be a joint value assignment to the variables of $\mathbf{U}$ of $\mathcal{B}_\phi$. We will distinguish between three possible scenarios:

1. $\mathbf{u} \in \{\#\}^n$, i.e., *all* variables are set to #
2. $\mathbf{u} \in \{\text{TRUE}, \text{FALSE}\}^n$, i.e., *none* of the variables are set to #
3. $\mathbf{u} \in \{\text{TRUE}, \text{FALSE}, \#\}^n$, $\mathbf{u} \notin \{\#\}^n$, and $\mathbf{u} \notin \{\text{TRUE}, \text{FALSE}\}^n$

Note that in case 3) $\Pr(\mathbf{u}, \mathbf{e}) = 0$ due to the constraints in the joint probability distributions of C_j. In case 2), if $\mathbf{u}$ does *not* satisfy ϕ, then also $\Pr(\mathbf{u}, \mathbf{e}) = 0$. If on the other hand $\mathbf{u}$ *does* satisfy ϕ, then the probability $\Pr(\mathbf{u}, \mathbf{e})$ equals $\frac{1}{N_{sat}(1+\epsilon)}$, where $1 \leq N_{sat} \leq 2^n$ denotes the number of satisfying truth assignments to ϕ. In case 1), if ϕ is satisfiable, then $\Pr(\mathbf{u}, \mathbf{e}) = \frac{\epsilon}{1+\epsilon}$; as ϵ was chosen to be strictly less than $\frac{1}{2^n}$, this probability is lower than the probability of any satisfying joint value assignment. However, when ϕ is not satisfiable, then $\Pr(\mathbf{u}, \mathbf{e}) = 1$.

Thus, the most probable explanation for evidence $\mathbf{e} = \wedge_{j=1}^{m} C_j = \text{TRUE}$ is either $\mathbf{u} \in \{\text{TRUE}, \text{FALSE}\}^n$ if ϕ is satisfiable, or $\mathbf{u} \in \{\#\}^n$ if ϕ is not satisfiable. Now assume that, when given $(\mathcal{B}_\phi, \mathbf{E}, \mathbf{e})$ as input, A outputs the value assignment of one of the unobserved variables in $\mathcal{B}_\phi$, that correspond to the value in the most probable explanation of $\mathcal{B}_\phi$. In case A outputs TRUE or FALSE, ϕ is satisfiable; in case A outputs $\#$, ϕ is not satisfiable. Hence, we can use A to solve 3SAT in polynomial time, concluding the proof.

3.2 Ordered Variables

We saw in the previous section that it is NP-hard to structure-approximate even a single variable of the most probable explanation in a Bayesian network. However, we assumed that the values of the variables in the network were *unordered*. In this section we assume a particular order on the values and investigate the consequences for the computational complexity of structure approximation.

Typically, in a Bayesian network some variables might have a 'natural' ordering, like a variable HEIGHT with values TALL, NORMAL and SMALL; these values are ordered SMALL $\preceq$ NORMAL $\preceq$ TALL. Other variables, like BLOODTYPE or ETHNICGROUP lack such an ordering. When a variable is ordered, it makes sense to redefine the distance measure: when HEIGHT is assigned the value TALL in the most probable explanation, NORMAL would be a better approximation than SMALL.

In the remainder we assume that all variables are ordered, and we introduce a *partial ordered lattice* [20] and a corresponding *lattice distance function*. The lattice includes all joint value assignments to the observable variables in the network and it captures the partial order between the assignments. The bottom of the lattice encodes the joint value assignment $\mathbf{m}$ such that $\mathbf{m} \preceq \mathbf{m}'$ for all $\mathbf{m}' \in \Omega(\mathbf{M})$. Likewise, the top of the lattice encodes the joint value assignment $\mathbf{m}''$ such that $\mathbf{m}' \preceq \mathbf{m}''$ for all $\mathbf{m}' \in \Omega(\mathbf{M})$. In general, a lattice element $L(\mathbf{m})$ encoding a joint value assignment $\mathbf{m}$ precedes another lattice element $L(\mathbf{m}')$ if and only if $\mathbf{m} \preceq \mathbf{m}'$. In Figure 3 an example (from [20]) is shown for two ternary variables X and Y.

A natural distance function comparing two joint value assignments $\mathbf{m}$ and $\mathbf{m}'$ would be the *distance in the lattice* between these assignments, i.e., the length of the shortest path from $L(\mathbf{m})$ to $L(\mathbf{m}')$. For example, the distance between x_2y_1 and x_1y_3 would be three. Note that this distance function, denoted by d_L, is a metric as the properties of Section 2.2 also hold for d_L. Using this distance function, we can find a trivial *guaranteed* $h(x)/d_L$-structure approximation with

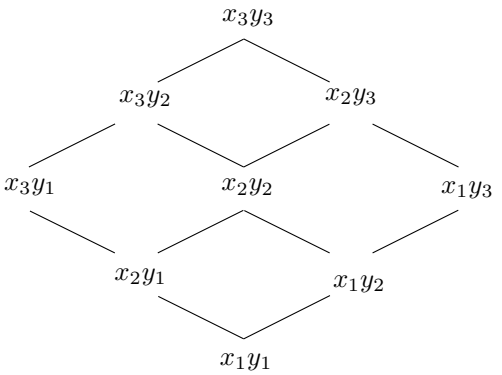


Fig. 3. A lattice describing the partial order of the joint value assignments to the variables X and Y

ordering for $h(x) = |\mathbf{M}| \cdot \lfloor \frac{c}{2} \rfloor$, rather than the *expected* $E(h(x)) = |\mathbf{M}| - \frac{|\mathbf{M}|}{c}$ without ordering, by always picking the 'middle' value in the order. We can, however, not expect to do better than $h(x) = |\mathbf{M}|$ for $c \geq 5$, unless $\mathsf{P} = \mathsf{NP}$:

Theorem 2. *MPE is $h(x)/d_L$-structure inapproximable for $h(x) = |\mathbf{M}| - 1$, unless $\mathsf{P} = \mathsf{NP}$.*

Proof. Similar as in the proof of Theorem 1, and using the same construction, we show that the existence of a polynomial-time algorithm A that can $h(x)/d_L$-structure-approximate MPE for $h(x) = |\mathbf{M}| - 1$ implies that we can decide 3SAT in polynomial time. We augment the construction used to prove Theorem 1 as follows: let all variables U_i have *five* values $\Omega(U_i) = \{\text{FALSE}, \text{TRUE}, \#, d_1, d_2\}$ in which d_1 and d_2 act as dummy variables. U_i is uniformly distributed, and the order of $\Omega(U_i)$ is $\text{FALSE} \preceq d_1 \preceq \# \preceq d_2 \preceq \text{TRUE}$. The conditional probability distribution of C_j is similar as in Theorem 1; in particular, any joint value assignment $\mathbf{U}$ that includes a dummy variable has probability $\Pr(C_j = \text{TRUE} \mid \mathbf{U}) = 0$. We claim that, for any $h(x)/d_L$-structure approximation with $h(x) \leq |\mathbf{M}|-1$, the majority of the variables that contain non-dummy values can be used to decide satisfiability of ϕ: if the (strict) majority of these variables has TRUE or FALSE as value, then the instance is satisfiable, otherwise the instance is unsatisfiable.

Observe that an approximation with $h(x) = |\mathbf{M}| - 1$ has at least *one* 'correct' variable, as any deviation from the MPE would increase $h(x)$ by at least one, i.e., every variable that has a value that is not equal to the MPE contributes a distance of 1 to $h(x)$. In particular, when one of the variables is correctly labeled with either # (for an unsatisfying instance) or TRUE or FALSE (for a satisfying instance), and the other variables have dummy values that are closest to the MPE value of that variable (i.e., d_1 for FALSE, d_2 for TRUE, and either d_1 or d_2 for #), then $h(x) = |\mathbf{M}| - 1$; clearly here a majority of the (non-dummy) variables correctly reflects the satisfiability of the instance.

Now we show that this property holds for every alteration to this joint value assignment that maintains that $h(x) = |\mathbf{M}| - 1$. We will demonstrate the case that ϕ is satisfiable; for unsatisfiable ϕ, the proof goes analogously.

- If we replace a dummy value with a # value, then $h(x)$ increases by one. We must also change another dummy value to TRUE or FALSE (whichever is closest) to maintain that $h(x) = |\mathbf{M}| - 1$, so still the majority of non-dummy variables has as value TRUE or FALSE.
- If we replace a TRUE or FALSE value to a # value, then $h(x)$ increases by two, and so two dummy variables need to be changed into TRUE or FALSE.

Thus, if A returns a $h(x)/d_L$-structure approximation with $h(x) \leq |\mathbf{M}| - 1$, then we can use the output to decide 3SAT: count the number TRUE or FALSE values and the number of #-values. If the first number is higher than the second, answer *yes*, else answer *no*. As A runs in polynomial time, this algorithm can decide 3SAT in polynomial time, hence $\mathsf{P} = \mathsf{NP}$.

4 Conclusion

In this paper we discussed structure approximations of MPE. In general, we cannot do better than just randomly guess the joint value assignment: we then would on average expect to guess $\frac{1}{c}$ of the variables correctly, where c is the cardinality of the variables. As it is NP-hard to determine the value of more than $\frac{1}{c}$ of the variables in the MPE, there is little room for improvement. We hypothesize (but could not prove) that it is even NP-hard to get an *expected* structure approximation that is strictly better than $|\mathbf{M}| - \frac{|\mathbf{M}|}{c}$.

Furthermore, we showed that it is NP-hard in general to obtain an approximation that determines even a *single* variable in the MPE. So, without information on the ordering of the values or restrictions on the network structure or probability distribution, if we want information on the structure of the MPE (in polynomial time), there are little alternatives than to compute it exactly.

However, if we do have information on the ordering of the values, we can do a bit better[5] than that. We showed that the simple strategy 'always stay in the middle' *guarantees* a $h(x)/d_L$-structure approximation for $h(x) = |\mathbf{M}| \cdot \lfloor \frac{c}{2} \rfloor$ in the worst case, which is better than the expected value if we would randomly guess the values. We showed that it is NP-hard to $h(x)/d_L$-structure approximate MPE for $h(x) = |\mathbf{M}| - 1$ and $c \geq 5$.

The gap between these two results (for $c = 5, h(x) = 2 \cdot |\mathbf{M}|$) might leave some room for improvement. There may be constrained situations where solving MPE exactly remains intractable, whereas a good structure approximation might be found in polynomial time. One suggestion, that we leave for future work, is to investigate whether it could help to use *monotonicity properties* in the network

[5] As one reviewer carefully pointed out, the Hamming and edit distances are not quite comparable for $c > 2$ as the edit distance will be on average larger with larger c, while the Hamming distance remains 1 whenever a mismatch occurs.

to get a better structure approximation; the NP-hardness proofs in this paper critically depend on non-monotone relations between the clause-nodes and the literal-nodes in the network. However, note that even when the hypothesis space is monotone in the evidence, obtaining evidence need not tell us anything about the most probable hypothesis. As an example, in the following conditional dependencies for H_1 and H_2, the most probable hypothesis given evidence e differs, even though both H_1 and H_2 are monotone in E:

$$\Pr(H_1 = \text{TRUE} \mid E) = \begin{cases} 0.2 \text{ if } E = \text{TRUE} \\ 0.1 \text{ if } E = \text{FALSE} \end{cases}$$

$$\Pr(H_2 = \text{TRUE} \mid E) = \begin{cases} 0.7 \text{ if } E = \text{TRUE} \\ 0.6 \text{ if } E = \text{FALSE} \end{cases}$$

Note, that $\text{argmax}_{H_1} \Pr(H_1, e = \text{TRUE}) = \text{FALSE}$, while $\text{argmax}_{H_2} \Pr(H_2, e = \text{TRUE}) = \text{FALSE}$.

Acknowledgments. The author wishes to thank Iris van Rooij, Linda van der Gaag, and Todd Wareham for helpful discussions on this topic, and the anonymous reviewers for their constructive comments. A previous version of this paper appeared in the 24th Benelux Conference on Artificial Intelligence [13].

References

1. Abdelbar, A.M., Hedetniemi, S.M.: Approximating MAPs for belief networks is NP-hard and other theorems. Artificial Intelligence 102, 21–38 (1998)
2. Baker, C.L., Saxe, R., Tenenbaum, J.B.: Action understanding as inverse planning. Cognition 113(3), 329–349 (2009)
3. Bodlaender, H.L., van den Eijkhof, F., van der Gaag, L.C.: On the complexity of the MPA problem in probabilistic networks. In: van Harmelen, F. (ed.) Proceedings of the 15th European Conference on Artificial Intelligence, pp. 675–679 (2002)
4. Chater, N., Oaksford, M.: Ten years of the rational analysis of cognition. Trends in Cognitive Sciences 3, 57–65 (1999)
5. Cooper, G.F.: The computational complexity of probabilistic inference using Bayesian belief networks. Artificial Intelligence 42(2), 393–405 (1990)
6. Feige, U., Langberg, M., Nissim, K.: On the hardness of approximating NP witnesses. In: Jansen, K., Khuller, S. (eds.) APPROX 2000. LNCS, vol. 1913, pp. 120–131. Springer, Heidelberg (2000)
7. Garey, M.R., Johnson, D.S.: Computers and Intractability. A Guide to the Theory of NP-Completeness. W. H. Freeman and Co., San Francisco (1979)
8. Gemela, J.: Financial analysis using Bayesian networks. Applied Stochastic Models in Business and Industry 17, 57–67 (2001)
9. Hamilton, M., Müller, M., van Rooij, I., Wareham, T.: Approximating solution structure. In: Demaine, E., Gutin, G.Z., Marx, D., Stege, U. (eds.) Structure Theory and FPT Algorithmics for Graphs, Digraphs and Hypergraphs, Dagstuhl Seminar Proceedings, vol. 07281 (2007)
10. Jensen, F.V., Nielsen, T.D.: Bayesian Networks and Decision Graphs, 2nd edn. Springer, New York (2007)

11. Kumar, R., Sivakumar, D.: Proofs, codes, and polynomial-time reducibilities. In: Proceedings of the Fourteenth Annual Conference on Computational Complexity, pp. 46–53. IEEE Computer Society (1999)
12. Kwisthout, J.: Most probable explanations in Bayesian networks: Complexity and tractability. International Journal of Approximate Reasoning 52(9), 1452–1469 (2011)
13. Kwisthout, J.: Structure approximation of most probable explanations in Bayesian networks. In: Uiterwijk, J.W.H.M., Roos, N., Winands, M.H.M. (eds.) Proceedings of the 24th Benelux Conference on Artificial Intelligence (BNAIC 2012), pp. 131–138 (2012)
14. Lacave, C., Díez, F.J.: A review of explanation methods for Bayesian networks. The Knowledge Engineering Review 17(2), 107–127 (2002)
15. Lucas, P.J.F., de Bruijn, N., Schurink, K., Hoepelman, A.: A probabilistic and decision-theoretic approach to the management of infectious disease at the ICU. Artificial Intelligence in Medicine 3, 251–279 (2000)
16. Millgram, E.: Coherence: The price of the ticket. Journal of Philosophy 97, 82–93 (2000)
17. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, Palo Alto (1988)
18. Shimony, S.E.: Finding MAPs for belief networks is NP-hard. Artificial Intelligence 68(2), 399–410 (1994)
19. Sticha, P.J., Buede, D.M., Rees, R.L.: Bayesian model of the effect of personality in predicting decisionmaker behavior. In: van der Gaag, L.C., Almond, R. (eds.) Proceedings of the Fourth Bayesian Modelling Applications Workshop (2006)
20. van der Gaag, L.C., Renooij, S., Geenen, P.L.: Lattices for studying monotonicity of Bayesian networks. In: Studený, M., Vomlel, J. (eds.) Proceedings of the Third European Workshop on Probabilistic Graphical Models, pp. 99–106 (2006)
21. van der Gaag, L.C., Renooij, S., Witteman, C.L.M., Aleman, B.M.P., Taal, B.G.: Probabilities for a probabilistic network: a case study in oesophageal cancer. Artificial Intelligence in Medicine 25, 123–148 (2002)
22. van Rooij, I., Kwisthout, J., Blokpoel, M., Szymanik, J., Wareham, T., Toni, I.: Communicating intentions: Computationally easy or difficult? Frontiers in Human Neuroscience 5(52), 1–18 (2011)
23. van Rooij, I., Wareham, T.: Intractability and approximation of optimization theories of cognition. Journal of Mathematical Psychology 56(4), 232–247 (2012)

Argumentation Based Dynamic Multiple Criteria Decision Making

Christophe Labreuche

Thales Research & Technology, Palaiseau, France
christophe.labreuche@thalesgroup.com

Abstract. An important aspect of decision making is that a decision is not made at a time assuming that the decision maker (DM) has all relevant information at hand at the same time. On the contrary making a decision results from a process during which (non) relevant and possibly conflicting information come at different instant. Multi-criteria decision making (MCDM) is a typical case of a dynamic process. We show in this paper how argumentation can help to represent this dynamics.

1 Introduction

Multiple Decision Making (MCDM) consists for a Decision Maker (DM) in selecting one option among several on the basis of several criteria. In this setting, the set of criteria is assumed to be known and fixed. In contrast to the normative, descriptive and prescriptive approaches to decision aid, *constructive* approaches correspond to situations where the set of criteria and the MCDM model are evolving over time [13,3,14]. Decision is not instantaneous, assuming that the DM has all relevant information at hand at the same time. On the contrary making a decision results from a process during which possibly conflicting information becomes available at different instants. In common language, the decision is the impact point of a sequence of highlights and milestones during which the options are being constructed, partial decisions are taken, a new actor provides information or advice, and the decision problem is reshaped, a new criterion appears useful, and so on. This constitutes the *decision process*. There are often big differences between the initial convictions of the DM and the final decision.

What is at the origin of this evolution? At the epistemic (knowledge) level, the evaluation of the options on the various criteria might be mistaken or new options might be considered. At the practical (preferences) level, as the DM's preferences are by nature subjective, one may say that no individual can provide arguments against the use of a particular decision model for the DM. Now, consider the example of a DM who wishes to buy an electronic device. The choice of the relevant criteria the DM shall use depends mostly on the *usage* he intends to make of the device. We interpret these usages as *goals* the DM has to pursue. Then the shift between the initial convictions of the DM and the final decision arises from (1) a mismatch between the usage/goal and the preference model he defined (e.g. the criteria that appear in the decision model are not representative of the usage), or (2) a mismatch between the preference model defined by the DM and the available options (in which case, there is no satisfactory option according to the

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 352–363, 2013.

preference model). In both cases, the DM has to revise his preference model. In particular, the DM probably needs to make concessions on the level of achievement he wants on some goals. The mechanism under which the DM makes concession is similar to the concessions agents make during a negotiation protocol [6,10,12]. The DM becomes aware of these mismatches by getting new information from experts (e.g. reading expert reviews on the electronic devise or discussing with experts). The DM will discover new usages relevant for him, importance of criteria given some usages, etc.

We propose to use argumentation to handle this dynamic process. Argumentation is useful as several experts are expected to give rise to conflicting arguments. The existing works combining multi-criteria approaches to argumentation [2,5,9,15] do not address our concern properly. In particular, they do not make a distinction between the goals and the criteria. Moreover, they do not address the concessions the DM needs to make during the dynamic decision process.

The paper is organized as follows. Section 2 develop the abstract general modelling of the decision process. The next section justifies our approach with the help of a detailed example on the purchase of a digital camera. Section 4 explains our (argumentation) model to represent the decision process. Section 5 illustrate this model on the motivating example. Lastly, we compare ourselves to related works and we conclude.

2 General Modeling of the Whole Decision Process

2.1 Abstract General Modeling of the Decision Process: Acceptability and Goals

The DM shall make a choice among a set $\mathcal{D}$ of *potential decisions (options)*. The choice of the DM will be based on a set $\mathcal{C} = \{c_1, \ldots, c_n\}$ of fundamental *points of view (criteria)*, which permit to meet the concerns of the DM. Each criterion $c \in \mathcal{C}$ is associated to a *descriptor (attribute)*, that is a set $\mathcal{X}_c$ describing the plausible impacts of options with respect to c. The DM will only base his decision on the value of options on the n attributes. The value of option d on attribute $\mathcal{X}_c$ is denoted by $x_c(d)$.

The decision model is characterized by some parameters p (e.g. weights on criteria) At this point, the decision model is only represented by a condition $acceptability_p(d)$ indicating whether $d \in \mathcal{D}$ is acceptable given parameters values p.

Our point is that the choice of parameters p We will develop in Section 3 an illustrative example on an individual willing to buy a digital camera. Criteria are not the same if the user wants to make sport or portrait pictures. These usages can be seen as *goals* that the DM wishes to pursue. Unlike traditional AI works [1,2], we make a clear distinction between goals and criteria. Goals correspond to usages the DM intends to make; these criteria are derived from these goals and the decision is made on the basis of criteria.

We denote by $\mathcal{G}$ the set of all potential goals for the DM. All possible usages will be potential goals. One may have other goals such as the will to minimize cost. An importance in scale $\mathbb{L}_{\text{imp}}$ is defined for each $g \in \mathcal{G}$ – depicting how important it is to fulfil the goal. Apart from that, we denote by e_g the level of achievement of goal g expressed in a scale $\mathbb{L}_{\mathcal{G}}$ composed of values "null", "low", "medium" and "high". We have $e \in {\mathbb{L}_{\mathcal{G}}}^{\mathcal{G}}$. As previously said, parameters p are derived from e by a function $T : {\mathbb{L}_{\mathcal{G}}}^{\mathcal{G}} \rightarrow \mathcal{P}$ such that $T(e) = p$.

2.2 Abstract General Modeling of the Decision Process: Concession

Given $e \in {\mathbb{L}_{\mathcal{G}}}^{\mathcal{G}}$, the acceptable options are the options $d \in \mathcal{D}$ such that condition $acceptability_{T(e)}(d)$ holds. This set might be empty. We interpret this as some conflicts that arise during the decision process, between the input data and the decision model.

Negotiation is a standard means during which a set of agents having conflicting points of view try to reach an agreement. The alternating offer protocol is very popular one in AI and Game Theory [8]. Let us consider a buyer-seller situation. Both agents define the ranges of price values that they find acceptable: the most preferable price $initP$ an agent will start with in the negotiation, and the limit price $limitP$ below/above which the agent will refuse to accept offers. When an agent has a negotiation deadline T, he will make offer $initP$ at time $t = 0$ and offer $limitP$ just before $t = T$. The offers he will make for intermediate times depend on his tactic concerning time [6,10,12]: *boulware* (the agent keeps his initial offer almost until the deadline, and makes concession only at the end), *conceder* (the agent concedes very rapidly and then offers $limitP$ till the deadline), *linear* (the agent makes concession linearly with time).

When there is no acceptable option at a given iteration of the decision process, the DM needs to make concessions on the expected level of achievement of some goals. As in a negotiation, the DM starts with some initial preferences, artifacts on which he is ready to give up, artifacts on which he will by no mean give up. Then the DM can define two levels of achievement for each goal g: the initial one $initP_g \in \mathbb{L}_{\mathcal{G}}$ the DM will start with in the decision process, and the limit threshold $limitP_g \in \mathbb{L}_{\mathcal{G}}$ below/above which the DM will refuse to concede. For goal $g \in \mathcal{G}$, we define a concession function

$$Conc_g : \mathbb{N} \times \mathbb{L}_{\mathcal{G}} \times \mathbb{L}_{\mathcal{G}} \to \mathbb{L}_{\mathcal{G}}. \tag{1}$$

More precisely, $Conc_g(k, initP_g, limitP_g)$ is the value of the required achievement level the DM is ready to accept at step k. The tactics boulware, conceder and linear provide examples of functions $Conc$. We assume that $Conc_g(1, initP_g, limitP_g) = initP_g$ and $Conc_g(T, initP_g, limitP_g) = limitP_g$, where T is the deadline. We define $C : \mathbb{N} \times {\mathbb{L}_{\mathcal{G}}}^{\mathcal{G}} \times {\mathbb{L}_{\mathcal{G}}}^{\mathcal{G}} \to 2^{{\mathbb{L}_{\mathcal{G}}}^{\mathcal{G}}}$ by

$$C(k, initP, limitP) = \times_{g \in \mathcal{G}} \Big[initP_g, Conc_g(T, initP_g, limitP_g) \Big] \tag{2}$$

where $[a, b]$ (with $a, b \in \mathbb{L}_{\mathcal{G}}$) is the set of elements of $\mathbb{L}_{\mathcal{G}}$ between a and b, and $C(k, initP, limitP) \subseteq {\mathbb{L}_{\mathcal{G}}}^{\mathcal{G}}$ is the set of possible concessions at iteration k of the decision process. The acceptable options after a concession has been made are:

$$acc\mathcal{D}(k) = \{d \in \mathcal{D} \ , \ \exists g \in C(k, initP, limitP) \text{ s.t. } acceptability_{T(g)}(d) \text{ holds}\}. \tag{3}$$

The decision process may stop whenever $acc\mathcal{D}(k) \neq \emptyset$. The preferred option in $acc\mathcal{D}(k)$ is determined by (i) minimizing the importance of the goals on which a concession is made, and (ii) maximizing the certainty of the information used for each option.

This general model of the decision process will be instantiated in the rest of the paper. We propose to use logic and argumentation rather than simply applying functions C, $acc\mathcal{D}$, ..., because (i) there are often conflicting information coming from several sources (which cannot be readily handled with standard MCDM), (ii) an argument contains its warrant, i.e. the explanation of its statement (which is not the case in MCDM).

2.3 Instantiation of the Abstract General Modeling: Case of the Weighted Sum

The aim of this paper is to emphasize on the synergy between MCDM and argumentation to represent a realistic dynamic decision process. To keep the paper simple for non-specialists in MCDM, we choose the simplest and most widely used MCDM model – namely the weighted sum – but our model can be used on any MCDM model.

The DM has a preference represented by the binary relation $\succsim_{\mathcal{X}_c}$ on attribute $\mathcal{X}_c$: $a \succsim_{\mathcal{X}_c} b$ if the DM finds that $a \in \mathcal{X}_c$ is at least as good as $b \in \mathcal{X}_c$. Criteria are supposed to be either not satisfied at all or completely satisfied. More precisely, criterion c is completely met iff $x_c \succsim_{\mathcal{X}_c} \mathbb{1}_c$, where $\mathbb{1}_c \in \mathcal{X}_c$ is a threshold. In order to synthesize the evaluations on all criteria, a scale $\mathbb{L}_{\text{imp}}$ expressing the relative *importance of each criterion* is also introduced. Scale $\mathbb{L}_{\text{imp}}$ is composed of values 0 (no importance), 1 (weak importance), 2 (medium importance), 3 (large importance) and 4 (very large importance). In the weighted sum model, an overall utility U is assigned to each option $d \in \mathcal{D}$:

$$U(d) = \sum_{c \in \mathcal{C},\, x_c(d) \succsim_{\mathcal{X}_c} \mathbb{1}_c} w_c \tag{4}$$

where $w_c \in \mathbb{L}_{\text{imp}}$ is the relative importance of criterion c. The parameters p of this model are composed of w_c and $\mathbb{1}_c$ for all $c \in \mathcal{C}$: $p = (w_c, \mathbb{1}_c)_{c \in \mathcal{C}}$.

An option is preferred to another one if its overall utility is larger. Finally, the DM cannot accept a decision if most of the criteria are not met. In this paper, condition $acceptability_p(d)$ is true (d is *acceptable* for the DM) if

$$U(d) > \lambda \sum_{c \in \mathcal{C}} w_c \qquad (\text{with } \tfrac{1}{2} < \lambda < 1). \tag{5}$$

3 Motivating Example

We consider here a typical example coming from daily life: John (the DM) wants to buy a digital camera. We take $\lambda = \frac{3}{4}$ in equation (5).

3.1 Step 1: Initialisation Phase – Definition of Usages and Criteria

John first defines his goals for the camera. His main usage will be for "landscape" photography as he likes trekking and climbing. He associates achievement level "high" to this usage. The second goal is "affordability" as John has limited budget. Then John defines the following three criteria according to his goals:

- c_1: sharpness of images. Attribute $\mathcal{X}_{c_1}$ is the number of pixels in Mega pixels (Mpx). John sets $\mathbb{1}_{c_1} = 10\,\text{Mpx}$ and its importance to 2 (medium).
- c_2: speed of camera. Attribute $\mathcal{X}_{c_2}$ is the number of images that the camera can take in a second (im/s). John sets $\mathbb{1}_{c_2} = 1\,\text{im/s}$ and its importance to 1 (weak).
- c_3: price. Attribute $\mathcal{X}_{c_2}$ is the price of the camera in Euros. John sets $\mathbb{1}_{c_3} = 200$ Euros and its importance to 2 (medium). John also says that his limit price he cannot go beyond is $400\,\text{Euros}$.

John assesses four options d_1, d_2, d_3, d_4. Acceptability condition equation (5) gives: $U(d) > \frac{15}{4}$ (see Table 1). Option d_1 completely meets the criteria and expectations of John. He wants to have advices from different experts to confirm his choice.

3.2 Step 2: Expertise on Sharpness

– Capture of the Expertise. John learns from an expert that *"If expectation is high on landscape, then threshold* $\mathbb{1}_{c_1}$ *should be* $16\,\mathrm{Mpx}$*"*, and *"If expectation is medium on landscape, then threshold* $\mathbb{1}_{c_1}$ *should be* $14\,\mathrm{Mpx}$*"*. The expert also says that *"If expectation is medium or high on landscape, then criterion speed is not important"*.

From this, John updates $\mathbb{1}_{c_1} = 16\,\mathrm{Mpx}$ and importance of c_2 to 0. Acceptability condition equation (5) gives: $U(d) > 3$. There is no acceptable option:

$$U(d_1) = 2\ ,\ U(d_2) = 2\ ,\ U(d_3) = 0\ ,\ U(d_4) = 2.$$

– Choice on the Concession. John realizes that his expectations yield no acceptable solution. He thinks of making a concession either on usage (give-up on "high on landscape" and consider "medium on landscape" instead) or on price (give-up on $\mathbb{1}_{c_3} = 200\,\mathrm{Euros}$ and consider $\mathbb{1}_{c_3} = 300\,\mathrm{Euros}$ instead). In the first case, he updates $\mathbb{1}_{c_1} = 14\,\mathrm{Mpx}$. We obtain:

For concession on usage: $U(d_1) = 2\ ,\ U(d_2) = 2\ ,\ U(d_3) = 2\ ,\ U(d_4) = 2$

For concession on price: $U(d_1) = 2\ ,\ U(d_2) = 2\ ,\ U(d_3) = 2\ ,\ U(d_4) = 2$

For concession on both: $U(d_1) = 2\ ,\ U(d_2) = 2\ ,\ U(d_3) = 4\ ,\ U(d_4) = 2$

John understands that making only one of these two concessions does not solve the problem as there is no admissible option in both situations. Hence he decides to make both concessions. Option d_3 appears then as a suitable camera.

3.3 Step 3: Need for "Adventure-Proof" Camera

– Capture of the Expertise. John discusses with a new expert that warns him that he needs a rugged (tough) camera: *"If you like climbing, you need to add* adventure *with expectation* high *as usage"*. Moreover, the experts says: *"If expectation is high on adventure, criterion* adventure-proofness *is very important"*. The new criterion c_4 (adventure-proofness) is associated to attribute $\mathcal{X}_{c_4}$ representing the height in meter (m) up to which the camera is shockproof. Finally, the expert says: *"If expectation is high on adventure, then the camera should be shockproof up to* $1.5\,\mathrm{m}$*"*.

Accordingly, John adds c_4 in $\mathcal{C}$, with importance 4 and $\mathbb{1}_{c_4} = 1.5\,\mathrm{m}$. Acceptability condition equation (5) gives: $U(d) > 6$. John considers two new cameras (d_5 and d_6).

Options d_5 and d_6 are just below the acceptability threshold (see Table 1).

Table 1. Values of the options on the criteria and utility U at steps 1 and 3. In columns with U, Yes/No in bracket tells whether option is acceptable

Options	c_1 (Mpx)	c_2 (im/s)	c_3 (Euros)	U (acc) at step 1	c_4 (m)	U (acc) at step 3
d_1	10	1	200	5 (Yes)	0.5	2 (No)
d_2	8	3	200	3 (No)	0.5	2 (No)
d_3	14	1	300	3 (No)	0.5	4 (No)
d_4	16	1	400	3 (No)	0.5	2 (No)
d_5	12	1	300		1.5	6 (No)
d_6	14	1	400		1.5	6 (No)

– Final Concession. John thinks of making a last concession either on usage (give-up on "medium on landscape" and consider "low on landscape" instead) or on price (give-up on $\mathbb{1}_{c_3} = 300\,\text{Euros}$ and consider $\mathbb{1}_{c_3} = 400\,\text{Euros}$ instead). John does not want to give-up on "medium on landscape" since it was his first motivation for buying a camera. Hence he accepts to make concession on price ($\mathbb{1}_{c_3} = 400\,\text{Euros}$) as $400\,\text{Euros}$ does not exceed his limit price. We obtain:

$$U(d_1) = 2\ ,\ U(d_2) = 2\ ,\ U(d_3) = 4\ ,\ U(d_4) = 4\ ,\ U(d_5) = 6\ ,\ U(d_6) = 8.$$

Only d_6 is acceptable. Finally, as John thinks he gets sufficient advices, he decides to buy the acceptable option d_6.

4 Argumentation Framework for the Decision Process

4.1 Knowledge and Goal Bases

In Argumentation frameworks applied to decision problems, two separate knowledge bases are usually constructed: one at the epistemic layer depicting the beliefs of the agent, and the other one at the practical layer representing the preferences of the agent [11]. Decision models are described as propositional formula.

In our case, the belief and preference layers are heterogeneous (the first one is symbolic and the second one is numeric). We propose to represent the multi-criteria preference model under the formalism of propositional logic in order to have a unique symbolic formalism to encompass both beliefs and preferences. Argumentation will then be used on all rules (both from the belief and preference layers).

The decision process is composed of different steps from step 1 to step T (the deadline). The current step number is k. Let $\mathcal{L}$ be a propositional language among which the propositional variables are:

- $Goal(g, \alpha)$ with $g \in \mathcal{G}$ and $\alpha \in \mathbb{L}_{\mathcal{G}}$, which is true if goal g has level of achievement α at current step k;
- $initAchievGoal(g, \alpha)$ with $g \in \mathcal{G}$ and $\alpha \in \mathbb{L}_{\mathcal{G}}$, which is true if goal g has initial required achievement level α;
- $limitAchievGoal(g, \alpha)$ with $g \in \mathcal{G}$ and $\alpha \in \mathbb{L}_{\mathcal{G}}$, which is true if goal g has limit required achievement level α;
- $Concession(g, k)$ with $g \in \mathcal{G}$, which is true if a concession on goal g is done at step k;
- $Att_c(d, \alpha)$ with $c \in \mathcal{C}$, $d \in \mathcal{D}$ and $\alpha \in \mathcal{X}_c$, which is true if the value of option d on attribute $\mathcal{X}_c$ is α (i.e. $x_c(d) = \alpha$);
- $Imp_c(\alpha)$ with $c \in \mathcal{C}$ and $\alpha \in \mathbb{L}_{\text{imp}}$, which is true if the importance of crit. c is α;
- $\mathbb{1}_c(\alpha)$ with $c \in \mathcal{C}$ and $\alpha \in \mathcal{X}_c$, which is true if the value of threshold $\mathbb{1}_c$ is α;
- $U(d, \alpha)$ with $d \in \mathcal{D}$ and $\alpha \in \mathbb{R}_+$, which is true if the utility $U(d)$ of option d is α;
- $Acc(d)$ with $d \in \mathcal{D}$, which is true if option d is acceptable according to the preference model of the DM;
- $Acc(\mathcal{D})$, which is true if there is at least one option in $\mathcal{D}$ that is acceptable.

Let us now give examples of rules at the belief and practical levels (taken from the illustrative example) based on the previous primitives. Function $T : {\mathbb{L}_{\mathcal{G}}}^{\mathcal{G}} \to \mathcal{P}$ (see Section 2.1) is described as rules. The first two rules are examples of such rules.

$$\begin{aligned}
&Goal(\text{landscape}, \text{high}) \ \to \ \mathbb{1}_{c_1}(16)\\
&Goal(\text{landscape}, \text{high}) \vee Goal(\text{landscape}, \text{medium}) \ \to \ Imp_{c_2}(0)\\
&\text{climbing} \ \to \ Goal(\text{adventure}, \text{high})
\end{aligned}$$

Let us now describe the rules at the preference level. First of all, if a criterion has importance 0, the DM is not obliged to define threshold $\mathbb{1}$ and we can set any value to $\mathbb{1}$

$$Imp_c(0) \ \to \ \mathbb{1}_c(0) \tag{6}$$

According to equation (4), the rule specifying U is:

$$\begin{aligned}
&Att_{c_1}(d, x_1) \wedge \ldots \wedge Att_{c_n}(d, x_n) \wedge \mathbb{1}_{c_1}(\alpha_1) \wedge \ldots \wedge \mathbb{1}_{c_n}(\alpha_n) \wedge\\
&\wedge Imp_{c_1}(w_1) \wedge \ldots \wedge Imp_{c_n}(w_n) \ \to \ U\Big(d, \sum_{i \in \{1,\ldots,n\}\,,\, x_i \succsim_{\mathcal{X}_c} \alpha_i} w_n\Big)
\end{aligned} \tag{7}$$

According to equation (5), the rule specifying $Acc(d)$ is:

$$\begin{aligned}
&Att_{c_1}(d, x_1) \wedge \ldots \wedge Att_{c_n}(d, x_n) \wedge \mathbb{1}_{c_1}(\alpha_1) \wedge \ldots \wedge \mathbb{1}_{c_n}(\alpha_n) \wedge\\
&\wedge Imp_{c_1}(w_1) \wedge \ldots \wedge Imp_{c_n}(w_n) \wedge\\
&\wedge \Big(\sum_{i \in \{1,\ldots,n\}\,,\, x_i \succsim_{\mathcal{X}_c} \alpha_i} w_n > \lambda \sum_{i \in \{1,\ldots,n\}} w_n \Big) \ \to \ Acc(d)
\end{aligned} \tag{8}$$

Finally predicate $Acc(\mathcal{D})$ is obtained by the following rule:

$$\vee_{d \in \mathcal{D}} \, Acc(d) \ \to \ Acc(\mathcal{D}) \tag{9}$$

4.2 Concession Management

During the decision process, the DM may adopt different concession tactics (boulware, conceder, linear) depending on the goals. He may for instance be boulware on one goal and conceder on another goal. Note that there is some relationship between the choice of these tactics for each goal, and the relative priority assigned to them. For instance, one might expect that the DM is not ready to concede rapidly for an important goal.

The next rule determines the step where the last concession occurred for goal g

$$\begin{aligned}
&Concession(g, k') \wedge \neg Concession(g, k'+1) \wedge \cdots \wedge \neg Concession(g, k)\\
&\wedge (k' \leq k) \to LastConcession(g, k')
\end{aligned} \tag{10}$$

If the latest concession occurred at step k', then the required achievement level at step k for goal g is equal to $Conc_g(k', initP, limitP)$ (i.e. the concession made at step k'):

$$\begin{aligned}
&LastConcession(g, k') \wedge initAchievGoal(g, initP_g) \wedge limitAchievGoal(g, limitP_g)\\
&\to Goal(g, Conc_g(k', initP_g, limitP_g))
\end{aligned} \tag{11}$$

where $Conc_g$ is defined in (1). In order to trigger rule (11) for all $k \geq 1$, we add the literal $Concession(g, 1)$ for all goals in the knowledge base, since making a concession at step 1 means taking the initial value $initP_g$ at this step.

4.3 Dynamics

At each step $k \geq 2$, the DM gets new information from a new expert and updates his bases accordingly. In the motivating example, the DM found an acceptable option at steps 1 and 2 (options d_1 and d_3 respectively). Hence, why did not he select the acceptable option and stop the decision process at one of these steps? At the beginning of the decision process, the DM sets a deadline T for the decision process. He did not stop before the deadline since, he feels that he is not expert in the field and he wants to get advices from a minimum number (denoted by $threshold$ later) of experts to confront his conviction and become confident in his decision. In our motivating example, the DM prefers to see at least two experts in different specialties before taking the final decision. The decision process stops when an acceptable option is obtained and the DM is confident in the model, or when the maximal number of steps is reached:

$$(k \geq T) \vee ((k \geq threshold) \wedge Acc(\mathcal{D})) \rightarrow stop. \tag{12}$$

At step k of the decision process, the DM has two bases:

- $\mathcal{B}_{\mathcal{K}}(k) = \{(s, \rho)\}$ where $s \in Wff(\mathbb{L})$ (well-formed formulas of language $\mathbb{L}$), is a knowledge base (including criteria and goals), and ρ is the certainty level associated to s. We assume ρ can take only two values: 1 (low certainty) or 2 (high certainty).
- $\mathcal{B}_{\mathcal{G}}(k) = \{(g, p)\}$ where $g \in \mathcal{G}$, is a set of goals, and p is the importance on g in scale $\mathbb{L}_{\text{imp}}$.

We denote by $\mathcal{B}^*_{\mathcal{K}}(k)$ and $\mathcal{B}^*_{\mathcal{G}}(k)$ the corresponding sets of propositions in which weights are ignored. We assume that $\mathcal{B}^*_{\mathcal{G}}(k) = \mathcal{G}$ so that $\mathcal{B}_{\mathcal{G}}(k)$ contains each goal exactly once. $\mathcal{B}_{\mathcal{K}}(k)$ contain rules (6) – (11) with certainty 2. $\mathcal{B}_{\mathcal{K}}(1)$ contains the initial convictions of the DM with certainty either 1 or 2. The DM assigns certainty 1 to the initial convictions for which he is not confident. At step k, the DM adds the rules provided by the experts with certainty 2.

4.4 Arguments

We derive the importance w_c and the threshold $\mathbb{1}_c$ for criteria from the required level of achievement of goals. According to equation (5), option d is acceptable if most of the criteria are met, and thus if most of the goals are fulfilled, thanks to the rules between goals and criteria artifacts. If the DM is more drastic and wants all his goals to be fulfilled, equation (5) shall be replaced by the condition $U(d) = \sum_{c \in \mathcal{C}} w_c$.

Once knowledge and goal bases are defined, we are looking for the subsets of these bases that entail the acceptability of a decision [1,2]. These are arguments.

Definition 1. *An* argument supporting a decision $d \in \mathcal{D}$ *is a pair* $a = \langle K, d \rangle$ *where*

(i) $K \subseteq \mathcal{B}^*_{\mathcal{K}}(k)$,
(ii) K *is consistent,*
(iii) $K \vdash Acc(d)$,
(iv) K *is minimal (w.r.t.* $\subseteq$*) among the sets satisfying (i), (ii) and (iii).*

K *is called the support of the argument and* d *its conclusion. Arguments correspond to elements in* $acc\mathcal{D}(k)$ *(see (3)).*

The importance of criteria is used to weigh up pros and cons for each option in (7) and (8). By contrast, importance of goals allows to select the most appropriate concession: we prefer to concede on a less important goal.

Definition 2. *Given an argument $a = \langle K, d \rangle$, we define $(cert(a), achiev(a))$ (the certainty and achievement levels of argument a respectively) by*

$$cert(a) = |\{(r,1) \in \mathcal{B}_{\mathcal{K}}(k) \,:\, r \in K\}| \quad \textit{(\# of beliefs in } K \textit{ of certainty } 1\textit{)}$$

$$achiev(a) = \sum_{(g,p) \in \mathcal{B}_{\mathcal{G}}(k) \,:\, LastConcession(g,k) \in K} p$$

For function $achiev$, we consider all goals for which a concession has been made at current iteration k. The importance of these goals are summed as there is a cumulative effect to make concessions on several goals at the same time. We want to minimize the value of both $achiev$ and $cert$ functions. We add predicates $Concession(g,k)$ and $\neg Concession(g,k)$ in the knowledge base $\mathcal{B}_{\mathcal{K}}(k)$ at step k. As we know after each iteration whether a concession has been made, we enter in $\mathcal{B}_{\mathcal{K}}(k)$ the correct value of these predicates for the previous steps (either $Concession(g,j)$ or $\neg Concession(g,j)$ for every $j < k$). In order to minimize $achiev$, we need to keep as few concessions at iteration k as possible in arguments.

We define the following order $\trianglerighteq$ over arguments:

$$a \trianglerighteq b \text{ if either } cert(a) < cert(b) \text{ or } [cert(a) = cert(b) \text{ and } achiev(a) < achiev(b)].$$

It is the lexicographic ordering, where we order first on the certainty levels and secondly on the achievement of goals (concessions made). We indeed prefer first to use as much certain knowledge as possible.

5 Application on the Motivating Example

5.1 Step 1: Initialisation Phase – Definition of Usages and Criteria

We assume that $T = 3$. Base $\mathcal{B}_{\mathcal{K}}(1)$ contains in particular the following knowledge

$(initAchievGoal(\text{landscape}, \text{high}), 2)$, $(limitAchievGoal(\text{landscape}, low), 2)$
$(initAchievGoal(\text{affordability}, \text{high}), 2)$, $(limitAchievGoal(\text{affordability}, low), 2)$
$(Goal(\text{affordability}, \text{high}) \rightarrow \mathbb{1}_{c_3}(200), 2)$, $(Concession(g,1), 2)$ $\forall g \in \mathcal{G}$
$(Goal(\text{affordability}, \text{medium}) \rightarrow \mathbb{1}_{c_3}(300), 2)$, $(Imp_{c_1}(2), 2)$
$(Goal(\text{affordability}, \text{low}) \rightarrow \mathbb{1}_{c_3}(400), 2)$, $(Imp_{c_2}(1), 1)$
$(Imp_{c_3}(2), 2)$, $(Imp_{c_4}(0), 1)$, $(\mathbb{1}_{c_1}(10), 1)$, $(\mathbb{1}_{c_2}(1), 1)$

John is not sure about the thresholds for criteria c_1 and c_2 and the importance of criteria c_2 and c_4, and assigns certainty 1 to these beliefs. There is only one argument $a_1 = \langle K_1, d_1 \rangle$, where $cert(a_1) = 4$, $K_1 \vdash U(d_1, 5)$ and $K_1 \vdash Acc(d_1)$. Hence d_1 is acceptable.

5.2 Step 2: Expertise on Sharpness

Base $\mathcal{B}_{\mathcal{K}}(2)$ contains in particular the following rules

$(Goal(\text{landscape}, \text{high}) \rightarrow \mathbb{1}_{c_1}(16), 2)$, $(Goal(\text{landscape}, \text{medium}) \rightarrow \mathbb{1}_{c_1}(14), 2)$
$(Goal(\text{landscape}, \text{low}) \rightarrow \mathbb{1}_{c_1}(12), 2)$
$(Goal(\text{landscape}, \text{high}) \vee Goal(\text{landscape}, \text{medium}) \rightarrow Imp_{c_2}(0), 2)$
$(Concession(g, 2), 2)\ \forall g \in \mathcal{G}$, $(\neg Concession(g, 2), 2)\ \forall g \in \mathcal{G}$

The argument with minimum certainty has $cert$ equal to 1. There is no acceptable decision with $cert$ equal to 1, when the DM does not make any concession. Then $a_2 = \langle K_2, d_3 \rangle$, with $\{Concession(\text{landscape}, 2), Concession(\text{affordability}, 2)\} \subseteq K_2$ and $cert(a_2) = 1$, is an argument, where we assume (under the linear tactic – see Section 2.2):

$$Conc_{\text{landscape}}(2, \text{high}, \text{low}) = \text{medium}$$
$$Conc_{\text{affordability}}(2, \text{high}, \text{low}) = \text{medium}$$

Hence the DM needs to make a concession on both landscape and affordability. Moreover, $K_2 \vdash U(d_3, 4)$. Hence we recover that d_3 is acceptable.

5.3 Step 3: Need for "Adventure-Proof" Camera

$\mathcal{B}_{\mathcal{G}}(3)$ is composed of $(\text{adventure}, 3)$, $(\text{landscape}, 2)$ and $(\text{affordibility}, 1)$. Base $\mathcal{B}_{\mathcal{K}}(3)$ contains in particular the following rules

$(initAchievGoal(\text{adventure}, \text{high}), 2)$, $(limitAchievGoal(\text{adventure}, \text{low}), 2)$
$(Goal(\text{adventure}, \text{high}) \rightarrow \mathbb{1}_{c_4}(1.5), 2)$, $(Goal(\text{adventure}, \text{low}) \rightarrow \mathbb{1}_{c_4}(0.5), 2)$
$(Concession(\text{affordability}, 2), 2)$, $(Concession(\text{landscape}, 2), 2)$
$(\neg Concession(\text{adventure}, 2), 2)$
$(Concession(g, 3), 2)\ \forall g \in \mathcal{G}$, $(\neg Concession(g, 3), 2)\ \forall g \in \mathcal{G}$

There are seven arguments with $cert$ equal to 0. At least one concession must be made at step 3 with these arguments. Considering the four arguments making the least concessions, we obtain the arguments $a_3 = \langle K_3, d_6 \rangle$, $a_3' = \langle K_3', d_3 \rangle$, $a_3'' = \langle K_3'', d_5 \rangle$, $a_3''' = \langle K_3''', d_6 \rangle$, with $cert(a_3) = cert(a_3') = cert(a_3'') = cert(a_3''') = 0$ and

$$Concession(\text{affordability}, 3) \in K_3$$
$$Concession(\text{adventure}, 3) \in K_3'$$
$$Concession(\text{landscape}, 3) \in K_3''$$
$$\{Concession(\text{affordability}, 3), Concession(\text{landscape}, 3)\} \subseteq K_3'''$$

with $achiev(a_3) = 1$, $achiev(a_3') = 3$, $achiev(a_3'') = 2$, $achiev(a_3''') = 3$. Hence a_3 is the most preferred argument according to $\trianglerighteq$ and thus option d_6 is selected. The process stops according to condition (12).

6 Related Work

Combination of MCDM and argumentation has an increased interest in AI in recent years. The first work on using argumentation for decision making is [7]. Even though this work uses a simple mechanism of pros and cons, it has been successfully applied to medicine applications [4]. A multi-criteria decision approach to decide among options is presented in [2], where criteria are fed with acceptable arguments pros and cons each option. Decision models are then applied to these sets of arguments. In [15], a logical language for represented a qualitative multi-criteria model is described. Some predicates are defined for each component/parameter of a decision model. Arguments can then be constructed.

Reference [9] is the first attempt to apply argumentation to assist the decision process. The main focus of this paper is on the relationship between the Decision Analyst (expert in decision techniques) and the Decision Maker (the client of the decision analyst). Moreover, the framework proposed by the authors allows specifying context-dependent multi-criteria decision frameworks (combining both the problem description and its formulation). A virtual selling agent helps a customer to choose an electronic devise that meets his needs in [5]. Firstly the profile of the customer is identified among a set of predefined profiles. The profile includes the intended usage by the customer of the electronic device. Then recommendations are generated until the customer accepts one or there is no recommendation left. A very simple multi-attribute decision model is used to compute the overall utility of each option from predefined scores allotted to each predefined profile on each attribute.

Compare to the previous works, our model focuses on the dynamics of decision process, and in particular the appearance of new individuals (experts) providing advices in order to help the decision maker to construct a decision model. We also consider the concessions the decision maker has to make during this dynamic process. In our view, decision aid is thus close to a negotiation process, except that the final decision is made only by one individual.

7 Conclusion and Perspectives

We have proposed an argumentation framework to represent a dynamic multi-criteria decision process. We aim at helping a DM who gathers information from different sources (expert review on Internet, ...) in order to increase his convictions and confidence in the best decision to take. At each iteration of the process, the DM integrates information from a new source, confronts it to his own knowledge base and makes partial decisions. The DM has some goals he wants to pursue (e.g. target price or usages on the electronic devise). The DM also defines intensity on the usage and more generally the required level of achievement on each goal. The experts basically explain what are the relevant criteria (with their importance and threshold value) given the goals and their associated level of achievement. Often, a DM starts with high expectations on many goals. Then he realizes that there is no suitable option. To go further, he needs to make concessions on his expectations. We have proposed a mechanism to suggest the less demanding concessions that yield acceptable options, in the spirit of what is done in negotiation protocols.

The current work can be improved in different ways. First of all, one can extent our framework to use more general models than the weighted sum. One only needs to define the acceptability function. For outranking models, this can be done in the spirit of ELECTRE TRI method (using a reference profile). Next, more refined argumentation frameworks could be used, for instance based on Dung like paradigm and computation of extensions. From an applicative side, our model is closer to human decision process. It allows representing the dynamics of decision making and in particular the evolution of the DM's convictions. We feel an implementation in the spirit of [7,4] could be possible.

References

1. Amgoud, L., Bonnefon, J.-F., Prade, H.: An argumentation-based approach to multiple criteria decision. In: Godo, L. (ed.) ECSQARU 2005. LNCS (LNAI), vol. 3571, pp. 269–280. Springer, Heidelberg (2005)
2. Amgoud, L., Prade, H.: Using arguments for making and explaining decisions. Artificial Intelligence 173, 413–436 (2009)
3. Bouyssou, D., Marchant, T., Pirlot, M., Tsoukiàs, A., Vincke, P.: Evaluation and decision models with multiple criteria: Stepping stones for the analyst. International Series in Operations Research and Management Science. Springer (2006)
4. Coulson, A.S., Glasspool, D., Fox, J., Emery, J.: Rags: A novel approach to computerized genetic risk assessment and decision support from pedigrees. Methods of Information in Medicine 40, 315–322 (2001)
5. Delecroix, F., Morge, M., Routier, J.-C.: A virtual selling agent which is persuasive and adaptive (2012)
6. Fatima, S., Jennings, N., Wooldridge, M.: An agenda-based framework for multi-issue negotiation. Artificial Intelligence Journal 152, 1–45 (2004)
7. Huang, J., Fox, J., Gordon, C., Jackson-Smale, A.: Symbolic decision support in medical car. Artificial Intelligence in Medicine 5, 415–430 (1993)
8. Osborne, M., Rubinstein, A.: A Course in Game Theory. MIT Press, Cambridge (1994)
9. Ouerdane, W., Dimopoulos, Y., Liapis, K., Moraitis, P.: Towards automating decision aiding through argumentation. Journal of Multi-Criteria Decision Analysis 18, 289–309 (2011)
10. Pruitt, D.: Negotiation Behaviour. Academic Press Inc., London (1982)
11. Rahwan, I., Amgoud, L.: An argumentation-based approach for practical reasoning. In: 5th International Joint Conference on Autonomous Agents & Multi Agent Systems, AAMAS 2006, Hakodate, Japan, pp. 347–354 (2006)
12. Raiffa, H.: The Art and Science of Negotiation. Harvard University Press, Cambridge (1982)
13. Simon, H.: A behavioural model of rational choice. Quarterly Journal of Economics 69, 99–118 (1955)
14. Tsoukiàs, A.: On the concept of decision aiding process. Annals of Operations Research 154, 3–27 (2007)
15. Visser, W., Hindriks, K.V., Jonker, C.M.: An argumentation framework for qualitative multicriteria preferences. In: Modgil, S., Oren, N., Toni, F. (eds.) TAFA 2011. LNCS, vol. 7132, pp. 85–98. Springer, Heidelberg (2012)

Conditional Beliefs in a Bipolar Framework

Jonathan Lawry and Trevor Martin

Department of Engineering Mathematics,
University of Bristol,
Bristol, UK
j.lawry@bris.ac.uk

Abstract. A framework for quantifying lower and upper bipolar belief is introduced, which incorporates aspects of stochastic and of semantic uncertainty as well as an indeterministic truth-model allowing for inherent linguistic vagueness at the propositional level. This is then extended to include lower and upper measures of conditional belief given information in the form of lower and upper truth-valuations. The properties of these measures are explored and their relationship with conditional belief in other uncertainty theories is highlighted.

1 Introduction

A defining feature of vague concepts is that they admit borderline cases which neither definitely satisfy the concept nor its negation. For example, there are some height values which would neither be classified as being absolutely *short* nor absolutely *not short*. For propositions involving vague concepts this naturally results in truth-gaps. In other words, there are cases in which a proposition is neither *absolutely true* nor *absolutely false* suggesting that a non-Tarskian notion of truth may be required to capture this aspect of vagueness. A model of this kind with distinct, although related, valuations for absolute truth and absolute falsity exhibits, what Dubois and Prade [1], refer to as *symmetric bivariate unipolarity*, whereby judgments are made according to two distinct evaluations on unipolar scales i.e. distinct evaluations about the truth value of a sentence and that of its negation. In the current context, we have a strong and a weak evaluation criterion where the former corresponds to *absolute truth* and the latter *not absolute falsity*. As with many examples of this type of bipolarity there is then a natural duality between the two evaluation criteria in that a proposition is absolutely true if and only if its negation is absolutely false.

The development of formal models incorporating truth-gaps has potentially important applications in artificial intelligence systems. For example, allowing for borderline cases can help to mitigate the risks associated with making forecasts [15]. In this context, a bipolar framework can form the basis of a decision theoretic model to enable natural language generation systems, such as automatic weather forecasters, to decide between different assertions with different levels of semantic precision, so as to minimize risk and maximize performance [5]. In multi-agent systems where agents need to reach consensus concerning a set of

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 364–375, 2013.

propositions, the use of borderline cases can allow agents to adapt their beliefs so as to reach a compromise with others, whilst maintaining a certain level of internal consistency [6]. Furthermore, in multi-agent dialogues a bipolar approach can help to distinguish between strong and weak viewpoints in opinion formation [8]. Another application area of growing importance is in the representation of so-called flexible specifications for adaptive autonomous systems. The deployment of autonomous systems in complex dynamic environments tends to naturally result in a tension between the requirement that the system's behaviour conforms to a predefined specification, and the need for it to be sufficiently flexible so as to cope with severe uncertainty and unexpected scenarios. For example, it might find itself in situations not envisaged by its designers, where all available actions result in some violation of its specification. In such cases, a more flexible form of specification may allow for some constraints to be only borderline satisfied in certain conditions. Furthermore, the blurring of concept boundaries in the interpretation would then permit some aspect of gradedness, potentially allowing the system to choose between different suboptimal possibilities.

In all of the above application areas the adequate representation of epistemic uncertainty combined with bipolarity is also of central importance. Typically we think of uncertainty as arising because of insufficient information about the state of the world. However, in the presence of vagueness there may also be semantic uncertainty due to partial knowledge of language conventions resulting in agents being unsure about conceptual boundaries. Here we extend bipolar belief measures, recently proposed in [7], which combine probabilistic uncertainty with truth-gaps as represented in Kleene's strong three-valued logic [4]. More specifically, the main contribution of this paper is the introduction of natural measures of conditional belief within this framework. We then discuss their properties and relate and contrast these measures to existing definitions of conditional belief in the literature such as in Dempster-Shafer theory and fuzzy logic.

An outline of the paper is as follows: Section 2 introduces valuation pairs as a bipolar truth-model based on Kleene's three-valued logic. Section 3 defines bipolar belief pairs in terms of probability distributions over the set of valuation pairs and shows their relationship to lower and upper membership functions in interval-valued fuzzy logic. Extending this idea, section 4 proposed definitions for conditional belief pairs and investigates their properties. Finally, in section 5 we have conclusions and further discussion of potential applications of the framework.

2 Valuation Pairs

In this section, we introduce valuation pairs as a bipolar model of truth which allows for the explicit representation of borderline cases. Typical examples are declarative sentences containing vague adjectives e.g. *low*, *tall*, *fast* etc, although truth-gaps can of course result from other sources of vagueness such as from verbs and nouns. We now propose to model truth-gaps by replacing a single binary, true or false, valuation on propositions with distinct lower and upper valuations

representing absolutely true and not absolutely false respectively. Borderline cases then correspond to those sentences in which the lower and upper valuation differ.

Let $\mathcal{L}$ be a language of propositional logic with connectives $\wedge$, $\vee$ and $\neg$ and propositional variables $P = \{p_1, \ldots, p_n\}$, and let $S\mathcal{L}$ denote the sentences of $\mathcal{L}$ as generated recursively from P by application of the connectives. A valuation pair on $S\mathcal{L}$ consists of two binary functions $\underline{v}$ and $\overline{v}$ representing lower and upper truth-values. The underlying idea is that $\underline{v}$ represents the strong criterion of *absolutely true* while $\overline{v}$ represents the weaker criteria of *not absolutely false*. In accordance with [11], we might think of a sentence being absolutely true as meaning that it can be uncontroversially asserted without any risk of censure, while being not absolutely false only means that it is acceptable to assert i.e. one can get away with such an assertion. For example, consider a witness in a court of law describing a suspect as being *short*. Depending on the actual height of the suspect this statement may be deemed as clearly true or clearly false, in which latter case the witness could be accused of perjury. However, there will also be an intermediate height range for which, while there may be doubt and differing opinions concerning the use of the description *short*, it would not be deemed as definitely inappropriate and hence the witness would not be viewed as committing perjury. In other words, for certain height values of the suspect, it may be acceptable to assert the statement p='the suspect was short', even though this statement would not be viewed as being absolutely true. One possible bipolar model of the concept *short* exhibiting such truth-gaps could be as follows: Let h be the height of the suspect and suppose that *short* is defined in terms of lower and upper thresholds $\underline{h} \leq \overline{h}$ on heights. In this case p is *absolutely true* if $h \leq \underline{h}$, *absolutely false* if $h > \overline{h}$ and *borderline* if $\underline{h} < h \leq \overline{h}$ (see figure 1).

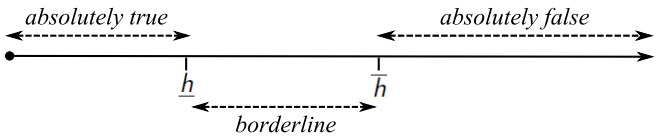


Fig. 1. A bipolar interpretation of the concept short

It is important to note that in this model truth-gaps corresponding to different lower and upper truth valuations are not the result of epistemic uncertainty concerning the state of the world but rather due to inherent flexibility in the underlying language conventions. In other words, a truth-gap (or middle truth-value in three-valued logic) does not represent an *uncertain* epistemic state. For example, given absolute certainty about suspect's height the proposition p may then be *known* to be borderline because of the inherent flexibility (or vagueness) in the definition of the concept short i.e. because $\underline{h} < h \leq \overline{h}$. The potential confusion resulting from applying many-valued logic to model epistemic uncertainty is highlighted by Dubois in [2]. In the sequel we shall emphasize the truth-value status of the intermediate case by using the term *borderline* rather than 'uncertain' or 'unknown' as originally suggested by Kleene [4].

Definition 1. *Kleene Valuation Pairs*
A Kleene valuation pair on $\mathcal{L}$ is a pair of functions $\boldsymbol{v} = (\underline{v}, \overline{v})$ where $\underline{v} : S\mathcal{L} \to \{0,1\}$ and $\overline{v} : S\mathcal{L} \to \{0,1\}$ such that $\underline{v} \leq \overline{v}$ and where $\forall \theta, \varphi \in S\mathcal{L}$ the following hold:

- $\underline{v}(\neg\theta) = 1 - \overline{v}(\theta)$ *and* $\overline{v}(\neg\theta) = 1 - \underline{v}(\theta)$
- $\underline{v}(\theta \wedge \varphi) = \min(\underline{v}(\theta), \underline{v}(\varphi))$ *and* $\overline{v}(\theta \wedge \varphi) = \min(\overline{v}(\theta), \overline{v}(\varphi))$
- $\underline{v}(\theta \vee \varphi) = \max(\underline{v}(\theta), \underline{v}(\varphi))$ *and* $\overline{v}(\theta \vee \varphi) = \max(\overline{v}(\theta), \overline{v}(\varphi))$

We use $\mathbb{V}$ to denote the set of all Kleene valuation pairs on $\mathcal{L}$.

The link to three-valued logic is clear when we view the three possible values of a valuation pair for a sentence as truth values i.e. $\mathbf{t} = (1,1)$ as absolutely true, $\mathbf{b} = (0,1)$ as borderline and $\mathbf{f} = (0,0)$ as absolutely false. From definition 1 we can then determine truth-tables for the connectives $\wedge$, $\vee$ and $\neg$ in terms of the truth-values $\{\mathbf{t}, \mathbf{b}, \mathbf{f}\}$ identical to those of Kleene's logic [4]. Shapiro [14] has recently proposed the use of Kleene's three-valued logic to model truth-gaps in vague predicates, arguing that Kleene's truth tables 'reflect the open-texture of vague predicates'. For example, if instead we were to adopt Lukasiewicz logic [10] this would mean that for two borderline propositional variables their conjunction would be absolutely false, even though neither conjunct was absolutely false. This would seem to be a totally unwarranted elimination of vagueness. One might of course consider a non-functional calculus for valuation pairs based, for example, on supervaluationist principles as explored in Lawry and Tang [5]. Another possibility would be to introduce many-valued logics with more than three truth-values. From the current perspective this would correspond to propositions being *borderline* to differing degrees. However, the representational utility of making such distinctions between borderline cases is not entirely clear, as is discussed in more details in [5].

Definition 2. *For $\theta, \varphi \in S\mathcal{L}$, θ and φ are equivalent, denoted $\theta \equiv \varphi$, if and only if $\forall \boldsymbol{v} \in \mathbb{V}$ $\boldsymbol{v}(\theta) = \boldsymbol{v}(\varphi)$.*

The following theorem identifies a number of well known equivalences from Kleene three-valued logic.

Theorem 1. *Important Equivalences [7]*
$\forall \theta, \varphi, \psi \in SL$ the following sentences are equivalent:

- *De Morgan's Laws:* $\neg(\theta \wedge \varphi) \equiv \neg\theta \vee \neg\varphi$ *and* $\neg(\theta \vee \varphi) \equiv \neg\theta \wedge \neg\varphi$
- *Double Negation:* $\neg(\neg\theta) \equiv \theta$
- *Idempotence:* $\theta \wedge \theta \equiv \theta$ *and* $\theta \vee \theta \equiv \theta$
- *Commutativity:* $\theta \vee \varphi \equiv \varphi \vee \theta$ *and* $\theta \wedge \varphi \equiv \varphi \wedge \theta$
- *Associativity:* $\theta \vee (\varphi \vee \psi) \equiv (\theta \vee \varphi) \vee \psi$ *and* $\theta \wedge (\varphi \wedge \psi) \equiv (\theta \wedge \varphi) \wedge \psi$
- *Distributivity:* $\theta \vee (\varphi \wedge \psi) \equiv (\theta \vee \varphi) \wedge (\theta \vee \psi)$ *and* $\theta \wedge (\varphi \vee \psi) \equiv (\theta \wedge \varphi) \vee (\theta \wedge \psi)$

Kleene valuation pairs do not completely satisfy the laws of non-contradiction and excluded middle in borderline cases. While it is the case that for any sentence $\varphi \in S\mathcal{L}$, $\underline{v}(\varphi \wedge \neg\varphi) = 0$ and $\overline{v}(\varphi \wedge \neg\varphi) = 1$ the same equalities do not necessarily

hold for the corresponding upper and lower valuations respectively. In fact, any such partial failure of the laws of non-contradiction and excluded middle exactly correspond with φ being a borderline case, as the following result shows.

Theorem 2. $\boldsymbol{v} \in \mathbb{V}$, $\boldsymbol{v}(\varphi) = \mathbf{b}$ *if and only if* $\overline{v}(\varphi \wedge \neg\varphi) = 1$ *if and only if* $\underline{v}(\varphi \vee \neg\varphi) = 0$.

Proof. ($\Rightarrow$) Suppose $\boldsymbol{v}(\varphi) = \mathbf{b}$ then $\underline{v}(\varphi) = 0 \Rightarrow \overline{v}(\neg\varphi) = 1$ and since also $\overline{v}(\varphi) = 1 \Rightarrow \overline{v}(\varphi \wedge \neg\varphi) = 1$. ($\Leftarrow$) $\overline{v}(\varphi \wedge \neg\varphi) = 1 \Rightarrow \overline{v}(\varphi) = 1$ and also $\Rightarrow \overline{v}(\neg\varphi) = 1 \Rightarrow \underline{v}(\varphi) = 0$. Furthermore, by duality and de Morgan's law (theorem 2) it follows that $\underline{v}(\varphi \vee \neg\varphi) = 0$ if and only if $\overline{v}(\varphi \wedge \neg\varphi) = 1$ as required.

We now define *semantic precision* as a natural partial ordering on $\mathbb{V}$. This concerns the situation in which one valuation pair admits more borderline cases than another but where otherwise their truth-valuations agree. More formally, valuation pair $\boldsymbol{v}_1$ is less semantically precise than $\boldsymbol{v}_2$ if they disagree only for some subset of sentences of $\mathcal{L}$, which being identified as either absolutely true or absolutely false by $\boldsymbol{v}_2$, are classified as borderline by $\boldsymbol{v}_1$.

Definition 3. *Semantic Precision*
$\boldsymbol{v}_1 \preceq \boldsymbol{v}_2$ *iff* $\forall\theta \in S\mathcal{L}$ $\underline{v_1}(\theta) \leq \underline{v_2}(\theta)$ *and* $\overline{v_1}(\theta) \geq \overline{v_2}(\theta)$.

Shapiro [14] proposed essentially the same ordering of interpretations which he refers to as *sharpening* i.e. $\boldsymbol{v}_1 \preceq \boldsymbol{v}_2$ means that $\boldsymbol{v}_2$ extends or sharpens $\boldsymbol{v}_1$. Here we shall refer to $\preceq$ as the *semantic precision* ordering on valuation pairs whereby, if $\boldsymbol{v}_1 \preceq \boldsymbol{v}_2$ then $\boldsymbol{v}_1$ tends to classify more sentences of $\mathcal{L}$ as *borderline* than $\boldsymbol{v}_2$. In other words, one might think of $\preceq$ as ordering valuation pairs according to their relative vagueness.

3 Belief Pairs

Within the proposed bipolar framework, uncertainty concerning the sentences of $\mathcal{L}$ effectively corresponds to uncertainty as to which is the correct Kleene valuation pair for $\mathcal{L}$. In practice, there are likely to be many different sources of this uncertainty, however one natural division of uncertainty types is as follows:

- *Semantic uncertainty* about the linguistic conventions defining concepts relevant to the sentences of $\mathcal{L}$. For example, an agent may be uncertain as to whether or not a proposition such as 'the suspect is short' is *absolutely true* or *not absolutely false* even if the suspect's height h is known precisely. For instance, this might manifest itself in terms of uncertainty about the exact values of the thresholds $\underline{h}$ and $\overline{h}$ (see figure 1). This uncertainty naturally arises from the distributed manner in which language is learnt through communications with other agents across a population of interacting agents.
- *Stochastic uncertainty* arising from a lack of knowledge concerning the state of the world. For example, being uncertain about the suspect's height h in the proposition 'the suspect is short'.

In general we view uncertainty as being epistemic in nature, resulting from a lack of knowledge concerning either, the state of the world to which propositions refer, or the linguistic conventions governing the assertability of propositions as part of communications. Viewing semantic uncertainty as being epistemic in nature requires that agents make the assumption that there is a correct underlying interpretation of the language $\mathcal{L}$, but about which they may be uncertain. This is a weaker version of the epistemic theory of vagueness as expounded by Timothy Williamson [16] referred to as the *epistemic stance* [9]. Williamson's theory assumes that for the extension of a vague concept there is a precise but unknown boundary between it and the extension of its negation. In contrast the epistemic stance corresponds to the more pragmatic view that individuals, when faced with decision problems about what to assert, find it useful as part of a decision making strategy to simply *assume* that there is an underlying correct interpretation of $\mathcal{L}$. In other words, when deciding what to assert agents behave as is the epistemic theory is correct. Another difference between the epistemic theory and our current approach is that the former assumes that the underlying truth model is classical while here we assume a bipolar model which can exhibit truth-gaps.

In the following definition we assume that uncertainty is quantified by a probability measure w on the set of Kleene valuation pairs $\mathbb{V}$.

Definition 4. *Kleene Belief Pairs [7]*
Let $\mathbb{V}$ be the set of all Kleene valuation pairs on $\mathcal{L}$ and let w be a probability distribution defined on $\mathbb{V}$ so that $w(\boldsymbol{v})$ is the agent's subjective belief that $\boldsymbol{v}$ is the true valuation pair for $\mathcal{L}$. Then $\boldsymbol{\mu} = (\underline{\mu}, \overline{\mu})$ is a Kleene belief pair where $\forall \theta \in S\mathcal{L}$, $\underline{\mu}(\theta) = w(\{\boldsymbol{v} \in \mathbb{V} : \underline{v}(\theta) = 1\})$ and $\overline{\mu}(\theta) = w(\{\boldsymbol{v} \in \mathbb{V} : \overline{v}(\theta) = 1\})$.

There is a clear rationality argument for defining belief measures in this manner when Kleene valuation pairs are the underlying truth model for $\mathcal{L}$. From a general result due to Paris [12], it follows that an agent can only avoid Dutch books where the outcomes of bets are dependent on lower (upper) Kleene valuations if their belief measures on $S\mathcal{L}$ correspond to lower (upper) belief measures as given in definition 4. This idea is explored in more detail in Lawry and Tang [5] in the context of lower and upper bets. The following theorem highlights a number of properties of Kleene belief pairs, including additivity. The latter property in particular distinguishes Kleene Belief pairs from Dempster-Shafer belief and plausibility measures [13] on $S\mathcal{L}$ which are not, in general, additive.

Theorem 3. *For all $\theta, \varphi \in S\mathcal{L}$, the following hold:*

- $\underline{\mu}(\theta) \leq \overline{\mu}(\theta)$
- $\underline{\mu}(\neg\theta) = 1 - \overline{\mu}(\theta)$ *and* $\overline{\mu}(\neg\theta) = 1 - \underline{\mu}(\theta)$.
- $\underline{\mu}(\theta \vee \varphi) = \underline{\mu}(\theta) + \underline{\mu}(\varphi) - \underline{\mu}(\theta \wedge \varphi)$ *and* $\overline{\mu}(\theta \vee \varphi) = \overline{\mu}(\theta) + \overline{\mu}(\varphi) - \overline{\mu}(\theta \wedge \varphi)$

It is also interesting to note that a special case of Kleene belief pairs has the same calculus as the interval (or type 2) fuzzy membership functions proposed by Zadeh [17]. This is the case of Kleene belief pairs in which there is only uncertainty about the level of semantic precision of the valuation pair. More formally we have the following result:

Theorem 4. *[7] Let w be a probability distribution on $\mathbb{V}$ for which $\{\boldsymbol{v} \in \mathbb{V} : w(\boldsymbol{v}) > 0\} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\}$ can be ordered such that $\boldsymbol{v}_1 \preceq \boldsymbol{v}_2 \ldots \preceq \boldsymbol{v}_k$. In this case $\boldsymbol{\mu}$ satisfies the following properties; $\forall \theta, \varphi \in S\mathcal{L}$,*

$\underline{\mu}(\theta \wedge \varphi) = \min(\underline{\mu}(\theta), \underline{\mu}(\varphi))$ *and* $\overline{\mu}(\theta \wedge \varphi) = \min(\overline{\mu}(\theta), \overline{\mu}(\varphi))$
$\underline{\mu}(\theta \vee \varphi) = \max(\underline{\mu}(\theta), \underline{\mu}(\varphi))$ *and* $\overline{\mu}(\theta \vee \varphi) = \max(\overline{\mu}(\theta), \overline{\mu}(\varphi))$

Example 1. Recall the example from section 2 concerning the proposition $p =$'the suspect is short', where the concept short is defined by two thresholds on height $0 \leq \underline{h} \leq \overline{h}$, so that an individual is absolutely *short* if their height is less than or equal to $\underline{h}$ and absolutely not short if their height is greater than $\overline{h}$. Hence, if the suspect's height is known to be h then an agent's beliefs about the interpretation of $\mathcal{L}$ can be modelled by a valuation pair $\boldsymbol{v}$ such that:

$\underline{v}(p) = 1$ if and only if $h \leq \underline{h}$ and $\overline{v}(p) = 1$ if and only if $h \leq \overline{h}$

We might further assume, perhaps reasonably in this case, that the agent's semantic uncertainty with regard to p is limited to uncertainty about the actual values of the thresholds $\underline{h}$ and $\overline{h}$. Further suppose that the agent's beliefs about these thresholds is represented by a joint probability density function f on $(\underline{h}, \overline{h})$ satisfying:

$$\int_0^\infty \int_{\underline{h}}^\infty f(\underline{h}, \overline{h}) \, \mathrm{d}\overline{h} \, \mathrm{d}\underline{h} = 1$$

Based on this the agent can define a lower measure of their belief in p, $\underline{\mu}(p)$, corresponding to the probability that the lower threshold $\underline{h} \geq h$ and similarly and upper measure, $\overline{\mu}(p)$, corresponding to the probability that the upper threshold $\overline{h} \geq h$ i.e.

$$\underline{\mu}(p) = \int_h^\infty \int_{\underline{h}}^\infty f(\underline{h}, \overline{h}) \, \mathrm{d}\overline{h} \, \mathrm{d}\underline{h} \text{ and } \overline{\mu}(p) = \int_h^\infty \int_0^{\overline{h}} f(\underline{h}, \overline{h}) \, \mathrm{d}\underline{h} \, \mathrm{d}\overline{h}$$

Now suppose that in this case the agent believes that $\underline{h}$ and $\overline{h}$ are independent variables both with triangular distributions centered around $130cm$ and $150cm$ respectively. More specifically; $f(\underline{h}, \overline{h}) = f_1(\underline{h}) \times f_2(\overline{h})$ where

$$f_1(\underline{h}) = \begin{cases} \frac{\underline{h}-120}{100} : \underline{h} \in [120, 130) \\ \frac{140-\underline{h}}{100} : \underline{h} \in [130, 140] \\ 0 : \text{ otherwise} \end{cases} \quad \text{and } f_2(\overline{h}) = \begin{cases} \frac{\overline{h}-140}{100} : \overline{h} \in [140, 150) \\ \frac{160-\overline{h}}{100} : \overline{h} \in [150, 160] \\ 0 : \text{ otherwise} \end{cases}$$

In this case the resulting values for $\underline{\mu}(p)$ and $\overline{\mu}(p)$ are shown in figure 2 as height h varies. Similarly, figure 3 shows the agent's belief that p is a borderline proposition, as quantified by $\overline{\mu}(p) - \underline{\mu}(p)$, for different values of h.

We can also consider the possibility that the agent is uncertain about the value of suspect's height. Suppose that the agent's knowledge about h is characterised

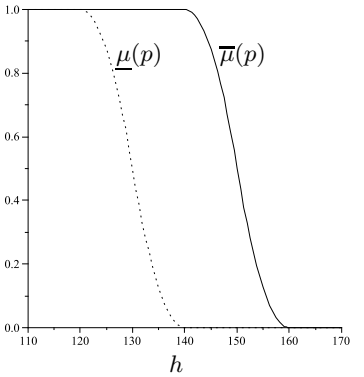


Fig. 2. Lower and upper belief values for proposition p as the suspect's height h varies

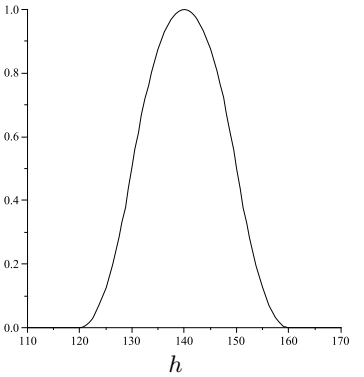


Fig. 3. Belief that p is a borderline proposition, given by $\overline{\mu}(p) - \underline{\mu}(p)$, as h varies

by a probability density function g and further suppose that h is taken to be independent of the thresholds $\underline{h}$ and $\overline{h}$. This additional uncertainty can then be included in the calculation of the lower and upper belief measures as follows:

$$\underline{\mu}(p) = \int_0^\infty \int_h^\infty \int_{\underline{h}}^\infty f(\underline{h}, \overline{h}) g(h) \, \mathrm{d}\overline{h} \, \mathrm{d}\underline{h} \, \mathrm{d}h \text{ and}$$

$$\overline{\mu}(p) = \int_0^\infty \int_h^\infty \int_0^{\overline{h}} f(\underline{h}, \overline{h}) g(h) \, \mathrm{d}\underline{h} \, \mathrm{d}\overline{h} \, \mathrm{d}h$$

For example, if g is a normal distribution with mean 140 and standard deviation 7 then $\underline{\mu}(p) = 0.1092$ and $\overline{\mu}(p) = 0.8908$.

4 Conditional Belief Pairs

In this section we propose a conditioning model by which agents can update their subjective belief pairs on the basis of new information concerning the absolute truth and absolute falsity of sentences in $\mathcal{L}$. In view of the inherently probabilistic nature of belief pairs, one obvious method is based on conditional probabilities. For this approach we assume that new knowledge takes the form of lower and upper valuation constraints, which it is then assumed that the *correct* valuation for $\mathcal{L}$ must satisfy. From the perspective of the above discussion on uncertainty in a bipolar context, we can think of such constraints as providing new information both about the state of the world and about the underlying interpretation of $\mathcal{L}$. This knowledge allows us to define conditional lower and upper belief measures by determining a posterior distribution on valuation pairs from the prior w, according to the standard definition of conditional probability

Definition 5. *Conditional Belief Pairs*
Suppose an agent obtains new knowledge regarding the assertability of sentences in $S\mathcal{L}$ in the form of a set K of constraints on lower and upper valuations of the following form:

$$K = \{\underline{v}(\theta_1) = 1, \ldots, \underline{v}(\theta_t) = 1, \overline{v}(\varphi_1) = 1, \ldots, \overline{v}(\varphi_s) = 1\}$$

Then we define lower and upper conditional belief measures conditional on K as follows:

$$\underline{\mu}(\theta|K) = \frac{w(\{\boldsymbol{v} \in \mathbb{V}(K) : \underline{v}(\theta) = 1\})}{w(\mathbb{V}(K))} \text{ and } \overline{\mu}(\theta|K) = \frac{w(\{\boldsymbol{v} \in \mathbb{V}(K) : \overline{v}(\theta) = 1\})}{w(\mathbb{V}(K))}$$

where $\mathbb{V}(K) \subseteq \mathbb{V}$ denotes the set of Kleene valuation pairs on $\mathcal{L}$ which satisfy the constraints K.

A possible source of knowledge constraints of the form given in definition 5, is from strong and weak assertion in agent dialogues [8]. For example, a witness might describe the suspect as 'absolutely short' or 'definitely short'. Alternatively, they might only be prepared to say that the suspect was 'possibly short' or 'short-ish'. The former might be regarded as strong assertions concerning the proposition p ='the suspect is tall' corresponding to the knowledge that $\boldsymbol{v}(p) = \mathbf{t}$. In contrast, the latter are weak assertions corresponding to $\boldsymbol{v}(p) \neq \mathbf{f}$ and $\boldsymbol{v}(p) = \mathbf{b}$ respectively. One can then envisage a knowledge base K as in definition 5, being derived from a dialogue with other agents consisting of such strong and weak assertions.

We now consider the special cases where $K = \{\underline{v}(\varphi) = 1\}$, $K = \{\overline{v}(\varphi) = 1\}$ and $K = \{\underline{v}(\varphi) = 0, \overline{v}(\varphi) = 1\}$ for some sentence $\varphi \in S\mathcal{L}$. Notice, that these correspond to the knowledge that $\boldsymbol{v}(\varphi) = \mathbf{t}$, $\boldsymbol{v}(\varphi) \neq \mathbf{f}$ and $\boldsymbol{v}(\varphi) = \mathbf{b}$ respectively.

Theorem 5

$$\underline{\mu}(\theta|\underline{v}(\varphi) = 1) = \frac{\underline{\mu}(\theta \wedge \varphi)}{\underline{\mu}(\varphi)} \text{ and } \overline{\mu}(\theta|\underline{v}(\varphi) = 1) = \frac{\overline{\mu}(\theta \vee \neg\varphi) - \overline{\mu}(\neg\varphi)}{1 - \overline{\mu}(\neg\varphi)}$$

Proof

$$\forall \theta, \varphi \in S\mathcal{L},\ \underline{\mu}(\theta|\underline{v}(\varphi) = 1) = \frac{w(\{\boldsymbol{v} \in \mathbb{V} : \underline{v}(\theta) = 1, \underline{v}(\varphi) = 1\})}{w(\{\boldsymbol{v} \in \mathbb{V} : \underline{v}(\varphi) = 1\})}$$

$$= \frac{w(\{\boldsymbol{v} \in \mathbb{V} : \underline{v}(\theta \wedge \varphi) = 1\})}{w(\{\boldsymbol{v} \in \mathbb{V} : \underline{v}(\varphi) = 1\})} \text{ by definition 1 } = \frac{\underline{\mu}(\theta \wedge \varphi)}{\underline{\mu}(\varphi)} \text{ by definition 4.}$$

In addition, by duality we have that:

$$\overline{\mu}(\theta|\underline{v}(\varphi) = 1) = 1 - \underline{\mu}(\neg\theta|\underline{\varphi} = 1) \text{by the above } = 1 - \frac{\underline{\mu}(\neg\theta \wedge \varphi)}{\underline{\mu}(\varphi)}$$

$$= \frac{\underline{\mu}(\varphi) - \underline{\mu}(\neg\theta \wedge \varphi)}{\underline{\mu}(\varphi)} = \frac{1 - \overline{\mu}(\neg\varphi) - 1 + \overline{\mu}(\neg(\neg\theta \wedge \varphi))}{1 - \overline{\mu}(\neg\varphi)}$$

$$= \frac{\overline{\mu}(\theta \vee \neg\varphi) - \overline{\mu}(\neg\varphi)}{1 - \overline{\mu}(\neg\varphi)} \text{ by de Morgan's law (theorem 1)}$$

Theorem 6

$$\underline{\mu}(\theta|\overline{v}(\varphi)=1)=\frac{\underline{\mu}(\theta\vee\neg\varphi)-\underline{\mu}(\neg\varphi)}{1-\underline{\mu}(\neg\varphi)} \text{ and } \overline{\mu}(\theta|\overline{v}(\varphi)=1)=\frac{\overline{\mu}(\theta\wedge\varphi)}{\overline{\mu}(\varphi)}$$

Proof. Similar to theorem 5.

Notice that the lower and upper conditions in theorem 6 have the same definition relative to the underlying belief measures as conditional belief and plausibility in Dempster-Shafer theory [13]. However, recall that Kleene belief pairs are not Dempster Shafer measures since, for example, they satisfy additivity (see theorem 3).

Theorem 7

$$\underline{\mu}(\theta|\boldsymbol{v}(\varphi)=\mathbf{b})=\frac{\underline{\mu}(\theta\vee\varphi\vee\neg\varphi)-\underline{\mu}(\varphi\vee\neg\varphi)}{1-\underline{\mu}(\varphi\vee\neg\varphi)} \text{ and } \overline{\mu}(\theta|\boldsymbol{v}(\varphi)=\mathbf{b})=\frac{\overline{\mu}(\theta\wedge\varphi\wedge\neg\varphi)}{\overline{\mu}(\varphi\wedge\neg\varphi)}$$

Proof

$$\overline{\mu}(\theta|\boldsymbol{v}(\varphi)=\mathbf{b})=\frac{w(\{\boldsymbol{v}:\boldsymbol{v}(\varphi)=\mathbf{b},\overline{\boldsymbol{v}}(\theta)=1\})}{w(\{\boldsymbol{v}:\boldsymbol{v}(\varphi)=(0,1)\})}=\frac{w(\{\boldsymbol{v}:\overline{\boldsymbol{v}}(\varphi\wedge\neg\varphi)=1,\overline{\boldsymbol{v}}(\theta)=1\})}{w(\{\boldsymbol{v}:\boldsymbol{v}(\varphi)=(0,1)\})}$$

$$\text{by theorem 2 } =\frac{w(\{\boldsymbol{v}:\overline{\boldsymbol{v}}(\theta\wedge\varphi\wedge\neg\varphi)=1\})}{w(\{\boldsymbol{v}:\overline{v}(\varphi\wedge\neg\varphi)=1\})}=\frac{\overline{\mu}(\theta\wedge\varphi\wedge\neg\varphi)}{\overline{\mu}(\varphi\wedge\neg\varphi)}$$

Also we have that,

$$\underline{\mu}(\theta|\boldsymbol{v}(\varphi)=\mathbf{b})=\frac{w(\{\boldsymbol{v}:\underline{\boldsymbol{v}}(\theta)=1,\boldsymbol{v}(\varphi)=\mathbf{b}\})}{w(\{\boldsymbol{v}:\boldsymbol{v}=\mathbf{b}\})}=\frac{w(\{\boldsymbol{v}:\underline{\boldsymbol{v}}(\theta)=1,\underline{\boldsymbol{v}}(\varphi\vee\neg\varphi)=0\})}{w(\{\boldsymbol{v}:\underline{\boldsymbol{v}}(\varphi\vee\neg\varphi)=0\})}$$

$$\text{by theorem 2} =\frac{w(\{\boldsymbol{v}:\underline{\boldsymbol{v}}(\varphi\vee\neg\varphi)=0\})-w(\{\boldsymbol{v}:\underline{\boldsymbol{v}}(\theta)=0,\underline{\boldsymbol{v}}(\varphi\vee\neg\varphi)=0\})}{w(\{\boldsymbol{v}:\underline{\boldsymbol{v}}(\varphi\vee\neg\varphi)=0\})}$$

$$=\frac{w(\{\boldsymbol{v}:\underline{\boldsymbol{v}}(\varphi\vee\neg\varphi)=0\})-w(\{\boldsymbol{v}:\underline{\boldsymbol{v}}(\theta\vee\varphi\vee\neg\varphi)=0\})}{w(\{\boldsymbol{v}:\underline{\boldsymbol{v}}(\varphi\vee\neg\varphi)=0\})}$$

$$=\frac{1-\underline{\mu}(\varphi\vee\neg\varphi)-(1-\underline{\mu}(\theta\vee\varphi\vee\neg\varphi))}{1-\underline{\mu}(\varphi\vee\neg\varphi)}=\frac{\underline{\mu}(\theta\vee\varphi\vee\neg\varphi)-\underline{\mu}(\varphi\vee\neg\varphi)}{1-\underline{\mu}(\varphi\vee\neg\varphi)}$$

Corollary 1. *Let w be a probability distribution on $\mathbb{V}$ for which $\{\boldsymbol{v}\in\mathbb{V}:w(\boldsymbol{v})>0\}=\{\boldsymbol{v}_1,\ldots,\boldsymbol{v}_k\}$ can be ordered such that $\boldsymbol{v}_1\preceq\boldsymbol{v}_2\ldots\preceq\boldsymbol{v}_k$. Then for $\theta,\varphi\in S\mathcal{L}$ it holds that:*

$$\underline{\mu}(\theta|\underline{\boldsymbol{v}}(\varphi)=1)=\begin{cases}\frac{\underline{\mu}(\theta)}{\underline{\mu}(\varphi)} & :\underline{\mu}(\theta)\leq\underline{\mu}(\varphi)\\ 1 & :\textit{otherwise}\end{cases} \textit{ and}$$

$$\overline{\mu}(\theta|\underline{\boldsymbol{v}}(\varphi)=1)=\begin{cases}\frac{\overline{\mu}(\theta)+\underline{\mu}(\varphi)-1}{\underline{\mu}(\varphi)} & :\overline{\mu}(\theta)+\underline{\mu}(\varphi)\geq 1\\ 0 & :\textit{otherwise}\end{cases}$$

$$\underline{\mu}(\theta|\overline{v}(\varphi)=1)=\begin{cases}\frac{\underline{\mu}(\theta)+\overline{\mu}(\varphi)-1}{\overline{\mu}(\varphi)} & :\underline{\mu}(\theta)+\overline{\mu}(\varphi)\geq 1\\ 0 & :\textit{otherwise}\end{cases} \textit{and}$$

$$\overline{\mu}(\theta|\overline{v}(\varphi)=1)=\begin{cases}\frac{\overline{\mu}(\theta)}{\overline{\mu}(\varphi)} & :\overline{\mu}(\theta)\leq\overline{\mu}(\varphi)\\ 1 & :\textit{otherwise}\end{cases}$$

Proof. Follows immediately from theorem 4 and theorems 5 and 6.

Notice that in corollary 1 $\underline{\mu}(\theta|\underline{v}(\varphi) = 1)$ and $\overline{\mu}(\theta|\overline{v}(\varphi) = 1)$ correspond to the Goguen implication operator [3] applied to the lower and upper belief values of θ and φ respectively.

Example 2. Recall the proposition $p =$'the suspect is short' as described in example 1. Now consider an additional proposition $q =$'the suspect is very short' where the concept *very short* is defined by lower and upper height thresholds $\underline{h}'$ and $\overline{h}'$. Further suppose that these thresholds are dependent on the thresholds of *short*, according to $\underline{h}' = 0.9\underline{h}$ and $\overline{h}' = 0.9\overline{h}$. Further suppose that, as in example 1, the semantic and stochastic uncertainty is modelled by the joint distribution f on the threshold $\underline{h}$ and $\overline{h}$, and the distribution g on h respectively. Now suppose that the agent learns that the suspect is *borderline very short*. How does this change their level of belief that the suspect is short? In other words, what are the values of the conditional beliefs $\underline{\mu}(p|\boldsymbol{v}(q) = \mathbf{b})$ and $\overline{\mu}(p|\boldsymbol{v}(q) = \mathbf{b})$? Notice that given the above definition of $\overline{h}'$ then it follows that $h \leq \overline{h}'$ implies that $h \leq \overline{h}$ and hence $\overline{\mu}(p|\boldsymbol{v}(q) = \mathbf{b}) = 1$. Now in this example $w(\{\boldsymbol{v} : \boldsymbol{v}(q) = b\})$ corresponds to the probability that $\underline{h}' \leq h \leq \overline{h}'$ or alternatively that $\overline{h} \geq \frac{h}{0.9}$ and $\underline{h} \leq \frac{h}{0.9}$. Hence, we have that:

$$w(\{\boldsymbol{v} : \boldsymbol{v}(q) = b\}) = \int_0^\infty \int_0^{\frac{h}{0.9}} \int_{\frac{h}{0.9}}^\infty f(\underline{h}, \overline{h}) g(h) \, \mathrm{d}\overline{h} \, \mathrm{d}\underline{h} \, \mathrm{d}h = 0.2625$$

Similarly we have that:

$$w(\{\boldsymbol{v} : \underline{v}(p) = 1, \boldsymbol{v}(q) = b\}) = \int_0^\infty \int_h^{\frac{h}{0.9}} \int_{\frac{h}{0.9}}^\infty f(\underline{h}, \overline{h}) g(h) \, \mathrm{d}\overline{h} \, \mathrm{d}\underline{h} \, \mathrm{d}h = 0.0888$$

Hence,

$$\underline{\mu}(p|\boldsymbol{v}(q) = \mathbf{b}) = \frac{0.0888}{0.2625} = 0.3383$$

In comparison with the values obtained in example 1 we see that both $\underline{\mu}(p|\boldsymbol{v}(q) = b) > \underline{\mu}(p)$ and $\overline{\mu}(p|\boldsymbol{v}(q) = b) > \overline{\mu}(p)$. Clearly then, learning that q is a borderline case is informative when trying to determine the truth value of p. This emphasises the difference in terms of conditioning between the two distinct interpretations of truth-gaps (or middle truth-values) either as being borderline cases due to inherent vagueness or as representing epistemic ignorance. Indeed, if all we were to learn was that the truth value of q was *unknown* then this would tell us nothing about the truth-value of p, and therefore conditioning would not result in any change to belief values.

5 Conclusion and Discussion

In this paper we have proposed definitions for lower and upper conditional belief pairs, extending the framework introduced in [7]. The properties of these measures has been investigated and the relationship to conditional belief in existing uncertainty theories has been highlighted.

The belief pairs framework, incorporating the conditional measures proposed in this paper, is sufficiently rich to capture aspects of both stochastic and semantic uncertainty together with indeterminism in the underlying truth model. This can permit the definition of more flexible rules and specifications for intelligent autonomous systems, as well as providing an enhanced model of decision making in the presence of both uncertainty and conceptual vagueness. For example, one can envisage flexible requirements concerning the relationship between a pair of propositions p and q which include the requirement that p must be *absolutely true* in those circumstances in which q is only *borderline true*. Furthermore, in the presence of significant uncertainty probabilistic requirements may be more appropriate in the form of constraints on lower and upper condition beliefs e.g. $\underline{\mu}(p|\boldsymbol{v}(q) = b) \geq \alpha$ for a suitable confidence level α. Future work will aim to explore the application of the belief pairs framework to the formal representation of flexible specifications and their verification.

Acknowledgements. This work is partially funded by EPSRC grant EP/J01205X/1.

References

1. Dubois, D., Prade, H.: An Introduction to Bipolar Representations of Information and Preference. Int. Journal of Intelligent Systems 23(8), 866–877 (2008)
2. Dubois, D.: On Ignorance and Contradiction Considered as Truth-Values. Logic Journal of the IGPL 16(2), 195–216 (2008)
3. Goguen, J.A.: The Logic of Inexact Concepts. Synthese 19, 325–373 (1968)
4. Kleene, S.C.: Introduction to Metamathematics. D. Van Nostrand Company Inc., Princeton (1952)
5. Lawry, J., Tang, Y.: On Truth-gaps, Bipolar Belief and the Assertability of Vague Propositions. Artificial Intelligence 191-192, 20–41 (2012)
6. Lawry, J., Dubois, D.: A Bipolar Framework for Combining Beliefs about Vague Propositions. In: Proceedings of 13th International Conference on Principles of Knowledge Representation and Reasoning, pp. 530–540 (2012)
7. Lawry, J., González-Rodríguez, I.: A Bipolar Model of Assertability and Belief. International Journal of Approximate Reasoning 52, 76–91 (2011)
8. Lawry, J.: Imprecise Bipolar Belief Measures Based on Partial Knowledge from Agent Dialogues. In: Deshpande, A., Hunter, A. (eds.) SUM 2010. LNCS, vol. 6379, pp. 205–218. Springer, Heidelberg (2010)
9. Lawry, J.: Appropriateness Measures: An Uncertainty Model for Vague Concepts. Synthese 161(2), 255–269 (2008)
10. Lukasiewicz, J.: O logice trojwartosciowej (On three-valued logic). Ruch Filozoficzny 5, 170–171 (1920)
11. Parikh, R.: Vague Predicates and Language Games. Theoria XI(27), 97–107 (1996)
12. Paris, J.B.: A Note on the Dutch Book Method. In: Proceedings of ISIPTA 2001, Ithaca, New York (2001)
13. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press (1976)
14. Shapiro, S.: Vagueness in Context. Oxford University Press (2006)
15. van Deemter, K.: Utility and Language Generation: The Case of Vagueness. Journal of Philosophical Logic 38, 607–632 (2009)
16. Williamson, T.: Vagueness. Routledge (1994)
17. Zadeh, L.A.: The Concept of a Linguistic Variable and its Application to Approximate Reasoning: I. Information Sciences 8, 199–249 (1975)

Detecting Marginal and Conditional Independencies between Events and Learning Their Causal Structure

Jan Lemeire[1,4], Stijn Meganck[1,3], Albrecht Zimmermann[2], and Thomas Dhollander[3]

[1] ETRO Department, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium
{jan.lemeire,stijn.meganck}@vub.ac.be
[2] Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3001 Leuven, Belgium
albrecht.zimmermann@cs.kuleuven.be
[3] D square N.V., Kempische Steenweg 297 bus 3, 3500 Hasselt, Belgium
thomas.dhollander@dsquare.be
[4] iMinds, Department of Future Media and Imaging, Gaston Crommenlaan 8 (box 102), 9050 Ghent Belgium

Abstract. Consider data given as a sequence of events, where each event has a timestamp and is of a specific type. We introduce a test for detecting marginal independence between events of two given types and for conditional independence when conditioned on one type. The independence test is based on comparing the delays between two successive events of the given types with the delays that would occur in the independent situation. We define a Causal Event Model (CEM) for modeling the event-generating mechanisms. The model is based on the assumption that events are either spontaneous or caused by others and that the causal mechanisms depend on the event type. The causal structure is defined by a directed graph which may contain cycles. Based on the independence test, an algorithm is designed to uncover the causal structure. The results show many similarities with Bayesian network theory, except that the order of events has to be taken into account. Experiments on simulated data show the accuracy of the test and the correctness of the learning algorithm when assumed that the spontaneous events are generated by a Poisson process.

1 Introduction

In this paper we consider the following problem. The data is a sequence of events $\mathcal{E} = \langle (E_1, t_1), (E_2, t_2), \ldots \rangle$ where E_i represents an event type and t_i, the time of occurrence (also called timestamp) of the ith event, is a real value $\in [0, T]$, with T the end time of the sequence. E_i take values from a finite set of event types, the event domain $\mathcal{D}$. Fig. 1 shows an example event sequence with $\mathcal{D} = \{A, B, C, D\}$. When there is no confusion possible we denote events (E_i, t_i) with lower case e_i. Event types are denoted with upper case and sets with boldface letters. The question is to infer (1) independencies and (2) causal relations between the events.

If events of type A can cause events of type B, which we write as $A \rightarrow B$, then sequences $\langle t_{A_1}, t_{A_2} \ldots t_{A_k} \rangle$ and $\langle t_{B_1}, t_{B_2} \ldots t_{B_l} \rangle$ are correlated, where t_{A_i} and t_{B_j} are the timestamps of the A and B events respectively. We want a test to identify such correlation. We also want a test to identify conditional independencies. For causal model

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 376–387, 2013.

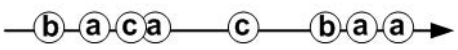


Fig. 1. Example event sequence

$A \rightarrow C \rightarrow B$, A is independent from B when conditioned on C, which we write as $A \perp\!\!\!\perp B|C$.

The problem has extensively been studied for series of continuous-valued dynamic variables, see for instance Granger causality [1]. Methods for analyzing sequences of events, on the other hand, have been studied in the data mining community. The main technique is episode mining where an *episode* is defined as an ordered tuple of events. Occurrences of episodes are counted and highly-frequent episodes are considered as relevant.

The independence test we propose here is based on the information given by the intervals between successive events of a given episode. The intervals measured from the data will be compared with the intervals in the case in which the events would be generated independently. If both interval arrays appear as being generated from different distributions, the events are correlated.

Our approach for learning the causal structure is similar to the approach as used in causal model theory in which a causal model is represented by a Bayesian network [2,3]. In Bayesian network theory, conditional independencies are defined over the joint probability distribution and a link is drawn between causality and dependencies through the causal Markov condition. The conditional independencies following from the causal structure can then be used to learn the causal structure from data.

In the next section we define a Causal Event Model for reflecting the event-generating mechanisms. We show that it is more general than current settings. In Section 3 we define marginal and conditional independence between events. Section 4 draws the link between causation and correlation in our framework. Section 5 defines the conditional independence tests. Section 6 gives a causal structure learning algorithm and Section 7 provides experiments with simulated data.

2 Causal Event Model

The model for the event-generating mechanisms is based on the following assumptions. (a) Events have exogenous causes (called *spontaneous events*) or are caused by other (*effect events*). (b) The causal mechanisms depend on the type of event. This does not mean that event c literally causes event e. It is possible that the mechanism responsible for generating event c (e.g. when a variable passes a certain threshold) affects another mechanism which triggers event e. In such case, the event related to the cause can happen after the effect event. Here, (c) we will assume that cause events happen before their effects. (d) The causal mechanism only generates one event (or none) of a specific type.

The effect event counterfactually depends on the cause events; if one of the causes would not have happened, the effect event would not have happened. The event sequence $\mathcal{E}$ can then be split up into two sequences: the spontaneous events $\mathcal{E}_s$ and the

effect events $\mathcal{E}_e$. Instantiations of events belong to either $\mathcal{E}_s$ or $\mathcal{E}_e$, however events of a certain type can occur in both. Each non-spontaneous event has one or more causes: an effect event is linked to one or more events. $\forall e_i \in \mathcal{E}_e, \exists \boldsymbol{c} \subset \mathcal{E} : \boldsymbol{c} \rightarrow e_i$. This is indicated in Fig. 2. We call it the *Causal Event Sequence Graph* (CESG). It constitutes a Directed Acyclic Graph (DAG).

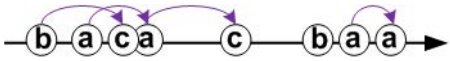


Fig. 2. Example event sequence with the causal relations between the events

On the other hand, a graph describing the mechanisms responsible for generating the effect events should not be cyclic. We only assumed that the mechanism depend on the event types, in the sense that events from a certain type are responsible for generating events from another type in the future. So, $A \rightarrow B$ means that some events of type B are caused by events of type A. If an A event, say e_i, causes a B event, say e_j, then t_j depends on t_i. This is represented by $P(\triangle^* t_B|A)$, a probability distribution over $\triangle^* t_B$ which is defined as the time interval between cause and effect, $t_j - t_i$ in the case of e_i causing e_j. The asterisk denotes that it is an interval between causally-connected events. The probability distribution can often be described by a Weibull distribution. It should be noted that the sum $P_{total} = \int_t P(\triangle^* t_B = t|A)dt$ can be smaller than 1, indicating that A in some cases does not generate B. By defining P over the time difference, time invariance is incorporated into our system.

The causal structure can be represented by a directed graph which can be cyclic, and can have bidirected edges or loops. Fig. 3 shows the causal structure responsible for the event sequence of Fig. 2. The parameterization is that for each node X and for all parents Pa of X, there is $P(\triangle^* t_X|Pa)$ which specifies a distribution of the time delay. These distributions represent the generation of X by Pa. This is shown in Fig. 4. If X has multiple parents, they can all independently generate X or the generation of X happens by a mutual occurrence of multiple parent events. $Pa_1 \ldots Pa_k \rightarrow X$ is described by $P(\triangle^* t_X|Pa_1, \triangle t_{Pa_2}, \ldots, \triangle t_{Pa_k})$. The distribution gives the time to X after the occurrence of Pa_1 and occurrence of Pa_i $(i = 2 \ldots k)$ with a time difference of $\triangle t_{Pa_i}$.

The directed graph together we the parameterization we call a *Causal Event Model* (CEM). The CEM can be considered as a generic template to produce the CESG, which is often called the 'rollout'.

2.1 Related Work

Temporal Nodes Bayesian Networks (TNBNs) [4] are a special kind of Bayesian network which are parameterized considering delays (relative times). When an initial (spontaneous) event occurs, its occurrence gives the reference time. The nodes represent variables. Events occur when variables pass a certain threshold. This is indeed often the case, but we do not want to make any assumption about the 'meaning' of the events and use event variables as nodes.

Networks of Probabilistic Events in Discrete Time (NPEDTs) [5] are also defined over event variables with a parameterization similar to ours. NPEDTs are, however,

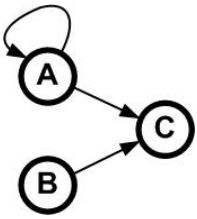


Fig. 3. Causal structure used for the experiments

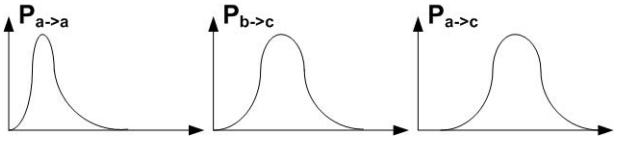


Fig. 4. Parameterization for some families of the causal structure of Fig. 3

more restrictive: each event can happen at most once and no self-references or cycles are allowed in the graph. By this restriction, NPEDTs are genuine Bayesian networks, while a CEM is not.

[6] uses a dynamic Bayesian network to model the relations between the event variables. The limitation of a dynamic Bayesian network is that you need to draw a link between a node and another node in the future. This fixes the time interval between cause and effect. [6] use the dynamic Bayesian network in combination with episode mining, but because of this limitation, they limit themselves to fixed-delay episodes. In our case we make no assumption about the time interval between the cause and effect event. We even allow continuous time intervals.

Finally, it must be noted that all 3 models here discussed discretize the time.

3 Independence Relationships

We define marginal and conditional independence on the distributions over the time intervals between successive events. This is motivated by the following. Events generate new events that will happen in the future. The causal mechanism defines the time interval between cause C and effect E, which we denoted by $\triangle^* t_E$. The knowledge of an event happening at time t contains information on the occurrence of the causally-related future events. We therefore consider the time to the first future occurrence of an event of a specific type. We denote this time delay as $\triangle^1 t_E$.

3.1 Interval to First Occurrence ($\triangle^1$)

We denote by $P_{ab}(\triangle^1 t_B | A)$ the probability that the first event of type B after a random time t happens at time $t + \triangle^1 t_B$ given that an event of type A occurred at time t. The subscript ab indicates that event b must happen after a. The definitions for independency will be based on this distribution instead of causal distributions $\triangle^* t$, because it

is measurable from data. Before that, two important consequences of the model have to be discussed.

It must be noted that if $A \rightarrow B$, this does not necessarily mean that $P(\triangle^* t_B|A)$ and $P_{ab}(\triangle^1 t_B|A)$ are the same, since the first event B after event A might be a spontaneous event or caused by other events, and accidently occurring right after A. The relation between $P(\triangle^* t_B|A)$ and $P_{ab}(\triangle^1 t_B|A)$ is calculated as follows. The probability for having the *first* occurrence of a B event at relative time t is the probability that B occurs at time t multiplied with the probability that no B occurred before that. This is expressed by the following equation. With $P(\triangle t_B = t|A)$ we denote the probability that a B event happens at time t after an A event.

$$P(\triangle^1 t_B = t|A) = P(\triangle t_B = t|A).(1 - P(\triangle^1 t_B < t|A)) \quad (1)$$

$$= (\sum_i P_i(\triangle^* t_B = t|A)).(1 - \int_{t'=0}^{t'=t} P(\triangle^1 t_B = t'|A)dt') \quad (2)$$

The first probability of the right hand side of Eq. 1 is determined by all possible direct causes of B (denoted by index i), the second is an integral adding all previous probabilities. It results in a recursive formula, given by Eq. 2. If the probability $P(\triangle t_B = t|A)$ is a constant, the result is an exponential distribution.

Next, assume the causal model is $A \rightarrow B$ and A is spontaneously generated by a Poisson process with rate λ. The first event B after an event A can then be (1) the event caused by that A or (2) a B event caused by another A event. For the latter holds that $P_2(\triangle t_B = t|A) = P(t_B = t) = \lambda$ since it is unrelated to A. $P(\triangle^1 t_B = t|A)$ is a combination of both given by above equation. The resulting distribution mainly depends on which distribution 'comes first'. The distribution with most of its weight for small delays greatly determines $P(\triangle^1 t_B = t|A)$.

3.2 Marginal Independence

Marginal independence is defined as follows:

$$A \underset{ab}{\perp\!\!\!\perp} B \Leftrightarrow P_{ab}(\triangle^1 t_B|A) = P(\triangle^1 t_B) \quad (3)$$

where $P(\triangle^1 t_B)$ is the probability that the first event of type B after time t happens at time $t + \triangle^1 t_B$ given a random time t.

An important difference with statistical independence defined over a joint probability distribution is that the order should be taken into account: $A \underset{ab}{\perp\!\!\!\perp} B$ means that knowledge about an A event has no information on the next B event, while $\underset{ba}{\perp\!\!\!\perp}$ is about information of an B event over the next A event. Hence:

$$A \underset{ab}{\perp\!\!\!\perp} B \not\Leftrightarrow A \underset{ba}{\perp\!\!\!\perp} B. \quad (4)$$

While it can be shown that symmetry holds for a given order:

$$A \underset{ab}{\perp\!\!\!\perp} B \Leftrightarrow B \underset{ab}{\perp\!\!\!\perp} A \quad (5)$$

A special case is autocorrelation. $P(\triangle^1 t_B|B)$ is the probability that the first event of type B after a random time t happens at time $t + \triangle^1 t_B$ given that another event of type B occurred at time t. To check for autocorrelation we check whether $P(\triangle^1 t_B|B) = P(\triangle^1 t_B)$. We denote B autocorrelated as $\overset{\frown}{B}$.

3.3 Conditional Independence

Conditional independence is also defined for a specific event ordering over its arguments. The order is described by an episode.

$$\begin{aligned} & A \underset{ep(A,B,\boldsymbol{S})}{\perp\!\!\!\perp} B|\boldsymbol{S} \Leftrightarrow \\ & P_{e=ep(A,B,\boldsymbol{S})}(\triangle^1 t_B|s^*, \triangle^1(\boldsymbol{S} \setminus s^*), \triangle^1 t_A) \\ & \quad = P_{e\setminus A}(\triangle^1 t_B|s^*, \triangle^1(\boldsymbol{S} \setminus s^*)) \end{aligned} \tag{6}$$

with $ep(A, B, \boldsymbol{S})$ an episode over A, B and $\boldsymbol{S}$, and s^* the first element of $\boldsymbol{S}$ in the episode, all $\triangle$s are defined with respect of s^*. $e \setminus A$ denotes the episode e from which A is removed. Note that the distributions do not depend on the choice of s^* among $\boldsymbol{S}$; it only sets the reference time.

4 Causation Implies Correlation and Vice Versa

In this section we draw the relation between causation and correlation as defined in the previous sections. The relation is grounded by the assumption that causally unrelated events are independent.

This assumption is also expressed by Reichenbach's principle: if A and B are correlated, then either A causes B, either B causes A or either there is a common cause of A and B.

In the following we will also assume that there are no unknown (latent) common causes. Together with the independence assumption this implies that there is no correlation if no causal relation in the model. Except that the spontaneous events from a specific type will be autocorrelated when their occurrence is not random.

4.1 Correlation and the Causal Event Sequence Graph

Consider $\mathcal{I}(cem)$ the conditional independencies of a causal event model, consider $CEM(G)$ all causal event models compatible with graph G. We are interested in the conditional independencies that hold for all $CEMs$ compatible with G (the intersection): $\mathcal{I}(G) = \cap_{CEM(G)}\mathcal{I}(cem)$. These independencies follow from the causal structure, independent from the parameterization. Specific parameterizations may lead to additional independencies.

The following theorem proves that the conditional independence statements from $\mathcal{I}(G)$ can be extracted from the Causal Event Sequence Graphs (CESG) compatible with G by d-separation. We recall the definition. $\boldsymbol{X}$ and $\boldsymbol{Y}$ are d-separated by $\boldsymbol{Z}$ if every path between $\boldsymbol{X}$ and $\boldsymbol{Y}$ is blocked by $\boldsymbol{Z}$. An (undirected) path is said to be blocked by $\boldsymbol{Z}$ if it contains a collider $\rightarrow \cdot \leftarrow$ whose descendants are not in $\boldsymbol{Z}$ or a non-collider $\rightarrow \cdot \rightarrow$ or $\leftarrow \cdot \rightarrow$ or $\leftarrow \cdot \leftarrow$ that is in $\boldsymbol{Z}$ [2].

Theorem 1. *$A \underset{ep(A,B,\boldsymbol{S})}{\perp\!\!\!\perp} B|\boldsymbol{S}$ is not in $\mathcal{I}(G)$ if and only if there is a subset of nodes in a Causal Event Sequence Graph compatible with G which forms an occurrence of the episode $ep(A, B, \boldsymbol{S})$ in the event sequence such that $a \not\perp b|\boldsymbol{s}$.*

Proof. If there is an active path between a and b in the sequence graph, we prove that t_a and t_b are bounded by the causal delay distributions of the network. Then there exists at least one parameterization which bounds the occurrence of b to the time of occurrence of a, such that the independence does not hold. For any triple $x \rightarrow y \rightarrow z$ along the path, t_z is bounded by t_y and also by t_x. The same applies for $x \leftarrow y \leftarrow z$. For any triple $x \leftarrow y \rightarrow z$ along the path, both t_z and t_x are bounded by t_y which makes them also depend on each other. For any triple $x \rightarrow y \leftarrow z$, t_y is bounded by t_x and t_z but this does not imply that t_x and t_z are dependent unless y or one of its descendants is conditioned on. In that case, t_y is known and together with t_x this gives information about t_z. Combining these bounds proves that an active path implies a conditional dependency. If, on the other hand, there is no path, then a and b events are assumed to be independent. If there is a path, but blocked by an event, say c, then t_c constrains t_b, but t_a does not further bounds t_c.

4.2 Correlation and the Causal Event Model

d-separation is not readily usable to identify conditional independencies from the Causal Event Model. A related criterion will be established here.

Definition 1 **(d-separation in CEM).** *A path p between two nodes A and B is said to be blocked by a set $\boldsymbol{S} = \{\boldsymbol{S}_1, \boldsymbol{S}_2, \boldsymbol{S}_3\}$, with $\boldsymbol{S}_1, \boldsymbol{S}_3 \subset \boldsymbol{E}$ and $\boldsymbol{S}_2 \subset \boldsymbol{E} \backslash \{A, B\}$, corresponding to an ordered episode $(\boldsymbol{s}_1, a, \boldsymbol{s}_2, b, \boldsymbol{s}_3)$ if:*

- *on p there is a fork $X \leftarrow Y \rightarrow Z$ and $Y \in S_1$*
- *on p there is a chain $X \rightarrow Y \rightarrow Z$ and $Y \in S_2$*

and

- *on p there is no collider $X \rightarrow Y \leftarrow Z$ for which either Y or any of its descendants $\in S_3$*

When all paths between A and B are blocked by $\boldsymbol{S} = \{\boldsymbol{S}_1, \boldsymbol{S}_2, \boldsymbol{S}_3\}$ we say that A is d-separated from B given $\boldsymbol{S}$ denoted as $A \underset{\boldsymbol{s}_1 a \boldsymbol{s}_2 b \boldsymbol{s}_3}{\perp} B|\boldsymbol{S}_1\boldsymbol{S}_2\boldsymbol{S}_3$, otherwise we call them d-connected denoted as $A \underset{\boldsymbol{s}_1 a \boldsymbol{s}_2 b \boldsymbol{s}_3}{\not\perp} B|\boldsymbol{S}_1\boldsymbol{S}_2\boldsymbol{S}_3$.

Theorem 2. *$A \underset{\boldsymbol{s}_1 a \boldsymbol{s}_2 b \boldsymbol{s}_3}{\perp\!\!\!\perp} B|\boldsymbol{S}_1\boldsymbol{S}_2\boldsymbol{S}_3 \Leftrightarrow A \underset{\boldsymbol{s}_1 a \boldsymbol{s}_2 b \boldsymbol{s}_3}{\perp} B|\boldsymbol{S}_1\boldsymbol{S}_2\boldsymbol{S}_3$, with $\boldsymbol{S}_1, \boldsymbol{S}_2, \boldsymbol{S}_3 \subset \boldsymbol{E}$.*

Proof. $\Leftarrow$
Assume $A \underset{\mathbf{s}_1 a \mathbf{s}_2 b \mathbf{s}_3}{\perp} B|\mathbf{S}_1\mathbf{S}_2\mathbf{S}_3$. Conditioning on events in $\mathbf{S}_3$ cannot block a path between A and B in the CEM. For each $e \in \mathbf{S}_1$, in the corresponding CESG e will appear before a and b. If in the CESG there is a causally directed path from e to both a and b then conditioning on e closes the path $a \leftarrow \ldots \leftarrow e \rightarrow \ldots \rightarrow b$. If there is no directed

Algorithm 1.. Marginal independence test for $A \underset{ab}{\perp\!\!\!\perp} B$

Given: Set of possible event types $\mathcal{D}$ and event sequence $\mathcal{E} = \langle (E_1, t_1), \ldots, (E_n, t_n) \rangle$

1. Count the number of occurrences of B in $S = n$.
2. Generate a new sequence S' with the same A events as in S, and add n events of type B with random timestamp $\in [0, T]$. If A and B are the same (self-correlation test), sequence S' should only contain the randomly generated events.
3. For both sequences S and S', generate the sequence of intervals I and I' between each occurrence of A and the first occurrence of B after that of A.
4. If the Kolmogorov-Smirnov test applied on I and I' returns 'equal', the test returns true (meaning independence).

path from e to both a and b in the CESG then a is trivially d-separated from b. Similar observations can be made for $e \in \mathbf{S}_2$, where there either is a causally directed path $a \rightarrow \ldots \rightarrow e \rightarrow \ldots \rightarrow b$ or a is again trivially d-separated from b in the corresponding CESG. Therefore $\Leftarrow$ follows from Theorem 1.

$\Rightarrow$

Assume $A \underset{\mathbf{s}_1 a \mathbf{s}_2 b \mathbf{s}_3}{\perp\!\!\!\perp} B|\mathbf{S}_1\mathbf{S}_2\mathbf{S}_3$ and $A \underset{\mathbf{s}_1 a \mathbf{s}_2 b \mathbf{s}_3}{\not\perp\!\!\!\perp} B|\mathbf{S}_1\mathbf{S}_2\mathbf{S}_3$. This implies that $A \underset{\mathbf{s}_1 a \mathbf{s}_2 b \mathbf{s}_3}{\perp} B|\mathbf{S}_1\mathbf{S}_2\mathbf{S}_3$ in the corresponding CESG (Theorem 1). Conditioning on events in $\mathbf{S}_1$ and $\mathbf{S}_2$ cannot d-connect A and B in the CEM, so $A \underset{ab\mathbf{s}_3}{\not\perp\!\!\!\perp} B|\mathbf{S}_3$. This means that there is a $E \in \mathbf{S}_3$ such that there is a causally directed path from both A and B to E, or an edge $A \rightarrow B$ in the CEM. This however is contradicted by the lack of such paths in the corresponding CESG.

5 Independence Test

The goal is to define a test which identifies $.\underset{ep}{\perp\!\!\!\perp}.|.$ such that in the generic case: $\perp \Leftrightarrow \underset{ep}{\perp\!\!\!\perp}$.

For testing $X \underset{\cdot\cdot}{\perp\!\!\!\perp} Y|\mathbf{Z}$, we have to compare the distribution $P_{ep(Y,\mathbf{Z})}(\triangle^1 t_Y|\mathbf{Z})$ reflecting the independence situation, with the actual distribution $P_{ep(X,Y,\mathbf{Z})}(\triangle^1 t_Y|\mathbf{Z}, X)$ estimated from the data. We will use the Kolmogorov-Smirnov test which works on the data directly. The test identifies whether two samples are drawn from the same distribution, without making any assumption about the distribution of data. The exact significance probability is calculated using the method of [7].

Note that all tests have linear complexity with respect to the sequence size.

5.1 Marginal Independence

The algorithm is described by Alg. 1. The algorithm measures the distribution over $\triangle^1$, which is different from that of $\triangle^*$ but as discussed in Sec. 3.1, $\triangle^*$ comes close to $\triangle^1$ if the average delta is smaller than that of the other causes.

Algorithm 2.. Conditional independence test for $A \underset{ep(abc)}{\perp\!\!\!\perp} B|C$

Given: Set of possible event types $\mathcal{D}$ and event sequence $\mathcal{E} = \langle (E_1, t_1), \ldots, (E_n, t_n) \rangle$

1. For each occurrence of episode $ep(abc)$, add interval $t_b - t_a$ to sequence I, add interval $t_c - t_a$ to sequence I_1 and $t_b - t_c$ to I_2.
2. Now shuffle sequence I_1 randomly such that the order of the elements gets completely different from that of I_2.
3. Construct sequence I' by adding the elements of I_1 to those of I_2 (interval i of I_1 is summed with interval i of I_2).
4. If the Kolmogorov-Smirnov test applied on I and I' returns 'equal', the test returns true (meaning conditional independence).

5.2 Conditional Independence

Next, we give an algorithm to test for independence when conditioned on one variable. The test is based on randomization of the intervals with respect to the time of occurrence of the conditioning variable. This creates the reference distribution. Alg. 2 describes the test procedure. If the occurrence of an A event is irrelevant for the occurrence of B when C is known, the time interval between A and B is irrelevant with respect of C. The distribution for the null hypothesis is then constructed by randomizing (swapping) the intervals between A and B.

6 Causal Structure Learning

Here we present a modified version of the PC algorithm to detect the causal structure called EPC: Algorithm 3. We define a complete directed CEM as a complete graph with all bi-directed edges and self-references for each variable. The description of the algorithm simplifies as each bi-directed edge $A \leftrightarrow B$ is considered as two edges $A \rightarrow B$ and $A \leftarrow B$. Since we can directly make a difference between these edges by looking at ab and ba episodes (through the asymmetry, see Section 2), we do not have to add an orientation phase or end up with a class of equivalent models under the given independencies such as the PC algorithm.

All CIs discovered in the data are following from the causal structure.

Theorem 3. *Under faithfulness, the EPC algorithm returns the correct CEM given an oracle for the independence tests.*

Proof. By faithfulness, no adjacent nodes can become independent when conditioned on any subset of other nodes. Non-adjacent nodes are either marginally independent or become independent when conditioned on one of the nodes along each path.

7 Experiments and Evaluation

In this section we experimentally analyze the accuracy of the independence test and the learning algorithm. It is then compared with the results obtained by episode mining.

Algorithm 3.. EPC

Given: Set of possible event types $\mathcal{D}$ and event sequence $\mathcal{E} = \langle (E_1, t_1), \ldots, (E_n, t_n) \rangle$

1. Initialization with complete directed CEM G over $\mathcal{D}$
2. For each edge $A \to B$ in G (including $\overset{\frown}{A}$, i.e. B=A),
   (Consider each bi-directed edge $A \leftrightarrow B$ as two edges $A \to B$ and $A \leftarrow B$)
   $\forall \mathbf{S}_1 \subset \mathcal{D}$ and $\forall \mathbf{S}_2 \subset \mathcal{D} \backslash \{B\}$:
   If $A \underset{\mathbf{s}_1 a \mathbf{s}_2 b}{\perp\!\!\!\perp} B | \mathbf{S}_1, \mathbf{S}_2$, remove $A \to B$ from G

7.1 Influence of Causal Delay and Sample Size

As discussed in Sec. 3.1, the delay between cause and effect plays an important role in identifying $\triangle^*$ from $\triangle^1$. We experimentally studied this with data generated from model $X \to Y \to Z$ with the following parameterization. X is generated by a Poisson process with rate 0.01 (meaning that on average every 100 time units an event occurs). The parameterization of both causal relations, $X \to Y$ and $Y \to Z$, is given by a Gaussian distribution with given mean and the standard deviation is set to the square root of the mean (a mean of 100 thus gives the same average delay as that of the Poisson processes). There is a probability of 0.2 that no effect event is generated. Table 1 shows the minimum episode occurrences necessary to correctly identify the given dependencies in 10 experiments. A dash means that the minimum count exceeded 4000.

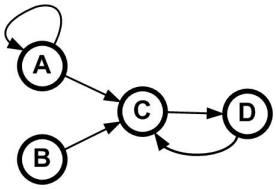


Fig. 5. Causal structure of the example event sequence

Secondly, data was generated from the model of Fig. 5. The parameterization was set as follow. Spontaneous events are generated for A, B and D by a Poisson process with rate 0.01. The parameterization of each causal relation (each edge), $P(\triangle^* E|C)$, is given by a Gaussian distribution with given mean and the standard deviation set to be the square root of the mean. For all causal relations there is a probability of 0.2 that no effect event is generated, except for $A \to A$ and $D \to C$ for which the probability of no effect is set to 0.8 to avoid cascading effects due to the cycles. The results are also shown in Table 1.

The results show clearly that the detection of dependencies is accurate for small sample sizes for simple models but becomes harder when multiple causes are into play, such as for detecting the dependency between A and D. The self-correlation of A is also hard to detect since the causal relation is only fired with low probability (0.2). Finally, it

must be noted that the test rarely makes errors on detecting conditional independencies. For model $X \to Y \to Z$ and over all experiments, testing $X \underset{xyz}{\perp\!\!\!\perp} Z|Y$ gave an accuracy of 99.3%. The same accuracy was obtained when testing $A \underset{acd}{\perp\!\!\!\perp} D|C$ in the second model.

Table 1. Number of episodes necessary to correctly identify the following (in)dependencies that hold for the given models with varying Gaussian mean. The lowest row shows the minimal sequence size to correctly learn model $X \to Y \to Z$.

mean	20	40	60	80	100	120
$X \underset{xy}{\not\perp\!\!\!\perp} Y$	19	20	58	59	56	138
$X \underset{xz}{\not\perp\!\!\!\perp} Z$	59	135	137	297	303	1242
$X \underset{xyz}{\not\perp\!\!\!\perp} Y\|Z$	185	186	185	89	90	87
$A \underset{ac}{\not\perp\!\!\!\perp} C$	25	75	378	790	3181	-
$A \underset{ad}{\not\perp\!\!\!\perp} D$	53	941	2341	-	-	-
$A \underset{abc}{\not\perp\!\!\!\perp} B\|C$	127	159	126	129	128	121
$A \underset{acd}{\not\perp\!\!\!\perp} C\|D$	583	565	267	260	263	257
$A \underset{aa}{\not\perp\!\!\!\perp} A$	375	1223	3695	-	-	-
learning	3068	686	685	677	668	690

7.2 Causal Structure Learning

The lowest row of Table 1 shows the minimal sequence size (number of events) to correctly learn model $X \to Y \to Z$ with the parameterization specified in the previous section. It shows that only a relatively small sample size is needed to learn simple models. The high sample size needed to learn the model with mean 20 is needed since the delays come close to being deterministic (small standard deviation) which results in violations of faithfulness.

The accuracy of the learning algorithm depends on the correctness of the independence test and the validity of faithfulness. This was confirmed by our experiments with randomly-generated models and different sample sizes. The following causes of failure were detected:

1. Large causal delays and small sample sizes increase the number of test errors, as discussed in the previous section. When for a single event multiple causes come into play, it's harder to detect the dependencies.
2. Exact violations of faithfulness or near-to-unfaithful situations. For example, when the causal delay is nearly deterministic. These cases are similar to the problems in learning causally-interpreted Bayesian networks. See for instance [8] for discussion of the problems and modifications of the PC algorithm to handle such violations.
3. Finally, our conditional independence test is limited to one conditioning variable. This means that if two variables are related by two difference causal paths, they are dependent and will not become independent when conditioned on only one variable. It introduces a false positive edge.

8 Conclusions

We created a procedure with linear complexity for testing marginal and conditional independence between events in event sequences. The test is accurate in detecting dependencies coming from causal relations if the average interval between cause and effects is smaller than that of spontaneous events or other causes. We defined a very general model, the Causal Event Model (CEM), to describe the underlying event-generating mechanisms. As opposed to other event models, it is not a Bayesian network since it allows cycles. Based on the conditional independencies an algorithm could be constructed to learn the correct causal structure under faithfulness and causal sufficiency.

Acknowledgements. This research was partly funded by the Prognostics for Optimal Maintenance (POM) project (grant nr. 100031; `www.pom-sbo.org`) which is financially supported by the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen).

References

1. Granger, C.W.J.: Investigating causal relations by econometric models and cross-spectral methods. Econometrica 37(3), 424–438 (1969)
2. Pearl, J.: Causality: Models, Reasoning, and Inference. Cambridge University Press (2000)
3. Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search, 2nd edn. Springer (1993)
4. Arroyo-Figueroa, G., Sucar, L.E.: Temporal bayesian network of events for diagnosis and prediction in dynamic domains. Applied Intelligence 23, 77–86 (2005)
5. Galán, S.F., Díez, F.J.: Networks of probabilistic events in discrete time. Int. J. Approx. Reasoning 30(3), 181–202 (2002)
6. Patnaik, D., Laxman, S., Ramakrishnan, N.: Inferring dynamic bayesian networks using frequent episode mining. CoRR (2009)
7. Nikiforov, A.: Algorithm as 288: Exact smirnov two-sample tests for arbitrary distributions. Applied Statistics 43, 265–284 (1994)
8. Lemeire, J., Meganck, S., Cartella, F., Liu, T.: Conservative independence-based causal structure learning in absence of adjacency faithfulness. Int. J. Approx. Reasoning 53(9), 1305–1325 (2012)

Measuring Incompleteness under Multi-valued Semantics by Partial MaxSAT Solvers

Yue Ma[1,⋆] and Qingfeng Chang[2]

[1] Technische Universität Dresden, Germany
[2] Chongqing University of Posts and Telecommunications, China
mayue@tcs.inf.tu-dresden.de, clheang@gmail.com

Abstract. Knowledge base metrics provide a useful way to analyze and compare knowledge bases. For example, inconsistency measurements have been proposed to distinguish different inconsistent knowledge bases. Whilst inconsistency degrees have been widely developed, the incompleteness of a knowledge base is rarely studied due to the difficulty of formalizing incompleteness. For this, we propose an incompleteness degree based on multi-valued semantics and show that it satisfies some desired properties. Moreover, we develop an algorithm to compute the proposed metric by reducing the problem to an instance of partial MaxSAT problem such that we can benefit from highly optimized partial MaxSAT solvers. We finally examine the approach over a set of knowledge bases from real applications, which experimentally shows that the proposed incompleteness metric can be computed practically.

1 Introduction

In knowledge engineering, it is often helpful to have metrics for measuring some aspects of a knowledge base [1–3]. Such metrics can convey the state of a knowledge base, thus enabling the comparison of the quality of different knowledge bases [1] or ranking ontologies in the Semantic Web [3]. Among often and widely studied metrics is the inconsistency degree [4–6, 1, 5, 7–9, 3, 10, 11] which can reflect how much conflict an inconsistent knowledge base has. While inconsistency degrees are to measure conflicting information caused by redundant information, there is rare metric that can measure how incomplete a knowledge base is, which is an interesting aspect specially for knowledge construction process [12].

To motivate the necessity for defining an incompleteness degree, consider the following three simple knowledge bases constructed by three propositional letters: $K_1 = \{p, q, r\}$, $K_2 = \{p \vee q, r\}$, and $K_3 = \{p \vee q \vee r\}$. Intuitively, the information that K_1 conveys is complete because we know that all the letters should be true, but K_2 is less complete since we are not sure about the certainty of p and q, though r is certainly true; And K_3 seems even less complete because there is no certain information about any of the three letters. In this paper, we are interested in defining an incompleteness metric that can distinguish different degrees of completeness of knowledge bases.

It seems that an information measure [13–16] is the right solution to our question. However, there is still no consensus about the meaning of "degree of information" [15].

⋆ We acknowledge financial support by the DFG Research Unit FOR 1513, project B1.

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 388–399, 2013.

Unlike [6, 1, 5, 7–9, 3, 10] where inconsistency is the target feature, our focus in this paper is on estimating the amount of uncertain information contained in a knowledge base, no matter it is consistent or inconsistent. In this line, [15] has proposed the "degree of ignorance" based on the the minimal effort necessary to disambiguate a knowledge base. Such a degree is defined in an "active" way based on action plans that can gain information during disambiguation. However it can be the case that there is no action plan, which leads to a degree of ignorance $+\infty$. Different from [15], in this paper we use a static way based on Belnap's four-valued semantics to get an incompleteness measurement which is always a normalized value between 0 and 1. Belnap's four-valued semantics [17, 18] has been frequently used in inconsistency degrees [1, 8–10]. However, these work together with the LPm semantics used in the degree of ignorance in [15] all depend on the truth value *both* and ignore the truth value *unknown* of Belnap's four-valued semantics. But *unknown* is of a key importance to define the incompleteness degree in this paper since it allows to express undefined status of a letter.

Once a metric is defined, it is expected that we can have an efficient way to compute such a metric [10, 11, 19]. So besides the analysis of the theoretical properties of the proposed incompleteness degree, we also give an algorithm for the computation, which is based on a linear reduction to a partial Max-Sat problem. Therefore, the computation of the defined incompleteness degree can benefit from cutting edge Max-Sat solvers.

The remainder of this paper is structured as follows: In Section 2, we recall Belnap's four-valued semantics and some satisfiability problems. Section 3 gives the definition of the proposed incompleteness degree and its properties. Section 4 gives the encoding based novel algorithm. Section 5 describes the implementation and evaluation. We conclude this paper and outlook our future work in Section 6.

2 Preliminaries

In this paper, we consider a propositional language $\mathcal{L}_{\mathcal{A}}$ with a finite set of propositional variables $\mathcal{A} = \{p_1, \ldots, p_n\}$. A literal is a variable p or its negation $\neg p$. A knowledge base is a set of propositional formulas built from $\mathcal{A}$. $Var(K)$ denotes the set of variables occurring in K and $|S|$ denotes the cardinality of a set S.

A clause $\gamma = l_1 \vee l_2 \vee \ldots \vee l_k$ is a disjunction of literals. A CNF formula is a conjunction of clauses, which is usually represented as a set of clauses $K = \{\gamma_1, \gamma_2, \ldots, \gamma_m\}$. In this paper, we consider knowledge bases in CNF format. Note that each propositional formula can be translated into CNF formula without loss of generality in the sense that satisfiability keeps unchanged.

Four-Valued Semantics. In this paper, we consider Belnap's logic that is shown of a particular importance among the family of many-valued logical systems [18]. Compared to two truth values used by classical semantics, the set of truth values for 4-valued semantics [17, 18] contains four elements: *true, false, unknown* and *both*, written by t, f, N, B, respectively. The value B thus can be understood to stand for both *true and false*, while N stands for *neither true nor false*, i.e. for the absence of any information about truth or falsity. The four truth values together with two orderings $\preceq_t$ and $\preceq_k$ defined below form a bilattice $\mathsf{FOUR} = (\{t, f, B, N\}, \preceq_t, \preceq_k)$:

$f \preceq_t N \preceq_t t, f \preceq_t B \preceq_t t, N$ and B are incomparable with respect to $\preceq_t$.
$N \preceq_k t \preceq_k B, N \preceq_k f \preceq_k B, t$ and f are incomparable with respect to $\preceq_k$.

In Belnap's logic, the four-valued semantics of connectives $\vee, \wedge$ are defined according to the upper and lower bounds of two elements based on the ordering $\preceq_t$, respectively, and the operator $\neg$ is defined as $\neg t = f, \neg f = t, \neg B = B$, and $\neg N = N$. By this definition, the following proposition holds:

Proposition 1. *The connectives $\vee, \wedge, \neg$ are monotonic with respect to $\preceq_k$. That is, for $x, y \in \{t, f, B, N\}$ satisfying $x \preceq_k y$, $\neg x \preceq_k \neg y$; For $x_1, x_2, y_1, y_2 \in \{t, f, B, N\}$ satisfying $x_1 \preceq_k x_2, y_1 \preceq_k y_2$, $(x_1 \circ y_1) \preceq_k (x_2 \circ y_2)$ for $\circ \in \{\wedge, \vee\}$.*

The designated set of FOUR is $\{t, B\}$. So a 4-valued interpretation I is a *4-model* of a knowledge base K denoted $I \models_4 K$ if and only if for each formula $\phi \in K$, $\phi^I \in \{t, B\}$. A knowledge base which has a 4-model is called *4-valued satisfiable*. A knowledge base K 4-valued entails a formula φ, written $K \models_4 \varphi$, if and only if each 4-model of K is a 4-model of φ. For simplicity, we call $K \models_4 \varphi$ the 4-valued reasoning and call $K \models \varphi$ under the classical semantics 2-valued reasoning. We denote $Mod4(K)$ the set of 4-models of a knowledge base K.

Example 1. Given a propositional knowledge base $K = \{p, \neg p \vee q, \neg q \vee r, \neg r, s \vee u\}$. Then the following three 4-valued models I_1, I_2 and I_3 satisfy K:

$$p^{I_1} = t, q^{I_1} = B, r^{I_1} = f, s^{I_1} = t, u^{I_1} = N;$$
$$p^{I_2} = B, q^{I_2} = f, r^{I_2} = B, s^{I_2} = t, u^{I_2} = N;$$
$$p^{I_3} = B, q^{I_3} = B, r^{I_3} = B, s^{I_3} = t, u^{I_3} = N.$$

Satisfiability Problems. Deciding if a knowledge base in CNF is satisfiable is called a satisfiability (SAT) problem which is NP-complete. Even though the SAT problem is intractable, the state of the art SAT solvers are highly optimized and can deal with large size inputs.

As an extension of SAT, partial Max-SAT (the partial maximum satisfiability problem) has gotten deep study recently. Formally, a partial MaxSAT problem is of the form $P = (H, S)$, where H is a set of clauses, called the hard part; And S is the other set of clauses, called the soft part. The objective is to ask for a classical variable assignment that satisfies all hard clauses in H together with the maximum number of the soft ones in S. That is, an answer should be a two-valued interpretation $\hat{I}$ such that $\hat{I} = \arg\max_I |\{\gamma \mid \gamma \in S, I \models \gamma, I \models H\}|$.

The state of the art partial MaxSAT solvers such as SAT4j MaxSAT [20], MSUnCore [21] and Clone [22] are highly optimized and scalable as shown in the third[1] and fourth[2] MaxSAT Evaluations. Moreover, they are free to download and to use for academic research purpose.

[1] `http://www.maxsat.udl.cat/08/`
[2] `http://www.maxsat.udl.cat/09/`

3 Incompleteness Degree by Multi-valued Semantics

In this section, we give the definition of incompleteness degree based on four-valued semantics. Moreover, some logical properties of the proposed incompleteness degree are discussed. Then we give a reduction from incompleteness degree to Partial Max-Sat problem such that incompleteness degrees can be computed via cutting-edge Partial Max-Sat solvers.

3.1 Definition

The definition of incompleteness degree is based on the intuition that the truth value N from the Belnap's four-valued semantics characterizes the lack of information. As a degree, we consider the ratio of undefined information with respect to the whole information, formally defined as follows

Definition 1. *Suppose I is a four-valued interpretation. The incompleteness degree of a knowledge base K with respect to I, written $IncomDegree(K, I)$, is a value in $[0, 1]$ defined as*

$$IncomDegree(K, I) = \frac{|\{p \in Var(K) \mid p^I = N\}|}{|Var(K)|},$$

where the numerator $\{p \in Var(K) \mid p^I = N\}$ is called the incomplete set of I with respect to K, written $Incomplete(K, I)$.

The above definition is interpretation dependent. Based on this definition, for a given knowledge base K, we can get an order among models of K as follows:

Definition 2. *Let I, I' be two 4-models of K, then we say I is preferred than I' for measuring incompleteness, written $I \geq_{prfd} I'$, if and only if*

$$IncomDegree(K, I) \geq IncomDegree(K, I').$$

That is, the preferred model gives a larger incompleteness degree than a less preferred model. Obviously, $\geq_{prfd}$ is a total order and we denote the most preferred model of K as $MostPrfd(K) = \{I \mid I \geq_{prfd} I' \text{ for } I' \in Mod4(K)\}$. The most preferred model is of a particular interest according to the following proposition:

Proposition 2. *Let K be a knowledge base and $I \in MostPrfd(K)$. Then for any 4-valued interpretation I' such that $I \leq_k I'$, we have I' is a 4-valued model of K.*

Proof. By Proposition 1, we know that for each formula α and any two interpretations $I_1 \leq_k I_2$, if I_1 four-valued satisfies α, so does I_2. By the definition of $MostPrfd(K)$, we have I satisfies all formulae in K, so the conclusion follows.

Example 1. *Suppose $K = \{p, p \vee q \vee r, \neg p\}$. Then among all possibilities, we have the following 4-models for K:*

$$p^{I_1} = B, q^{I_1} = f, r^{I_1} = f;$$
$$p^{I_2} = B, q^{I_2} = N, r^{I_2} = f;$$
$$p^{I_3} = B, q^{I_3} = N, r^{I_3} = N;$$

It is easy to see that $I_3 \geq_{prfd} I_2 \geq_{prfd} I_1$. *Since 2 over 3 propositional letters are assigned to be* N *and* p *cannot be valued* N *because* p *is in* K, *we can see that* $I_3 \in MostPrfd(K)$.

To get rid of the interpretation dependence as in Definition 1, we define the incompleteness degree as the maximized incompleteness degree over 4-valued models of a knowledge base as follows:

Definition 3. *Given a knowledge base* K *and its most preferred four-valued model set* $MostPrfd(K)$, *the incompleteness degree of* K *is defined as following:*

$$IncomDegree(K) = IncomDegree(K, I) \text{ for } I \in MostPrfd(K).$$

It is obvious that this is a well-defined definition by the definition of $MostPrfd(K)$.

Example 2. *(Example 1 continued)*
Since $I_3 \in MostPrfd(K)$, *we have* $IncomDegree(K) = 2/3$.

Note that K in Example 1 is inconsistent because it contains both p and $\neg p$, but under Definition 1, it does not imply that its incompleteness degree is 0. Intuitively, this is because the confliction only occurs in one propositional letter, but the information about other letters is still unknown which indeed contributes the nonzero incompleteness degree.

Moreover, we can see that such defined incompleteness degree is to maximize the number of propositional letters assigned to N as shown by the following corollary:

Corollary 1. *Let* K *be a knowledge base. Then we have*

$$IncomDegree(K) = \max_{I \models_4 K} \{IncomDegree(K, I)\}.$$

This captures the intuition that more preferred models contain less redundant fake information for measuring incompleteness degree. For example, in Example 1, I_3 is more preferred than other two interpretations I_1 and I_2 which over-optimistically assign complete information (i.e. non N value) to letters. By Proposition 2, we know that once I_3 is a four-valued model of K, so are I_1 and I_2. So I_3 is of interest to be used to calculate the incompleteness degree of K.

Example 3. *Consider the three knowledge bases* K_1, K_2, K_3 *defined in the introduction. We have the following three most preferred models for each as follows (*I_i *for* K_i*):*

$$p^{I_1} = t, q^{I_1} = t, r^{I_1} = t$$
$$p^{I_1} = N, q^{I_1} = t, r^{I_1} = t$$
$$p^{I_1} = N, q^{I_1} = N, r^{I_1} = t$$

So we have $IncomDegree(K_1) = 0$, $IncomDegree(K_2) = 1/3$, $IncomDegree(K_3) = 2/3$, *which coincides with the intuition given in the introduction.*

3.2 Properties of Incompleteness Degree

Next we give some properties of the defined incompleteness degree. Assume that a set of propositional letters is given, namely Σ, and two knowledge bases K, K' are given and satisfy $Var(K) = Var(K') = \Sigma$.

Proposition 1. *If* $K \models_4 K'$*, then* $IncomDegree(K) \leq IncomDegree(K')$.

Proof. By $K \models_4 K'$, we have $Mod4(K) \subseteq Mod4(K')$. Thus, for any given 4-model $I \in MostPrfd(K)$, $I \in Mod4(K')$ holds. For any 4-model $I' \in MostPrfd(K')$, by the definition of the most preferred model, $Incomplete(K, I') \geq Incomplete(K, I)$. The conclusion follows.

This proposition means that a logically weaker (under 4-valued semantics) knowledge base is more incomplete than a stronger one.

Proposition 3. *If* $K \models_4 K'$*, then* $IncomDegree(K) = IncomDegree(K \cup K')$.

Proof. By $K \models_4 K'$, we have $K \models_4 K \cup K'$ and $K \cup k' \models_4 K$. By Proposition 1, we have both $IncomDegree(K) \leq IncomDegree(K \cup K')$ and $IncomDegree(K \cup K') \leq IncomDegree(K)$ hold. So $IncomDegree(K) = IncomDegree(K \cup K')$.

This proposition shows that enhanced with inferred knowledge (under 4-valued semantics) does not decrease the incompleteness degree of a knowledge base.

Obviously, Proposition 1 can have the following corollary hold, which shows that the proposed incomplete degree is semantics based, instead of syntax based.

Corollary 2. *If* $K \models_4 K'$ *and* $K' \models_4 K$, $IncomDegree(K) = IncomDegree(K')$.

We have the following proposition which characterizes a set of knowledge bases whose incompleteness degrees are zero.

Proposition 4. *Given a knowledge base* K*, if* $p \in K$ *and/or* $\neg p \in K$ *for all* $p \in Var(K)$*, then* $IncomDegree(K) = 0$.

Proof. If the conclusion does not hold, there exists a 4-interpretation $I \in MostPrfd(K)$ and a variable $p \in Var(K)$ such that $p^I = N$, which contradicts the definition of I and the assumption that either p or $\neg p$ appears in K.

4 Algorithm for Incompleteness Degree

To compute the proposed incompleteness degree, we propose a novel algorithm which encodes the problem to the partial Max-SAT problem, which is linear in the size of input knowledge bases.

Given a knowledge base $K = \{\gamma_i \mid i = 1, \ldots, n\}$ over variables set $\mathcal{A}$, it is well-known that the 4-valued reasoning on K can be simulated by the 2-valued reasoning

on $4(K)$, where $4(\cdot)$ is the transformation function from (a set of) clauses to (a set of) clauses defined as follows [23]:

$$\begin{aligned} 4(\{\gamma_1, \gamma_2, \ldots, \gamma_n\}) &= \{4(\gamma_1), 4(\gamma_2), \ldots, 4(\gamma_n)\}\ ; \\ 4(l_1 \vee \ldots \vee l_k) &= 4(l_1) \vee \ldots \vee 4(l_k)\ ; \\ 4(p) &= +p\ ; \\ 4(\neg p) &= -p\ . \end{aligned}$$

That is, $4(K)$ is a knowledge base over variables $\mathcal{A}^{+}_{-} = \{+p, -p \mid p \in Var(K)\}$. Obviously, computing $4(K)$ from K can be done in linear time.

A 4-valued interpretation I on $\mathcal{A}$ can also be seen as a 2-valued interpretation on variables $\mathcal{A}^{+}_{-}$. The corresponding relation can be described as follows:

$$\begin{aligned} p^I = B \text{ iff } +p^I = t \text{ and } -p^I = t; \\ p^I = f \text{ iff } +p^I = f \text{ and } -p^I = t; \\ p^I = t \text{ iff } +p^I = t \text{ and } -p^I = f; \\ p^I = N \text{ iff } +p^I = f \text{ and } -p^I = f. \end{aligned}$$

In the rest, we will refer to either of these two views without explicit explanation.

Theorem 1. *[23] Given a propositional knowledge base K and a 4-valued interpretation I, we have $I \models_4 K$ iff $I \models 4(K)$.*

Example 2. Let $K = \{\neg p, p \vee q, \neg q, r\}$. We have $4(K) = \{-p, +p \vee +q, -q, +r\}$. Consider the interpretation $I_1 = \{+p, -p, -q, +r\}$. I_1 can be seen as a 4-interpretation on $\{p, q, r\}$ with $p^{I_1} = B, q^{I_1} = f, r^{I_1} = t$. I_1 can also be viewed as a 2-interpretation on $\{+p, -p, +q, -q, +r, -r\}$ which assigns variables in I_1 true and other variables false, i.e. in the following way:

$$\begin{aligned} +p^{I_1} = t, -p^{I_1} = t, +q^{I_1} = f, \\ -q^{I_1} = t, +r^{I_1} = t, -r^{I_1} = f. \end{aligned}$$

It is easy to check that $I_1 \models_4 K$ and $I_1 \models 4(K)$.

Proposition 5. *Let K be a knowledge base over $\mathcal{A}$ and I be a 4-valued model of K. Denote $b(K, I) = \{p \in Var(K) \mid +p^I = f \text{ and } -p^I = f\}$. Then the incompleteness degree of K under 4-valued semantics can be computed by 2-valued semantics over $\mathcal{A}^{+}_{-}$ as follows:*

$$IncomDegree(K, I) = \frac{|b(K, I)|}{|Var(K)|}\ ;$$

$$IncomDegree(K) = \max_{I \models 4(K)} IncompleteDegree(K, I) = \frac{\max\limits_{I \models 4(K)} |b(K, I)|}{|Var(K)|}\ .$$

Proof. By Definition 3 and the fact that $p^I = N$ iff $+p^I = f$ and $-p^I = f$, this corollary holds obviously.

Based on Proposition 5, we can see that the computation of $\max_{I \models 4(K)} |b(K, I)|$ is the key to compute the incompleteness degree. We have

$$\begin{aligned}\max_{I \models 4(K)} |b(K,I)| &= \max_{I \models 4(K)} |\{p \mid p \in \mathit{Var}(K), +p^I = f \text{ and } -p^I = f\}| \\ &= \max_{I \models 4(K)} |\{p \mid p \in \mathit{Var}(K), \neg + p^I = t \text{ and } \neg - p^I = t\}| \\ &= \max_{I \models 4(K)} |\{p \mid p \in \mathit{Var}(K), (\neg + p \wedge \neg - p)^I = t\}|. \end{aligned} \quad (1)$$

Note that the conditional part in Equation 1 above (i.e. $(\neg + p \wedge \neg - p)^I = t$) is not a clause yet thus not acceptable by MaxSet solvers. For this, we need to introduce some auxiliary fresh propositional letters as following:

$$Aux_p = \neg + p \wedge \neg - p$$

Then we have

$$\begin{aligned}&\max_{I \models 4(K)} |\{p \mid p \in \mathit{Var}(K), (\neg + p \wedge \neg - p)^I = t\}| \\ &= \max_{I \models 4(K)} |\{Aux_p \mid p \in \mathit{Var}(K), (Aux_p)^I = t\}|\end{aligned}$$

Now we are ready to compute incomplete degrees by using partial Max-SAT problem solvers. This is based on the following reduction to a partial Max-SAT instance:

Definition 4. *Given a propositional knowledge base $K = \{\gamma_1, \ldots, \gamma_n\}$, $\mathit{Var}(K) = \{p_1, \ldots, p_m\}$, the corresponding partial Max-SAT problem for the 4-semantics based incompleteness degree $IncompleteDegree$, written $P(K) = (H(K), S(K))$, is defined as follows:*

$$\begin{aligned}H(K) &= H_1(K) \cup H_2(K) \cup H_3(K) \cup H_4(K), \text{ where ;} \\ H_1(K) &= \{4(\gamma) \mid \gamma \in K\}; \\ H_2(K) &= \{\neg Aux_p \vee \neg + p \mid p \in \mathit{Var}(K)\}; \\ H_3(K) &= \{\neg Aux_p \vee \neg - p \mid p \in \mathit{Var}(K)\}; \\ H_4(K) &= \{+p \vee -p \vee Aux_p \mid p \in \mathit{Var}(K)\}. \\ S(K) &= \{Aux_p \mid p \in \mathit{Var}(K)\}.\end{aligned}$$

Then we have the following theorem.

Theorem 2. *Given a knowledge base K, suppose I is a solution to the partial Max-SAT problem $P(K)$. Let $b(I, K) = |\{Aux_p \mid p \in \mathit{Var}(K) \text{ and } Aux_p^I = t\}|$ and $m(K) = |\mathit{Var}(K)|$. Then we have that $IncompleteDegree(K) = b(I, K)/m(K)$.*

Proof. By the definition of $P(K)$, I satisfies that for any other J, $b(I, K) \leq b(J, K)$. By Proposition 5, the conclusion follows.

Theorem 2 can be described by the following algorithm. The algorithm first generates $P(K)$ in line 4 to line 13, then computes a solution of $P(K)$ by calling a partial Max-SAT solver in line 14, and computes the value of incompleteness degree by Theorem 2 in line 15 to 16.

Algorithm 1.. Computing $IncompleteDegree$ by Partial Max-SAT Solver

procedure $IncompleteDegree(K)$
  $P \leftarrow \{\}$
  $m \leftarrow |Var(K)|$
  for all Clause $\gamma \in K$ **do**
    $P.addHardClause(4(\gamma))$
  end for
  for all Variable $p \in Var(K)$ **do**
    Create a fresh variable Aux_p
    $P.addHardClause(\neg Aux_p \vee \neg + p)$
    $P.addHardClause(\neg Aux_p \vee \neg - p)$
    $P.addHardClause(\neg + p \vee -p \vee Aux_p)$
    $P.addSoftClause(Aux_p)$
  end for
  $I \leftarrow$ PartialMaxSATSolver(P)
  $b = |\{Aux_p \mid {Aux_p}^I = t\}|$
  return b/m
end procedure

Corollary 1 (Correctness of Algorithm 1). *For any given knowledge base K, Algorithm 1 is sound and complete for computing the incompleteness degree of K under Belnap's four-valued semantics. That is, $Algorithm1(K) = IncompleteDegree(K)$, where $Algorithm1(K)$ is the value returned by Algorithm 1 with K as the input.*

Proof. This conclusion easily follows from Theorem 2.

Next example gives a further illustration of Algorithm 1.

Example 3. (Example 1 continued)
We have $4(K) = \{+p, +p \vee +q \vee +r, -p\}$. Then, by Definition 4, the hard clause set of $P(K)$ is $\{+p, +p \vee +q \vee +r, -p\} \cup \{\neg Aux_p \vee \neg + p, \neg Aux_q \vee \neg + q, \neg Aux_r \vee \neg + r\} \cup \{\neg Aux_p \vee \neg - p, \neg Aux_q \vee \neg - q, \neg Aux_r \vee \neg - r\} \cup \{\neg + p \vee -p \vee Aux_p, \neg + q \vee -q \vee Aux_q, \neg + r \vee -r \vee Aux_r\}$, and the soft clause set of $P(K)$ is $\{Aux_p, Aux_q, Aux_r\}$. For $P(K)$, we have the following one optimized solution I_0 by a partial Max-SAT solver:

$$+p^{I_0} = t, -p^{I_0} = t, +q^{I_0} = f,$$
$$-q^{I_0} = f, +r^{I_0} = f, -r^{I_0} = f.$$
$$Aux_p^{I_0} = f, Aux_q^{I_0} = t, Aux_r^{I_0} = t.$$

The corresponding 4-model of K is $p^{I_0} = B, q^{I_0} = N, r^{I_0} = N$, from which we have that $IncompleteDegree(K) = 2/3$ by Algorithm 1, thus coinciding with its theoretical value.

5 Evaluation

This section describes the experimental results to show the efficiency of our encoding algorithm. To this end, we used three state of the art partial Max-SAT solvers, namely SAT4j MaxSAT [20], MsUncore [21] and Clone [22], to implement our encoding algorithms.

Table 1. Results of Algorithm 4 on Different Types of Instances

Instance					Encoding Algorithm		
Instance Name	#V	#C	$IncomDegree$	n(N)	sat4j	msuncore	clone
small0	3	4	0	0	2.085125	0.262398	0.622703
small1	4	6	0.25	1	0.476389	0.202738	0.644446
small2	1	1	0	0	0.462012	0.203198	0.640497
small3	2	3	0	0	0.454506	0.195095	0.628428
small4	2	1	0.5	1	0.48092	0.207973	0.656463
small5	3	6	0.33	1	0.568903	0.267385	0.650628
small6	2	4	0.5	1	0.521615	0.206352	0.640252
small7	3	4	0	0	0.450428	0.197144	0.605292
small8	3	5	0	0	0.471399	0.196071	0.630425
small9	3	5	0.33	1	0.462075	0.201086	0.644681
small10	2	4	0	0	0.474247	0.197673	0..623865
small11	3	6	0	0	0.460687	0.201238	0.613063
K010	20	120	0	0	0.688807	0.40357	0.989557
K020	40	440	0	0	1.013349	0.656627	1.647413
K050	100	2600	0	0	2.419134	1.790165	3.45362
K100	200	10200	0	0	4.707065	2.53466	6.982243
K200	400	40400	0	0	11.496605	4.953519	18.681155
C168_FW_SZ_41	1698	5387	0.25795053	438	*	18.98164	*
C168_FW_SZ_66	1698	5401	0.25795053	438	*	21.328969	*
C168_FW_SZ_75	1698	5422	0.25795053	438	*	27.438378	*
C168_FW_SZ_107	1698	6599	0.2585394582	439	*	25.461582	*
C168_FW_UT_714	1909	7487	0.3305395495	631	*	21.587304s	*
C168_FW_UT_851	1909	7491	0.3305395495	631	*	23.083428	*
C168_FW_UT_854	1909	7486	0.3305395495	631	*	39.160652	*
C168_FW_UT_855	1909	7485	0.33053954	631	*	22.047771	*
C168_FW_UT_2469	1909	7500	0.3305395495	631	*	24.34484	*
C170_FR_RZ_32	1659	4956	0.2085593731	346	*	18.517225	*
C170_FR_SZ_58	1659	5001	0.2079566004	345	*	18.740518	*
C202_FS_RZ_44	1750	6199	0.2714285714	475	*	24.049521	*
C202_FS_SZ_121	1750	6181	0.2714285714	475	*	19.589966	*
C202_FS_SZ_74	1750	6355	0.2714285714	475	19.929274	*	*

The experiments were performed on an Intel Pentium(R) Dual-Core (2.10GHz) machine with 2G Memory running Ubuntu and the results were shown in Tables 1. We ran every instance against each solver with a timeout of 120 seconds and "*" is used to indicate the occurrence of a timeout. We use three different types of instances for the evaluation, as shown in Table 1:

Type 1 (names starting with "small"). Manually constructed instances of a small size of clauses and variables.

Type 2 (names starting with "K"). The data set that is used in [24], that is, inputs are $K_N = \{p_i, q_j, \neg p_i \vee \neg q_j \mid 1 \leq i, j \leq N\}$ for $N = 10, 20, 50, 100$. Obviously, $|Var(K_N)| = 2N$ and $|K_N| = N^2 + 2N$.

Type 3 (names starting with "C"). A large set of unsatisfiable CNF benchmarks from automotive product configuration [25], each of which encodes a set of available configurations for a product, along with constraints enforcing a specific property to be checked. Due to space limitations, only part of the results of this dataset are shown in this paper.

The meaning of each column of Table 1 is given as follows:

– "name": the name of the instance used as test datum;
– "#V" and "#C": the number of variables and clauses in the instance;

- "$IncomDegree$": the values of incompleteness degrees ;
- "n(N)": the cardinality of the incomplete set of the most preferred models
- "Encoding Algorithm": time consumed in seconds by encoding algorithms based on each partial Max-SAT solver.

From Table 1, we have the following observations:

1. For the small instances of Type 1 and instances of Type 2, all the solvers can handle them in a short time, though msuncore was faster than the other two for all these instances. But note that sat4j can be slow due to its java implementation.
2. The instances from [24] are inconsistent, but this is not the reason that their incompleteness degrees are all zero. Instead, they are zero because of Proposition 4. Indeed, Proposition 4 gives us a pre-processing way to detect such zero incompleteness degree cases, an extension that can be easily added into Algorithm 4.
3. The instances from real applications can be handled by the solvers. In particular, msuncore worked better than sat4j and clone for most of the instances except that sat4j outperformed than others for the instance C202_FS_SZ_74.

6 Conclusion and Future Work

In this paper, we have proposed a novel metric that can measure the degree of incompleteness of a propositional knowledge base, no matter if it is inconsistent or not. Based on the Belnap's four-valued semantics, this metric is semantic dependent which can allows for syntax variance. Some desired properties of this metric are shown. To compute it in practice, we have constructed a linear reduction from the computation of incompleteness degree to a Partial MaxSAT problem, thus state-of-the-art MaxSAT solvers can be used. Experiments on some manfully made and real data from industry applications have been performed, which shows that such an algorithm can be expected useful in practice.

In the future, we will study the computational complexity of the proposed metric. Since it seems not tractable in general case, we will try to study approximating algorithms to compute the incompleteness degree such that any knowledge bases of large sizes can be dealt with. Meanwhile, we will investigate other possible ways for measuring incompleteness because very few exist. We will apply such incompleteness degrees to the applications such as to guide the process of ontology generation, for which such metrics may be necessarily extended to Description Logics.

References

1. Hunter, A.: Measuring inconsistency in knowledge via quasi-classical models. In: Proc. of AAAI 2002, pp. 68–73. AAAI Press (2002)
2. Hunter, A.: How to act on inconsistent news: Ignore, resolve, or reject. Data & Knowledge Engineering 57, 221–239 (2006)
3. Zhou, L., Huang, H., Qi, G., Ma, Y., Huang, Z., Qu, Y.: Measuring inconsistency in DL-Lite ontologies. In: Proc. of WI 2009, pp. 349–356. Springer (2009)
4. Knight, K.: Measuring inconsistency. Journal of Philosophical Logic 31(1), 77–98 (2002)

5. Hunter, A., Konieczny, S.: Approaches to measuring inconsistent information. In: Bertossi, L., Hunter, A., Schaub, T. (eds.) Inconsistency Tolerance. LNCS, vol. 3300, pp. 191–236. Springer, Heidelberg (2005)
6. Grant, J.: Classifications for inconsistent theories. Notre Dame Journal of Formal Logic 19, 435–444 (1978)
7. Grant, J., Hunter, A.: Measuring inconsistency in knowledgebases. Journal of Intelligent Information Systems 27, 159–184 (2006)
8. Ma, Y., Qi, G., Hitzler, P., Lin, Z.: Measuring inconsistency for description logics based on paraconsistent semantics. In: Mellouli, K. (ed.) ECSQARU 2007. LNCS (LNAI), vol. 4724, pp. 30–41. Springer, Heidelberg (2007)
9. Grant, J., Hunter, A.: Analysing inconsistent first-order knowledge bases. Artificial Intelligence 172, 1064–1093 (2008)
10. Xiao, G., Lin, Z., Ma, Y., Qi, G.: Computing inconsistency measurements under multi-valued semantics by partial max-sat solvers. In: Proc. of KR (2010)
11. Xiao, G., Ma, Y.: Inconsistency measurement based on variables in minimal unsatisfiable subsets. In: Proc. of ECAI, pp. 864–869 (2012)
12. Baader, F., Ganter, B., Sattler, U., Sertkaya, B.: Completing description logic knowledge bases using formal concept analysis. In: Proc. of IJCAI 2007. AAAI Press (2007)
13. Lozinskii, E.L.: Resolving contradictions: A plausible semantics for inconsistent systems. J. Autom. Reasoning 12, 1–32 (1994)
14. Wong, P., Besnard, P.: Paraconsistent reasoning as an analytic tool. Journal of the Interest Group in Propostional Logic, pp. 217–229 (2001)
15. Konieczny, S., Lang, J., Marquis, P.: Quantifying information and contradiction in propositional logic through test actions. In: Proceedings of the 18th International Joint Conference on Artificial Intelligence, IJCAI 2013, pp. 106–111 (2003)
16. Knight, K.M.: Two information measures for inconsistent sets. Journal of Logic, Language and Information 12, 227–248 (2003)
17. Belnap, N.D.: A useful four-valued logic. In: Modern Uses of Multiple-Valued Logics, pp. 7–73, Reidel (1977)
18. Arieli, O., Avron, A.: The value of the four values. Artificial Intelligence 102, 97–141 (1998)
19. McAreavey, K., Liu, W., Miller, P., Meenan, C.: Tools for finding inconsistencies in real-world logic-based systems. In: Proc. of STAIRS 2012, pp. 192–203 (2012)
20. Berre, D.L.: SAT4J: A Satisfiability Library for Java (2009), `http://www.sat4j.org`
21. Marques-Silva, J.: The msuncore maxsat solver. Technical report, CASL/CSI, University College Dublin (2009)
22. Pipatsrisawat, K., Darwiche, A.: Clone: Solving weighted Max-SAT in a reduced search space. In: Orgun, M.A., Thornton, J. (eds.) AI 2007. LNCS (LNAI), vol. 4830, pp. 223–233. Springer, Heidelberg (2007)
23. Cadoli, M., Schaerf, M.: On the complexity of entailment in propositional multivalued logics. Annals of Mathematics and Artificial Intelligence 18, 29–50 (1996)
24. Ma, Y., Qi, G., Xiao, G., Hitzler, P., Lin, Z.: An anytime algorithm for computing inconsistency measurement. In: Karagiannis, D., Jin, Z. (eds.) KSEM 2009. LNCS, vol. 5914, pp. 29–40. Springer, Heidelberg (2009)
25. Sinz, C., Kaiser, A., Küchlin, W.: Formal methods for the validation of automotive product configuration data. Artificial Intelligence for Engineering Design, Analysis and Manufacturing 17, 75–97 (2003)

On the Tree Structure Used by Lazy Propagation for Inference in Bayesian Networks

Anders L. Madsen[1,2] and Cory Butz[3]

[1] HUGIN EXPERT A/S, Aalborg, Denmark
anders@hugin.com
[2] Department of Computer Science, Aalborg University, Denmark
[3] Department of Computer Science, University of Regina, Canada
butz@cs.uregina.ca

Abstract. Lazy Propagation (LP) is a propagation scheme for belief update in Bayesian networks based upon Shenoy-Shafer propagation. So far the secondary computational structure has been a junction tree (or strong junction tree). This paper describes and shows how different tree structures can be used for LP. This includes the use of different junction trees and the maximal prime subgraph decomposition organised as a tree. The paper reports on the results of an empirical evaluation on a set of real-world Bayesian networks of the performance impact of using different tree structures in LP. The results indicate that the tree structure can have a significant impact on both time and space performance of belief update.

Keywords: Bayesian networks, inference, tree structure.

1 Introduction

A Bayesian network (BN) is an efficient knowledge base for representing uncertain knowledge [22, 3, 8, 9]. It consists of a graph specifying dependence and independence relations over a set of variables and a set of conditional probability distributions (CPDs) encoding the strengths of the dependence relations effectively combining elements of probability and graph theory. Due to the intuitive graphical nature of BNs, they have and are being used for handling uncertainty in a wide range of domains.

The key element in handling uncertainty with BNs is to perform probabilistic inference or belief update, i.e., to compute posterior probabilities given (partial or incomplete) information about the state of the domain. As both exact and approximate probabilistic inference in BNs are NP-hard [2, 4], methods that in the worst case have exponential complexity are justified (unless P=NP). Methods such as Variable Elimination (VE) [29] (equivalent to Fusion [27] and Bucket elimination [5]), Symbolic Probabilistic Inference (SPI) [24, 11] and Arc-Reversal (AR) [20, 25] are often referred to as *direct methods* as they focus on computing a single posterior marginal by manipulating the set of CPDs directly. On the other hand, methods such as Lauritzen-Spiegelhalter propagation [10], HUGIN

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 400–411, 2013.

propagation [7] and Shenoy-Shafer propagation [28] are referred to as *indirect methods* as they focus on computing all posterior marginals by passing messages in a secondary computational structure.

A number of hybrid algorithms combining direct and indirect methods have been proposed such as, for instance, *factor trees* [1] and LP [18]. LP is based on a Shenoy-Shafer propagation scheme using a direct method for message computation [12–14]. In [19] an algorithm for decomposing a BN into its maximal prime subgraphs is presented. The work reported in this paper was motivated by the potential use of the maximal prime subgraph decomposition (MPD) organised into a tree as a computational structure of LP. We evaluate the use of different tree structures in LP. This includes evaluating the potential use of the MPD organised into a tree, using a (near-) optimal junction tree versus a non-optimal junction tree and a junction tree with a single node as the tree structure. The results of an extensive empirical evaluation indicate that the tree structure can have a significant impact on both time and space performance of belief update.

The rest of the paper is organised as follows. Section 2 contains preliminaries. Section 3 presents the VE and LP algorithms as used in this paper and Section 4 describes the use of LP on different tree structures. Section 5 describes the design of the empirical evaluation and the results. Section 6 discusses the findings presented in this paper. Our conclusions are contained in Section 7.

2 Preliminaries

A (discrete) *BN* $\mathcal{N} = (\mathcal{X}, G, \mathcal{P})$ consists of a set of random variables $\mathcal{X}$, an acyclic, directed graph (DAG) $G = (V, E)$ where $V \sim \mathcal{X}$ is the set of vertices and E is the set of edges and a set of CPDs $\mathcal{P}$. It represents a factorization of a joint probability distribution into a set of conditionals:

$$P(\mathcal{X}) = \prod_{X \in \mathcal{X}} P(X \,|\, \mathrm{pa}(X)), \tag{1}$$

where $\mathrm{pa}(X)$ denotes the parents of X in G and $\mathrm{fa}(X) = \mathrm{pa}(X) \cup \{X\}$.

Belief update is defined as the task of computing the posterior marginal distribution $P(X \,|\, \epsilon)$ for each non-observed variable $X \in \mathcal{X}$ given a set of evidence ϵ. An *evidence function* $f(X)$ is used to force an evidence variable X to its observed state x by assigning the value 1 to x and 0 otherwise. The set of observed variables is denoted $\mathcal{X}_\epsilon$. Barren variables are variables that are neither evidence nor target variables and have only barren descendants, if any [25].

A *probability potential* on *domain* $\mathrm{dom}(\phi) = \mathcal{Y}$ is a function ϕ such that $\phi(y) \geq 0$, for each configuration $y \in \mathcal{Y}$ and at least one $\phi(y) > 0$ [26]. A *conditional probability potential* ϕ of H given T is a probability potential of H when T is known where $\mathrm{dom}(\phi) = H \cup T$ is divided into head variables H denoted $\mathrm{head}(\phi)$ and tail variable T denoted $\mathrm{tail}(\phi)$. That is, $\mathrm{head}(\phi)$ and $\mathrm{tail}(\phi)$ are the conditioned and conditioning variables of $\mathrm{dom}(\phi)$, respectively

The *domain graph* representation $G(\phi) = (V, E)$ of a potential ϕ has vertices $V = \mathrm{dom}(\phi)$ and edges $E = \{(H_1, H_2), (H_2, H_1) \,|\, H_1, H_2 \in \mathrm{head}(\phi)\} \cup \{(T, H) \,|$

$H \in \text{head}(\phi), T \in \text{tail}(\phi)\}$. The notion of barren variables can be extended to domain graphs [13].

Let G be an undirected graph. A *clique* C is a *maximal, complete subgraph* of G. If the vertices V of a undirected graph G can be partitioned into a triple (V', S, V'') of nonempty sets where S is a complete separator of V' and V'' in G such that every path from a vertex in V' to a vertex in V'' includes a vertex in S, then G is *decomposable*; otherwise G is *prime*. A subgraph $G(U)$ of a graph $G = (V, E)$ is a *maximal prime subgraph* of G, if $G(U)$ is prime and $G(W)$ is decomposable for all W with $U \subset W \subseteq V$ [19]. The set of maximal prime subgraphs of a Bayesian network $\mathcal{N} = (\mathcal{X}, G, \mathcal{P})$ are defined with respect to G^M.

A junction tree representation $T = (\mathcal{C}, \mathcal{S})$ of $\mathcal{N}$ with cliques $\mathcal{C}$ and separators $\mathcal{S}$ is constructed from a triangulated graph G^T produced by triangulating the moral graph G^M of G. The size $s(C)$ of a clique (separator) $C \in \mathcal{C}$ ($S \in \mathcal{S}$) is defined as the combined state space size of C (S), i.e., $s(C) = \prod_{X \in C} ||X||$. The size of a junction tree T is defined as $s(T) = \sum_{C \in \mathcal{C}} s(C)$ and T over $\mathcal{N}$ is *optimal* if $s(T) \leq s(T')$ for any T' over $\mathcal{N}$. We denote an optimal junction as $\hat{T}$. The number of cliques in $\mathcal{C}$ is denoted $|\mathcal{C}|$. A junction tree with $|\mathcal{C}| = 1$ is denoted T_1.

The algorithm of [19] produces a cluster tree from a junction tree T by recursively aggregating cliques connected by incomplete separators (in G^M) to larger clusters where T should be minimal. The resulting cluster tree is referred to as the *MPD tree* $T' = (\mathcal{C}', \mathcal{S}')$ with clusters $\mathcal{C}'$ and (complete) separators $\mathcal{S}'$.

A junction tree $T = (\mathcal{C}, \mathcal{S})$ is initialised by associating each CPD $P \in \mathcal{P}$ with the smallest clique $A \in \mathcal{C}$ such that $\text{dom}(P) \subseteq A$. The set of CPDs associated with a cluster C' is defined by the aggregated cliques producing it.

For example, consider Asia [10] with BN $\mathcal{N} = (\mathcal{X}, G, \mathcal{P})$. Figure 1 shows G (i), G^M (ii), an optimal junction tree (not showing separators) $\hat{T} = (\hat{\mathcal{C}}, \hat{\mathcal{S}})$ with $|\hat{\mathcal{C}}| = 6$, $s(\hat{T}) = 40$ and $\max s(C) = 8$ (iii) and the MPD tree $T' = (\mathcal{C}', \mathcal{S}')$ with $|\mathcal{C}'| = 5$, $s(T') = 40$ and $\max s(C') = 16$ (iv). Each $P \in \mathcal{P}$ is associated with a clique $C \in \mathcal{C}$ that can hold it and BEL is the only clique with no P associated.

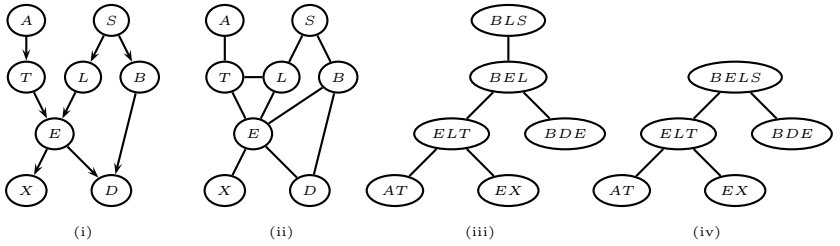


Fig. 1. (i) Asia (ii) G^M (iii) an optimal junction tree (iv) the MPD

In Asia only two cliques (BLS and BEL) are aggregated to form a single cluster ($BELS$) in $\mathcal{C}'$. The other cliques $\hat{\mathcal{C}} \setminus \{BLS, BEL\}$ remain clusters in $\mathcal{C}'$ as their adjacent separators are complete in G^M.

3 Belief Update

There exists a number of different approaches to exploiting the decomposition of $P(\mathcal{X})$ in (1) to perform belief update (efficiently). As mentioned above, this paper considers VE and LP.

3.1 Variable Elimination

In VE, a posterior marginal probability distribution $P(X \,|\, \epsilon)$ for a non-observed variable X is, in principle, computed by normalising:

$$P(X, \epsilon) = \sum_{Y \neq X} \prod_{P \in \mathcal{P}} P \prod_{Z \in \mathcal{X}_\epsilon} f(Z), \tag{2}$$

where $f(Z)$ is an evidence potential reflecting the instantiation of Z.

Barren variables are removed before variable elimination is performed. *Barren variables* have the property that when eliminated they produce a uniform likelihood over the conditioning variables and can therefore be eliminated without performing any computations. Also, we will assume that in (2) distributions are instantiated to reflect the evidence ϵ (as opposed to summing out $\mathcal{X}_\epsilon$).

The order $\rho = (Y_1, \ldots, Y_{|\mathcal{X} \setminus \{X\}|})$ in which the variable eliminations are performed is the *elimination order*. The elimination order can be identified using a range of different algorithms. Since optimal triangulation is NP-hard, heuristics are often used. The *fill-in-weight* (*fiw*) heuristic [6], for instance, aims to minimise the sum of the weights of the fill-in edges produced by a node elimination operation, i.e., $s_{fiw}(X) = \sum_{(Y_i, Y_j) \in F} ||Y_i|| \cdot ||Y_j||$, where F is the set of fill-ins added by the elimination of X.

3.2 Lazy Propagation

LP [12–14] is based on a Shenoy-Shafer scheme where messages are passed in two phases over a junction tree representation T of the BN $\mathcal{N} = (\mathcal{X}, G, \mathcal{P})$ to propagate the evidence ϵ. After initialisation and prior to message passing, each $P \in \mathcal{P}$ such that $\mathrm{dom}(\phi) \cap \mathcal{X}_\epsilon \neq \emptyset$ is instantiated to reflect ϵ. Each clique C holds an initial clique potential $\Phi_C = \{P_{i_1}, \ldots, P_{i_n}\}$ which is a set of instantiated CPDs. Propagation of evidence is the process of *collecting* and *distributing* messages to and from a chosen root of T. When VE is used for message (marginal) computation, the message passed from clique A to clique B is computed as

$$\Phi_{A \to B} = \sum_{A \setminus B} (\Phi_A \cup \bigcup_{C \in \mathrm{adj}(A) \setminus \{B\}} \Phi_{C \to A}), \tag{3}$$

where $\mathrm{adj}(A)$ are the cliques adjacent to A in T. Prior to computing (3) barren variables and potentials corresponding to domain graphs over variables all separated from B given ϵ are removed. Notice that the result is a set of potentials. Notice also that the moralization step of the junction tree compilation in effect

ignores a lot of the information contained in a DAG. A key element in LP is to use information from the DAG to improve efficiency of inference. After message passing has terminated, $P(X \mid \epsilon)$ can be computed from any $C \in \mathcal{C}$ or $S \in \mathcal{S}$ such that $X \in C$ or $X \in S$. This version of LP is referred to as LPVE.

4 Tree Structure

In previous work on LP, the secondary computational tree structure of LP has been a junction tree (or in some cases a strong junction tree [17, 13]). In [19], the authors suggest using the MPD of $\mathcal{N}$ as the computational tree structure of LP. This paper evaluates the impact on performance of using different tree structures in LP. This includes different junction trees and MPD trees. The tree structure of LP is, in principle, a structure for caching intermediate results.

4.1 Junction Tree

There exists a number of different heuristics for generating a junction tree representation T of $\mathcal{N} = (\mathcal{X}, G, \mathcal{P})$. Some algorithms such as, for instance, *fiw* are based on node elimination where the next node elimination operation is based on the node with lowest scores, some are based on decomposing the graph G into its minimal separators and others are based on exhaustive search [21].

A special case is when the junction tree has only a single clique. LP over a single clique is, in principle, equivalent to VE. In a junction tree $T_1 = (\{\mathcal{X}\}, \emptyset)$ there is no caching of intermediate results. Each marginal $P(X \mid \epsilon)$ is computed from $\mathcal{P}$ given ϵ after removing barren variables and potentials corresponding to variables separated from X given ϵ. The computations corresponds to (2).

Proposition 1. *Let $A = \mathcal{X}$ be the single cluster in a junction tree $T = (\mathcal{C} = \{\mathcal{X}\}, \emptyset)$. We have*

$$P(A, \epsilon) = \prod_{\phi \in \Phi_A} \phi \prod_{i=1}^{n} f_i, \tag{4}$$

where $\Phi_A = \mathcal{P}$ is the set of potentials associated with A.

Proof. (4) is (1) now with evidence functions. □

4.2 Maximal Prime Subgraph Decomposition Tree

The nodes of the MPD tree T' represent maximal prime subgraphs in G^M whereas the nodes of a junction tree represent maximal complete subgraphs in G^T. As mentioned in Section 2, a MPD tree T' can be constructed from a junction tree T representing any minimal triangulation G^T of G by iteratively aggregating adjacent cliques connected by an incomplete separator in G^M. By construction the structure of T' is equivalent to T up to complete separators in G^M. Each cluster $C' \in \mathcal{C}'$ represents a connected set of cliques in $\mathcal{C}$.

Propagation of evidence in a MPD tree T' is similar to propagation of evidence in a junction tree as described in Section 3.2. Messages are *collected* to and *distributed* from a chosen root of T' where messages are computed as in (3).

Proposition 2. *Let A be a cluster in a MPD tree, let S be a neighboring separator and let $\epsilon = \{\epsilon_1, \ldots, \epsilon_n\}$ be the evidence. After a full round of message passing, we have*

$$P(A, \epsilon) = \prod_{\phi \in \Phi_A} \phi \prod_{i=1}^{n} f_i \prod_{C \in \mathrm{adj}(A)} \prod_{\phi' \in \Phi_{C \to A}} \phi',$$

$$P(S, \epsilon) = \prod_{\phi \in \Phi_{S \to A}} \phi \prod_{\phi' \in \Phi_{S \leftarrow A}} \phi',$$

where Φ_A is the set of potentials associated with A, $\Phi_{C \to A}$ is the set of potentials passed to A, and $\Phi_{S \to A}$ and $\Phi_{S \leftarrow A}$ are the sets of potentials passed over S.

Proof. The MPD tree T' corresponds to a triangulated graph G^T of G^M, where each maximal prime subgraph is made complete and its set of nodes is equivalent to the cliques of the junction tree created from G^T. □

The maximal prime subgraph can be independently triangulated to produce an optimal triangulation if each maximal prime subgraph is optimally triangulated. This does, however, not take into account the independence and barren variable properties induced by a specific set of evidence. Hence, an optimal junction tree may not be the best tree structure for belief update using LP given a specific set of evidence.

4.3 Example

In Asia of Figure 1 (i), consider the calculation of $P(D)$ and $P(E)$ using three different structures, namely, a single cluster tree, an optimal tree $\hat{T}$ in Figure 1 (iii), and the MPD tree in Figure 1 (iv). Using a single cluster with all variables to guide the computation will identify X as barren relative to $P(D)$ and eliminate the remaining variables, for example, in the order $\rho_D = (A, T, L, S, E, B)$ and identify $\{D, B, X\}$ as barren relative to $P(E)$ and eliminate the remaining variables, for instance, in the order $\rho_E = (A, T, L, S)$. Notice the amount of repeated computations. In an optimal tree $\hat{T}$, $P(E)$ and $P(D)$ can be computed from clique BDE after a collect to it (even though this may not be the optimal choice as $P(E)$ can be computed more efficiently from EX). Variables A and X are identified as barren relative to the message passed to BDE while B is not identified as barren as the elimination of S creates the potential $\phi(B, L)$. Note that the elimination of A and T is *cached* at BEL eliminating the repeated computations. Lastly, in the MPD tree T', the situation is the same as for $\hat{T}$ except that in the cluster $BELS$ there are more degrees of freedom to determine ρ than in the clique BEL as the latter case corresponds to restricting ρ (in $BELS$) to have S as the first variable. This may in some cases be suboptimal.

5 Experimental Analysis

We give an empirical evaluation of the performance impact of using different secondary computational structures in LP based upon a set of real-world BNs.

5.1 Setup

Table 1[1] shows statistics on the BNs used in the evaluation and their optimal (or believed to be near-optimal) junction tree ($\hat{T}$), a junction tree generated using s_{fiw} (T_{fiw}) and the MPD tree (T'), respectively. In the table $|\mathcal{Y}|$ is the cardinality of $\mathcal{Y}$ and sizes are on a log-scale in base 10. The junction trees have been generated using the *total weight* and *fill-in-weight* heuristics as implemented in the HUGIN tool [6, 16]. The test set consists of networks of different size and complexity in terms of the size of the tree structure.

For each network, one hundred sets of evidence have been generated at random. For each evidence set, LPVE computes the posterior marginal distribution of each non-evidence variable. The same set of evidence sets is used to evaluate each tree structure for a specific network. In the experiments, the *fiw* heuristic is applied to determine the online elimination order when computing messages and posterior marginals [15].

The experiments were performed using a Java implementation (Java (TM) SE Runtime Environment, Standard Edition (build 1.7.0_10-b18)) running on a Linux Ubuntu 12.10 (kernel 3.5.0-21-generic) PC with an Intel Core i7(TM) 920 Processor (2.67GHz) and 12 GB RAM.

5.2 Results

Table 2 presents the time performance results of the evaluation for $\hat{T}$, T_{fiw}, T_1 and T', respectively. Table 2 shows the sample average run-time in seconds and the sample variance for propagating one hundred sets of evidence generated at random, for each network and each type of secondary computational structure.

Table 3 shows size of the largest potential created during belief update using tree structures $\hat{T}$, T_{fiw}, T_1 and T', respectively. The table shows the sample average and variance when propagating one hundred sets of randomly generated evidence, for each network and each type of secondary computational structure. The time performance measurements include time for finding the on-line triangulation orders and do not include time used to generate the secondary computational structure. It is expected that on-line triangulation is more expensive for T_1 and T' than for $\hat{T}$ and T_{fiw}.

Notice that two different implementations of the *fill-in-weight* heuristic have been used. The junction trees have been generated using the HUGIN tool while the online triangulation have been generated using our own implementation. This may in part explain why T_1 produces a larger average largest potential size than the largest clique in T_{fiw}. The total cost of online triangulation is expected

[1] The size of the largest cluster for Diabetes cannot be represented using a Java double.

Table 1. Description of test BNs, $\hat{T}$, T_{fiw} and T' where * means that the triangulation is optimal, ** means that the triangulation has been created using a maximum of 200,000 separators and no * means that the best known triangulation is used

$\mathcal{N}$	$\lvert\mathcal{X}\rvert$	$\lvert\hat{\mathcal{C}}\rvert$	$\lvert\mathcal{C}_{fiw}\rvert$	$\lvert\mathcal{C}'\rvert$	max $s(\hat{C})$	max $s(C_{fiw})$	max $s(C')$	$s(\hat{T})$	$s(T_{fiw})$	$s(T')$
3nt*	58	41	41	22	3.5	3.7	16.8	4.1	4.4	16.8
Barley*	48	36	36	14	6.9	6.9	29.5	7.2	7.3	29.5
Diabetes	413	337	337	77	4.9	5.5	-	7.0	7.1	-
Hepar_II*	70	58	58	55	2.6	2.6	2.9	3.4	3.4	3.5
KK*	50	38	38	15	6.8	6.8	30.2	7.1	7.2	30.2
Mildew*	35	29	28	15	6.1	6.6	20.6	6.5	7.0	20.6
Munin1	189	162	160	70	7.6	7.9	69.2	7.9	8.3	69.2
Munin2	1,003	854	860	48	5.2	5.7	189.6	6.3	6.7	189.6
Munin3	1,044	904	904	53	5.2	5.2	174.5	6.5	6.5	174.5
Munin4	1,041	877	875	49	5.7	5.9	221.9	6.9	7.1	221.9
Water*	32	21	19	9	5.8	6.2	13.3	6.5	6.6	13.3
andes**	223	180	175	79	4.8	5.4	40.0	5.3	5.6	40.0
cc145*	145	140	140	13	3.0	3.0	3.0	3.6	3.6	3.6
cc245*	245	235	235	23	5.4	5.4	6.0	5.8	5.8	6.3
hailfinder*	56	43	43	29	3.5	3.5	11.6	4.0	4.0	11.6
medianus*	56	44	44	15	5.7	5.7	28.4	6.1	6.2	28.4
oow*	33	22	22	6	6.3	6.8	21.7	6.8	7.3	21.7
oow_bas*	33	19	19	8	5.7	6.2	18.4	6.3	6.6	18.4
oow_solo*	40	29	28	9	6.2	7.2	24.2	6.7	7.5	6.3
pathfinder*	109	91	91	86	4.5	4.5	6.8	5.3	5.3	6.8
sacso**	2,371	1,229	1,175	98	5.2	6.4	107.5	6.0	6.8	107.5
ship*	50	35	35	10	6.6	8.1	35.6	7.4	8.4	35.6
system_v57*	85	75	72	26	4.8	6.7	57.9	6.1	6.8	57.9
win95pts*	76	50	50	33	2.7	2.7	9.3	3.4	3.4	9.3

to be higher for T_1 as elimination orders on average are expected to be *longer* for this structure in the following sense. For T_1 all variables are in a single clique. This means that to compute any posterior marginal $P(X \mid \epsilon)$ all variables $\mathcal{X} \setminus \{X\}$ have to be eliminated (in principle) and the elimination order has length $|\mathcal{X}| - 1$. Using T_{fiw}, on the other hand, each marginal $P(X \mid \epsilon)$ is computed from any clique or separator containing X. Since the number of variables in the largest clique is usually much smaller than $|\mathcal{X}|$, the elimination orders are usually much shorter for T_{fiw}. The implementation of the online triangulation has not been optimised to cope with large domain graphs.

Observe that for a few networks average time performance on T_1 and T' is much worst than T_{fiw} and $\hat{T}$ with a high variance. For a few evidence sets the time performance is significantly worse for these structures. For instance, the average run-time performance on Diabetes is high with a high variance.

T_1 seems to have the worst time performance except for a few instances, while time performance of T' in one case is much worse than T_1 (as well as T_{fiw} and $\hat{T}$) and in a number of cases is comparable with the performance of T_{fiw} and $\hat{T}$. In almost all cases time performance of T_{fiw} is similar to the performance of $\hat{T}$,

Table 2. Run-time in seconds (mean ± standard deviation)

$\mathcal{N}$	$\hat{T}$	T_{fiw}	T_1	T'
3nt*	0.03±0.00	0.03±0.00	0.05±0.04	0.04±0.00
Barley*	0.13±0.18	0.15±0.21	0.34±0.64	0.33±0.61
Diabetes	0.45±0.39	0.47±0.41	27.69±72.84	93.83±282.90
Hepar_II*	0.05±0.00	0.05±0.03	0.1±0.07	0.05±0.00
KK*	0.12±0.15	0.14±0.18	0.38±0.63	0.22±0.32
Mildew*	0.06±0.06	0.08±0.10	0.22±0.67	0.17±0.50
Munin1	0.84±1.99	1.49±4.40	4.49±19.47	3.98±18.53
Munin2	0.6±0.25	0.61±0.26	8.37±11.50	1.6±1.90
Munin3	0.81±0.41	0.81±0.41	25.14±46.30	8.06±17.20
Munin4	0.75±0.38	0.76±0.41	15.25±22.70	3.96±6.13
Water*	0.08±0.08	0.07±0.06	0.09±0.11	0.09±0.11
andes**	0.19±0.09	0.18±0.09	0.69±0.69	0.51±0.39
cc145*	0.12±0.06	0.12±0.06	0.14±0.10	0.11±0.06
cc245*	0.27±0.12	0.26±0.12	0.35±0.30	0.26±0.12
hailfinder*	0.04±0.00	0.04±0.00	0.09±0.07	0.04±0.00
medianus*	0.05±0.03	0.06±0.04	0.1±0.14	0.09±0.13
oow*	0.1±0.12	0.14±0.22	0.17±0.44	0.13±0.24
oow_bas*	0.05±0.04	0.07±0.08	0.08±0.10	0.06±0.06
oow_solo*	0.1±0.12	0.29±0.61	0.79±3.29	0.55±2.14
pathfinder*	0.15±0.11	0.15±0.11	0.15±0.13	0.14±0.11
sacso**	0.66±0.25	0.67±0.26	44.51±76.62	1.85±2.20
ship*	0.25±0.46	1.4 ±4.90	1.42±5.13	0.88±3.76
system_v57*	0.09±0.05	0.12±0.14	0.71±1.56	0.6±1.35
win95pts*	0.05±0.00	0.05±0.00	0.12±0.07	0.08±0.04

while $\hat{T}$ is better than T_{fiw} in a few cases. On the other hand, in many cases the space performance of T_1 and T' is better than the space performance of T_{fiw} and $\hat{T}$. There are a few significant exceptions though, which is surprising.

6 Discussion and Analysis

Traditionally, LP has been based on message passing in a junction tree representation of a BN. This paper has described and evaluated how different tree structures can be used for LP. This includes different junction trees, MPD trees and junction trees with a single clique.

The identification of the MPD tree can be relatively efficient compared to finding the optimal junction tree, which can be a relatively expensive operation. The MPD is identified using a minimal triangulation and a linear search guided by the junction tree. The classical triangulation algorithm LEX M [23] can be used to determine a minimal triangulation with time complexity $\mathcal{O}(ne)$, where n is the number of vertices and e is the number of edges in the graph [23]. The complexity of constructing the MPD tree from a minimal junction tree is $\mathcal{O}(n^2)$ [19]. In the evaluation, we have generated MPD trees from the junction trees generated using *total-weight*.

Table 3. Size of largest potential (mean ± standard deviation)

$\mathcal{N}$	$\hat{T}$	T_{fiw}	T_1	T'
3nt*	2.9±3.0	2.9±3.1	2.6±2.7	2.6±2.7
Barley*	5.7±6.2	5.7±6.2	5.2±5.6	5.3±5.7
Diabetes	4.6±4.5	5.0±5.1	6.3±7.0	7.2±7.9
Hepar_II*	2.0±2.1	2.0±2.1	2.0±2.1	2.0±2.1
KK*	5.7±6.1	5.7±6.1	5.3±5.7	5.3±5.7
Mildew*	5.3±5.6	5.6±6.0	5.4±5.9	5.4±5.9
Munin1	6.3±6.8	6.6±7.0	6.0±6.7	6.1±6.8
Munin2	4.3±4.5	4.6±5.0	3.6±3.9	3.6±3.9
Munin3	4.6±4.8	4.6±4.8	5.1±5.5	5.1±5.5
Munin4	5.0±5.2	5.2±5.4	4.9±5.3	4.9±5.3
Water*	4.9±5.2	4.9±5.2	4.6±5.0	4.6±5.0
andes**	3.5±3.9	3.5±3.9	2.8±3.1	2.8±3.1
cc145*	2.2±2.3	2.2±2.3	2.2±2.3	2.2±2.3
cc245*	4.0±4.2	4.0±4.2	3.9±4.2	3.9±4.2
hailfinder*	3.0±3.1	3.0±3.1	2.8±3.0	2.8±3.0
medianus*	4.6±5.0	4.7±5.1	4.4±5.3	4.4±5.3
oow*	5.4±5.7	5.9±6.3	5.8±6.5	5.5±6.2
oow_bas*	4.9±5.2	5.4±5.7	5.1±5.6	5.1±5.6
oow_solo*	5.5±5.7	6.1±6.5	5.8±6.3	5.9±6.6
pathfinder*	3.8±4.0	3.8±4.0	3.8±4.0	3.8±4.0
sacso**	3.8±4.1	4.3±4.7	3.4±3.9	3.4±3.8
ship*	5.9±6.1	6.9±7.5	6.5±7.3	5.7±6.2
system_v57*	4.5±4.4	5.5±6.0	5.8±6.4	5.9±6.5
win95pts*	2.1±2.2	2.1±2.2	1.9±2.0	1.9±2.0

A junction tree is a caching structure. It caches in the separator potentials the results of intermediate variable elimination operations. This may give $\hat{T}$ an advantage over T_1 which has to identify an complete elimination order for each posterior marginal. On the other hand, $\hat{T}$ is *wide enough* to accommodate any set of evidence. This may be a disadvantage compared to T_1, which can exploit all information in the structure of the evidence. The results reported in this paper indicates that for only a few networks the time performance is insensitive to the tree structure, e.g., for *pathfinder* and *Water* the four structures considered produce almost equal time performance. In some cases the time performance is almost the same for $\hat{T}$, T' and T_{fiw}. This is the case, e.g., for *Hepar_II* and *hailfinder*. In other cases, the time performances of $\hat{T}$ and T_{fiw} are similar, whereas the time performances of T' and T_1 are much worse. This is the case, e.g., for *Barley*, *Munin4* and *Mildew*. In some cases the time performance of T_1 or/and T' is poor compared to the other algorithms. In these cases, the time performance variance is very high. This indicates that the time performance is poor on a few sets of evidence producing a high average time performance. The poor time performance is due to large potentials created during belief update and the large potentials are created due to a poor elimination order. It should be noted that in some cases $\hat{T}$ is not known to be optimal (finding the optimal triangulation is infeasible as the number of minimum separators in G^M is large).

In general, the evaluation illustrates that the tree structure can have a significant impact on performance. In most cases, $\hat{T}$ and T_{fiw} produce the best results. In almost all cases (except one) T_1 produced the worst results. Notice that in some cases T_1 produces a larger largest potential than T_{fiw}.

7 Conclusion

This paper has considered the impact of the secondary computational structure used by LP in belief update. The results of the empirical evaluation indicate that the tree structure can have a significant impact on both time and space performance of belief update. The structures $\hat{T}$ and T_{fiw} most often produced the best performance on the networks considered in the evaluation.

Future work includes assessing the impact of using a binary tree structure such as the binary join tree [27] as well as evaluating different variants of LP such as LP using AR or SPI as the message computation algorithm. In addition, the option to consider *almost* complete separators as complete should be considered in order to divide large maximal prime subgraphs into smaller clusters, i.e., to increase the level of caching in the tree structure.

Acknowledgments. We would like to thank the reviewers for their insightful comments, which have improved the quality of the paper.

References

1. Bloemeke, M., Valtorta, M.: A Hybrid Algorithm to compute Marginal and Joint Beliefs in Bayesian Networks and Its Complexity. In: Proc. of the UAI, pp. 16–23 (1998)
2. Cooper, G.F.: The computational complexity of probabilistic inference using Bayesian belief networks. Artificial Intelligence 42(2-3), 393–405 (1990)
3. Cowell, R.G., Dawid, A.P., Lauritzen, S.L., Spiegelhalter, D.J.: Probabilistic Networks and Expert Systems. Springer (1999)
4. Dagum, P., Luby, M.: Approximating probabilistic inference in Bayesian belief netwoks is NP-hard. Artificial Intelligence 60, 141–153 (1993)
5. Dechter, R.: Bucket elimination: A unifying framework for probabilistic inference. Artificial Intelligence 113(1-2), 41–85 (1999)
6. Jensen, F.V.: HUGIN API Reference Manual. HUGIN EXPERT A/S, Reference Manual for the HUGIN version 7.7 (2012), `http://www.hugin.com`
7. Jensen, F.V., Lauritzen, S.L., Olesen, K.G.: Bayesian updating in causal probabilistic networks by local computations. Computational Statistics Quarterly 4, 269–282 (1990)
8. Jensen, F.V., Nielsen, T.D.: Bayesian Networks and Decision Graphs, 2nd edn. Springer (2007)
9. Kjærulff, U.B., Madsen, A.L.: Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis, 2nd edn. Springer (2012)
10. Lauritzen, S.L., Spiegelhalter, D.J.: Local computations with probabilities on graphical structures and their application to expert systems. Journal of the Royal Statistical Society, B 50(2), 157–224 (1988)

11. Li, Z., D'Ambrosio, B.: Efficient Inference in Bayes Networks as a Combina torial Optimization Problem. Int. J. of Approximate Reasoning 11(1), 55–81 (1994)
12. Madsen, A.L.: An empirical evaluation of possible variations of lazy propagation. In: Proc. of the UAI, pp. 366–373 (2004)
13. Madsen, A.L.: Variations Over the Message Computation Algorithm of Lazy Propagation. IEEE TSMC Part B 36(3), 636–648 (2006)
14. Madsen, A.L.: Improvements to Message Computation in Lazy Propagation. Int. J. of Approximate Reasoning 51(5), 499–514 (2010)
15. Madsen, A.L., Butz, C.J.: On the Importance of Elimination Heuristics in Lazy Propagation. In: Sixth European Workshop on Probabilistic Graphical Models, pp. 227–234 (2012)
16. Madsen, A.L., Jensen, F.V., Kjærulff, U.B., Lang, M.: Hugin - the tool for bayesian networks and influence diagrams. International Journal on Artificial Intelligence Tools 14(3), 507–543 (2005)
17. Madsen, A.L., Jensen, F.V.: Lazy Evaluation of Symmetric Bayesian Decision Problems. In: Proc. of the UAI, pp. 382–390 (1999)
18. Madsen, A.L., Jensen, F.V.: Lazy propagation: A junction tree inference algorithm based on lazy evaluation. Artificial Intelligence 113(1-2), 203–245 (1999)
19. Olesen, K.G., Madsen, A.L.: Maximal Prime Subgraph Decomposition of Bayesian Networks. IEEE TSMC Part B 32(1), 21–31 (2002)
20. Olmsted, S.M.: On representing and solving decision problems. PhD thesis, Department of Engineering-Economic Systems, Stanford University, CA (1983)
21. Ottosen, T.J., Vomlel, J.: All roads lead to Rome - New search methods for the optimal triangulation problem. Int. J. of Approximate Reasoning 53(9), 1350–1366 (2012)
22. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Series in Representation and Reasoning. Morgan Kaufmann Publishers, San Mateo (1988)
23. Rose, D.J., Tarjan, R.E., Lueker, G.S.: Algorithmic aspects of vertex elimination on graphs. SIAM Journal of Computing 5(2), 266–283 (1976)
24. Shachter, R., D'Ambrosio, B., DelFavero, B.: Symbolic probabilistic inference in belief networks. In: Proc. Eighth National Conference on AI, pp. 126–131 (1990)
25. Shachter, R.D.: Evaluating influence diagrams. Operations Research 34(6), 871–882 (1986)
26. Shafer, G.R.: Probabilistic Expert Systems. SIAM (1996)
27. Shenoy, P.P.: Binary join trees for computing marginals in the Shenoy-Shafer architecture. Int. J. of Approximate Reasoning 17(2-3), 239–263 (1997)
28. Shenoy, P.P., Shafer, G.: Axioms for probability and belief-function propagation. In: Proc. of the UAI, pp. 169–198 (1990)
29. Zhang, N.L., Poole, D.: A simple approach to bayesian network computations. In: Proc. of the Canadian Conference on AI, pp. 171–178 (1994)

Hierarchical Model for Rank Discrimination Measures

Christophe Marsala[1] and Davide Petturiti[2]

[1] LIP6, Université Pierre et Marie Curie (Paris 6), France
christophe.marsala@lip6.fr
[2] Dip. Matematica e Informatica, Università di Perugia, Italy
davide.petturiti@dmi.unipg.it

Abstract. In this paper we focus on rank discrimination measures, i.e., functions able to quantify the discrimination power of an attribute w.r.t. the class, taking into account the monotonicity of the class w.r.t. the attribute. These measures are used in decision tree induction in order to enforce a local form of monotonicity of the class w.r.t. the splitting attribute and are characterized by a noticeable robustness to non-monotone noise present in the data. More precisely, here we present a hierarchical model in order to single out which properties a function must satisfy to be a rank discrimination measure, providing in this way a framework for the construction of new measures.

Keywords: Rank Discrimination Measure, Monotone Classification, Decision Tree Induction.

1 Introduction

Monotone classification is a relatively recent topic in machine learning founding its roots on problems deriving from economy, social sciences and medicine. Indeed, in these domains it is quite common to consider a set of objects $\Omega = \{\omega_1, \ldots, \omega_n\}$ described by attributes a_j's each ranging in a totally ordered set X_j and labelled by a function λ ranging itself in a totally ordered set of classes C. This is motivated by the fact that the introduction of order structures increases the expressive power of the decision model, allowing the representation of semantic concepts such as preference, priority, importance and so on. Moreover, it enables the decision model to highlight gradual dependencies between attributes and the set of classes.

It is easily seen that the just described problem has a deep analogy with the standard classification problem [17,3], anyway, in this context an additional monotonicity constraint on the classifier is imposed, in a way to express the intuitive idea that *objects with better attribute values should not be labelled with a worse class.*

More formally, denoting with X the description space generated by the X_j's, the monotone classification problem (see, e.g., [16]) consists in determining a monotone extension $\lambda' : X \to C$, of a monotone consistent labelling function

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 412–423, 2013.

$\lambda : E \to C$ defined on a set of examples $E \subseteq X$. Nevertheless, real data do not guarantee in general any form of consistence, that is, λ could not be monotone on E or, even worse, λ could be just a relation on $E \times C$.

Hence, proceeding in a strict sense, a solution to the problem requires to assume that the dataset is monotone consistent: thus, a preprocessing of the data could be necessary with ensuing loss of information [16,5]. Previous remark suggests that the monotone consistency assumption is quite strong for real applications. For this, methods not imposing any restriction on the dataset have been proposed in the relevant literature but they have as primary goal the monotonicity of the classifier not considering the classification accuracy as a primary objective. Moreover, in [2] it is shown that the existing monotone classifiers [1,4,6] are deeply affected by the presence of non-monotone noise in the data.

It is our opinion that a "monotone" classifier should not require any particular assumption on the data but it should be able to work on the original dataset as it is; at the same time we think that such a classifier should be able to exploit the possible monotonicity present in the data in a way to increase its prediction power and reach a better understanding of the data structure.

In this work we consider decision tree classifiers, whose construction is usually carried on using an *inductive algorithm* which builds the tree from the root to the leaves by a recursive *partitioning* of the dataset. It is well-known that the existing algorithms are essentially distinguished by the *discrimination measure* [14,13] they use for the data splitting and that the measures typically used for this task are not sensitive to monotonicity [15,10]. More precisely, our objective is to build a decision tree able to exploit "somehow" the possible monotonicity present in the data, but, since no monotonicity hypothesis is asked on the input dataset, we need to relax the requirements. Indeed, the "global" monotonicity constraint acts on the final classifier λ' and so it is particularly difficult to enforce during an inductive procedure, since at each step a single attribute can be taken into account.

Here we adopt a completely greedy approach: at each step of the construction we choose the attribute a_j enforcing the most the *local monotonicity constraint*, that is for every $\omega_i, \omega_h \in \Omega$,

$$a_j(\omega_i) \leq a_j(\omega_h) \;\Rightarrow\; \lambda(\omega_i) \leq \lambda(\omega_h),$$

where $a_j(\omega_i)$ and $\lambda(\omega_i)$ are the values of the attribute a_j and the labelling function λ on the object ω_i. As a consequence, we cannot expect a globally monotone classifier at the end of the procedure. Moreover, to accomplish such a construction, we need discrimination measures able to quantify the monotonicity of λ with respect to a_j which are, at the same time, robust to non-monotone noise in the data.

In [15] we applied the same generalization procedure proposed in [10] for the case of Shannon entropy, to other two well-known discrimination measures, such as the Gini dispersion index [3] and the Yuan and Shaw ambiguity measure [18], moreover we directly introduced a third measure that we called *pessimistic*. In the same paper we showed that only the first and the last among the new

functions behave as proper rank discrimination measures, i.e., they are sensitive to monotonicity and robust to non-monotone noise. To prove the effectiveness of the new measures we created the binary tree classifier RDMT(H^*) [15], which is parametrized by a rank discrimination measure H^* among the studied ones. Such classifier is implemented in Java using the WEKA framework and is essentially inspired to the REMT classifier described in [10]. An experimental analysis on artificial and real data showed that our classifier is able to exploit the possible monotonicity in the dataset: it can compete with non-monotone classifiers in accuracy [17,3] and it is much more robust to noise than monotone classifiers [1,4,6].

In this paper we present a hierarchical model in the spirit of [14,13,12] in a way to highlight which properties a function must satisfy to be called a rank discrimination measure. This model constitutes also a basis for the introduction of further new measures as we show by two examples.

The paper is organized as follows. In Section 2 we recall the rank discrimination measures introduced in [15], while in Section 3 we present the hierarchical model characterizing the functional structure of such measures. Finally, in Section 4 we present two new rank discrimination measures that we use in RDMT(H^*) to perform a comparative analysis with other well-known monotone classifiers on artificial data.

2 Rank Discrimination Measures

Consider a set of objects $\Omega = \{\omega_1, \dots, \omega_n\}$ described by a family $\mathcal{A} = \{a_1, \dots, a_m\}$ of *attributes* ranging in a finite totally ordered set (also called *true criteria* in [4]), that is for every $j = 1, \dots, m$, a_j is a function on Ω ranging in $X_j = \{x_{j_1}, \dots, x_{j_{t_j}}\}$ with $t_j > 1$ and $(X_j, \leq)$ totally ordered. Assume also a *labelling function* $\lambda : \Omega \to C$ is given, where $C = \{c_1, \dots, c_k\}$ is a set of *classes* with $k > 1$ and $(C, \leq)$ also totally ordered.

Let us stress that, for $i = 1, \dots, n$, every ω_i can be represented by a $(m+1)$-tuple $(a_1(\omega_i), \dots, a_m(\omega_i), \lambda(\omega_i))$, obtaining a *set of examples*, moreover the description space $X = X_1 \times \cdots \times X_m$ forms a *lattive* $(X, \leq)$ where for every $x, y \in X$,

$$x \leq y \;\Leftrightarrow\; x_j \leq y_j, \text{ for } j = 1, \dots, m. \tag{1}$$

Remark 1. To avoid cumbersome notation we use the same symbol $\leq$ for all the orders: the context will clarify which relation we refer to.

We stress that every attribute $a_j \in \mathcal{A}$ as well as the labelling function λ determine a partition of Ω whose elements are denoted, respectively, as

$$\{a_j = x_{j_s}\} = \{\omega_h \in \Omega \;:\; a_j(\omega_h) = x_{j_s}\}, s = 1, \dots, t_j,$$
$$\{\lambda = c_q\} = \{\omega_h \in \Omega \;:\; \lambda(\omega_h) = c_q\}, q = 1, \dots, k,$$

moreover, the same partitions can be object-wise written, denoting for every $\omega_i \in \Omega$

$$[\omega_i]_{a_j} = \{\omega_h \in \Omega \ : \ a_j(\omega_i) = a_j(\omega_h)\},$$
$$[\omega_i]_{\lambda} = \{\omega_h \in \Omega \ : \ \lambda(\omega_i) = \lambda(\omega_h)\},$$

where for every $\omega_h \in [\omega_i]_{a_j}$ it holds $[\omega_h]_{a_j} = [\omega_i]_{a_j}$ and, analogously, for every $\omega_h \in [\omega_i]_{\lambda}$ it holds $[\omega_h]_{\lambda} = [\omega_i]_{\lambda}$.

Following the procedure showed in [10], in [15] we provided the following object-wise writing of the Shannon and Gini discrimination measures, respectively.

Proposition 1. *Let* $p_s = \frac{|\{a_j = x_{j_s}\}|}{|\Omega|}$ *and* $p_{q,s} = \frac{|\{\lambda = c_q\} \cap \{a_j = x_{j_s}\}|}{|\Omega|}$*:*

$$H_S(\lambda|a_j) = \sum_{s=1}^{t_j} p_s \left(- \sum_{q=1}^{k} \left(\frac{p_{q,s}}{p_s} \right) \log_2 \left(\frac{p_{q,s}}{p_s} \right) \right)$$
$$= \sum_{i=1}^{|\Omega|} \frac{1}{|\Omega|} \left(- \log_2 \left(\frac{|[\omega_i]_{\lambda} \cap [\omega_i]_{a_j}|}{|[\omega_i]_{a_j}|} \right) \right);$$
$$H_G(\lambda|a_j) = \sum_{s=1}^{t_j} p_s \left(1 - \sum_{q=1}^{k} \left(\frac{p_{q,s}}{p_s} \right)^2 \right)$$
$$= \sum_{i=1}^{|\Omega|} \frac{1}{|\Omega|} \left(1 - \frac{|[\omega_i]_{\lambda} \cap [\omega_i]_{a_j}|}{|[\omega_i]_{a_j}|} \right).$$

Now the generalization procedure proposed in [10] refers to the concept of *dominance* originally introduced in the context of rough sets (see [7,8]) by the notion of *dominant set* generated, respectively, by a_j and λ. For every $\omega_i \in \Omega$, define

$$[\omega_i]^{\leq}_{a_j} = \{\omega_h \in \Omega \ : \ a_j(\omega_i) \leq a_j(\omega_h)\}, \tag{2}$$
$$[\omega_i]^{\leq}_{\lambda} = \{\omega_h \in \Omega \ : \ \lambda(\omega_i) \leq \lambda(\omega_h)\}. \tag{3}$$

At this point the rank versions of the previously introduced discrimination measures is simply obtained substituting in the object-wise writing, the equivalence classes $[\omega_i]_{\lambda} \cap [\omega_i]_{a_j}$ and $[\omega_i]_{a_j}$ with the corresponding dominant sets. Definition 1 reports the rank version of Shannon and Gini discrimination measures, respectively (we keep conditional notation just for uniformity).

Definition 1

$$H^*_S(\lambda|a_j) = \sum_{i=1}^{|\Omega|} \frac{1}{|\Omega|} \left(- \log_2 \left(\frac{|[\omega_i]^{\leq}_{\lambda} \cap [\omega_i]^{\leq}_{a_j}|}{|[\omega_i]^{\leq}_{a_j}|} \right) \right);$$
$$H^*_G(\lambda|a_j) = \sum_{i=1}^{|\Omega|} \frac{1}{|\Omega|} \left(1 - \frac{|[\omega_i]^{\leq}_{\lambda} \cap [\omega_i]^{\leq}_{a_j}|}{|[\omega_i]^{\leq}_{a_j}|} \right).$$

In Definition 1, the ratio $\frac{|[\omega_i]^{\leq}_{\lambda} \cap [\omega_i]^{\leq}_{a_j}|}{|[\omega_i]^{\leq}_{a_j}|}$ is a satisfaction measure of the *local monotonicity constraint* for a fixed $\omega_i \in \Omega$, which quantifies the validity of

$$a_j(\omega_i) \leq a_j(\omega_h) \;\Rightarrow\; \lambda(\omega_i) \leq \lambda(\omega_h),$$

for every $\omega_h \in \Omega$. Indeed, it is easy to show that such ratio is 1 if and only if the local monotonicity constraint for a fixed ω_i is completely satisfied.

In the rest of the paper, to simplify notation, for a fixed $a_j \in \mathcal{A}$ and λ denote:

$$\mathrm{dsr}(\omega_i) = \frac{|[\omega_i]^{\leq}_{\lambda} \cap [\omega_i]^{\leq}_{a_j}|}{|[\omega_i]^{\leq}_{a_j}|}, \tag{4}$$

$$\mathrm{mindsr}(\omega_i) = \frac{\min\limits_{\omega_h \in [\omega_i]_{a_j}} |[\omega_h]^{\leq}_{\lambda} \cap [\omega_h]^{\leq}_{a_j}|}{|[\omega_i]^{\leq}_{a_j}|}, \tag{5}$$

$$\mathrm{maxdsr}(\omega_i) = \frac{\max\limits_{\omega_h \in [\omega_i]_{a_j}} |[\omega_h]^{\leq}_{\lambda} \cap [\omega_h]^{\leq}_{a_j}|}{|[\omega_i]^{\leq}_{a_j}|}, \tag{6}$$

$$\mathrm{avgdsr}(\omega_i) = \frac{\sum\limits_{\omega_h \in [\omega_i]_{a_j}} \frac{|[\omega_h]^{\leq}_{\lambda} \cap [\omega_h]^{\leq}_{a_j}|}{|[\omega_i]_{a_j}|}}{|[\omega_i]^{\leq}_{a_j}|}. \tag{7}$$

Notice that the function dsr considers only the object ω_i, while the functions mindsr, maxdsr and avgdsr consider all the objects "in the same conditions" for what concerns the attribute a_j, that is those belonging to the equivalence class $[\omega_i]_{a_j}$. In particular, it holds for every $\omega_h \in [\omega_i]_{a_j}$, $\mathrm{mindsr}(\omega_h) = \mathrm{mindsr}(\omega_i)$, $\mathrm{maxdsr}(\omega_h) = \mathrm{maxdsr}(\omega_i)$ and $\mathrm{avgdsr}(\omega_h) = \mathrm{avgdsr}(\omega_i)$.

Remark 2. In [15] we provided also the rank generalization of the Yuan and Shaw measure but since the obtained function is not a good rank discrimination measure we will not present it here.

Finally, in [15] we directly introduced the following measure that, due to its conservative nature, has been called *pessimistic*.

Definition 2

$$H^*_P(\lambda|a_j) = \sum_{i=1}^{|\Omega|} \frac{1}{|\Omega|} \left(-\frac{\log_2 (\mathrm{mindsr}(\omega_i))}{\mathrm{mindsr}(\omega_i)} \right).$$

3 Hierarchical Model for Rank Discrimination Measures

In the spirit of [14,13,12] we aim to develop a *hierarchical model* for rank discrimination measures, with the goal of isolating which properties a function must satisfy to be a measure of this type. As a side effect, the definition of a

hierarchical model is also important since it provides a base for creating new measures.

Indeed, after a careful look all the measures presented so far share a common functional structure, in which we can distinguish three functions F^*, G^* and H^*, composed hierarchically. In particular, for fixed λ and a_j, the H^*-layer considers all the objects in Ω, while both the G^*-layer and the F^*-layer take into account a single object ω_i. Table 1 lists the different layers for the measures introduced so far. We will use subscripts S, G and P to refer to layers F^*, G^* and H^* of each measure.

Table 1. Hierarchical model for measures H^*_S, H^*_G and H^*_P

Layer	Shannon	Gini	Pessimistic
F^*	$\mathrm{dsr}(\omega_i)$		$\mathrm{mindsr}(\omega_i)$
G^*	$-\log_2 F^*(\omega_i)$	$1 - F^*(\omega_i)$	$-\frac{\log_2 F^*(\omega_i)}{F^*(\omega_i)}$
H^*	$\sum_{i=1}^{\lvert\Omega\rvert} \frac{1}{\lvert\Omega\rvert} G^*(F^*(\omega_i))$		

The F^*-layer is a function quantifying the validity of the local monotonicity constraint of λ with respect to a_j for a fixed $\omega_i \in \Omega$, i.e., it measures the satisfaction of $a_j(\omega_i) \leq a_j(\omega_h) \;\Rightarrow\; \lambda(\omega_i) \leq \lambda(\omega_h)$, for every $\omega_h \in \Omega$. F^* must satisfy the following conditions for every $\omega_i \in \Omega$:

(F1) $\mathrm{mindsr}(\omega_i) \leq F^*(\omega_i) \leq \mathrm{maxdsr}(\omega_i)$;
(F2) if $F^*(\omega_i) = 1$, then $a_j(\omega_i) \leq a_j(\omega_h) \;\Rightarrow\; \lambda(\omega_i) \leq \lambda(\omega_h)$, for every $\omega_h \in \Omega$;
(F3) if $[\omega_i]^{\leq}_{\lambda} \cap [\omega_i]^{\leq}_{a_j} \subseteq [\omega_h]^{\leq}_{\lambda} \cap [\omega_h]^{\leq}_{a_j}$ and $[\omega_i]_{a_j} = [\omega_h]_{a_j}$, then $F^*(\omega_i) \leq F^*(\omega_h)$.

Notice that condition **(F1)** implies $F^*(\omega_i) \in (0, 1]$. From a semantic point of view, condition **(F1)** imposes two natural boundaries to $F^*(\omega_i)$ which are determined by objects belonging to $[\omega_i]_{a_j}$; condition **(F2)** requires that $F^*(\omega_i)$ is equal to 1 only in the case of complete satisfaction of the local monotonicity constraint for ω_i; finally, condition **(F3)** is a monotonicity requirement related to other objects in $[\omega_i]_{a_j}$. It is immediate to verify that dsr, mindsr and avgdsr satisfy conditions **(F1)**–**(F3)**, and so F^*_S, F^*_G and F^*_P. On the contrary, maxdsr can fail to satisfy **(F2)**. From previous discussion, we have that for every $\omega_i \in \Omega$

$$\frac{1}{|\Omega|} \leq F^*_P(\omega_i) \leq F^*_G(\omega_i) = F^*_S(\omega_i) \leq 1. \tag{8}$$

Going on, the G^*-layer is a strictly decreasing transformation of the F^*-layer, and it is a real function defined on $(0, 1]$. Putting $f_i = F^*(\omega_i)$, G^* must satisfy the following conditions:

(G1) $G^*(f_i) \in [0, +\infty)$;
(G2) G^* is a strictly decreasing function of f_i;
(G3) $G^*(1) = 0$;

Notice that G^*_G, G^*_S and G^*_P satisfy conditions **(G1)**–**(G3)**, moreover on the interval $(0,1]$, G^*_P dominates G^*_S which, in turn, dominates G^*_G. Considering (8), for every $\omega_i \in \Omega$ we also have

$$G^*_G(F^*_G(\omega_i)) \leq G^*_S(F^*_S(\omega_i)) \leq G^*_P(F^*_P(\omega_i)). \tag{9}$$

Finally, the H^*-layer is an aggregation operator of the G^*-layers corresponding to objects in Ω, and thus it is a real function defined on $[0,\infty)^n$. Putting $g_i = G^*(F^*(\omega_i))$ for $i = 1,\dots,n$, H^* must satisfy the following conditions:

(H1) $H^*(g_1,\dots,g_n) \in [0,+\infty)$;
(H2) $H^*(g_1,\dots,g_n) = H^*(g_{\sigma(1)},\dots,g_{\sigma(n)})$ for every permutation σ;
(H3) if $g_i \leq g'_i$, then $H^*(g_1,\dots,g_i,\dots,g_n) \leq H^*(g_1,\dots,g'_i,\dots,g_n)$;
(H4) $H^*(g_1,\dots,g_n) = 0$ if and only if $g_i = 0$ for $i = 1,\dots,n$.

Again, it is easily seen that the arithmetic mean satisfies conditions **(H1)**–**(H4)**, nevertheless, it is not the only possible choice, indeed, also the maximum operator and the quadratic mean satisfy such conditions.

Next proposition summarizes some properties of H^*_G, H^*_S and H^*_P.

Proposition 2. *The following statements hold:*

(i) $H^*_G(\lambda|a_j) \leq H^*_S(\lambda|a_j) \leq H^*_P(\lambda|a_j)$;
(ii) $0 \leq H^*_G(\lambda|a_j) < \frac{|\Omega|-1}{|\Omega|}$;
(iii) $0 \leq H^*_S(\lambda|a_j) < \log_2(|\Omega|)$;
(iv) $0 \leq H^*_P(\lambda|a_j) < |\Omega|\log_2(|\Omega|)$.

Proof. All the properties follow by inequalities (8) and (9). In particular, for properties *(ii)–(iv)* the upper bound cannot be reached since the F^*-layers of objects in Ω cannot be simultaneously all equal to $\frac{1}{|\Omega|}$.

The layered decomposition we just presented suggests the following definition of a general rank discrimination measure.

Definition 3. *Let F^*, G^* and H^* be functions satisfying conditions* **(F1)**–**(F3)**, **(G1)**–**(G3)** *and* **(H1)**–**(H4)**, *respectively, then we call* **rank discrimination measure**

$$H^*(\lambda|a_j) = H^*(G^*(F^*(\omega_1)),\dots,G^*(F^*(\omega_n))).$$

In next theorem we prove that a rank discrimination measure defined as in Definition 3 reaches its minimum value 0 if and only if λ is monotonic w.r.t. a_j.

Theorem 1. *Let F^*, G^* and H^* be functions satisfying conditions* **(F1)**–**(F3)**, **(G1)**–**(G3)** *and* **(H1)**–**(H4)**, *respectively, then $H^*(\lambda|a_j) = 0$ if and only if λ is monotone with respect to a_j, that is for every $\omega_i, \omega_h \in \Omega$,*

$$a_j(\omega_i) \leq a_j(\omega_h) \;\Rightarrow\; \lambda(\omega_i) \leq \lambda(\omega_h).$$

Proof. Condition **(H4)** implies that the H^*-layer is 0 if and only if the G^*-layer related to each $\omega_i \in \Omega$ is equal 0 and by virtue of conditions **(G2)** and **(G3)** this can happen if and only if the corresponding F^*-layer is equal to 1. Finally, by conditions **(F1)** and **(F2)** the F^*-layer is equal to 1 for every ω_i if and only if the local monotonicity constraint of λ w.r.t. a_j is satisfied for every ω_i.

4 New Rank Discrimination Measures

Definition 3 enables us to introduce new rank discrimination measures, as the two proposed in next definition.

Definition 4

$$H^*_M(\lambda|a_j) = \max_{i=1,\ldots,|\Omega|} \left\{1 - \mathrm{dsr}(\omega_i)^2\right\};$$

$$H^*_Q(\lambda|a_j) = \sqrt{\sum_{i=1}^{|\Omega|} \frac{1}{|\Omega|} \left(1 - \mathrm{avgdsr}(\omega_i)\right)^2}.$$

It is easily verified that functions H^*_M and H^*_Q respect all the conditions in Definition 1. Table 2 lists the hierarchical decomposition of the two new measures.

Table 2. Hierarchical model for measures H^*_M and H^*_Q

Layer	M	Q
F^*	$\mathrm{dsr}(\omega_i)$	$\mathrm{avgdsr}(\omega_i)$
G^*	$1 - F^*(\omega_i)^2$	$1 - F^*(\omega_i)$
H^*	$\max_{i=1,\ldots,\lvert\Omega\rvert} \{G^*(F^*(\omega_i))\}$	$\sqrt{\sum_{i=1}^{\lvert\Omega\rvert} \frac{1}{\lvert\Omega\rvert} G^*(F^*(\omega_i))^2}$

4.1 Induced Order Structures

Keeping in mind the use of a rank discrimination measure, once new measures are introduced, it is extremely important to investigate the order structure they induce on the family of attributes $\mathcal{A}$. In particular, the new measures have an individual meaning only in the case they are not a monotone transformation of other existing measures. In [15] we showed that this does not hold for H^*_S, H^*_G and H^*_P. Example 1 shows that this is not true neither for H^*_M and H^*_Q, indeed all the presented measures induce a different total preorder $\leq_{H^*}$ on $\mathcal{A}$ (where H^* stands for a rank discrimination measure) and so they determine decision trees with different shapes.

Example 1. Consider the set of objects $\Omega = \{\omega_1, \ldots, \omega_5\}$ together with attributes a_1 ranging in $\{0, 1, 2, 3\}$ and a_2 ranging in $\{0, 1\}$, and the labelling function λ ranging in $\{0, 1, 2, 3\}$.

If a_1,a_2 and λ are defined as in Table 3 (a) then we have $H^*_S(\lambda|a_1) = 0.86$, $H^*_S(\lambda|a_2) = 0.96$, $H^*_G(\lambda|a_1) = 0.37$, $H^*_G(\lambda|a_2) = 0.44$, $H^*_P(\lambda|a_1) = 3.86$, $H^*_P(\lambda|a_2) = 4.17$, $H^*_M(\lambda|a_1) = 0.93$, $H^*_M(\lambda|a_2) = 0.88$, $H^*_Q(\lambda|a_1) = 0.71$ and $H^*_Q(\lambda|a_2) = 0.81$. Hence $a_1 <_{H^*} a_2$ for $H^* \in \{H^*_S, H^*_G, H^*_P, H^*_Q\}$ while $a_1 >_{H^*_M} a_2$.

Table 3. Definition of a_1, a_2 and λ

(a)

	a_1	a_2	λ
ω_1	3	1	1
ω_2	0	0	3
ω_3	2	1	0
ω_4	1	1	2
ω_5	1	0	3

(b)

	a_1	a_2	λ
ω_1	1	0	0
ω_2	0	0	2
ω_3	0	1	3
ω_4	3	1	1
ω_5	2	0	0

On the other hand, if a_1,a_2 and λ are defined as in Table 3 (b) then we have $H^*_S(\lambda|a_1) = 0.72$, $H^*_S(\lambda|a_2) = 0.46$, $H^*_G(\lambda|a_1) = 0.28$, $H^*_G(\lambda|a_2) = 0.22$, $H^*_P(\lambda|a_1) = 4.64$, $H^*_P(\lambda|a_2) = 2.78$, $H^*_M(\lambda|a_1) = 0.96$, $H^*_M(\lambda|a_2) = 0.84$, $H^*_Q(\lambda|a_1) = 0.60$ and $H^*_Q(\lambda|a_2) = 0.62$. Hence $a_1 <_{H^*_Q} a_2$, while $a_1 >_{H^*} a_2$ for $H^* \in \{H^*_S, H^*_G, H^*_P, H^*_M\}$.

It is important to notice that since a rank discrimination measure is used at each step of an inductive algorithm, it has only a partial view on the input dataset so the classifier obtained at the end of the construction is not globally monotone, in general. Nevertheless, no form of monotonicity (neither the local one) is guaranteed by standard discrimination measures.

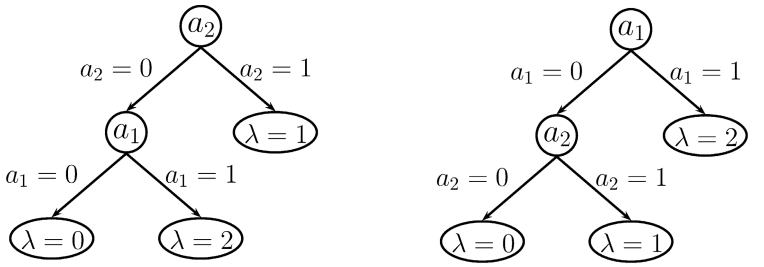
Next example shows the inductive construction of a decision tree starting from a monotone consistent dataset. Both the classical measures H_S and H_G and the new measures H^*_M and H^*_Q are used, stressing that the first two are completely insensitive to monotonicity while the last two (as well as H^*_S, H^*_G and H^*_P) give rise to a globally monotone classifier in this specific case.

Example 2. Consider the set of objects $\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4\}$ described by a_1, a_2 and a_3 ranging in $\{0, 1\}$, and labelled by λ ranging in $\{0, 1, 2\}$, defined as in Table 4.

Table 4. Definition of a_1, a_2, a_3 and λ

	a_1	a_2	a_3	λ
ω_1	0	0	1	0
ω_2	0	1	0	1
ω_3	0	1	1	1
ω_4	1	0	1	2

Considering measures H_G and H_S we compute $H_G(\lambda|a_1) = 0.33$, $H_G(\lambda|a_2) = 0.25$, $H_G(\lambda|a_3) = 0.5$, $H_S(\lambda|a_1) = 0.68$, $H_S(\lambda|a_2) = 0.5$ and $H_S(\lambda|a_3) = 1.18$. Hence, both measures select a_2 for splitting. For the next step, a leaf with label $\lambda = 1$ is added in the right branch, while for the left branch we compute $H_G(\lambda|a_1) = H_S(\lambda|a_1) = 0$, $H_G(\lambda|a_3) = 0.5$ and $H_S(\lambda|a_3) = 1$, so also in this case both measures select a_1 for splitting and the procedure stops with the tree shown in Figure 1 (a). It is easily verified that the resulting classifier $\lambda' : X \to C$ is not globally monotone, since $(1, 0, 1) \leq (1, 1, 1)$ and $2 = \lambda'(1, 0, 1) > \lambda'(1, 1, 1) = 1$.



(a) Non-globally monotone (b) Globally monotone

Fig. 1. Trees obtained using H_G, H_S or H_M^*, H_Q^*, respectively

On the other hand, with H_M^* and H_Q^* we get $H_M^*(\lambda|a_1) = 0.43$, $H_M^*(\lambda|a_2) = 0.93$, $H_M^*(\lambda|a_3) = 0.88$, $H_Q^*(\lambda|a_1) = 0.47$, $H_Q^*(\lambda|a_2) = 0.55$ and $H_Q^*(\lambda|a_3) = 0.72$. This implies the selection of a_1 for splitting, by both measures. Now a leaf with label $\lambda = 2$ is added in the right branch, while for the left branch we have $H_M^*(\lambda|a_2) = H_Q^*(\lambda|a_2) = 0$, $H_M^*(\lambda|a_3) = 0.75$ and $H_M^*(\lambda|a_3) = 0.60$. Also in this case both measures select a_2 for splitting and the procedure stops with the tree shown in Figure 1 (b), which coincides with the one obtained using H_G^*, H_S^* and H_P^*. In this case the resulting classifier is globally monotone.

4.2 Experimental Analysis

To have a first idea of the effectiveness of the two new measures we tested our binary tree classifier RDMT(H^*) [15] with $H^* \in \{H_G^*, H_S^*, H_P^*, H_M^*, H_Q^*\}$, comparing it with other monotone classifiers having a WEKA[1] (version 3-6-0) implementation: we used the Ordinal Learning Model (OLM) [3], the Ordinal Stochastic Dominance Learner (OSDL) [3] and the Ordinal Class Classifier (OCC) [6] (this last classifier is a monotone meta-classifier for which we used C4.5 [17] as basic classifier).

We executed tests on artificial data, producing datasets with an increasing number of monotone attributes. For $k = 1, \ldots, 10$, we generated a dataset of 1000 examples on 10 attributes, where a_j is a uniform random variable on $\{1, \ldots, 10\}$, $j = 1, \ldots, 10$, and the labelling function is defined as $\lambda = \max_{j=1,\ldots,k} a_j$. Clearly, for $k = 10$ the corresponding dataset is monotone consistent due to monotonicity of maximum operator. Each test has been executed performing a stratified 10-folds cross-validation with the same seed for the pseudo-casual number generator and using default WEKA settings for OLM, OSDL and OCC. Figure 2 displays graphics of correctly classified instances, or *CCI* for short.

Figure 2 highlights that RDMT(H^*), for every H^*, performs generally better than OLM, OSDL, OCC: there is only a slightly better behaviour of OCC with respect to RDMT(H_M^*). In particular, the best results are always obtained with

[1] `http://www.cs.waikato.ac.nz/ml/weka/`

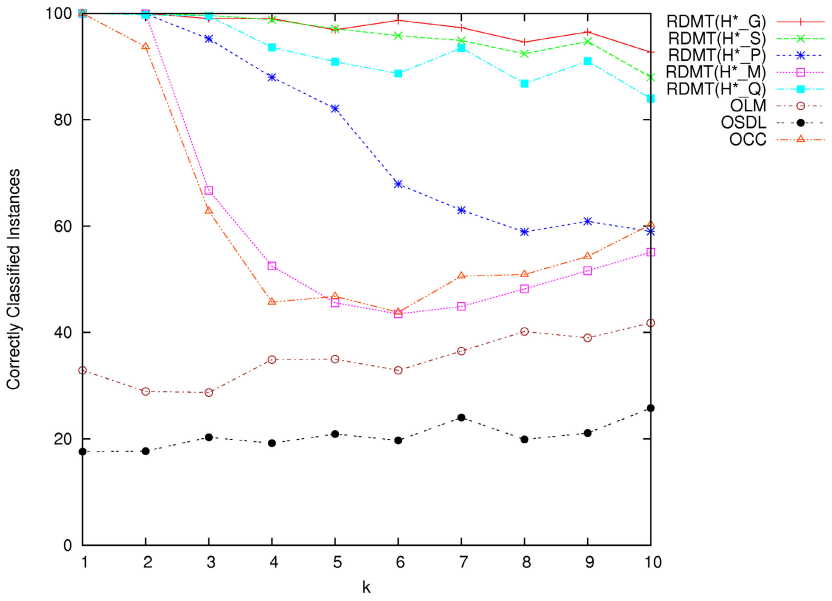


Fig. 2. *CCI* results of RDMT(H^*), OLM, OSDL, OCC

H^*_G, while the worst with H^*_M, moreover, for this experiment there is an evident overtaking of measures H^*_G, H^*_S and H^*_Q with respect to H^*_P and H^*_M.

5 Conclusions

In this paper we presented a hierarchical model for the validation of rank discrimination measures used in the inductive construction of decision tree classifiers, in a way to impose a local form of monotonicity. The properties a function must satisfy to be a rank discrimination measure have been singled out, allowing in this way the creation of two new rank discrimination measures. In future work, we aim at using those new measures to build decision trees in applications where monotonicity of the class related to the attributes is important, such as in medical applications [12]. A deeper experimental study, like the one done in [15], as well as the fuzzification of these measures are also envisaged.

References

1. Ben-David, A., Sterling, L., Pao, Y.H.: Learning and classification of monotonic ordinal concepts. Comp. Int. 5(1), 45–49 (1989)
2. Ben-David, A., Sterling, L., Tran, T.: Adding monotonicity to learning algorithms impair their accuracy. Exp. Sys. with App. 36(3, Part 2), 6627–6634 (2009)
3. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and Regression Trees. Chapman and Hall/CRC, Boca Raton (1984)

4. Cao-Van, K., De Baets, B.: Growing decision trees in an ordinal setting. Int. J. of Int. Sys. 18(7), 733–750 (2003)
5. Feelders, A.: Monotone Relabeling in Ordinal Classification. In: IEEE Int. Conf. on Data Mining 2010 (ICDM 2010), pp. 803–808 (2010)
6. Frank, E., Hall, M.: A simple approach to ordinal classification. In: De Raedt, L., Flach, P.A. (eds.) ECML 2001. LNCS (LNAI), vol. 2167, pp. 145–156. Springer, Heidelberg (2001)
7. Greco, S., Matarazzo, B., Slowinski, R.: Rough approximation by dominance relations. Int. J. of Int. Sys. 17(2), 153–171 (2002)
8. Greco, S., Matarazzo, B., Slowinski, R.: Rough sets methodology for sorting problems in presence of multiple attributes and criteria. Europ. J. of Op. Res. 138(2), 247–259 (2002)
9. Hu, Q., Che, X., Zhang, L., Zhang, D., Guo, M., Yu, D.: Rank entropy based decision trees for monotonic classification. IEEE Trans. on Knowledge and Data Engineering 24(11), 2052–2064 (2011)
10. Hu, Q., Guo, M., Yu, D., Liu, J.: Information entropy for ordinal classification. Science China Inf. Sci. 53, 1188–1200 (2010)
11. Hu, Q., Pan, W., Zhang, L., Zhang, D., Song, Y., Guo, M., Yu, D.: Feature selection for monotonic classification. IEEE Trans. on Fuzzy Systems 20(1), 69–81 (2012)
12. Marsala, C.: Gradual fuzzy decision trees to help medical diagnosis. In: IEEE Int. Conf. on Fuzzy Systems 2012 (FUZZ-IEEE 2012), pp. 1–6 (2012)
13. Marsala, C., Bouchon-Meunier, B.: Ranking attributes to build fuzzy decision trees: a comparative study of measures. In: IEEE Int. Conf. on Fuzzy Systems 2006 (FUZZ-IEEE 2006), pp. 1777–1783 (2006)
14. Marsala, C., Bouchon-Meunier, B.: Quality of measures for attribute selection in fuzzy decision trees. In: IEEE Int. Conf. on Fuzzy Systems 2010 (FUZZ-IEEE 2010), pp. 1–8 (2010)
15. Marsala, C., Petturiti, D.: Rank Discrimination Measures for Monotone Decision Tree Induction. Information Sciences (submitted)
16. Potharst, R., Feelders, A.J.: Classification trees for problems with monotonicity constraints. SIGKDD Exp. Newsletter 4(1), 1–10 (2002)
17. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
18. Yuan, Y., Shaw, M.J.: Induction of fuzzy decision trees. Fuz. Sets and Sys. 69(2), 125–139 (1995)

Extreme Points of the Credal Sets Generated by Elementary Comparative Probabilities

Enrique Miranda[1] and Sébastien Destercke[2]

[1] University of Oviedo, Dept. of Statistics and O.R., Oviedo, Spain
mirandaenrique@uniovi.es
[2] CNRS, HEUDIASYC Joint Research Unit, Compiègne, France
sebastien.destercke@hds.utc.fr

Abstract. When using convex probability sets (or, equivalently, lower previsions) as models of uncertainty, identifying extreme points can be useful to perform various computations or to use some algorithms. In general, sets induced by specific models such as possibility distributions, linear vacuous mixtures or 2-monotone measures may have extreme points easier to compute than generic convex sets. In this paper, we study extreme points of another specific model: comparative probability orderings between the elements of a finite space. We use these extreme points to study the properties of the lower probability induced by this set, and connect comparative probabilities with other uncertainty models.

Keywords: Comparative probabilities, credal sets, 2-monotone capacities, belief functions, regular extension, imprecise mass functions.

1 Introduction

In the last decades, there has been a growing interest on imprecise probability models as alternative models to probability in situations where the available information is vague or scarce. This type of models includes for instance belief functions [1], possibility measures [2], 2- and n-monotone capacities [3] or probability boxes [4]. All the above examples can be seen as instances of coherent lower and upper previsions [5].

The adequacy of each of these models for a particular problem depends, among other things, on the interpretation we are giving to our uncertainty. In this paper, we consider a *robust Bayesian* interpretation [6]: we assume the existence of a precise, but unknown, probability model, and work with the set of probability measures that are compatible with the available information. This gives rise to a *credal set*, as considered by Levi in [7].

Here, we consider the case where the information is expressed by means of a *comparative probability model* [8]: we consider a finite probability space Ω and assume that we are given judgements of the type "the probability of A is at least as great as that of B". Comparative probabilities have been deemed of particular interest within the context of subjective probability theory [9,10,11]; see also [5, Section 4.5] for a study from the point of view of coherent lower previsions. One of their advantages is that they seem well suited for modelling qualitative judgements.

In spite of this, there are only few works dealing with the numerical and practical aspects of comparative probabilities [12]. One reason for this is that it is not easy to

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 424–435, 2013.

summarize the set of probabilities associated to the comparative assessments, for instance by means of a lower and an upper probability, and this renders it difficult to summarize the information about the probability of an event of interest. In this paper, we solve this problem by characterizing the comparative probability models by means of the extreme points of their associated credal sets. This is a problem that has been studied for other types of imprecise probability models, such as 2-monotone capacities [13], possibility measures [14], probability intervals [15] and belief functions [16]. In this paper, we focus on probability sets generated by comparisons between singletons. Focusing on this particular case allows us to derive nice graphical characterizations, and we provide some practical examples where this special case may be useful. There is only one partial result for this type of assessments [17], and we generalize it in this paper.

After giving some preliminary results in Section 2, we shall see in Section 3 that, when the comparison judgements are made on the probabilities of the singletons, a graphical representation of these judgements makes it easy to derive the extreme points of the associated credal sets. In Section 4, we use this result to discuss some practical aspects of these models: we establish tight lower and upper bounds on the number of extreme points; investigate their relationship with other imprecise probability models; provide algorithms for the computation of these extreme points; and discuss the computation of conditional lower probabilities and the merging of multiple comparison judgements. Some additional remarks related to the practical use of these models and their extensions are provided in Section 5.

2 Preliminaries

Consider a finite space $\mathscr{X} = \{x_1, \ldots, x_n\}$, modelling the set of outcomes of some experiment. In this paper, we assume that our information about these outcomes can be modelled by means of *comparative probability orderings of the states*, i.e., statements of the type "the probability of x_i is at least as great as that of x_j". Hence, we shall represent the available information by means of a subset $\mathscr{L}$ of $\{1, \ldots, n\} \times \{1, \ldots, n\}$.

The set of probability measures compatible with this information is given by

$$\mathscr{P}(\mathscr{L}) = \{p \in \mathbb{P}_{\mathscr{X}} : \forall (i,j) \in \mathscr{L}, p(x_i) \geq p(x_j)\}, \tag{1}$$

where $\mathbb{P}_{\mathscr{X}}$ denotes the set of all probabilities on the power set of $\mathscr{X}$.

For the purposes of this paper, it shall be useful to represent these assessments by means of a graph $\mathscr{G} = (\mathscr{X}, \mathscr{L})$ where the nodes are the elements of $\mathscr{X}$ and we draw an edge between x_i and x_j when $(i,j) \in \mathscr{L}$.

Example 1. Consider the space $\mathscr{X} = \{x_1, \ldots, x_5\}$ and the set of assessments $\mathscr{L} = \{(1,3), (1,4), (2,5), (4,5)\}$. Its associated graph $\mathscr{G}$ is given by Figure 1. ♦

Note that the set $\mathscr{P}(\mathscr{L})$ determined by Eq. (1) is always non-empty, because it includes for instance the uniform probability distribution. It is interesting to compare it with the set

$$\mathscr{P}(\mathscr{K}) = \{p \in \mathbb{P}_{\mathscr{X}} : \forall (i,j) \in \mathscr{L}, p(x_i) > p(x_j)\},$$

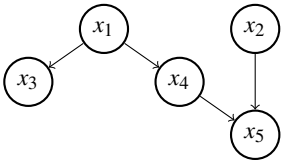


Fig. 1. Graph $\mathscr{G}$ of Example 1

i.e., with the credal set associated by *strict* elementary probability comparisons, which appear also sometimes in the literature. Since $\mathscr{P}(\mathscr{L})$ is a closed convex polytope in $\mathbb{R}^n$, it follows from basic convex analysis that $\mathscr{P}(\mathscr{L})$ corresponds to the closure of $\mathscr{P}(\mathscr{K})$ when the latter set is non-empty, and that $\mathscr{P}(\mathscr{K})$ is the topological interior of $\mathscr{P}(\mathscr{L})$. The non-emptiness of $\mathscr{P}(\mathscr{K})$ is easy to characterise.

Proposition 1. *$\mathscr{P}(\mathscr{K}) \neq \emptyset$ if and only if $\mathscr{G}$ is acyclic.*

Hence, our results in this paper will also allow us to characterize the set $\mathscr{P}(\mathscr{K})$. As we shall see in Remark 1, we can also deal with assessments of equality between the probabilities, which correspond to a cycle in $\mathscr{G}$.

3 Extreme Points of $\mathscr{P}(\mathscr{L})$

Consider a finite space $\mathscr{X} = \{x_1, \ldots, x_n\}$ and a subset $\mathscr{L}$ of $\{1, \ldots, n\} \times \{1, \ldots, n\}$, and let $\mathscr{P}(\mathscr{L})$ be the set it determines by means of Eq. (1). Any of the probability measures in $\mathscr{P}(\mathscr{L})$ is completely determined by its mass function, and as a consequence it can be seen as an element of the n-th dimensional Euclidean space. Then, $\mathscr{P}(\mathscr{L})$ is a closed convex subset of $\mathbb{R}^n$ in the Euclidean topology, that corresponds thus to the closed convex hull of its set of extreme points. We shall determine these extreme points by means of the graphical representation we have established in Section 2.

We shall make two assumptions on the graph $\mathscr{G}$ associated to $\mathscr{L}$:

(G1) The first one is that $\mathscr{G}$ is acyclic, meaning that there are no assumptions of equality between the probability of two different states.
(G2) The second is that $\mathscr{G}$ is connected, so for every $i \neq j$ there is an undirected path in $\mathscr{G}$ that connects the nodes x_i and x_j.

Remark 1. The results we obtain can be used to characterise the general case. On the one hand, when $\mathscr{G}$ has cycles, we have some assumptions of equality $P(x_i) = P(x_j)$ between the probabilities of two elements x_i, x_j in our possibility space. It is not difficult to determine the structure of the set $\mathscr{P}(\mathscr{L})$ in that case: for each of the assumptions of equality, we consider one of the elements x_i in the corresponding set and store the number of elements n_i in $\mathscr{X}$ that are assumed to have the same probability as x_i; from this we derive the simplified space $\mathscr{X}' \subset \mathscr{X}$ for which the graph $\mathscr{G}'$ satisfies (G1).

By this, we can establish a one-to-one correspondence between the sets $\mathscr{P}(\mathscr{L}) \subseteq \mathbb{P}_{\mathscr{X}}$ and $\mathscr{P}(\mathscr{L}') \subseteq \mathbb{P}_{\mathscr{X}'}$: any probability $P := (p_1, \ldots, p_n)$ in $\mathscr{P}(\mathscr{L})$ induces the probability P' on $\mathscr{P}(\mathscr{L}')$, with $P'(x_i) = P(x_i) \cdot n_i$. Then, once we determine the distributions

of the extreme points associated to the graph $\mathscr{G}'$, we just have to 'expand' this graph by reversing the above correspondence between the probabilities.

On the other hand, if $\mathscr{G}$ does not satisfy (G2), we can decompose it as a union of its weakly connected components $\mathscr{G}_1,\ldots,\mathscr{G}_k$. For each of these components we can characterise their associated extreme points in the form we shall give below, and then the extreme points associated to $\mathscr{G}$ will be the union of the sets of extreme points in each of these subgraphs. ♦

In order to characterise the extreme points of $\mathscr{P}(\mathscr{L})$, we are going to consider a number of lemmas:

Lemma 1. *Any extreme point p of $\mathscr{P}(\mathscr{L})$ corresponds to a uniform probability measure over some subset $A \subseteq \mathscr{X}$.*

For every subset A of $\mathscr{X}$, we shall denote by P_A the uniform probability measure on A, that is associated to the mass function

$$P_A(x_i) = \begin{cases} \frac{1}{|A|} & \text{if } x_i \in A \\ 0 & \text{otherwise} \end{cases}$$

for any $i \in \{1,\ldots,n\}$. Using the acyclic graph $\mathscr{G}$, we can now characterize those subsets $A \subseteq \mathscr{X}$ for which P_A is an extreme point of $\mathscr{P}(\mathscr{L})$. For every $x_j \in \mathscr{X}$, we shall denote by $H(x_j)$ the set of ancestors of x_j, i.e., those nodes x_i such that there is a directed path from x_i to x_j in $\mathscr{G}$. By an abuse of notation, we shall also consider that x_j is an ancestor of itself, i.e., we shall assume that $x_j \in H(x_j)$ for all j. Finally, for every $A \subseteq \mathscr{X}$, we shall denote $H(A) := \cup_{x\in A} H(x)$.

The following lemma gives further insight onto which uniform probabilities may be extreme points of the credal set $\mathscr{P}(\mathscr{L})$.

Lemma 2. *1. If $A \neq H(A)$, then P_A is not an extreme point of $\mathscr{P}(\mathscr{L})$.*
2. If there are $C_1, C_2 \subseteq A$ such that $H(C_1) \cap H(C_2) = \emptyset$ and $H(C_1) \cup H(C_2) = H(A)$, then $P_{H(A)}$ is not an extreme point on $\mathscr{P}(\mathscr{L})$.

Next, if B is a subset of A and $H(B) = H(A)$, both A and B give rise to the same probability measure $P_{H(B)} = P_{H(A)}$. This is related to the notion of strongly connected nodes:

Definition 1. *Two nodes x_i, x_j in the graph $\mathscr{G}$ are said to be* strongly connected *when there is a directed path from x_i to x_j, or viceversa, and are called* strongly disconnected *otherwise.*

Equivalently, x_i, x_j are strongly connected when either $x_i \in H(x_j)$ or $x_j \in H(x_i)$. This allows us to establish the following result:

Theorem 1. *If $P_{H(A)}$ is an extreme point of $\mathscr{P}(\mathscr{L})$, then there is some $B \subseteq A$ with $H(B) = H(A)$ and such that any two nodes in B are strongly disconnected. Thus, the set of extreme points coincide with the set of probabilities $P_{H(A)}$ generated by sets A*

(EXT1) composed of strongly disconnected nodes of $\mathscr{G}$ and
(EXT2) that cannot be decomposed as in Lemma 2.

Remark 2. An interesting related result has been established in [17], in the context of credal classification. The author considers the credal set determined by the comparisons of the probabilities of the states, and computes the lower probability of the set A of elements with no predecessor in $\mathscr{G}$. In order to do this, she provides results analogous to our Lemmas 1 and 2, and then in [17, Theorem B.2.2] she establishes which of the elements in $\mathscr{P}(\mathscr{L})$ attain the lower probability of A.

Theorem 1 subsumes these results, in the sense that we give the explicit form of the extreme points (from which we may determine also the lower probability of any other set, as well as the lower prevision induced by a comparative probability model). Note moreover that we have showed that not all the uniform probability distributions $P_{H(A)}$ determine an extreme point of $\mathscr{P}(\mathscr{L})$. ♦

Example 2. The extreme points generated by Example 1 are summarised in Table 1.

Table 1. Extreme points of Example 1

		p				
A	$H(A)$	x_1	x_2	x_3	x_4	x_5
$\{x_1\}$	$\{x_1\}$	1	0	0	0	0
$\{x_2\}$	$\{x_2\}$	0	1	0	0	0
$\{x_3\}$	$\{x_1,x_3\}$	1/2	0	1/2	0	0
$\{x_4\}$	$\{x_1,x_4\}$	1/2	0	0	1/2	0
$\{x_5\}$	$\{x_1,x_2,x_4,x_5\}$	1/4	1/4	0	1/4	1/4
$\{x_1,x_2\}$	$\{x_1,x_2\}$	1/2	1/2	0	0	0
$\{x_2,x_3\}$	$\{x_1,x_2,x_3\}$	1/3	1/3	1/3	0	0
$\{x_2,x_4\}$	$\{x_1,x_2,x_4\}$	1/3	1/3	0	1/3	0
$\{x_3,x_4\}$	$\{x_1,x_3,x_4\}$	1/3	0	1/3	1/3	0
$\{x_3,x_5\}$	$\{x_1,x_2,x_3,x_4,x_5\}$	1/5	1/5	1/5	1/5	1/5
$\{x_2,x_3,x_4\}$	$\{x_1,x_2,x_3,x_4\}$	1/4	1/4	1/4	1/4	0

4 Practical Aspects

4.1 Number of Extreme Points

Since extreme points correspond to uniform distributions over certain subsets $A \subseteq \mathscr{X}$, we immediately see that an upper bound of the number of extreme points is $2^{|\mathscr{X}|}$. Note that this is significantly lower than the maximal number of extreme points generated by lower coherent probabilities, known to be $|\mathscr{X}|!$ [18]. We next show that this number of extreme points can be reduced even further (recall that we are assuming throughout that the graph $\mathscr{G}$ associated with $\mathscr{L}$ satisfies (G1) and (G2)):

Theorem 2. *The maximum number of extreme points of $\mathscr{P}(\mathscr{L})$ is $2^{(|\mathscr{X}|-1)}$, and the minimum number is $|\mathscr{X}|$. Each of these bounds can be attained.*

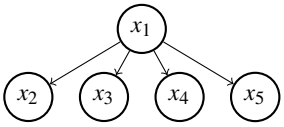


Fig. 2. Graph $\mathscr{G}$ for x_1 =modal value

To see that the upper bound given by the above theorem can indeed be reached, consider the case where a single modal value is provided. Figure 2 illustrates the situation.

Interestingly, the upper bound given in Theorem 2 is the same as the number of extreme points of the credal set associated to a possibility measure, as showed in [14, Section 5]. Our intuition for this is that possibility measures also determine an order between the singletons, by means of their associated possibility distributions. On the other hand, an example where the lower bound is reached is the case where $\mathscr{L}$ forms a complete ordering of singletons $\{x_1,\ldots,x_n\}$ (this is the case considered in [5, P. 195]; note that the result there is now a particular case of Theorem 1).

4.2 Extraction Algorithm

Using the results of Section 3, we can propose a pseudo-algorithm to extract extreme points, summarised in Algorithm 1.

Algorithm 1. Extreme point search

Input: Set $\mathscr{L}$ of comparisons
Output: Extreme points of $\mathscr{P}(\mathscr{L})$

List $\leftarrow \emptyset$;
for $i = 1,\ldots,n$ **do**
  Build extreme points corresponding to $H(x_i)$;
  List $\leftarrow \{x_i\}$;
Candidate set $\leftarrow$ List ;
for $i = 2,\ldots,n$ **do**
  List $\leftarrow \emptyset$;
  foreach *set B in Candidate set* **do**
    for $i = 1,\ldots,n$ **do**
      if x_i *is strongly disconnected from the elements B and* $H(B\cup\{x_i\})$ *is a new extreme point* **then**
        Add $H(B\cup\{x_i\})$ to extreme points ;
        List $\leftarrow B\cup\{x_i\}$;
      Candidate set $\leftarrow$ List ;

Implementing this algorithm mainly requires to be able, for a given set B, to check whether elements of B are strongly disconnected and to compute $H(B)$. An instrumental tool to do this is the matrix M corresponding to the transitive closure $\mathscr{C}(\mathscr{L}) \subseteq \mathscr{X}\times\mathscr{X}$

of $\mathscr{L}$, with $M(i,j) = 1$ iff $(i,j) \in \mathscr{C}(\mathscr{L})$. M can be efficiently computed by applying Warshall algorithm (see [19]) to matrix L with $L(i,j) = 1$ iff $(i,j) \in \mathscr{L}$.

Once this is done, checking whether two elements x_i, x_j are strongly disconnected can be done in linear time. Checking that B is made of strongly disconnected elements is equivalent to check whether all pairs of elements $x_i, x_j \in B$ are strongly disconnected, hence at most in quadratic time. As $H(B) = \cup_{x \in B} H(x)$, computing $H(B)$ is also linear. This means that the complexity of the loop going from Line 10 to 13 in Algorithm 1 is quadratic.

Algorithm 1 also tries to minimize the number of sets of nodes to check by reducing the search to sets that are not known to be sets containing connected nodes, rather than making a naïve search among all subsets $B \subseteq \mathscr{X}$. Summarizing, the whole algorithm complexity depends on the number of extreme points to extract, hence is at worst NP-hard (see Theorem 2), at best quadratic (as loop 10-13 is).

4.3 n-Monotonicity

Next, we investigate in more detail the set of probabilities $\mathscr{P}(\mathscr{L})$ from the point of view of the theory of coherent lower previsions developed in [5]. Since the set $\mathscr{P}(\mathscr{L})$ is a closed convex set of probabilities, its lower envelope $\underline{P}$, given by

$$\underline{P}(A) = \min\{P(A) : P \in \mathscr{P}(\mathscr{L})\} \; \forall A \subseteq \mathscr{X} \tag{2}$$

is a *coherent lower probability*. As such, it can be given a behavioural interpretation in terms of acceptable betting rates.

Coherent lower probabilities include as particular cases most of the imprecise probability models that we can find in the literature, such as 2-monotone capacities, belief functions, or necessity measures; see [20] for more details. In particular, a coherent lower probability is 2-monotone when for any $A, B \subseteq \mathscr{X}$ we have

$$\underline{P}(A \cup B) + \underline{P}(A \cap B) \geq \underline{P}(A) + \underline{P}(B). \tag{3}$$

These are also called *convex* functions on Choquet capacities of order 2 [3,21]. When $|\mathscr{X}| \leq 3$, a coherent lower probability on $\mathscr{P}(\mathscr{X})$ is always 2-monotone [22], and as consequence this is also true for the comparative probability models we consider in this paper. On the other hand, when $|\mathscr{X}| \geq 4$, there exist coherent lower probabilities on $\mathscr{P}(\mathscr{X})$ which are not 2-monotone. We next show that, in general, the coherent lower probabilities induced by comparative probability models will not be 2-monotone.

Example 3. Consider $\mathscr{X} = \{x_1, x_2, x_3, x_4\}$ and $\mathscr{L} = \{(1,2), (1,3), (2,4), (3,4)\}$. From Theorem 1, the extreme points of $\mathscr{P}(\mathscr{L})$ are associated to the mass functions

$$\left\{ (\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}), (\frac{1}{2}, \frac{1}{2}, 0, 0), (\frac{1}{2}, 0, \frac{1}{2}, 0), (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0), (1,0,0,0) \right\};$$

as a consequence, if we consider the events $A = \{x_1, x_3\}$ and $B = \{x_1, x_4\}$, we see that

$$\underline{P}(A \cup B) + \underline{P}(A \cap B) = 1/2 + 1/4 < \underline{P}(A) + \underline{P}(B) = 1/2 + 1/3.$$

Hence, $\underline{P}$ violates the 2-monotonicity condition. ♦

From this, we can deduce that belief functions, that are in particular 2-monotone, are not expressive enough to represent comparative probability models.

On the other hand, from a convex set of probability measures we can also determine lower and upper expectation functionals. Similarly to Eq. (2), the real-valued functional $\underline{P}$ given by

$$\underline{P}(f) = \min\{P(f) : P \in \mathscr{P}(\mathscr{L})\} \tag{4}$$

for any function $f : \mathscr{X} \to \mathbb{R}$ is called a *coherent lower prevision*. Here, we are also using P to denote the expectation functional associated to the probability measure P, given by $P(f) = \sum_{x\in\mathscr{X}} f(x)p(x)$.

Similarly to Eq. (3), a coherent lower prevision is called 2-monotone when

$$\underline{P}(f \vee g) + \underline{P}(f \wedge g) \leq \underline{P}(f) + \underline{P}(g)$$

for any $f, g : \mathscr{X} \to \mathbb{R}$, where $\vee$ denotes the point-wise maximum and $\wedge$ denotes the point-wise minimum. This type of lower previsions has been studied in detail in [22,23]. They are interesting, because, unlike coherent lower previsions, they can be computed as the Choquet integral with respect to the lower probability that is their restriction to events. Moreover, 2-monotonicity has been showed to be equivalent to comonotone additivity [23, Theorem 15]. However, we can prove that the coherent lower prevision associated to a one-to-one comparison model is not 2-monotone as soon as $\mathscr{X}$ has more than two elements:

Theorem 3. *Consider a space $\mathscr{X}$ with $|\mathscr{X}| \geq 3$, and let $\mathscr{L}$ be a number of probability comparisons on the elements of $\mathscr{X}$ whose associated graph satisfies (G1) and (G2). Let $\underline{P}$ be the coherent lower prevision determined by* (4)*. Then $\underline{P}$ is not* 2*-monotone.*

Although it is an open problem at this stage, we think that with similar arguments to those in the proof it can be showed that a comparative probability model on the singletons never determines a 2-monotone lower prevision even when the associated graph $\mathscr{G}$ violates (G1) or (G2).

4.4 Conditioning

A classical operation when dealing with uncertainty is that of conditioning. Here we will study the problem of computing lower conditional probabilities $\underline{P}(A|B)$ from the credal set $\mathscr{P}(\mathscr{L})$. Out of the many possible notions we can consider in this case, we think that the most intuitive under the robust Bayesian interpretation we are considering in this paper is that of *regular extension* [5, Appendix J], that produces

$$\underline{P}(A|B) = \inf_{P\in\mathscr{P}(\mathscr{L})} \{P(A|B) : P(B) > 0\}, \tag{5}$$

where $P(A|B)$ is obtained from P through Bayes Rule of conditioning.

Note that in order to apply this rule, we need that there is some probability measure P in $\mathscr{P}(\mathscr{L})$ such that $P(B) > 0$ (or, in other words, that the *upper* probability $\overline{P}(B)$ is positive); but this is no restriction in the case of comparative probabilities, because there will always be an extreme point P of $\mathscr{P}(\mathscr{L})$ for which $P(B) > 0$: it suffices to

consider $P_{H(x_i)}$ with $x_i \in B$. On the contrary, the lower probability $\underline{P}(B)$ will be positive if and only if for any $x_i \in \mathscr{X}$ it holds that $B \cap H(x_i) \neq \emptyset$, i.e., if and only if B contains all the nodes without a predecessor.

To attain the conditional lower probability $\underline{P}(A|B)$ given by Eq. (5), we need to find the extreme point for which $P(B)$ is positive and the fraction $P(A\cap B)/P(B)$ minimal. This can be done easily by the procedure described in Algorithm 2. Note that it is sufficient to concentrate on extreme points generated by subsets C of $B \setminus A$, as we want to minimise the ratio $|H(C)\cap B\cap A|/|H(C)\cap B|$. From this, we easily derive the following algorithm:

Algorithm 2. Conditional Probability computation

Input: Set $\mathscr{L}$ of comparisons
Output: Lower conditional probability $P(A|B)$ with $A \subset B$

```
Cond ← 1 ;
foreach Set C ⊆ B \ A do
    Value ← |H(C)∩(A∩B)|/|H(C)∩B| ;
    if Value < Cond then Cond← Value
Return Cond ;
```

4.5 Multiple Source Merging

When multiple sources provide different comparisons, for instance when two different experts provide assessments $\mathscr{L}_1$ and $\mathscr{L}_2$, it becomes necessary to merge them in a single representation. The two most common rules to do so are the conjunction and disjunction, that respectively come down to computing $\mathscr{P}(\mathscr{L}_1) \cap \mathscr{P}(\mathscr{L}_2)$ and $CH(\mathscr{P}(\mathscr{L}_1) \cup \mathscr{P}(\mathscr{L}_2))$, where CH denotes the convex hull (the disjunction usually producing non-convex probability sets). Our next result shows that simple operations on $\mathscr{L}_1$ and $\mathscr{L}_2$ can provide exact or approximated results of these operations.

Proposition 2. *1. The disjunctively merged set $CH(\mathscr{P}(\mathscr{L}_1) \cup \mathscr{P}(\mathscr{L}_2))$ is such that $CH(\mathscr{P}(\mathscr{L}_1) \cup \mathscr{P}(\mathscr{L}_2)) \subseteq \mathscr{P}(\mathscr{L}_1 \cap \mathscr{L}_2)$, and the inclusion can be strict.*
2. The conjunctively merged set satisfies $\mathscr{P}(\mathscr{L}_1) \cap \mathscr{P}(\mathscr{L}_2) = \mathscr{P}(\mathscr{L}_1 \cup \mathscr{L}_2)$.

5 Practical Examples and Extensions

In this section, we propose some particular examples of situations where elementary comparative probability models can be used, and discuss some possible extensions.

5.1 Imprecise Mass Functions

Elementary comparative probability models can be related to the work on imprecise mass functions discussed by Augustin [24] and Denoeux [25]. Recall that a belief function $\underline{P}$ on the power set of $\mathscr{X}$ is uniquely determined by its associated *basic probability assignment* m, by means of the formula [1]

$$\underline{P}(A) = \sum_{E \subseteq A} m(E). \tag{6}$$

The basic probability assignment $m(E)$ of a set E represents the weight of the available evidence supporting that the outcome of the experiment belongs to E. It holds that $\sum_{E\subseteq\mathscr{X}} m(E) = 1$, so we may regard m as the probability mass function of some probability measure on $\mathscr{P}(\mathscr{P}(\mathscr{X}))$. This arises for instance in the context of finite random sets.

We can then use our results to build imprecise mass functions. If we have assessments of the type $m(A_i) \geq m(A_j)$, we may consider the set of the mass functions compatible with these assessments. This is a convex set of probability measures whose extreme points can be determined by means of Theorem 1. Note that, by means of Eq. (6), each of these mass functions determines a belief function, that in turn is equivalent to a convex set of probability measures on $\mathscr{P}(\mathscr{X})$. Hence, a convex set of mass functions also induces a convex set of probabilities on $\mathscr{P}(\mathscr{X})$ [24]; however, its lower probability will not be, in general, a belief function (nor, as we can deduce from Example 3, 2-monotone).

This can be useful for instance in the context of inner/outer measures [26]. We may think of an infinite space $\mathscr{X}$ that is partitioned into n sets $A_1,\ldots,A_n$, and where a probability measure $P(A_i)$ is associated to each set A_i. Such an assessment induces on the power set of $\mathscr{X}$ a set of probabilities that can be described by $m(A_i) = P(A_i)$. In this situation, comparative statements between the probabilities $P(A_i)$ are equivalent to comparative statements between the masses $m(A_i)$, and the set of extreme masses can then be derived using our results.

5.2 Extension to General Comparative Probability Models: Some Comments

The most important extension of our work would be to consider arbitrary comparative probability models, where we allow for comparisons between any pair of events (the case of partitions is treated in Section 5.1), that is to allow any comparison $P(A) \geq P(B)$ with $A,B \subseteq \mathscr{X}$. These are the models studied extensively in [8,10,11], amongst others.

Note that, when considering comparative probability models, we can assume that the sets A,B we compare are disjoint, since the assessments $P(A) \geq P(B)$ and $P(A \setminus B) \geq P(B \setminus A)$ are equivalent. However, the existence of a probability compatible with the assessments is no longer trivial, and therefore the associated set $\mathscr{P}(\mathscr{L})$ may be empty: think for instance of the case of $\mathscr{X} = \{x_1,x_2,x_3\}$ and the assessments $P(\{x_1\}) \geq P(\{x_2,x_3\}), P(\{x_2\}) \geq P(\{x_1,x_3\})$ and $P(\{x_3\}) \geq P(\{x_1,x_2\})$. These are equivalent to $P(\{x_1\}) \geq 0.5, P(\{x_2\}) \geq 0.5$ and $P(\{x_3\}) \geq 0.5$, and there is no probability measure satisfying all these conditions simultaneously.

When $\mathscr{P}(\mathscr{L})$ is non-empty, then it is a closed convex set which is characterized by its finite number of extreme points. However, as the next example shows, we cannot expect the extreme points of such assessments to be as simple as the extreme points generated by the comparison of the probabilities of the states. In particular, the extreme points of the associated credal sets will not be necessarily associated with uniform probability distributions over some subsets, and finding an easy graphical representation from which they could be extracted seems hard.

Example 4. Consider $\mathscr{X} = \{x_1, x_2, x_3\}$ and the assessments $P(\{x_2\}) \geq P(\{x_1\})$ and $P(\{x_1, x_2\}) \geq P(\{x_3\})$, and let $\mathscr{P}$ be the credal set determined by these assessments. The extreme points of $\mathscr{P}$ are given by the mass functions

$$\{(0,1,0), (1/2,1/2,0), (1/4,1/4,1/2), (0,1/2,1/2)\}. \qquad \blacklozenge$$

6 Conclusions

Comparative probability models constitute a useful approach to modelling uncertain information about a probability model, especially when the available information is of a qualitative nature. However, most of the works in the literature about these models have focused on axiomatizing those comparative probability models that can be associated to a set of probability measures. In this paper, we have deepened on the link between elementary comparative probability models and imprecise probabilities, by: (a) characterizing the structure of the set of probability measures associated to a comparative probability model, and (b) studying the properties of the lower probability induced by this set. Interestingly, we have showed that this lower probability may not be 2-monotone, from which it follows that 2-monotone capacities (and in particular belief functions, or possibility measures) are not expressive enough to be able to deal with this type of qualitative information. Moreover, we have showed that the maximum number of extreme points is similar to the maximal number of extreme points of credal sets induced by possibility measures, and smaller than those induced by 2-monotone capacities or belief functions.

We have also suggested some practical situations where this model can be useful, such as the elicitation of modal or least probable values or imprecise mass functions. However, this model remains quite simple and of limited expressiveness; it would be desirable to determine to which extent the results presented in this paper can be extended to the case of general comparisons between disjoint events, discussed in Section 5.2. Another important open problem would be to provide algorithms for the computation of the lower prevision induced by a comparative probability model, and to study in detail the applications of these results in fields such as qualitative decision making.

Acknowledgements. The research in this paper has been supported by project MTM2010-17844 and by an invited professorship provided by Universidad de Oviedo. We would like to thank Marco Zaffalon for making us aware of the work carried out in [17], as well as for other useful suggestions. We also thank Erik Quaeghebeur for stimulating discussion.

References

1. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, Princeton (1976)
2. Dubois, D., Prade, H.: Possibility Theory. Plenum Press, New York (1988)
3. Choquet, G.: Theory of capacities. Annales de l'Institut Fourier 5, 131–295 (1953–1954)
4. Ferson, S., Kreinovich, V., Ginzburg, L., Myers, D.S., Sentz, K.: Constructing probability boxes and Dempster-Shafer structures. Technical Report SAND2002–4015, Sandia National Laboratories (January 2003)

5. Walley, P.: Statistical Reasoning with Imprecise Probabilities. Chapman and Hall, London (1991)
6. Berger, J.O.: An overview of robust Bayesian analysis. Test 3, 5–124 (1994); With discussion
7. Levi, I.: The enterprise of knowledge. MIT Press, Cambridge (1980)
8. Koopman, B.: The axioms and algebra of intuitive probability. Annals of Mathematics 41, 269–292 (1940)
9. Fine, T.: Theories of Probability. Academic Press, New York (1973)
10. Suppes, P.: The measurement of belief. Journal of the Royal Statistical Society. Series B (Methodological) 36, 160–191 (1974)
11. Walley, P., Fine, T.L.: Varieties of modal (classificatory) and comparative probability. Synthese 41, 321–374 (1979)
12. Regoli, G.: Comparative probability and robustness. Lecture Notes-Monograph Series, pp. 343–352 (1996)
13. Chateauneuf, A., Jaffray, J.Y.: Some characterizations of lower probabilities and other monotone capacities through the use of Möbius inversion. Mathematical Social Sciences 17(3), 263–283 (1989)
14. Miranda, E., Couso, I., Gil, P.: Extreme points of credal sets generated by 2-alternating capacities. International Journal of Approximate Reasoning 33(1), 95–115 (2003)
15. de Campos, L.M., Huete, J.F., Moral, S.: Probability intervals: a tool for uncertain reasoning. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 2, 167–196 (1994)
16. Dempster, A.P.: Upper and lower probabilities induced by a multivalued mapping. Annals of Mathematical Statistics 38, 325–339 (1967)
17. Gulordava, K.: Empirical evaluation of credal classifiers. Master's thesis, University of Lugano (June 2010); Supervised by Zaffalon, M., and Corani, G.
18. Wallner, A.: Extreme points of coherent probabilities in finite spaces. International Journal of Approximate Reasoning 44(3), 339–357 (2007)
19. Warshall, S.: A theorem on Boolean matrices. Journal of the ACM 9(1), 11–12 (1962)
20. Walley, P.: Towards a unified theory of imprecise probability. International Journal of Approximate Reasoning 24, 125–148 (2000)
21. Denneberg, D.: Non-Additive Measure and Integral. Kluwer Academic, Dordrecht (1994)
22. Walley, P.: Coherent lower (and upper) probabilities. Statistics Research Report 22, University of Warwick, Coventry (1981)
23. de Cooman, G., Troffaes, M.C.M., Miranda, E.: n-Monotone exact functionals. Journal of Mathematical Analysis and Applications 347, 143–156 (2008)
24. Augustin, T.: Generalized basic probability assignments. International Journal of General Systems 4, 451–463 (2005)
25. Denoeux, T.: Reasoning with imprecise belief structures. International Journal of Approximate Reasoning 20, 79–111 (1999)
26. Fagin, R., Halpern, J.Y.: Uncertainty, belief, and probability. In: Computational Intelligence, pp. 1161–1167. Morgan Kaufmann (1989)

MCMC Estimation of Conditional Probabilities in Probabilistic Programming Languages

Bogdan Moldovan, Ingo Thon, Jesse Davis, and Luc de Raedt

Department of Computer Science, Katholieke Universiteit Leuven, Belgium

Abstract. Probabilistic logic programming languages are powerful formalisms that can model complex problems where it is necessary to represent both structure and uncertainty. Using exact inference methods to compute conditional probabilities in these languages is often intractable so approximate inference techniques are necessary. This paper proposes a Markov Chain Monte Carlo algorithm for estimating conditional probabilities based on sampling from an AND/OR tree for ProbLog, a general-purpose probabilistic logic programming language. We propose a parameterizable proposal distribution that generates the next sample in the Markov chain by probabilistically traversing the AND/OR tree from its root, which holds the evidence, to the leaves. An empirical evaluation on several different applications illustrates the advantages of our algorithm.

1 Introduction

Probabilistic programing languages (PPLs) embed probabilistic concepts into programming languages. They provide high-level constructs for specifying models that can capture both uncertainty and structure. Examples of PPLs include ProbLog [12,2], PRISM [22], BLOG [15], Church [6], and IBAL [17].

This paper focuses on ProbLog, a probabilistic extension of the logic programming language Prolog, based on Sato's distribution semantics [20]. A ProbLog program represents a distribution over possible worlds. Consequently, unlike in Prolog, the success or failure of a query is not deterministic. A central inference problem is computing the probability that a query succeeds conditioned on some given evidence. Unfortunately, computing such probabilities exactly for high dimensional realistic problems is unfeasible, only approximation techniques providing a polynomial time solution [1]. One of the most popular sampling techniques used by many PPLs [21,6,15] is Markov chain Monte Carlo (MCMC) [1].

We present an MCMC approach tailored to computing the conditional probability of a ProbLog query. Computing conditional probabilities in PPLs has, with a few exceptions [3], not yet received much attention in the literature.

Several challenges arise when designing a ProbLog MCMC algorithm. First, as ProbLog is a programming language, the possible worlds can be infinite, making it impossible to sample complete worlds. Our MCMC approach samples partial possible worlds (i.e., assignments to subsets of the random variables in model) which correspond to proofs. Second, ProbLog explicitly deals with the disjoint

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 436–448, 2013.

sum problem in contrast to other PPLs (e.g., Prism) that make the mutually exclusiveness assumption to avoid this NP-hard problem. The disjoint sum problem arises when two proofs overlap. We solve this using the Karp and Luby algorithm [9]. By not making the mutually exclusiveness assumption, a user can write a ProbLog program that more easily models a richer problem setting. Third, only those possible worlds that agree with the evidence are relevant for approximating the conditional probability. We employ an AND/OR tree rooted at the evidence, representing all such possible worlds, and probabilistically traverse the tree to generate only those samples where the evidence holds. The AND/OR tree is needed to deal with ProbLog's underlying non-deterministic nature, also distinguishing our approach from those applied to functional programming languages. Finally, in contrast to some other languages, we also provide support for numeric random variables and discrete distributions.

2 Background

We first review some basic concepts of logic programming: An atom $pred(t_1, ..., t_n)$ consists of a predicate $pred/n$ of arity n and t_i terms. A term is either a (lowercase) *constant*, a (uppercase) *variable*, or a functor $func/n$ applied on n terms. A *definite clause* is an expression of the form $h \leftarrow b_1, ..., b_n$, where h and the b_i are atoms. It states that h is true whenever all b_i are true. If n is 0, we have a fact $f \leftarrow$, which expresses that f is true. A *substitution* $\theta = \{X_1 = t_1, ..., X_n = t_n\}$ maps each variable X_i to a term t_i. Applying a substitution θ to an atom a yields $a\theta$, in which each occurrence of X_i in a is replaced with t_i.

A ProbLog [12,2] program consists of a set of labeled facts $p_i :: c_i$, where p_i is a probability value and c_i a fact, and a set of definite clauses. Each ground instance of such a fact represents a random variable that is true with probability p_i. We use the following ProbLog program as a running example in the paper:

```
0.05 :: burglary.
0.01 :: earthquake.
0.7 :: hears_alarm(john).
0.6 :: hears_alarm(mary).
```

```
alarm :- burglary.
alarm :- earthquake.
calls(Pers) :- alarm, hears_alarm(Pers).
```

It has the random variables: $burglary$, $earthquake$, $hears\_alarm(john)$ and $hears\_alarm(mary)$, and states that there is an alarm whenever there is burglary or an earthquake. The last clause states that if there is an alarm and a person hears the alarm, that person will call.

To model univariate discrete distributions (e.g., uniform, Poisson), we also allow for discrete distribution probabilistic facts $X \sim \phi :: f$. X is a logical variable appearing in atom f and ϕ a probability density function. Currently only the uniform and Poisson distributions are implemented. For example, $X \sim uniform(7) :: apples(X)$ specifies that $apples(X)$ is true with X sampled from the set of integers between 1 and 7 with equal probability. Only for the sampled value of X will $apples(X)$ be true. Each grounding of all the variables (except X) in f denotes a random variable. In ProbLog, all random variables (discrete distributions or probabilistic facts) are assumed marginally independent.

The semantics of the ProbLog program is then given by probability distributions over subsets of the facts f_i (called subprograms) and sample values for the numeric variables in the uniform and Poisson distributions. Each ground probabilistic fact $p :: f$ specifies an atomic choice, i.e., we can choose to include f as a fact (with probability p) or its negation $\overline{f}$ (with probability $1-p$), where $\overline{f}$ is the predicate denoting the explicit negation of f. These negated predicates may also occur in the background knowledge, allowing us to deal with explicit negation on probabilistic facts. For a uniform distribution, X will be sampled from the discrete uniform distribution and $f(x)$ will be included as a fact, where x is the sampled value for X. Poisson distributions are treated similarly.

The resulting set of facts is called a *total choice* [18] when we have included a fact for *all* random variables, and a *partial choice* otherwise. To each total or partial choice we can associate a probability. This is simply the product of the probabilities of the atoms chosen for inclusion in the total or partial choice, as these random variables are marginally independent. For example, the probability of the total choice $T_1 = \{burglary, \overline{earthquake}, hears\_alarm(john), \overline{hears\_alarm(mary)}\}$ is $0.05 \times .99 \times .7 \times .4$.

The distribution over total choices induces a probability distribution P over possible worlds, which also defines the (success) probability $P_s(q)$ of a query q (conjunction of atoms) as $P_s(q) = P(\{w|q \text{ is true in the possible world } w\})$. Continuing our example, the probability of *alarm* is equal to the probability that it is true in the 2^4 possible worlds. Rather than enumerating these worlds explicitly, one would compute the proofs of the query and observe that *alarm* is true exactly when *earthquake* or *burglary* is true. The partial choices corresponding to the two proofs are sometimes called *explanations*. So:

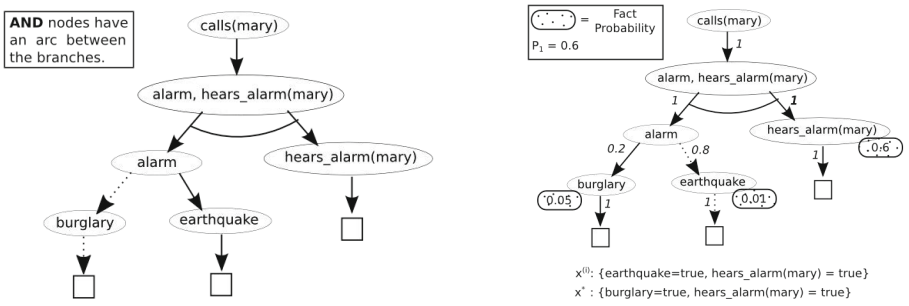
$$P_s(alarm) = P_s(burglary \vee earthquake)$$
$$= P_s(burglary \vee (\overline{burglary} \wedge earthquake)) = 0.05 + (.95 \times .01)$$

This derivation also illustrates the *disjoint sum* problem, as we have to make the two arguments of the disjunction *mutually exclusive* before we can correctly compute the probability of the query. This is a #P complete problem [23].

3 AND/OR Trees

Our MCMC algorithm relies on the notion of an AND/OR tree for definite programs [10]. Let T be a definite clause program and $? - e$ an evidence query. The AND/OR tree for ProbLog $pTree(e)$ of the given query is a tree with root e whose nodes are divided into two disjunctive sets, the set of *AND* nodes and the set of *atomic choice* nodes. Each node contains a query. Leafs of $pTree(e)$ are either an empty clause ($\square$) or a failure (for leaf $? - a_{leaf}$ no clause head in T unifies with a_{leaf}). The nodes $? - a_1, ..., ? - a_n$ constitute the children of an *AND* node $? - a_1, ..., a_n$. An atomic choice node can be of three types: exclusive OR-nodes, probabilistic atoms and discrete distribution atoms. An *OR* node $? - a$ has a child $? - (a_1, ..., a_n)\theta$ if and only if there is a definite clause $a \leftarrow a_1, ..., a_n$ in T and a substitution θ such that $a'\theta = a$. An atomic choice node $? - a$ (or $? - \overline{a}$) for a probabilistic atom $p :: a'$ adds $a'\theta$ (or $\overline{a'}\theta$) with $a\theta = a'\theta$ as a

fact to T, and imposes the constraint that $\overline{a'}\theta$ (or $a'\theta$) will never be added to T. Similarly, an atomic choice node $?-a$ for a discrete distribution atom $X \sim \phi :: f$ adds $a\theta\{X = v\}$ as a fact to T for one possible value v in the distribution ϕ with $a\theta\{X = v\} = f\theta\{X = v\}$, and imposes the contraint that facts will not be added for any other value than v. Since these facts are now added to T, they prove the node containing these probabilistic atomic choices, thus the child of this node is $\square$. Figure 1a illustrates $pTree(e)$ on our running example for $e = calls(mary)$. An AND/OR tree is obtained by starting with the root e and recursively expanding each node for the definite clause program.



(a) A solution tree (solid line arrows) (b) Highlighted solution tree for $x^{(i)}$.

Fig. 1. Example AND/OR tree for the evidence: $calls(mary)$

A *solution tree* S in the AND/OR tree $pTree(e)$ is a subtree such that 1) e is the root of S, 2) the children of all *AND* nodes that are in S are also in S, 3) all *OR* nodes that are in S have exactly one child that is also in S, and 4) all the leaves are $\square$. A solution tree is consistent with regard to random variables and atomic choices (e.g., it will not contain two atoms a and $\overline{a}$ and there cannot be two different values assigned to the same discrete distribution atom). A solution tree represents one particular proof of e. Since e is true with respect to every solution tree in $pTree(e)$, every solution tree implies a model of the evidence e. Figure 1a highlights with solid line arrows the solution tree corresponding to the partial possible world $\{earthquake = true, hears\_alarm(mary) = true\}$.

4 MCMC Algorithm Overview

Computing the conditional probability of a query in ProbLog is defined as:

Given: a ProbLog program T, a set of observed (evidence) atoms e, a query q
Do: Calculate $P(q|e)$

As it is often intractable to compute $P(q|e)$ exactly, we propose an MCMC [1,8,14] approach. Each state in the Markov Chain is a (partial) possible world. The estimate of $P(q|e)$ is obtained by dividing the number of partial possible worlds where e is true and q is entailed by the number of partial possible worlds where e is true. Two key challenges arise when designing the MCMC algorithm.

The first challenge is designing a proposal distribution that, as often as possible, constructs states that agree with e, as only these are relevant for estimating $P(q|e)$. We exploit the fact that each partial possible world meeting this criteria corresponds to a solution tree in $pTree(e)$. Given the previous solution tree, our proposal distribution builds a new one to propose as the candidate next state.

Secondly, two partial possible worlds can overlap, i.e., sampling their unassigned variables can lead to the same full possible world. If two such overlapping partial worlds are counted as distinct then that full possible world would be overcounted, skewing the probability estimate. We adapt ideas from the Karp and Luby algorithm [9] to identify overlapping worlds.

Our algorithm is similar to the standard MCMC algorithm in [1]. Until a stop criteria is met, each iteration proposes a candidate state, which is checked for overlap with previously seen states. If there is no overlap, or it can be resolved, we calculate the acceptance probability, and advance to the next state accordingly.

4.1 Proposing a New State

Our Markov chain samples solution trees from $pTree(e)$. We exploit the intuition that small changes in the solution tree are more likely to lead to another solution than a big jump by probabilistically favouring reusing parts of the current proof for e. Each proof requires making decisions at OR and atomic choice nodes. We stay close to the previous state by (1) following the same branch at an OR node with probability P_1, and (2) making the same atomic choice with probability P_2.

P_1 and P_2 are user defined parameters; higher values encourage more reuse between consecutive solution trees. Parameter choice depends on the problem. If $pTree(e)$ contains many solution trees with few shared branches, lower parameter values are better to encourage faster solution space exploration. If $pTree(e)$ contains only few solution trees or they share many branches, higher values are better to favour reuse. If solution trees are more evenly spead out in $pTree(e)$, parameters have smaller impact. Any non-zero values lead to eventual exploration of all solution trees in $pTree(e)$. Only these are relevant to estimate $P(q|e)$.

Algorithm 1 outlines the recursive procedure *prove* for proposing a new solution. Its parameters are: N (current node), S_{old} (previous solution tree), and S_{new} (tree under construction). It begins at the root node e and, depending on the type of the current node N, it recursively traverses $pTree(e)$ as follows:

AND node: Recursively call *prove* on each of the node's children because proving e requires proving each child. Return the conjunction of the results.

OR node: (At least) one of the children $c_1, \ldots, c_n$ of N needs to be proved in order for e to be true. To favour reuse, if N occurs in S_{old}, then pick the same child c as in S_{old} with probability P_1 and return $prove(c)$. Otherwise, pick c_i uniformly at random between $c_1, \ldots, c_n$ and return $prove(c_i)$.

Atomic choice node: To favour reuse, if N occurs in S_{old}, then with probability P_2 pick the same value for the random variable as in S_{old}. Otherwise pick a value for it from its probability distribution. For probabilistic atoms, only one value (either true or false) makes e true, so we are forced to pick this value for the proof to succeed. Add the atom in N to T and return true.

Empty clause: Return true.
Failure: No clause head in T unifies with the atom in the node. Return false.

Function *prove* returns true if it finds a solution tree, and false otherwise. If *prove* returns true, the partial possible world associated with S_{new} is the candidate next state. Otherwise, the candidate next state is identical to the current state: $x^* = x^{(i)}$, where we know e is entailed. Subsection 4.3 shows why this is advantageous.

Algorithm 1. bool $prove(N, S_{old}, var\ S_{new})$

```
   Require: Global vars: pTree(e)
 1 Add node N to S_new
 2 if N is AND node ? − a_1, ..., a_n then return  ⋀_i prove(a_i, S_old, S_new)
 3 else if N is OR node with n children then
 4 |  if N is in S_old then
 5 |  |  c_old = child of N in S_old
 6 |  |  with prob. P_1: return prove(c_old, S_old, S_new)
 7 |  let c_1, ..., c_n be the children of N in pTree(e)
 8 |  pick i uniformly from [1...n]
 9 |  return prove(c_i, S_old, S_new)
10 else if N is atomic choice then
11 |  if N is in S_old then
12 |  |  with prob. P_2: pick same value for random variable as in S_old
13 |  else pick value randomly from its prob distribution
14 |  add atom N to ProbLog program
15 |  return true
16 else if □ then return true
17 else return false
```

4.2 Handling Overlapping Partial Worlds

S_{new} represents one proof or explanation for e. Two different explanations for e are not necessarily mutually exclusive (i.e., they overlap). This occurs if, in both explanations, there exists a setting to the unassigned variables that produces the same *full* possible world. This is known as the disjoint sums of product problem.

We illustrate this problem on our example with e={$alarm$=$true$}. There are two solution trees corresponding to the partial possible worlds {$burglary$=$true$} and {$earthquake$=$true$}. Each partial possible world represents a *set* of full possible worlds. The partial world {$burglary$=$true$} represents the two full worlds: {$burglary$=$true$, $earthquake$=$false$} and {$burglary$=$true$, $earthquake$=$true$}. Similarly, the partial world {$earthquake$=$true$} represents the two full worlds: {$burglary$=$false$, $earthquake$=$true$} and {$burglary$=$true$, $earthquake$=$true$}. The full world {$burglary$=$true$, $earthquake$=$true$} is represented by both these two partial worlds. Two partial worlds overlap if they both can represent the same full world. Treating them as distinct (i.e., non-overlapping) will cause this full world to be counted twice, leading to an incorrect probability estimate.

To solve the disjoint sums problem we use the idea from the Karp and Luby algorithm [9]. Each possible world is assigned to exactly one of its explanations.

This assignment is defined as positive, leading to the world being accepted. We then use sampling of the unassigned variables in a partial world to resolve overlap. When a candidate sample is proposed, we assign it to its explanation represented by S_{new}, obtaining a pair of a possible world and an explanation. For each possible world, only one such pair is positive. As samples are obtained, we build a list of unique positive pairs, and check new candidate samples against this. If the sample overlaps with a previous one from the list assigned to a different explanation, we attempt to remove overlap as follows. We pick a variable from the previous possible world which is unassigned in the proposed world. Then we extend the proposed world by setting this variable to a value drawn from its distribution. We repeat this procedure until (1) we arrive at a world with no overlap which we save in the list and propose as the candidate state, or (2) no variable in the previous world is unassigned in our proposed world and there is still overlap. In the second case, we reject the sample and propose the current state instead. Intuitively, we reject sample contributions from the overlapping world. It was shown [9] that this results in an accurate estimate for $P(e)$.

Assume $\{earthquake{=}true\}$ is the first sampled possible world, assigned to the same explanation. If $\{burglary{=}true\}$ is the next sample, we identify an overlap and draw a value for *earthquake*. If $earthquake{=}true$, the full world overlaps with the first sample and we reject it. If $earthquake{=}false$, the overlap is eliminated. We then propose $\{burglary{=}true, earthquake{=}false\}$ and assign the world to the explanation $\{burglary{=}true\}$.

4.3 Computing the Acceptance Probability

The Markov chain advances by accepting a candidate state x^* with probability $A = min\{1, \frac{P(x^*)Q(x^{(i)}|x^*)}{P(x^{(i)})Q(x^*|x^{(i)})}\}$, and otherwise remains in the same state ($x^{(i+1)} = x^{(i)}$) [1]. $P(\cdot)$ is the probability of a state (i.e., partial possible world), and $Q(\cdot|\cdot)$ is the probability of transitioning from one state to another. We illustrate these calculations using the example in Figure 1b, where each choice branch is labeled with its probability of being selected in x^* given $x^{(i)}$.

Computing $P(\cdot)$: The probability of a partial world $w = \{c_1 = v_1, \ldots, c_n = v_n\}$ is: $P_{world}(w) = \prod_{i=1}^{n} P_{s_i}$, where P_{s_i} is probability that fact c_i takes on value v_i. Thus $P(\{burglary = true, hears\_alarm(mary) = true\}) = 0.05 \times 0.6 = 0.03$.

Computing $Q(\cdot|\cdot)$: $Q(x^*|x^{(i)})$ is the product of the probabilities of all the choices made when constructing S_{new} from S_{old} since the choices at each node type are made independently. Algorithm 2 shows *computeQ*, a recursive algorithm similar in structure to Algorithm 1, with S_{old} (previous solution tree) and S_{new} (proposed solution tree) as parameters. To compute $Q(x^{(i)}|x^*)$ we call *computeQ* and swap the order of the parameters. In our example in Figure 1b, at the top OR node the probability of the choice is 1 (node has only one child). Next, at the AND node, we multiply the probabilities obtained by recursively calling *computeQ* on each child. The atomic choice *hears_alarm(mary)* must be true for the proof to succeed, so there is no choice and we return 1. At the OR

Algorithm 2. float $computeQ(N, S_{old}, S_{new})$

if N *is AND node* $? - a_1, ..., a_n$ **then return** $\prod_i computeQ(a_i, S_{old}, S_{new})$
else if N *is OR node with* n *children* **then**
  c_{new} = child of N in S_{new}
  if N *and* c_{new} *are in* S_{old} **then**
    return $(P_1 + \frac{1-P_1}{n}) * computeQ(c_{new}, S_{old}, S_{new})$
  else if N *is in* S_{old} *but* c_{new} *is not in* S_{old} **then**
    return $(\frac{1-P_1}{n}) * computeQ(c_{new}, S_{old}, S_{new})$
  else return $(\frac{1}{n}) * computeQ(c_{new}, S_{old}, S_{new})$; `// none in` S_{old}
else if N *is atomic choice* **then**
  Let P_{new} be the prob of the value of random variable in S_{new}
  if N *is in* S_{old} *with same sampled value* **then**
    return $P_2 + (1 - P_2) * P_{new}$
  else if N *is in* S_{old} *with different sampled value* **then**
    return $(1 - P_2) * P_{new}$
  else return P_{new}; `//` N `is not in` S_{old}
else return 1; `// if` $\square$

node *alarm*, given parameter $P_1 = 0.6$, the probability of picking the child *burglary* is $\frac{1-0.6}{2} = 0.2$. We reach the atomic choice *burglary*, and we return 1. The product of all the choices made is: $Q(x^*|x^{(i)}) = 1 \times ((1) \times (0.2 \times 1)) = 0.2$.

Computing A**:** In our running example, the acceptance probability will be: $A = min\{1, \frac{0.03 \times 0.2}{0.006 \times 0.2}\} = 1$ and the proposed sample will be accepted.

If a traversal does not reach a solution, or if overlap cannot be resolved, the proposed state is the current state. This greatly simplifies the algorithm. In this case, computing $Q(x^*|x^{(i)})$ would have needed to sum over the probabilities of all paths in $pTree(e)$ where e is not entailed. However, the ratio $\frac{Q(x^{(i)}|x^*)}{Q(x^*|x^{(i)})} = 1$ (since $x^* = x^{(i)}$). Thus $A = 1$ and the MCMC chain advances with $x^{(i+1)} = x^{(i)}$.

5 Related Work

The use of MCMC techniques is popular in the literature on PPLs and statistical relational learning. Many languages (e.g., Blog [15], Church [6], Alchemy [13], Prism [22]) offer an inference algorithm based on MCMC. Our MCMC approach has the important difference that it needs to deal with the disjoint sum problem. The above mentioned techniques assume that the probability of a function or predicate call can be approximated by counting/weighting the number of succesful execution traces of the program. Doing this in the ProbLog context will lead to overcounting of partial worlds and possibly incorrect probability values larger than one. In Blog or Church this is a valid assumption as the underlying programming language is functional (i.e., deterministic). In Prism, one assumes mutually exclusive explanations so the problem does not arise. We solve this using the Karp and Luby algorithm [9], previously used in the ProbLog context in DNF sampling [11], but not with an MCMC approach. Aditionally, by proposing

states probabilistically, we eventually fully explore the state space and can tackle a bigger set of problems, while Alchemy and MLNs [13,19] combine MCMC with satisfiability testing to have an MC-SAT algorithm that can also tackle problem domains with deterministic or near-deterministic dependencies [19].

Wingate [24] proposed a general MCMC technique for obtaining a probability distribution over program execution traces, together with a general method of transforming arbitrary programming languages into PPLs. To compute a conditional probability, one would need to do rejection sampling on all these execution traces sampled from the unconditioned program. By comparison, our AND/OR tree based approach for estimating conditional probabilities attempts to guide the Markov chain towards the solution space of the conditioning query.

We want to stress that the use of AND/OR trees here is completely different than in the work by Dechter & probabilistic theorem proving (PTP) [4,5], in that, we employ the traditional trees used in theorem proving, whereas Dechter & PTP employ the special data structures used in a knowledge compilation setting. These data structures impose different requirements than the AND/OR trees and are aimed at optimizing some operations (e.g., weighted model counting).

6 Empirical Evaluation

The goal of the experimental evaluation is to explore how our MCMC approach:

Goal 1: compares to existing ProbLog inference techniques
Goal 2: compares to other PPLs when faced with hard constraints
Goal 3: copes with Poisson and uniform distributions
Implementation was in Yap-6 Prolog. Experiments were run on computers with Intel Core $i7-2600$ $3.4GHz$ processors, $8MB$ cache, and $16GB$ memory. Parameters were set to $P_1 = 0.6$, $P_2 = 0.4$, but varying them had minimal impact on performance on the three considered problem domains.

Goal 1: Comparison to the Following Existing ProbLog Inference Algorithms:

ProbLog Exact is the current exact inference implementation for ProbLog, which can scale to tens of thousands of proofs [12].

ProbLog MC is a naive Monte-Carlo method that samples possible worlds for a ProbLog program[12]. We reject the ones where the evidence does not hold.

ProbLog MC-SAT is the state-of-the-art approach to approximate inference in ProbLog [3]. It converts a ProbLog program to a CNF theory and then runs the MC-SAT inference algorithm [16].

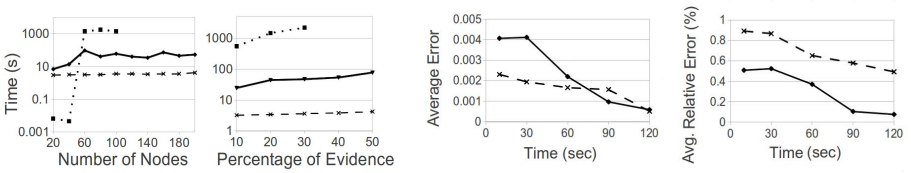
We use WebKB[1], a large data set about university webpages. The knowledge base (KB) contains deterministic knoweldge about the set of words present on the pages and links between pages. We only consider the overall 20 most commonly occuring words in all documents, not including stem words (e.g., the, of, a). The query is a ground wordclass/3 atom. For each setting we randomly generate 20 KBs and average results. Timeout is 1 hour, and this value is used in case of

[1] `http://www.cs.cmu.edu/~webkb/`

a timeout when computing average runtime. The lines in the graphs stop when more than half the runs time out. We run MCMC and MC-SAT for 100,000 samples. ProbLog MC is setup with a 95% confidence interval width of 0.01.

We first compare how run time varies with the number of pages in the domain. We vary the number of pages from 20 to 200. For each page, with 20% probability, we include its true class (e.g. course, staff, etc.) in the evidence. Figure 2a(left) shows the results, where ProbLog MC is not included as it cannot solve any task. ProbLog Exact cannot solve domains with more than 100 pages. MCMC and MC-SAT can solve any WebKB graph size, MC-SAT being faster.

In a second task, we compare how run time varies with the amount of evidence. We keep the number of pages constant at 100, but vary the probability that page's class is included in e from 10% to 50%. ProbLog MC cannot solve this task either, ProbLog exact can only solve settings with a smaller amount of evidence, while MCMC and MC-SAT can consistently solve any setting, as shown in Figure 2a(right). MC-SAT is also faster on this setting.



(a) Runtime comparisons (b) Error comparisons

Fig. 2. WebKB: MCMC is continuous line, Exact dotted, MC-SAT dashed

Since MC-SAT is faster than MCMC in producing the same number of samples, we investigate their accuracy next. We use the first task for the largest subset of pages (100) where we can obtain Exact probabilities. We run both methods for the same amount of time (i.e., MC-SAT produces more samples) and compare the errors in the predicted probabilities, as shown in Figure 2b. After the burn-in period, MCMC average error is about the same as MC-SAT, but its average relative error with respect to the exact probability is smaller.

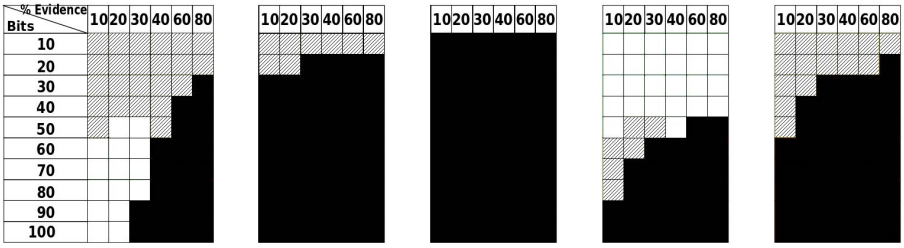
Goal 2: Comparison with other inference engines: the **Bher** implementation of Church [6] and the **Alchemy** [13] implementation of Markov logic [19].

As a test domain, we use Hamming codes, a family of linear error-correcting codes [7] containing data and parity bits. Instead of considering error correction or detection, we predict the values of certain bits given other bits as evidence. This is intended as an illustration of how algorithms cope with hard constraints (PPLs are not necessarily the best way to infer missing bit values). Hard constraints are important in PPLs, yet many approaches struggle.

We vary the number of bits in randomly produced Hamming codes from 10 to 100, and the percentage of bits included in the evidence from 10% to 80%. The query is one of the data bits. We run all sampling algorithms for 100,000 samples. Figure 3 shows a runtime comparison against the inference engines mentioned

above, also including ProbLog Exact and MC. White means the method is the fastest, striped that it solves the problem but is not the fastest, and black means timeout (1 hour) or invalid answer. For smaller domains MCMC run time versus Exact is overestimated as we converge faster than 100,000 samples.

Bher's MCMC algorithm has difficulty with the hard constraints in this problem and cannot switch between the two non-zero probability states, returning a probability of either 0 or 1. Solving this problem requires running inference multiple times and averaging results (100 times 1,000 samples). For Alchemy with the MC-SAT inference algorithm, the CNF conversion times out for any domain with more than 9 bits. MCMC can solve more problems than any of the other four approaches. In this task, we outperform them because we propose states probabilistically which eventually allows full exploration of the state space.



(a) MCMC (b) Bher (c) Church (d) Exact (e) MC

Fig. 3. Runtime: White=fastest, Striped=solves problem, Black=timeout/error

Goal 3: Poisson and Uniform Distributions We model a single server queue, showing a practical problem using these distributions. We assume that (1) the expected number of customer arrivals is 4 (i.e., Poisson distribution with $\lambda = 4$), and (2) the number of customers served is uniformly distributed between 1 and 8. At time t_0 the number of customers in the queue is 10. At $t_5 = t_0 + 5$, we observe (i.e., e) 12 customers. We want to find the posterior distributions of the number of customers in the queue at t_2 and number of customers served at t_3.

We ran our MCMC algorithm 20 times, each with 500,000 samples. The average runtime was 12 minutes. Figure 4 shows the prior and posterior distributions for the two queries. The posterior puts more weight on a higher number of customers in the queue at t_2 and a smaller number of customers served at t_3.

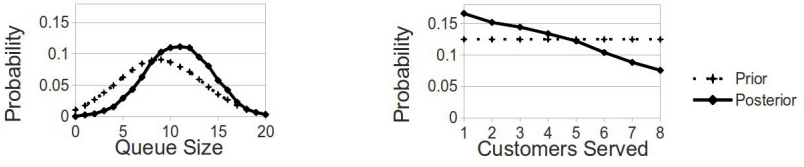


Fig. 4. Different Probability Distributions: at t_2 (left), at t_3 (right)

7 Conclusion

We presented an MCMC algorithm for estimating the conditional probability of a query given evidence in ProbLog. Our proposal distribution proposes candidate states by sampling solution trees from an AND/OR tree. Handling potential overlap between partial worlds is solved by employing ideas from the Karp and Luby algorithm. We provide support for Poisson and uniform distributions. We outperform existing ProbLog inference techniques on the considered tasks.

Acknowledgements. Bogdan Moldovan is supported by the IWT (agentschap voor Innovatie door Wetenschap en Technologie). This work is supported by the European Community's 7th Framework Programme, grant agreement First-MM-248258.

References

1. Andrieu, C., de Freitas, N., Doucet, A., Jordan, M.I.: An introduction to MCMC for machine learning. Machine Learning 50 (2003)
2. De Raedt, L., Kimmig, A., Toivonen, H.: Problog: A probabilistic Prolog and its application in link discovery. In: IJCAI, pp. 2462–2467 (2007)
3. Fierens, D., Van den Broeck, G., Thon, I., Gutmann, B., De Raedt, L.: Inference in probabilistic logic programs using weighted CNF's. In: UAI (2011)
4. Gogate, V., Dechter, R.: AND/OR importance sampling. In: UAI (2008)
5. Gogate, V., Domingos, P.: Probabilistic theorem proving. CoRR, abs/1202.3724 (2012)
6. Goodman, N., Mansinghka, V.K., Roy, D.M., Bonawitz, K., Tenenbaum, J.B.: Church: a language for generative models. In: UAI, pp. 220–229 (2008)
7. Hamming, R.W.: Error detecting and error correcting codes. Bell System Technical J. 29, 147 (1950)
8. Hastings, W.K.: Monte Carlo sampling methods using Markov chains and their applications. Biometrika 57, 97–109 (1970)
9. Karp, R.M., Luby, M.: Monte-carlo algorithms for enumeration and reliability problems. In: FOCS, pp. 56–64. IEEE Computer Society (1983)
10. Kersting, K., De Raedt, L.: Bayesian logic programs. CoRR, cs.AI/0111058 (2001)
11. Kimmig, A.: A Probabilistic Prolog and its Applications. PhD thesis, Informatics Section, Department of Computer Science, KU Leuven, Belgium (November 2010)
12. Kimmig, A., Demoen, B., De Raedt, L., Santos Costa, V., Rocha, R.: On the implementation of the probabilistic logic programming language ProbLog. Theory and Practice of Logic Programming 11, 235–262 (2011)
13. Kok, S., Sumner, M., Richardson, M., Singla, P., Poon, H., Lowd, D., Wang, J., Nath, A., Domingos, P.: The alchemy system for statistical relational AI. Technical report, Dept. of Computer Science and Engineering, U. of Washington, WA (2010)
14. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equations of state calculations by fast computing machine. Journal of Chemical Physics 21, 1087–1091 (1953)
15. Milch, B., Marthi, B., Russell, S., Sontag, D., Ong, D.L., Kolobov, A.: BLOG: Probabilistic models with unknown objects. In: IJCAI, pp. 1352–1359 (2005)
16. Park, J.D.: Using weighted MAX-SAT engines to solve MPE. In: AAAI/IAAI, pp. 682–687. AAAI Press, Menlo Park (2002)

17. Pfeffer, A.: IBAL: A probabilistic rational programming language. In: IJCAI (2001)
18. Poole, D.: The independent choice logic for modelling multiple agents under uncertainty. Artif. Intell. 94(1-2), 7–56 (1997)
19. Richardson, M., Domingos, P.: Markov logic networks. Machine Learning 62(1-2), 107–136 (2006)
20. Sato, T.: A statistical learning method for logic programs with distribution semantics. In: ICLP, pp. 715–729. MIT Press (1995)
21. Sato, T.: A general MCMC method for bayesian inference in logic-based probabilistic modeling. In: IJCAI, pp. 1472–1477. IJCAI/AAAI (2011)
22. Sato, T., Kameya, Y.: PRISM: A language for symbolic-statistical modeling. In: IJCAI, pp. 1330–1339 (1997)
23. Valiant, L.G.: The complexity of enumeration and reliability problems. SIAM Journal on Computing 8, 410–421 (1979)
24. Wingate, D., Stuhlmüller, A., Goodman, N.D.: Lightweight implementations of probabilistic programming languages via transformational compilation. Journal of Machine Learning Research - Proceedings Track 15, 770–778 (2011)

Sorted-Pareto Dominance and Qualitative Notions of Optimality*

Conor O'Mahony and Nic Wilson

Cork Constraint Computation Centre,
University College Cork, Ireland
{c.omahony,n.wilson}@4c.ucc.ie

Abstract. Pareto dominance is often used in decision making to compare decisions that have multiple preference values – however it can produce an unmanageably large number of Pareto optimal decisions. When preference value scales can be made commensurate, then the Sorted-Pareto relation produces a smaller, more manageable set of decisions that are still Pareto optimal. Sorted-Pareto relies only on qualitative or ordinal preference information, which can be easier to obtain than quantitative information. This leads to a partial order on the decisions, and in such partially-ordered settings, there can be many different natural notions of optimality. In this paper, we look at these natural notions of optimality, applied to the Sorted-Pareto and min-sum of weights case; the Sorted-Pareto ordering has a semantics in decision making under uncertainty, being consistent with any possible order-preserving function that maps an ordinal scale to a numerical one. We show that these optimality classes and the relationships between them provide a meaningful way to categorise optimal decisions for presenting to a decision maker.

1 Introduction

In a decision-making task, it is often the case that the basis for comparing decisions involves more than one preference value (e.g., evaluations of multiple criteria in multi-criteria decision making, evaluations by more than one agent in multi-agent decision making, or considerations of different states in decision making under uncertainty), and therefore we have a preference vector for each decision. In these cases, Pareto dominance is an often used preference relation, where a decision Pareto dominates another if it is at least as good as the other in every component (comparing the preference vectors component-wise), and a decision is Pareto optimal if it is not Pareto dominated by any other [18, Ch. 2]. For example, for minimising costs, where costs are on an ordered scale $T = (low, med, hi)$, and we have three decisions with preference vectors: $a = (low, hi)$, $b = (med, low)$ and $c = (med, hi)$, we can see that both a and b Pareto dominate c, and also, since a and b do not Pareto dominate each other, they are both Pareto optimal. This relation is not very discerning though, and often the set of

* This material is based upon works supported by the Science Foundation Ireland under Grant No. 08/PI/I1912.

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 449–460, 2013.

Pareto optimal decisions is very large. However, if the preference scales used in each component are commensurate (or can be normalised as such), then we can compare decisions by sorting the preference vectors first and then performing the component-wise comparison – which leads to a more discerning relation. For example, if a and b are sorted in non-descending order, i.e., (low, hi) and (low, med) respectively, the second vector now dominates the first, and therefore we now have only one undominated decision w.r.t. this new relation, which we call the Sorted-Pareto relation. This leads to a smaller, more manageable set of Sorted-Pareto optimal solutions, that are still Pareto optimal, as shown in [13].

If the scale T is quantitative, or we have information that gives a quantitative mapping for T, e.g., we have a mapping $f : T \to \mathbb{R}^+$, then the decisions could be compared by summing the preference vector values and seeing which decisions have the smallest sum of costs, i.e., the min-sum of weights. However, often the preference information available is only of an ordinal or qualitative nature, as it can be easier to obtain such information, e.g., there may be uncertainty over exact values, or it may be easier to elicit qualitative preference information from a decision maker [12]. Sorted-Pareto relies only on ordinal or qualitative information, and therefore can be used in these qualitative decision making situations. In addition, for any mapping $f : T \to \mathbb{R}^+$, where f is order-preserving w.r.t. scale T, we show that Sorted-Pareto is compatible with any such mapping.

In a partially ordered setting, such as in the situation just described, there can be different natural notions of optimality. The framework in [21] describes some of these notions, for qualitative decision making under uncertainty, where there are different possible scenarios in a given problem. This gives us classes of decisions that are not dominated by any other decision, decisions that are possibly optimal or possibly strictly optimal, (i.e., optimal in some scenario), and decisions that are optimal in all scenarios. Sorted-Pareto connects to Weighted Constraints Satisfaction Problems (WCSP) [17, Ch. 9] and Bayesian Networks [15] where we only have ordinal information, and in these frameworks the possibly optimal decisions are those that are min-sum optimal for some compatible WCSP, or are the complete assignments that are most probable in some compatible Bayesian Network. In this paper, we look at the relationship between Sorted-Pareto and min-sum of weights in Section 3, and in Section 4 we then examine these different natural notions of optimality from [21] and apply them to the Sorted-Pareto and min-sum of weights case. In Section 5, we show how to generate these optimality classes for Sorted-Pareto, and in Section 6 we present some experimental results.

2 Preliminaries

We assume a minimising context, where lower preference values are preferred. A preference relation $\preceq$ on a set $\mathcal{A}$ is a binary relation that gives an ordering over $\mathcal{A}$, i.e., given any $\alpha, \beta \in \mathcal{A}$, if $\alpha \preceq \beta$, then α is preferred to β according to $\preceq$. Relation $\preceq$ is a *preorder*, if it is reflexive ($\alpha \preceq \alpha$, for all $\alpha \in \mathcal{A}$) and transitive (i.e., if $\alpha \preceq \beta$ and $\beta \preceq \gamma$, then $\alpha \preceq \gamma$). Relation $\preceq$ is a *total preorder*, if it is complete (i.e., either $\alpha \preceq \beta$, or $\beta \preceq \alpha$, or both, for all $\alpha, \beta \in \mathcal{A}$) and

transitive. For a preorder $\preceq$ on $\mathcal{A}$, we have a corresponding strict relation $\prec$, and a corresponding equivalence relation $\equiv$, defined respectively as: $\alpha \prec \beta$ if and only if $\alpha \preceq \beta$ and $\beta \not\preceq \alpha$; and $\alpha \equiv \beta$ if and only if $\alpha \preceq \beta$ and $\beta \preceq \alpha$.

We consider situations where the following preference information is available for some finite set of decisions $\mathcal{A}$. Let $\mathcal{S} = \{1, \ldots, m\}$ be a finite set, where each $i \in \mathcal{S}$ labels some aspect of the decisions in $\mathcal{A}$ for which a preference can be expressed. Let T be a scale, totally ordered by relation $\leq$. Let $\alpha_i \in T$ represent a preference value for decision $\alpha \in \mathcal{A}$ in aspect i. Let $\alpha = (\alpha_1, \ldots, \alpha_m)$ be the preference vector of m preference values (to ease notation, we interchangeably use "α" as meaning a decision $\alpha \in \mathcal{A}$, or as meaning the evaluation vector $\alpha = (\alpha_1, \ldots, \alpha_m)$). Let $\alpha^{\uparrow} = (\alpha_{(1)}, \ldots, \alpha_{(m)})$ be the sorted preference vector such that $\alpha_{(1)} \leq \ldots \leq \alpha_{(m)}$, i.e., the values are ordered w.r.t. the scale T. For any two preference vectors α and β: $\alpha \leq \beta$ if and only if $\alpha_i \leq \beta_i$ for all $i \in \{1, \ldots, m\}$; and $\alpha < \beta$ if and only if $\alpha_i \leq \beta_i$ for all $i \in \{1, \ldots, m\}$, and there exists $j \in \{i, \ldots, m\}$ such that $\alpha_j < \beta_j$.

3 Sorted-Pareto and Min-Sum of Weights

In this section, we recall definitions for Sorted-Pareto dominance from [13], and show how this ordering relates to min-sum of weights. For all $\alpha, \beta \in \mathcal{A}$, decision α *Weak Sorted-Pareto dominates* β, written as $\alpha \preccurlyeq_{\text{SP}} \beta$, if and only if $\alpha^{\uparrow} \leq \beta^{\uparrow}$. Decision α *Sorted-Pareto dominates* β, written as $\alpha \prec_{\text{SP}} \beta$, if and only if $\alpha^{\uparrow} < \beta^{\uparrow}$, or in terms of $\preccurlyeq_{\text{SP}}$, if and only if $\alpha \preccurlyeq_{\text{SP}} \beta$ and $\beta \not\preccurlyeq_{\text{SP}} \alpha$. Decision α is *Sorted-Pareto equivalent* to β, written as $\alpha \equiv_{\text{SP}} \beta$, if and only if $\alpha^{\uparrow} = \beta^{\uparrow}$, or in terms of $\preccurlyeq_{\text{SP}}$, if and only if $\alpha \preccurlyeq_{\text{SP}} \beta$ and $\beta \preccurlyeq_{\text{SP}} \alpha$. Let $[\alpha]_{\text{SP}}$ denote the SP-equivalence class of $\alpha \in \mathcal{A}$, where $[\alpha]_{\text{SP}} = \{\beta \in \mathcal{A} : \alpha \equiv_{\text{SP}} \beta\}$ Decision α is *Sorted-Pareto optimal* (or *undominated*) if and only if there is no $\beta \in \mathcal{A}$ such that $\beta \prec_{\text{SP}} \alpha$.

Min-Sum of Weights. We consider situations in which there is additional quantitative preference information available, i.e., we have a function $f : T \to \mathbb{R}^+$. In such cases, we can order the set of decisions by using the min-sum of weights, defined as follows.

For some $f : T \to \mathbb{R}^+$, for all $\alpha, \beta \in \mathcal{A}$, decision α is *min-sum preferred* to β, written as $\alpha \leq_f \beta$, if and only if $\sum_{i=1}^{m} f(\alpha_i) \leq \sum_{i=1}^{m} f(\beta_i)$. Decision α is *strictly min-sum preferred* to β, written as $\alpha <_f \beta$, if and only if $\sum_{i=1}^{m} f(\alpha_i) < \sum_{i=1}^{m} f(\beta_i)$. Decision α is *min-sum equivalent* to β, written as $\alpha \equiv_f \beta$, if and only if $\sum_{i=1}^{m} f(\alpha_i) = \sum_{i=1}^{m} f(\beta_i)$. The relation $\leq_f$ forms a total preorder on a set of decisions $\mathcal{A}$. Decision α is *min-sum-optimal* for f if and only if for all $\beta \in \mathcal{A}$, $\alpha \leq_f \beta$.

3.1 Relating Sorted-Pareto and Min-Sum of Weights

Let F be the set of all possible weight functions $f : T \to \mathbb{R}^+$ such that $f \in F$ if and only if f is monotonic w.r.t. T, i.e., $u \leq v \Leftrightarrow f(u) \leq f(v)$ for all $u, v \in T$. Define the order relation $\leq_F$ on $\mathcal{A}$ as, for all $\alpha, \beta \in \mathcal{A}$, $\alpha \leq_F \beta \Leftrightarrow \alpha \leq_f \beta$, for all f monotonic w.r.t. T. From Theorem 1 in [13], we have that $\leq_F = \preccurlyeq_{\text{SP}}$.

Now, let F' be the set of all possible weight functions such that $f \in F'$ if and only if f is *strictly* monotonic w.r.t. T, i.e., $u < v \Leftrightarrow f(u) < f(v)$ for all $u, v \in T$. Define the order relation $\leq_{F'}$ as, for all $\alpha, \beta \in \mathcal{A}$, $\alpha \leq_{F'} \beta \Leftrightarrow \alpha \leq_f \beta$, for all f strictly monotonic w.r.t. T. Define $<_{\cap_{F'}}$ as the intersection of all $<_f$ such that $f \in F'$, i.e., $<_{\cap_{F'}} = \bigcap_{f \in F'} <_f$ so for all $\alpha, \beta \in \mathcal{A}$, $\alpha <_{\cap_{F'}} \beta$ if and only if for all $f \in F'$, $\alpha <_f \beta$. We have the following results (proofs are in an extended version of the paper [14]).

Theorem 1. $\preccurlyeq_{\text{SP}} = \leq_F = \leq_{F'}$

Corollary 1. $\prec_{\text{SP}} = <_{\cap_{F'}}$

4 Qualitative Notions of Optimality

In this section, we look at the different notions of optimality from the qualitative decision making framework in [21], which we use to describe the relationship between Sorted-Pareto and min-sum of weights. A Multiple Ordering Decision Structure (MODS) is a tuple $\mathcal{G} = \langle \mathcal{A}, \mathcal{P}, \{\preccurlyeq_p : p \in \mathcal{P}\}\rangle$, where $\mathcal{A}$ is a set of decisions, $\mathcal{P}$ is a set of possible scenarios, and for each $p \in \mathcal{P}$, relation $\preccurlyeq_p$ is a total preorder on $\mathcal{A}$, with corresponding strict and equivalence relations $\prec_p$ and $\equiv_p$ respectively.

For any instance of this framework, we have the following relations that always hold in general. Decision α *necessarily dominates* β, written $\alpha \preccurlyeq_N \beta$, if and only if $\alpha \preccurlyeq_p \beta$, for all $p \in \mathcal{P}$. Relation $\preccurlyeq_{\text{N}}$ is the intersection of $\preccurlyeq_p$ over all $p \in \mathcal{P}$. Relation $\preccurlyeq_N$ has corresponding strict and equivalence relations $\prec_N$ and $\equiv_N$ respectively. Let $[\alpha]_{\text{N}}$ denote the N-equivalence class of $\alpha \in \mathcal{A}$, where $[\alpha]_{\text{N}} = \{\beta \in \mathcal{A} : \alpha \equiv_N \beta\}$. Decision α *necessarily strictly dominates* β, written $\alpha \prec_{\text{NS}} \beta$, if and only if $\alpha \prec_p \beta$ for all $p \in \mathcal{P}$. Relation $\prec_{\text{NS}}$ is the intersection of $\prec_p$ over all $p \in \mathcal{P}$.

Optimality Classes. We now look at different notions of optimality for the general case. Decision α is *necessarily optimal* if and only if $\alpha \preccurlyeq_{\text{N}} \beta$ for all $\beta \in \mathcal{A}$. The set of these decisions is denoted by $\text{NO}(\mathcal{G})$. Decision α is *necessarily strictly optimal* if and only if $\alpha \prec_{\text{NS}} \beta$ for all $\beta \in \mathcal{A} \setminus [\alpha]_{\text{N}}$. The set of these decisions is denoted by $\text{NSO}(\mathcal{G})$. Decision α is *possibly optimal* if and only if there exists $p \in \mathcal{P}$ such that $\alpha \preccurlyeq_p \beta$ for all $\beta \in \mathcal{A}$. The set of these decisions is denoted by $\text{PO}(\mathcal{G})$. Decision α is *possibly strictly optimal* if and only if there exists $p \in \mathcal{P}$ such that $\alpha \prec_p \beta$ for all $\beta \in \mathcal{A} \setminus [\alpha]_{\text{N}}$. The set of these decisions is denoted by $\text{PSO}(\mathcal{G})$. A decision α is in $\text{CD}(\mathcal{G})$, if and only if for all $\beta \in \mathcal{A}$, there exists $p \in \mathcal{P}$ such that $\alpha \preccurlyeq_p \beta$. $\text{CD}(\mathcal{G})$ are the decisions that are undominated w.r.t. $\prec_{\text{NS}}$. A decision α is in $\text{CSD}(\mathcal{G})$ if and only if for all $\beta \in \mathcal{A} \setminus [\alpha]_{\text{N}}$, there exists $p \in \mathcal{P}$ such that $\alpha \prec_p \beta$. $\text{CSD}(\mathcal{G})$ are the decisions that are undominated w.r.t. $\prec_{\text{N}}$. We also have the following optimality classes, which are intersections between existing classes. $\text{NOPSO}(\mathcal{G})$ is the intersection of $\text{NO}(\mathcal{G})$ and $\text{PSO}(\mathcal{G})$. $\text{PO}'(\mathcal{G})$ is the intersection of $\text{PO}(\mathcal{G})$ and $\text{CSD}(\mathcal{G})$.

Figure 1 shows precisely the subclass relationships between these optimality classes that always hold in the general case, as given by Theorem 1 in [21]. [21] also gives an example of strict subclass relationships between each of the optimality classes.

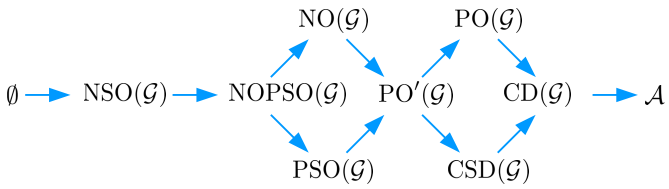


Fig. 1. Subclass relationships ($\subseteq$) between classes that always hold in general

4.1 Sorted-Pareto MODS

Recall from Section 3, where we define F be the set of all possible weight functions $f : T \to \mathbb{R}^+$ such that f is monotonic w.r.t. T. We also have that for all $\alpha, \beta, \in \mathcal{A}$, $\alpha \preccurlyeq_{\text{SP}} \beta \Leftrightarrow \alpha \leq_f \beta$ for all $f \in F$, i.e., α Weak Sorted-Pareto dominates β if and only if α is min-sum-preferred to β for all $f \in F$. This gives us the Sorted Pareto MODS $\mathcal{S} = \langle \mathcal{A}, F, \{\leq_f : f \in F\}\rangle$, where the set of scenarios is the set F of possible order-preserving weight functions $f : T \to \mathbb{R}^+$, and the set of possible orderings is that given by the min-sum of weights orderings for all possible weight functions, i.e., the set $\{\leq_f : f \in F\}$.

For the Sorted-Pareto MODS $\mathcal{S}$, we have the following relations. Decision α necessarily dominates β if and only if $\alpha \leq_f \beta$ for all $f \in F$. Since $\alpha \leq_f \beta$, for all $f \in F \Leftrightarrow \alpha \preccurlyeq_{\text{SP}} \beta$, this gives us the result in Proposition 1.

Proposition 1. *For* MODS $\mathcal{S}$, $\preccurlyeq_{\text{N}} = \preccurlyeq_{\text{SP}}$

Since we have that $\alpha \prec_{\text{SP}} \beta$ if and only if $\alpha \preccurlyeq_{\text{SP}} \beta$ and $\beta \not\preccurlyeq_{\text{SP}} \alpha$, then we also have that $\prec_{\text{N}} = \prec_{\text{SP}}$. Decision α necessarily strictly dominates β if and only if $\alpha <_f \beta$ for all $f \in F$. Since from Corollary 1, $<_{\cap_{F'}} = \prec_{\text{SP}}$, and $<_{\cap_{F'}}$ is defined as the intersection of all $<_f$ such that $f \in F'$ (and $F' \subseteq F$), then we have the result in Proposition 2.

Proposition 2. *For* MODS $\mathcal{S}$, $\prec_{\text{NS}} = \prec_{\text{N}} = \prec_{\text{SP}}$

We have an equivalence relation for each $f \in F$, i.e., $\alpha \equiv_f \beta$ if and only if $\sum_{i=1}^{m} f(\alpha_i) = \sum_{i=1}^{m} f(\beta_i)$. We also have an equivalence relation $\equiv_F$, which is the intersection of $\equiv_f$ over all $f \in F$, i.e., $\equiv_F$ is equal to $\bigcap_{f \in F} \equiv_f$, so $\alpha \equiv_F \beta$ if and only if they are equivalent over all possible choice of f. Let $[\alpha]_F$ denote the F-equivalence class of $\alpha \in \mathcal{A}$, where $[\alpha]_F = \{\beta \in \mathcal{A} : \alpha \equiv_F \beta\}$. Since we have from Theorem 1 in [13] that $\leq_F$ is equal to $\preccurlyeq_{\text{SP}}$, i.e., $\preccurlyeq_{\text{SP}} = \bigcap_{f \in F} \leq_f$, then we have that $\equiv_F$ is equal to $\equiv_{\text{SP}}$, i.e., $\equiv_{\text{SP}}$ is the intersection of $\equiv_f$ over all $f \in F$, which gives us the result in Proposition 3.

Proposition 3. *For* MODS $\mathcal{S}$, $\equiv_N = \equiv_{\text{SP}}$

Sorted-Pareto Optimality Classes. We now look at the notions of optimality that are applicable for the Sorted-Pareto MODS $\mathcal{S}$. Decision α is in $\text{NO}(\mathcal{S})$ if and only if for all $\beta \in \mathcal{A}$, for all $f \in F$, $\alpha \leq_f \beta$, i.e., if and only if $\alpha \preccurlyeq_{\text{SP}} \beta$ for all $\beta \in \mathcal{A}$. Decision α is in $\text{NSO}(\mathcal{S})$ if and only if for all $\beta \in \mathcal{A} \setminus [\alpha]_F$, for all $f \in F$, $\alpha <_f \beta$, i.e., if and only if $\alpha \prec_{\text{SP}} \beta$ for all $\beta \in \mathcal{A} \setminus [\alpha]_{\text{SP}}$.

These definitions and Proposition 2 give us the result in Proposition 4.

Proposition 4. *For* MODS $\mathcal{S}$, $\text{NSO}(\mathcal{S}) = \text{NOPSO} = \text{NO}(\mathcal{S})$

Decision α is in $\text{CD}(\mathcal{S})$ if and only if for all $\beta \in \mathcal{A}$, there exists $f \in F$ such that $\alpha \leq_f \beta$. Decision α is in $\text{CSD}(\mathcal{S})$ if and only if for all $\beta \in \mathcal{A} \setminus [\alpha]_F$, there exists $f \in F$ such that $\alpha <_f \beta$.

Since in the general case $\text{CD}(\mathcal{G})$ are the decisions that are undominated w.r.t. $\prec_{\text{NS}}$ and $\text{CSD}(\mathcal{G})$ are the decisions that are undominated w.r.t. $\prec_{\text{N}}$, and also from Proposition 1 we have $\prec_{\text{NS}} = \prec_{\text{N}}$, then this gives us the result in Proposition 5.

Proposition 5. *For* MODS $\mathcal{S}$, $\text{CSD}(\mathcal{S}) = \text{CD}(\mathcal{S})$

Decision α is in $\text{PO}(\mathcal{S})$ if and only if there exists $f \in F$ such that for all $\beta \in \mathcal{A}$, $\alpha \leq_f \beta$. Decision α is in $\text{PSO}(\mathcal{S})$ if and only if there exists $f \in F$ such that for all $\beta \in \mathcal{A} \setminus [\alpha]$, $\alpha \leq_f \beta$. Let $\text{PO}'(\mathcal{S}) = \text{PO}(\mathcal{S}) \cap \text{CSD}(\mathcal{S})$ and $\text{NOPSO}(\mathcal{S}) = \text{NO}(\mathcal{S}) \cap \text{PSO}(\mathcal{S})$.

Since in the general case $\text{PO}(\mathcal{G}) \subseteq \text{CD}(\mathcal{G})$, and since we have from Proposition 5 that $\text{CSD}(\mathcal{S}) = \text{CD}(\mathcal{S})$, this gives us the result in Proposition 6.

Proposition 6. *For* MODS $\mathcal{S}$, $\text{PO}(\mathcal{S}) \subseteq \text{CSD}(\mathcal{S})$.

Given these results, we now look at the subclass relationship between the optimality classes for the Sorted-Pareto instance of the MODS framework. Propositions 1-6 and definitions give us that $(\text{NSO}(\mathcal{S}) = \text{NOPSO}(\mathcal{S}) = \text{NO}(\mathcal{S})) \subseteq \text{PSO}(\mathcal{S}) \subseteq (\text{PO}(\mathcal{S}) = \text{PO}'(\mathcal{S})) \subseteq (\text{CSD}(\mathcal{S}) = \text{CD}(\mathcal{S}))$, as shown in Figure 2.

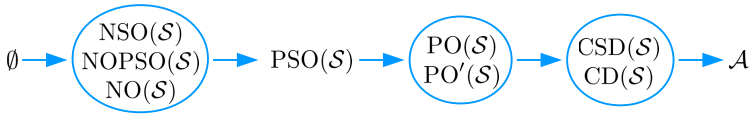


Fig. 2. Subclass relationships ($\subseteq$) between classes for MODS $\mathcal{S}$

Now we consider the case where there exists a decision that is necessarily optimal, i.e., when $\text{NO}(\mathcal{S}) \neq \emptyset$. Proposition 5 in [21] gives us that if $\text{NO}(\mathcal{S}) \neq \emptyset$, then we have $\text{NO}(\mathcal{S}) = \text{CSD}(\mathcal{S})$, and therefore we have a single SP-equivalence class, where the decisions are all equivalent. This gives us the result in Proposition 7.

Proposition 7. *For* MODS $\mathcal{S}$, *if* $\text{NO}(\mathcal{S}) \neq \emptyset$, *then* $\text{NSO}(\mathcal{S}) = \text{NO}(\mathcal{S}) = \text{PSO}(\mathcal{S}) = \text{PO}(\mathcal{S}) = \text{CSD}(\mathcal{S}) = \text{CD}(\mathcal{S}) \subseteq \mathcal{A}$

5 Computing Optimality Classes for MODS $\mathcal{S}$

In this section, we look at methods for generating the different optimality classes for Sorted-Pareto MODS $\mathcal{S}$. Here we assume that there is some procedure to generate CSD($\mathcal{S}$), i.e., that calculates the preference vectors for each decision and compares them using Sorted-Pareto dominance to generate the set of decisions that are non-dominated. For example, the branch and bound search algorithms detailed in [13] do exactly this; however other search procedures can be used. From CSD($\mathcal{S}$), NO($\mathcal{S}$) can be calculated by comparing all the solutions in CSD($\mathcal{S}$) with one another to see if any Sorted-Pareto dominate all others. In our experimental results in Section 6, we use the procedure outlined in [13] to calculate CSD($\mathcal{S}$), where the algorithm has been modified to maintain the set of currently non-dominated preference vectors, each preference vector mapping to the corresponding equivalence class of decisions. This leads to a substantial improvement in computation times as it results in a reduction in the number of dominance checks performed by the algorithm.

Calculating PO($\mathcal{S}$) and PSO($\mathcal{S}$). We want to determine if some decision α in CSD($\mathcal{S}$) is possibly optimal, i.e., there exists some weight function $f \in F$ such that $\alpha \leq_f \beta$ for all $\beta \in \text{CSD}(\mathcal{S})$. We can formulate this problem as a linear program P, as follows. Only certain elements on the scale T appear in any of the preference vectors for the decisions in CSD($\mathcal{S}$); let T' denote this set, i.e., $T' = \{i \in \beta : \beta \in \text{CSD}(\mathcal{S})\}$. For each of these elements $i \in T'$ we have a linear program variable w_i, representing an unknown weight. Since the scale T is totally ordered, then on these weights we have constraints of the form $w_i < w_j$, where $i < j$. For all $\beta \in \text{CSD}(\mathcal{S})$, we have a linear expression $\omega(\beta)$ as a sum in terms of the unknown weight variables, i.e., $\omega(\beta) = \sum_{i \in \beta} w_i$. For α to be possibly optimal, we require, for each $\beta \in \text{CSD}(\mathcal{S})$, that $\omega(\alpha) \leq \omega(\beta)$. Therefore we have a set P of linear inequalities, which consists of, (i) $w_i < w_j$, for all $i, j \in T'$, where $i < j$, and (ii) $\omega(\alpha) \leq \omega(\beta)$, for all $\beta \in \text{CSD}(\mathcal{S})$. If P has a feasible solution, then there exists some weights that make $\alpha \leq \beta$ for all $\beta \in \text{CSD}(\mathcal{S})$, i.e., α is possibly optimal.

In order to check this using a standard linear program solver, we need to convert to an equivalent problem which only has non-strict inequalities. Therefore, we create a linear program P' as follows, where $c > 0$ is some arbitrary strictly positive real number, for example, let us choose $c = 1$. Then, for any constraint in P of the form $w_i < w_j$, we have a constraint in P' with the form $w_j - w_i \geq c$, and for any constraint in P of the form $\omega(\alpha) \leq \omega(\beta)$, we have a constraint of the form $\omega(\beta) - \omega(\alpha) \geq 0$. We then solve the linear program P', and this has a solution if and only if P has a solution, and α is possibly optimal.

We can also determine if some solution is possibly strictly optimal, i.e., there exists f such that for all $\beta \in \text{CSD}(\mathcal{S}) \setminus [\alpha]$, $\alpha <_f \beta$. We have a set Q of linear inequalities for this problem, which consists of, (i) $w_i < w_j$, for all $i, j \in T'$, where $i < j$ and, (ii) $\omega(\alpha) < \omega(\beta)$, for all $\beta \in \text{CSD}(\mathcal{S})$. We again create a modified linear program Q' as follows: for any constraint in Q of the form $w_i < w_j$, we

have a constraint in Q' with the form $w_j - w_i \geq c$, and for any constraint in Q of the form $\omega(\alpha) < \omega(\beta)$, we have a constraint in Q' of the form $\omega(\beta) - \omega(\alpha) \geq c$. We then solve the linear program Q', and this has a solution if and only if Q has a solution, and α is possibly strictly optimal.

Proposition 8. *The set of linear inequalities P has a solution if and only if linear program P' has a solution. The set of linear inequalities Q has a solution if and only if linear program Q' has a solution.*

6 Experimental Results

In this section, we calculate the optimality classes CSD($\mathcal{S}$), PO($\mathcal{S}$), PSO($\mathcal{S}$), and NO($\mathcal{S}$) for some randomly generated and benchmark instances (details of the generation process are in the extended version of the paper [14]). As detailed in Section 5, we use the branch and bound algorithm from [13] to generate CSD($\mathcal{S}$) and NO($\mathcal{S}$), and we solve linear programs to generate PO($\mathcal{S}$) and PSO($\mathcal{S}$). The instances used are Weighted Constraint Satisfaction problems (WCSP) [17, Ch.9], where, for a set of problem variables, each variable can be assigned a value from its domain, and a complete assignment to all of the variables is a solution to the problem (which corresponds to a decision). There is also a set of weighted constraints which associate weights to these assignments, and these correspond to the preference levels of the solutions.

Table 1. Average size of optimality sets over 50 random instances, n denotes problem size, sc denotes size of preference vector (increasing), $|T|$ denotes size of scale

		CSD($\mathcal{S}$)			PO($\mathcal{S}$)		
n	sc	$\|T\| = 3$	$\|T\| = 5$	$\|T\| = 7$	$\|T\| = 3$	$\|T\| = 5$	$\|T\| = 7$
20	48	9.12 (2.90)	9.48 (6.78)	21.68 (19.28)	7.60 (2.62)	8.22 (5.88)	17.44 (15.70)
24	69	9.94 (3.70)	11.76 (10.18)	55.18 (51.30)	8.90 (3.46)	9.72 (8.36)	36.88 (34.50)
28	95	10.12 (4.52)	15.96 (14.04)	67.54 (63.88)	8.32 (3.86)	11.54 (10.32)	40.30 (38.44)
32	124	9.56 (5.00)	21.44 (19.28)	114.58 (110.96)	6.76 (3.86)	13.44 (12.42)	58.38 (56.82)
36	158	10.60 (5.68)	30.42 (27.04)	145.36 (143.02)	8.22 (4.64)	18.24 (16.58)	68.16 (67.06)
40	195	10.20 (5.38)	24.12 (23.00)	135.14 (134.44)	8.50 (4.62)	15.94 (15.38)	63.96 (63.84)

		PSO($\mathcal{S}$)			NO($\mathcal{S}$)		
n	sc	$\|T\| = 3$	$\|T\| = 5$	$\|T\| = 7$	$\|T\| = 3$	$\|T\| = 5$	$\|T\| = 7$
20	48	6.52 (2.30)	8.02 (5.78)	17.42 (15.68)	0.20 (0.10)	0.00 (0.00)	0.00 (0.00)
24	69	6.58 (2.92)	9.44 (8.18)	36.80 (34.46)	0.02 (0.02)	0.00 (0.00)	0.00 (0.00)
28	95	6.14 (3.24)	11.36 (10.14)	40.24 (38.40)	0.02 (0.02)	0.00 (0.00)	0.00 (0.00)
32	124	5.60 (3.36)	13.12 (12.16)	58.14 (56.68)	0.02 (0.02)	0.00 (0.00)	0.00 (0.00)
36	158	5.44 (3.72)	17.88 (16.26)	67.96 (66.92)	0.02 (0.02)	0.00 (0.00)	0.00 (0.00)
40	195	6.18 (3.78)	15.32 (14.92)	63.86 (63.76)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)

Random Instances. For these random instances: n denotes problem size, i.e., the number of variables; sc denotes the size of the preference vector for each solution, i.e., the number of weighted constraints; and $|T|$ denotes the size of the

ordinal scale used. Each set of problems was generated with 3 different scales, with $|T| = 3, 5$ and 7.

Table 1 shows the average size of the optimality classes (and the average number of equivalence classes in parentheses) for 50 random instances for problem size $n = 20, 24, \ldots, 40$. The size of the preference vector sc (i.e., the number of weighted constraints) is varied as a parameter of the problem size. It can be observed that $\mathrm{PO}(\mathcal{S})$ is usually smaller than $\mathrm{CSD}(\mathcal{S})$, with $\mathrm{PSO}(\mathcal{S})$ smaller again, and in nearly all cases $\mathrm{NO}(\mathcal{S})$ is empty.

Table 2 shows the average size of the optimality classes (and the average number of equivalence classes in parentheses) for 50 random instances for problem size $n = 20, 24, \ldots, 40$. The size of the preference vector sc is fixed at 10 for all instances. In these problems, the size of the $\mathrm{CSD}(\mathcal{S})$ sets are much larger than in Table 1, and often the same size as the $\mathrm{PO}(\mathcal{S})$ set. However the equivalence classes are much smaller, indicating that for these problems there are a large number of equivalent optimal solutions in each optimality class. Often $\mathrm{NO}(\mathcal{S})$ is non-empty, indicating a single equivalence class of necessarily optimal solutions, and in these cases we have $\mathrm{CSD}(\mathcal{S}) = \mathrm{PO}(\mathcal{S}) = \mathrm{PSO}(\mathcal{S}) = \mathrm{NO}(\mathcal{S})$.

Table 2. Average size of optimality sets over 50 random instances, n denotes problem size, sc denotes size of preference vector (fixed), $|T|$ denotes size of scale

		$\mathrm{CSD}(\mathcal{S})$			$\mathrm{PO}(\mathcal{S})$		
n	sc	$\lvert T\rvert = 3$	$\lvert T\rvert = 5$	$\lvert T\rvert = 7$	$\lvert T\rvert = 3$	$\lvert T\rvert = 5$	$\lvert T\rvert = 7$
20	10	190.16 (1.44)	121.38 (2.12)	116.84 (3.56)	190.16 (1.44)	118.50 (2.06)	115.12 (3.42)
24	10	330.88 (1.66)	191.08 (2.36)	242.48 (3.98)	330.88 (1.66)	190.84 (2.32)	227.26 (3.78)
28	10	379.14 (1.52)	196.32 (1.96)	201.68 (3.14)	373.86 (1.50)	186.06 (1.86)	191.22 (3.00)
32	10	642.56 (1.62)	393.72 (2.32)	354.36 (3.72)	642.56 (1.62)	393.72 (2.32)	344.16 (3.54)
36	10	925.92 (1.48)	709.56 (2.08)	663.32 (3.20)	925.92 (1.48)	697.56 (2.02)	652.30 (3.14)
40	10	1177.10 (1.54)	904.24 (2.18)	779.72 (3.06)	1177.10 (1.54)	904.24 (2.18)	779.72 (3.06)
		$\mathrm{PSO}(\mathcal{S})$			$\mathrm{NO}(\mathcal{S})$		
n	sc	$\lvert T\rvert = 3$	$\lvert T\rvert = 5$	$\lvert T\rvert = 7$	$\lvert T\rvert = 3$	$\lvert T\rvert = 5$	$\lvert T\rvert = 7$
20	10	167.54 (1.38)	118.50 (2.06)	114.02 (3.40)	86.94 (0.62)	36.40 (0.38)	15.36 (0.14)
24	10	275.30 (1.58)	190.84 (2.32)	227.26 (3.78)	96.10 (0.44)	15.50 (0.22)	3.88 (0.12)
28	10	373.06 (1.48)	186.06 (1.86)	191.22 (3.00)	164.94 (0.52)	80.56 (0.38)	18.14 (0.22)
32	10	638.78 (1.60)	392.76 (2.30)	344.04 (3.52)	183.22 (0.42)	11.42 (0.18)	5.92 (0.08)
36	10	925.92 (1.48)	697.56 (2.02)	652.30 (3.14)	150.98 (0.52)	78.00 (0.32)	18.32 (0.12)
40	10	1166.54 (1.52)	904.24 (2.18)	779.72 (3.06)	383.24 (0.50)	265.18 (0.32)	96.92 (0.14)

Benchmark Instances. Table 3 shows the size of the optimality classes (and the number of equivalence classes in parentheses), when applied to some modified WCSP instances from the Celar Radio-Link Frequency Assignment problem benchmark, where again problem size is denoted by n, sc denotes the size of the preference vector, and T denotes the size of the scale. These instances have been modified by adding random binary hard constraints to the problem, to limit the expected number of solutions to around $10,000$. $\mathrm{PO}(\mathcal{S})$ is usually smaller than $\mathrm{CSD}(\mathcal{S})$, but $\mathrm{PSO}(\mathcal{S})$ is only very seldom smaller than $\mathrm{PO}(\mathcal{S})$. In all of these instances, $\mathrm{NO}(\mathcal{S})$ is empty.

Table 3. Size of optimality sets for modified CELAR benchmark instances, n denotes problem size, sc denotes size of preference vector, $|T|$ denotes size of scale

Instance	n	sc	$\|T\|$	CSD($\mathcal{S}$)	PO($\mathcal{S}$)	PSO($\mathcal{S}$)	NO($\mathcal{S}$)
CELAR6-SUB0*	16	207	5	17 (16)	12 (11)	12 (11)	0 (0)
CELAR6-SUB1*	14	300	5	24 (20)	13 (11)	11 (9)	0 (0)
CELAR6-SUB2*	16	353	5	20 (12)	19 (11)	19 (11)	0 (0)
CELAR6-SUB3*	18	421	5	4 (3)	4 (3)	4 (3)	0 (0)
CELAR6-SUB4*	22	477	5	6 (6)	6 (6)	6 (6)	0 (0)
CELAR7-SUB0*	16	188	5	10 (10)	8 (8)	8 (8)	0 (0)
CELAR7-SUB1*	14	300	5	15 (11)	14 (10)	14 (10)	0 (0)
CELAR7-SUB2*	16	353	5	10 (8)	7 (5)	7 (5)	0 (0)
CELAR7-SUB3*	18	421	5	19 (15)	13 (9)	13 (9)	0 (0)
CELAR7-SUB4*	22	477	5	8 (8)	5 (5)	5 (5)	0 (0)

Discussion. One possible approach to choosing which decisions to present to a decision maker is to calculate CSD($\mathcal{S}$) first, and from this set, NO($\mathcal{S}$) can be easily derived. If NO($\mathcal{S}$) is not empty, then there are one or more equivalent decisions which are preferred to all other decisions for any choice of f, and these are prime candidates for presenting to a decisions maker. However, if NO($\mathcal{S}$) is empty, then PO($\mathcal{S}$) or PSO($\mathcal{S}$) can be computed and presented, these sets are often much smaller than CSD($\mathcal{S}$). PO($\mathcal{S}$) is the set of decisions that are min-sum optimal for some possible f, and thus are good candidates to present to a decision maker. If the PO($\mathcal{S}$) set is large, and there is a small number of equivalence classes, then a representative solution for each equivalence class could be chosen to present to a decision maker, since this would give a decision maker a choice between non-equivalent solutions that are possibly min-sum-optimal.

7 Related Work

As well as our own work [13,21], on which this work builds, Larichev and Moshkovich [11] use Sorted-Pareto in the context of normalising different criteria scales, and Kaci and Prade [10] use it in preference handling using possibilistic logic. Both Perny and Spanjaard [16] and Bossong and Schweigert [4] look at preference based search for generating sets of optimal solutions for shortest path problems, which is related to Sorted-Pareto as previously outlined in [13]. The Sorted Pareto relation extends the Pareto dominance relation [18], and computing the Sorted-Pareto optimal set is viable when preference level scales are commensurate, since calculating the Pareto optimal set can be prohibitive. Some works that approximate the Pareto optimal set in constraints problems include Torrens and Faltings [20], however this requires quantitative information as it performs a sum of weights on the preference vector, and Gavanelli [8] uses a branch and bound algorithm similar to what is used in [13]. Sorted-Pareto is reminiscent of Lorenz dominance [19], and is extended by preference relations that perform a lexicographic comparison on reordered vectors of preference levels, such as Lexicographic Min-Max [7] in multicriteria optimisation, and Leximin [6]. These lexicographic orderings place excessive emphasis on the

worse preference values, since they ignore better values when comparing two decisions, whereas Sorted-Pareto compares over all values. Bouveret and Lemaître [5] looks at depth first branch and bound algorithms for the computation of Leximin optimal solutions. The notions of optimality in the MODS framework are partly inspired by Gelain et al. [9], who investigate optimality for interval-valued constraints, however we assume only qualitative or ordinal information. Also, the MODS framework relates to decision making under complete uncertainty or ignorance (such as in Arrow [1]), since there is no quantitative information assumed on the importance or likelihood of scenarios.

8 Conclusion

In this paper, we looked at Sorted-Pareto dominance, a preference relation that assumes only qualitative information, and based on the correspondence between Sorted-Pareto and decision making under uncertainty, we argue that there are other natural notions of optimal decision. Specifically, we look at decisions that are undominated, i.e, CSD($\mathcal{S}$), the solutions that are optimal and strictly optimal in one (or more) scenarios, i.e., PO($\mathcal{S}$) and PSO($\mathcal{S}$) respectively, and the solutions that are optimal in all scenarios, i.e., NO($\mathcal{S}$). We explore the relations between these notions of optimality and show how to compute them for the Sorted-Pareto ordering and the min-sum of weights case. The experimental results show, that in some cases, NO($\mathcal{S}$) is non-empty, and these are the decisions that would be of most interest to a decision maker. However, in other cases, no such decisions exist, and then PO($\mathcal{S}$) and PSO($\mathcal{S}$) are of interest to a decision maker since these are the decisions that are optimal or strictly optimal in some scenario. The Sorted-Pareto ordering connects with Weighted Constraint Satisfaction problems (WCSP) [17, Ch. 9] (or similarly, with Generalised Additive Independence decompositions [2]), where a problem has only weights on an ordinal scale T; each such problem has a set of compatible proper weighted constraints problems, based on mapping the ordinal scale $T \to \mathbb{R}^+$. Sorted-Pareto is also connected to Bayesian Networks [15], where in a given network we only have ordinal probabilistic information and therefore we have an associated set of compatible Bayesian Networks. In a Weighted CSP with ordinal weights, the decisions that are possibly optimal are those that are min-sum optimal in some compatible weighted constraints problem, and in a Bayesian Network with ordinal probabilities, the possibly optimal decisions are those assignments that are most probable in some compatible Bayesian Network. In the context of decision making under uncertainty, we argue that these decisions would certainly be of interest to a decision maker.

References

1. Arrow, K.J., Hurwicz, L.: An Optimality Criterion for Decision Making under Ignorance. Basil Blackwell, Oxford (1972)
2. Bacchus, F., Grove, A.J.: Graphical models for preference and utility. In: Proc. UAI 1995, pp. 3–10 (1995)

3. Berkelaar, M., Eikland, K., Notebaert, P.: lp_solve, Open source (Mixed-Integer) Linear Programming system. Software (May 1, 2004), http://lpsolve.sourceforge.net/5.5/
4. Bossong, U., Schweigert, D.: Minimal paths on ordered graphs. Mathematica Slovaca 56, 23–31 (2006)
5. Bouveret, S., Lemaître, M.: Computing leximin-optimal solutions in constraint networks. Artif. Intell. 173(2), 343–364 (2009)
6. Dubois, D., Fargier, H., Prade, H.: Refinements of the maximin approach to decision-making in a fuzzy environment. Fuzzy Sets and Systems 81, 103–122 (1996)
7. Ehrgott, M.: A characterization of lexicographic max-ordering solutions. Technical report, University of Kaiserslautern, Department of Mathematics (1996)
8. Gavanelli, M.: An implementation of Pareto optimality in CLP(FD). In: CP-AI-OR - International Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimisation Problems (2002)
9. Gelain, M., Pini, M.S., Rossi, F., Brent Venable, K., Wilson, N.: Interval-valued soft constraint problems. Annals of Mathematics and Artificial Intelligence 58, 261–298 (2010)
10. Kaci, S., Prade, H.: Mastering the processing of preferences by using symbolic priorities in possibilistic logic. In: Proc. ECAI 2008, pp. 376–380 (2008)
11. Larichev, O.I., Moshkovich, H.M.: ZAPROS-LM – a method and system for ordering multiattribute alternatives. EJOR 82(3), 503–521 (1995)
12. Moshkovich, H.M., Mechitov, A.I., Olson, D.L.: Ordinal judgments in multiattribute decision analysis. EJOR 137(3), 625–641 (2002)
13. O'Mahony, C., Wilson, N.: Sorted-Pareto Dominance: an extension to Pareto Dominance and its application in Soft Constraints. In: Proc. ICTAI 2012 (2012)
14. O'Mahony, C., Wilson, N.: Sorted-Pareto Dominance and qualitative notions of optimality (extended version). Technical report, Cork Constraint Computation Centre (4C), University College Cork (2013), http://4c.ucc.ie/~comahony/docs/sorted_pareto_QNOO_extended.pdf
15. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann, San Fran. (1988)
16. Perny, P., Spanjaard, O.: A preference-based approach to spanning trees and shortest paths problems. EJOR 162, 584–601 (2005)
17. Rossi, F., van Beek, P., Walsh, T. (eds.): Handbook of Constraint Programming. Elsevier Science Inc., New York (2006)
18. Sen, A.K.: Collective choice and social welfare. North-Holland Publishing Co., Amsterdam (1970)
19. Shorrocks, A.F.: Ranking income distributions. Economica 50(197), 3–17 (1983)
20. Torrens, M., Faltings, B.: Using soft CSPs for approximating Pareto-optimal solution sets. In: Proc. AAAI 2002 Workshop: Preferences in AI and CP: Symbolic Approaches (2002)
21. Wilson, N., O'Mahony, C.: The relationships between qualitative notions of optimality for decision making under logical uncertainty. In: Proc. 22nd Irish Conference on Artificial Intelligence and Cognitive Science, AICS 2011 (2011)

A First-Order Dynamic Probability Logic

Zoran Ognjanović[1], Aleksandar Perović[2], and Dragan Doder[3]

[1] Mathematical Institute of Serbian Academy of Sciences and Arts,
Kneza Mihaila 36, 11000 Belgrade, Serbia
zorano@mi.sanu.ac.rs
[2] University of Belgrade, Faculty of Transport and Traffic Engineering,
Vojvode Stepe 305, 11000 Belgrade, Serbia
pera@sf.bg.ac.rs
[3] University of Belgrade, Faculty of Mechanical Engineering,
Kraljice Marije 16, 11000 Belgrade, Serbia
ddoder@mas.bg.ac.rs

Abstract. We introduce a Hilbert-style first-order dynamic probability logic and prove the strong completeness theorem for the class of rigid measurable models.

1 Introduction

First-order dynamic probability logic can be understood as a reasoning tool that involves classical logic enriched with dynamic and probability operators. The subject itself has naturally emerged from development and extensive application of probabilistic algorithms in late seventies and early eighties of the twentieth century. Due to the modal nature of dynamic and probability operators, a first-order dynamic logic is deeply rooted and closely related to first-order modal logic and its "derivatives": first-order dynamic logic, first-order probability logic and, to some extent, to a branching time first-order temporal probability logic.

The main purpose of this work is to present a Hilbert-style formalization of first-order dynamic probability logic. Developed syntax enables expressions such as "if α is possible after termination of the program a, then probability that α will be true after the termination is positive", which we formally code by $\langle a\rangle\alpha \rightarrow P^a_{>0}\alpha$.

In order to achieve the strong completeness, we have introduced infinitary inference rules with countably many premises, primarily in order to syntactically express the real valued probabilities and the connection between the modal operator $[a]$ and its reflexive and transitive closure $[a^*]$. The necessity of such approach is discussed below in the following subsection.

1.1 Axiomatization Issues

The standard modal basis for dynamic logics is the K modal logic (substitutional instances of tautologies plus the K-axiom $[a](\alpha \rightarrow \beta) \rightarrow ([a]\alpha \rightarrow [a]\beta)$). The natural approaches for the introduction of probabilities that we have adopted is

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 461–472, 2013.

to define a local probability space (probability space associated to a particular world w and a particular label a) on subsets of worlds that are accessible from the underlying world. As a consequence, for every world w the set of all accessible worlds from w must be nonempty. In modal terms, all models must be serial. Hence, the modal core must be extended to the modal logic D, i.e. K plus the D-axiom $[a]\alpha \rightarrow \langle a\rangle\alpha$. The more serious issue lies in the problem of the axiomatization of the reflexive and transitive closure of the accessability relation $R(a)$, i.e. in the formal description of the connection between the operators $[a]$ and $[a^*]$. This connection is precisely expressed by

$$[a^*]\alpha \Leftrightarrow \bigwedge_{n=0}^{\infty} [a]^n\alpha.$$

Hence, some form of infinity seems to be natural for the complete axiomatization. Similarly as in the cases of probability and temporal logics (see [1, 6–9]), we have adopted the approach to work in the very tame fragment of $L_{\omega_1,\omega}$-logic: the formulas are finite sequences of symbols, and we use certain type of infinitary inference rules with countably many premises. It turns out that the adequate form of the rule to satisfy the above semantic requirement and to prove $T \vdash \alpha \Rightarrow [a]T \vdash [a]\alpha$, where $[a]T = \{[a]\beta \mid \beta \in T\}$ is

$$\frac{\theta([a]^n\alpha),\ n \in \omega}{\theta([a^*]\alpha)},$$

where θ is such that $\theta\left(\bigwedge_{n=0}^{\infty}[a]^n\alpha\right) \Leftrightarrow \bigwedge_{n=0}^{\infty}\theta([a]^n\alpha)$.

Somewhat similar axiomatization issue is induced by the requirement that probability functions are real valued. In order to explain this, we will need a bit more information about the notation: our probability operators are of the form $P^a_{\geqslant r}$, where $r \in [0,1]\cap\mathbb{Q}$ and $P^a_{\geqslant r}\alpha$ reads "the probability of the set of all worlds satisfying α that are accessible by a is at least r", or more compactly as "the a-probability of α is at least r". Now it is easy to construct finitely satisfiable theories that propagates non-Archimedean probabilities. One such theory is

$$\{P^a_{\neq\frac{1}{2}}\alpha\} \cup \{P^a_{\geqslant\frac{1}{2}+\frac{1}{n}}\alpha \wedge P^a_{\leqslant\frac{1}{2}+\frac{1}{n}}\alpha \mid n \in \omega \text{ and } n \geqslant 2\},$$

which says that the a-probability of the formula α is not equal, but infinitely close to $\frac{1}{2}$. In order to render such theories inconsistent, i.e. to "destroy" all finitely satisfiable theories that propagate existence of proper infinitesimals, we need a rule of the following form: " if a-probability of α is infinitely close to the rational number $r \in [0,1]$, then it must be equal to r". The exact form of this so called Archimedean rule (preserves the Archimedean structure of the ordering) is given in the section devoted to axiomatization.

Another consequence of finitary form of formulas is the fact that the σ-additivity cannot be formally expressed. Thus, by probability we actually mean finitely additive probability. Besides the real valued codomain, we do not impose any other restriction on probability functions.

Finally, due to the presence of infinitary inference rules, the standard completion technique (Lindenbaum's theorem) has to be modified in the following way: if the current theory is inconsistent with the current formula and that formula can be derived by one of infinitary inference rules, than at least one premise should be blocked.

1.2 Related Work

The body of the work relevant for the study of dynamic and probability logic is quite staggering, so we will just mention the very few among the closely related ones. We start with the excellent overview of modal, temporal and dynamic logics presented by Colin Stirling in [12], where he has quite accurately emphasize one of the central problems in the axiomatization of the dynamic logics: the second order nature of the operator of the reflexive and transitive closure.

A complete axiomatization of propositional dynamic logic with a qualitative probability operator was given by D. P. Guelev in [3]. Another very interesting complete axiomatization of propositional dynamic logic presented as infinitary Gentzen system was proposed recently by G. Renardel de Lavalette, B. Kooi and R. Verburge in [10]. Though independently constructed, their completion technique is essentially the same as the one presented in our papers [1, 6–9] and it is a natural generalization of the classical techniques of Lindenbaum (maximization of a theory) and Henkin (construction of the canonical model).

In [11] J. Sack considers extensions of probabilistic dynamic epistemic logic. In spite the fact that there are no theorems presented in this paper, it offers important insights.

In [5] B. P. Kooi presented a propositional propositional dynamic probability logic and proved the corresponding simple completeness theorem (a formula is a theorem iff it is valid). The formal system presented in [5] is incomplete in the sense that there exist consistent unsatisfiable theories. One such example is $T = \{P_a(q) > 0\} \cup \{P_a(q) < 2^{-n} \ : \ n = 1, 2, 3, \ldots\}$ (here q is any propositional variable). Moreover, the axiomatization of the reflexive and transitive closure $[a^*]$ of the modal operator $[a]$ is not carried out in [5]. The formalization presented here resolves mentioned issues.

Up to our knowledge, the first Hilbert-style formalization of a first order dynamic probability logic was due to Y. A. Feldman and D. Harel presented in [2], where the authors have developed a rich and quite expressive syntax and proved the completeness theorem. The only "flaw" of the approach proposed by Feldman and Harel is in the self duality of dynamic operators (the formula $\neg[a]\alpha \leftrightarrow [a]\neg\alpha$ is a theorem of logic constructed in [2]), which is counterintuitive. As a consequence of the self duality, dynamic operators behave like the next operator in discrete linear time temporal logic.

The rest of the paper is structured as follows: in Section 2 syntax and semantics of our logic is introduced. Section 3 contains an axiomatization. logic. Strong completeness theorem is proved in Section 4, using Henkin-like construction. Concluding remarks are given in Section 5.

2 Syntax and Semantics

Let Π be a nonempty countable set of atomic programs. The set of *labels* (programs) $a \in \mathcal{L}$ is the smallest set containing atomic programs which is closed under the following formation rules: if a and b are labels, then $a;b$, $a \cup b$, and a^* are also labels. Intuitively, $\langle \mathcal{L}, ;, \cup, ^* \rangle$ is interpreted as a countable Kleene algebra, where ; and $\cup$ are binary operations on $\mathcal{L}$, called *sequential composition* and *choice*, respectively, while * is an unary operation on $\mathcal{L}$, called *iteration*. We will denote labels by a, b and c, indexed if necessary.

2.1 Syntax

Let $\mathcal{Q}$ be the set of rational numbers. A first order language for probabilistic dynamic logic $PrDL^{fo}$ of the sort $\mathcal{L}$ is any language which contains:

- the set of variables $Var = \{x, y, z, \ldots\}$;
- for every integer $k \geq 0$, k-ary relation symbols $P_0^k, P_1^k, \ldots$, and k-ary function symbols $F_0^k, F_1^k, \ldots$;
- Boolean connectives $\neg$ and $\wedge$, $(\forall)$, comma and parentheses,
- for each $a \in \mathcal{L}$, modal operator $[a]$, and
- for each $a \in \mathcal{L}$ and $r \in [0,1] \cap \mathcal{Q}$, probabilistic operator $P^a_{\geq r}$.

The sets of *terms* and *atomic formulas* are defined in a usual way. The set of *formulas* is the smallest set that contains all atomic formulas, closed under the formation rules: if α and β are formulas and $a \in \mathcal{L}$, then $\neg\alpha$, $\alpha \wedge \beta$, $(\forall x)\alpha$, $[a]\alpha$ and $P^a_{\geq r}\alpha$ are also formulas. The intended meaning of the formula $P^a_{\geq r}\alpha$ is "the probability that α will be true after termination of the program a is at least r". We will denote the set of all formulas by $For(PrDL^{fo})$. *Sentences* are formulas without free variables.

In order to simplify notation, we use the classical abbreviations for the Boolean connectives $\vee$, $\rightarrow$ and $\leftrightarrow$, and for quantifier $\exists$. Also, we introduce the usual convention: $\top$ and $\bot$ are abbreviations for $\alpha \vee \neg\alpha$ and $\alpha \wedge \neg\alpha$, respectively. Moreover, for each $a \in \mathcal{L}$, the dual of $[a]$ is the modal operator $\langle a \rangle$, defined as $\langle a \rangle \alpha \equiv \neg[a]\neg\alpha$. The other probabilistic operators are defined as follows: $P^a_{<r}\alpha$ is $\neg P^a_{\geq r}\alpha$, $P^a_{\leq r}\alpha$ is $P^a_{\geq 1-r}\neg\alpha$, $P^a_{>r}\alpha$ is $\neg P^a_{\leq r}\alpha$, and $P^a_{=r}\alpha$ is $P^a_{\geq r}\alpha \wedge P^a_{\leq r}\alpha$.

For a set of formulas T and $a \in \mathcal{L}$, we will denote the set $\{[a]\alpha \mid \alpha \in T\}$ by $[a]T$. Also, if $a \in \mathcal{L}$ and $k \in \omega$, we define $[a]^k\alpha$ as follows: $[a]^0\alpha \equiv \alpha$ and $[a]^{k+1}\alpha \equiv [a]([a]^k\alpha)$.

Let K be any symbol not belonging to the language of $PrDL^{fo}$. In the axiomatization of the logic $PrDL^{fo}$, we will use a special class of formulas $For(K)$, defined as the smallest set with the following properties:

- $K \in For(K)$,
- if $\varphi \in For(K)$, $\alpha \in For(PrDL^{fo})$ and $a \in \mathcal{L}$, then $(\alpha \wedge \varphi), (\alpha \vee \varphi), (\alpha \rightarrow \varphi), [a]\varphi \in For(K)$.

We will denote the formulas from $For(K)$ by φ, ψ and θ, possibly with indices. Note that $For(K) \cap For(PrDL^{fo}) = \emptyset$. Every formula $\varphi \in For(K)$ has exactly one appearance of K. We will denote the formula obtained from φ by replacing K with a $PrDL^{fo}$-formula α by $\varphi(\alpha)$. Obviously, $\varphi(\alpha) \in For(PrDL^{fo})$.

2.2 Semantics

The logic $PrDL^{fo}$ uses possible world semantics. Namely, a *$PrDL^{fo}$-model* $\mathcal{M}$ as a special kind of Kripke model $\langle S, R, D, I, Pr\rangle$ where:

- S is a non-empty set of *possible worlds*,
- R assigns to each atomic program a a so called serial binary relation $R(a)$ on S (for all $s \in S$ there is $t \in S$ such that $s\,R(a)\,t$);
- D is a non empty domain;
- I associates an interpretation $I(s)$ to every $s \in S$, such that for all j and k:
  1. $I(s)(F_j^k)$ is a function from D^k to D;
  2. for every $s' \in S$, $I(s)(F_j^k) = I(s')(F_j^k)$, and
  3. $I(s)(P_j^k)$ is a k-ary relation on D;
- Pr associates to every $s \in S$ end every $a \in \mathcal{L}$ a probability space $Pr(s,a) = \langle H(s,a), \mu(s,a)\rangle$ such that:
  - $H(s,a)$ is an algebra of subsets of $W(s,a) = \{s' \in S \mid sR(a)s'\}$, (i.e., it contains $W(s,a)$ and it is closed under complements and finite union);
  - $\mu(s,a) : H(s,a) \longrightarrow [0,1]$ is a finitely additive probability measure:
    - $\mu(s,a)(W(s,a)) = 1$,
    - $\mu(s,a)(A \cup B) = \mu(s,a)(A) + \mu(s,a)(B)$, if $A \cap B = \emptyset$.

Note that we use fixed domain models with rigid terms. This assumption actually the objectual interpretation for first order modal logics, and it is necessary restriction if want to preserve validity of all first-order axioms [4].

2.3 Satisfiability Relation

A *variable valuation* v assigns a function $v(s) : Var \longrightarrow D$ to every possible world s, i.e., $v(s)(x) \in D$. If $s \in S$, and $d \in D$, then $v[d/x]_s$ is the valuation identical to the valuation v, with the exception that $v[d/x]_s(s)(x) = d$. The value of a term t in a world s with respect to v (denoted by $I(s)(t)_v$) is defined recursively:

- if t is a variable x, then $I(s)(x)_v = v(s)(x)$,
- if $t = F_j^k(t_1, \ldots, t_k)$, then $I(s)(t)_v = I(s)(F_j^k)(I(s)(t_1)_v, \ldots, I(s)(t_k)_v)$.

The satisfiability of a formula α in a world s of a model $\mathcal{M}$ under a valuation v, denoted by $(\mathcal{M}, s, v) \models \alpha$, is defined as follows:

- $(\mathcal{M}, s, v) \models P_j^k(t_1, \ldots, t_k)$ iff $\langle I(s)(t_1)_v, \ldots, I(s)(t_k)_v\rangle \in I(s)(P_j^k)$,
- $(\mathcal{M}, s, v) \models \alpha \wedge \beta$ iff $(\mathcal{M}, s, v) \models \alpha$ and $(\mathcal{M}, s, v) \models \beta$,
- $(\mathcal{M}, s, v) \models \neg\alpha$ iff $(\mathcal{M}, s, v) \not\models \alpha$,
- $(\mathcal{M}, s, v) \models (\forall x)\alpha$ iff $(\mathcal{M}, s, v[d/x]_s) \models \alpha$, for every $d \in D$,

- For any atomic label (atomic program) $a \in \Pi$ we say that $(\mathcal{M}, s, v) \models [a]\alpha$ iff $(\mathcal{M}, s', v) \models \alpha$ for all s' such that $s\, R(a)\, s'$;
- $(\mathcal{M}, s, v) \models [a;b]\alpha$ iff $(\mathcal{M}, s, v) \models [a][b]\alpha$;
- $(\mathcal{M}, s, v) \models [a \cup b]\alpha$ iff $(\mathcal{M}, s, v) \models [a]\alpha$ and $(\mathcal{M}, s, v) \models [b]\alpha$;
- $(\mathcal{M}, s, v)[a^*]\alpha$ iff $(\mathcal{M}, s, v) \models [a]^n\alpha$ for all $n \in \omega$;
- $(\mathcal{M}, s, v) \models P^a_{\geq r}\alpha$ iff $\mu(s, a)(\{s' \in W(s, a) \mid (\mathcal{M}, s', v) \models \alpha\}) \geq r$.

If $(\mathcal{M}, s, v) \models \alpha$ holds for every valuation v, we write $(\mathcal{M}, s) \models \alpha$. Also, we say that a sentence α is *satisfiable* if there is a world s in a model $\mathcal{M}$ such that $(\mathcal{M}, s) \models \alpha$. A set T of sentences is satisfiable if there is a world s in a model $\mathcal{M}$ such that $(\mathcal{M}, s) \models \alpha$ holds for every $\alpha \in T$.

The possible problem is that for the set $\{s' \in W(s, a) \mid (\mathcal{M}, s', v) \models \alpha\}$ might not belong to $H(s, a)$. To overcome this problem, we will consider only so-called *measurable models*, i.e., models with the following property:

$$\{s' \in W(s, a) \mid (\mathcal{M}, s', v) \models \alpha\} \in H(s, a), \text{ for every } \alpha \in For(PrDL^{fo}).$$

Remark 1. Though it may seem that the introduced semantics deviates from the standard one, it faithfully follows it. Namely, if $R(a)$ and $R(b)$ are serial, then trivially $R(b) \circ R(a)$, $R(a) \cup R(b)$ and $\bigcup_{n\in\omega} R(a)^n$ are also serial. Also, it is well known that for the standard semantics of dynamic logics the following holds

- $(\mathcal{M}, s, v) \models [a;b]\alpha$ iff $(\mathcal{M}, s, v) \models [a][b]\alpha$;
- $(\mathcal{M}, s, v) \models [a \cup b]\alpha$ iff $(\mathcal{M}, s, v) \models [a]\alpha$ and $(\mathcal{M}, s, v) \models [b]\alpha$;
- $(\mathcal{M}, s, v)[a^*]\alpha$ iff $(\mathcal{M}, s, v) \models [a]^n\alpha$ for all $n \in \omega$.

Hence, relations $R(a)$ and the corresponding satisfiability of $[a]$-formulas for the non atomic labels are uniquely determined by the atomic relations $R(b)$ ($b \in \Pi$ is an atomic label) and the corresponding satisfiability of $[b]$-formulas.

Consequently, our semantics coincides with the standard semantics for dynamic logics.

2.4 Tests

Dynamic logics are usually presented with an additional set of labels of the form $\alpha?$, called *tests*. Then models has the additional restriction for R:

[R?] $R(\alpha?) = \{(s, s) \in S \mid (\mathcal{M}, s) \models \alpha\})$.

The corresponding axiomatization contains the axiom

[Ax?] $[\alpha?]\beta \leftrightarrow (\alpha \rightarrow \beta)$.

By $[Ax?]$, this extension of the language is obviously extension by definition. Thus, our paper will not deal with those labels. The results of the paper would hold (without any changes in the proofs), if we include those formulas in the language, $[R?]$ as a semantical clause and $[Ax?]$ in the axiomatization.

3 The Axiomatization of $PrDL^{fo}$

Axiom schemas

A1. the instantiations of the propositional tautologies
A2. $(\forall x)(\alpha \to \beta) \to (\alpha \to (\forall x)\beta)$, where x is not free in α
A3. $(\forall x)\alpha(x) \to \alpha(t/x)$, where $\alpha(t/x)$ is obtained by substituting all free occurrences of x in $\alpha(x)$ by the term t which is free for x in $\alpha(x)$
A4. $[a](\alpha \to \beta) \to ([a]\alpha \to [a]\beta)$
A5. $[a;b]\alpha \leftrightarrow [a][b]\alpha$
A6. $[a \cup b]\alpha \leftrightarrow ([a]\alpha \wedge [b]\alpha)$
A7 . $[a^*]\alpha \to [a]^n\alpha, \quad n \in \omega,$
A8. $[a]\alpha \to \langle a\rangle\alpha$
A9. $P^a_{\geq 0}\alpha$
A10. $[a]\alpha \to P^a_{\geq 1}\alpha$
A11. $P^a_{\leq s}\alpha \to P^a_{<t}\alpha$, $t > s$
A12. $P^a_{<s}\alpha \to P^a_{\leq s}\alpha$
A13. $(P^a_{\geq s}\alpha \wedge P^a_{\geq r}\beta \wedge P^a_{\geq 1}(\neg\alpha \vee \neg\beta)) \to P^a_{\geq \min(1,s+r)}(\alpha \vee \beta)$
A14. $(P^a_{\leq s}\alpha \wedge P^a_{<r}\beta) \to P^a_{<s+r}(\alpha \vee \beta)$, $s + r \leq 1$
A15. $(\forall x)[a]\alpha(x) \to [a](\forall x)\alpha(x)$

Inference rules

R1. from $\{\alpha, \alpha \to \beta\}$ infer β
R2. from α infer $(\forall x)\alpha$
R3. from a theorem α infer $[a]\alpha$
R4. from the set of premises

$$\{\beta \to \varphi(P^a_{\geq r-\frac{1}{n}}\alpha) \mid n \in \omega, r - \frac{1}{n} \geq 0\}$$

infer $\beta \to \varphi(P^a_{\geq r}\alpha)$
R5. from the set of premises

$$\{\beta \to \varphi([a]^n\alpha) \mid n \in \omega\}$$

infer $\beta \to \varphi([a^*]\alpha)$

Let us briefly discuss the above axioms and inference rules. The axiom system can be divided into three parts. The first part characterizes the classical first-order logic (the axioms A1–A3, together with the rules R1 and R2). The axioms A4–A6 are the usual axioms for dynamic logic [12]. The first one is modal K-axiom, while the other two axioms characterize sequential composition and choice. A7 captures the property that $R(a)$ is serial. The axioms A9, A11–A14 are a-variants of the axioms for probabilistic reasoning, used in our previous research [8, 9]. The axioms A10 connects probabilistic and dynamic operators. Finally, the axiom A15 is the well known Barcan formula.

The rule R1 is Modus Ponens, and the rule R2 is Goedel's Generalization. The rule R3 is the restricted of local modal Necessitation. The rules R4 and R5 are

infinitary inference rules. The first one is a modification of so called Archimedean rule presented in [8, 9]. Its purpose is to forbid nonstandard probabilistic values of the formulas. The second one, together with the axiom A7, characterizes iteration operator. The infinitary rules are of the form presented above because of the three reasons:

– the rules generalize the following two rules which are syntactical counterparts of important semantical properties of our logic:

$$\frac{\{P^a_{\geq r-\frac{1}{n}}\alpha \mid n \in \omega, r-\frac{1}{n} \geq 0\}}{P^a_{\geq r}\alpha},$$

$$\frac{\{[a]^n\alpha \mid n \in \omega\}}{[a^*]\alpha}.$$

They can be obtained from R4 and R5, respectively, by setting $\beta = \top$ and $\varphi = K$.
– The implicative form of the rules is standard trick that allows easy proof of Deduction theorem for infinitary logic.
– Generalization to all implicative formulas obtained by $For(K)$ allows the proof of Theorem 4, which is essential for the proof of Completeness theorem.

We say that α is a *theorem* of the logic $PrDL^{fo}$, and write $\vdash_{PrDL^{fo}} \alpha$, if there is an at most countable sequence of formulas $\alpha_0, \alpha_1, \ldots, \alpha$, such that every α_i is an axiom, or it is derived from the preceding formulas by an inference rule. A formula α is *deducible* from a set T of formulas ($T \vdash_{PrDL^{fo}} \alpha$) if there is an at most countable sequence of formulas $\alpha_0, \alpha_1, \ldots, \alpha$, such that every α_i is an axiom or a formula from T, or it is derived from the preceding formulas by an inference rule, with exception that the inference rule R3 can be applied to theorems only. The corresponding sequence of formulas $\alpha_0, \alpha_1, \ldots, \alpha$ is the *proof* of $T \vdash_{PrDL^{fo}} \alpha$. A set T of sentences is *consistent* if there is at least one formula which is not deducible from T. T is *inconsistent* iff it is not consistent. In the rest of the paper, we will omit $PrDL^{fo}$ in $\vdash_{PrDL^{fo}}$ because it's clear from context. Note that the length of inference may be any successor ordinal lesser than the first uncountable ordinal ω_1.

In the proof of Completeness theorem, we will use the special class of sentences, called saturated theories. A set T of sentences is *maximal* if for every sentence α, either $\alpha \in T$ or $\neg\alpha \in T$. A set T of sentences is *saturated* if it is consistent, maximal and satisfies the condition: if $\neg(\forall x)\alpha(x) \in T$, then for some term t, $\neg\alpha(t) \in T$.

As it is usually the case in logic, the soundness part of the completeness theorem (every syntactical consequence is also a semantical consequence of any given theory) can be verified by the straightforward induction on the length of inference. The same is true for the deduction theorem, so we will omit the corresponding proofs.

Theorem 2 (Soundness). *If T is a set of sentences and α is a sentence, then $T \vdash \alpha$ implies $T \models \alpha$.*

Theorem 3 (Deduction theorem). *Let T be a set of sentences and let α be a sentence. Then $T \cup \{\alpha\} \vdash \beta$ implies $T \vdash \alpha \to \beta$.*

The following theorem will play the essential role in the proof of Completeness theorem.

Theorem 4. *Let T be a set of sentences and let $T \vdash \alpha$. Then $[a]T \vdash [a]\alpha$.*

Proof. We will use the induction on the depth of the derivation of α from T. The cases when we apply the inference rule R1 is trivial, as well as the case when we use the rule R2, because it can be applied to theorems only. Suppose that $T \vdash (\forall x)\alpha$ is obtained from $T \vdash \alpha$ by the inference rule R2. Then we have
$T \vdash \alpha$ (by the assumption),
$[a]T \vdash [a]\alpha$ (by the induction hypothesis),
$[a]T \vdash (\forall x)[a]\alpha$ (by R2),
$[a]T \vdash [a](\forall x)\alpha$ (by A15).
Assume that $T \vdash \beta \to \varphi([a^*]\alpha)$ is obtained by R5. Then we have
$T \vdash \beta \to \varphi([a]^n\alpha)$ (by the assumption),
$[a]T \vdash [a](\beta \to \varphi([a]^n\alpha))$ (by the induction hypothesis),
$[a]T \vdash [a](\beta \to \varphi([a^*]\alpha))$ (by R5, applied on $\psi(K) = \top \to [a](\beta \to \varphi(K))$).
The case when we apply R4 can be solved in a similar way. □

Theorem 5.

1. $\vdash [a^*]\alpha \to (\alpha \wedge [a][a^*]\alpha)$
2. $\vdash [a^*](\alpha \to [a]\alpha) \to (\alpha \to [a^*]\alpha)$

Proof. $(\mathcal{M}, s) \models [b]\psi([a]^n\alpha)$

1. $[a^*]\alpha \vdash [a]^n\alpha$ for all $n \in \omega$, so $[a^*]\alpha \vdash \alpha$ and $[a^*]\alpha \vdash [a][a]^n\alpha$ for all $n \in \omega$. By R5 ($\varphi = \top \to [a]K$) we obtain $[a^*]\alpha \vdash [a][a^*]\alpha$. Now result follows from Theorem 3.
2. $[a^*](\alpha \to [a]\alpha) \wedge \alpha \vdash [a]^n\alpha$ for all $n \in \omega$ (by A4), so $[a^*](\alpha \to [a]\alpha) \wedge \alpha \vdash [a^*]\alpha$ (by R4). By Theorem 3 we have $[a^*](\alpha \to [a]\alpha) \vdash \alpha \to [a^*]\alpha$ and $\vdash [a^*](\alpha \to [a]\alpha) \to (\alpha \to [a^*]\alpha)$. □

The formulas in 1. and 2. from the previous theorem are axioms in standard axiomatizations of dynamic logic [12]. Thus, all theorems of standard dynamic logic are also theorem of (dynamic part of) $PrDL^{fo}$.

4 Completeness

Completeness is proved in three steps. First we extend any consistent set of sentences T to a saturated set T^*. Then we use T^* we construct the canonical model $\mathcal{M}^*$. Finally, we show that $\mathcal{M}^*$ is a model of T.

Let T be a consistent set of $PrDL^{fo}$-sentences. Let $\{\alpha_i \mid i \in \omega\}$ be an enumeration of all sentences of $PrDL^{fo}$ and C a countably infinite set of constants symbols not belonging to the language of $PrDL^{fo}$. We define a completion T^* of T in the extended language recursively:

1. $T_0 = T$.
2. For every $i \in \omega$,
   (a) If $T_i \cup \{\alpha_i\}$ is consistent, then $T_{i+1} = T_i \cup \{\alpha_i\}$.
   (b) Otherwise, if α_i is of the form $\beta \to \varphi(P^a_{\geq r}\alpha)$, then $T_{i+1} = T_i \cup \{\beta \to \neg\varphi(P^a_{\geq r-\frac{1}{k}}\alpha)\}$ for some k such that T_{i+1} is consistent.
   (c) Otherwise, if α_i is of the form $\beta \to \varphi([a^*]\alpha)$, then $T_{i+1} = T_i \cup \{\beta \to \neg\varphi([a]^m\alpha)\}$ for some m such that T_{i+1} is consistent.
   (d) Otherwise, if α_i is of the form $\neg(\forall x)\beta(x)$, then $T_{i+1} = T_i \cup \{\neg\beta(c)\}$ for some $c \in C$ such that T_{i+1} is consistent.
   (e) Otherwise, $T_{i+1} = T_i$.
3. $T^* = \bigcup_{i=0}^{\infty} T_i$.

Lemma 6. *T^* is well defined.*

Proof. We must show that m, k and c introduced in step 2 exist. Suppose that there is no such an m. Then $T_i \cup \{\beta \to \neg\varphi([a]^m\alpha)\}$ is inconsistent for all m, so, by Theorem 3 $T_i \vdash \neg\beta \to \neg\varphi([a]^m\alpha)$ for all m. Then, by propositional reasoning, $T_i \vdash \beta \to \varphi([a]^m\alpha)$ for all m. Thus, by R5, $T_i \vdash \beta \to \varphi([a^*]\alpha)$, which contradicts the assumption. Similarly, if $T_i \cup \{\beta \to \neg\varphi(P^a_{\geq r-\frac{1}{k}}\alpha)\} \vdash \bot$ for all k, by Theorem 3 and propositional reasoning we obtain $T_i \vdash \beta \to \varphi(P^a_{\geq r-\frac{1}{k}}\alpha)$ for all k. From R4 we obtain $T_i \vdash \beta \to \varphi(P^a_{\geq r}\alpha)$; a contradiction. The proof of existence of c is standard. □

Theorem 7. *Let T be a consistent set of sentences in the language of $PrDL^{fo}$ and C a countably infinite set of new constant symbols. Then T can be extended to a saturated set T^* in the extended language.*

Proof. Each T_i is consistent. Suppose that there is a sentence α such that both $\alpha \notin T^*$ and $\neg\alpha \notin T^*$. If $\alpha = \alpha_i$ and $\neg\alpha = \alpha_j$., by construction of T^* we obtain $T_i \vdash \neg\alpha$ and $T_j \vdash \alpha$. Then $T_{\max\{i,j\}} \vdash \alpha \wedge \neg\alpha$, so $T_{\max\{i,j\}}$ is inconsistent; a contradiction. So, T^* is maximal. We will prove that T^* is deductively closed, i.e., that $T^* \vdash \alpha$ implies $\alpha \in T^*$. Any axiom is consistent with any consistent set, so it is enough to prove that T^* is closed under the inference rules. The only possible problem is with the infinitary rules.

In order to prove closeness under R4, we must show that whenever $\{\beta \to \varphi(P^a_{\geq r-\frac{1}{n}}\alpha) \mid n \in \omega, r - \frac{1}{n} \geq 0\} \subseteq T^*$, also $\beta \to \varphi(P^a_{\geq r}\alpha) \in T^*$. Suppose not. By maximality of T^* we have $\neg(\beta \to \varphi(P^a_{\geq r}\alpha)) \in T^*$, so $\beta \in T^*$ and $\neg\varphi(P^a_{\geq r}\alpha) \in T^*$. Consequently, $\{\varphi(P^a_{\geq r-\frac{1}{n}}\alpha) \mid n \in \omega, r - \frac{1}{n} \geq 0\} \subseteq T^*$. Also, if $\alpha_i = \beta \to \varphi(P^a_{\geq r}\alpha)$, then there is k such that $\beta \to \neg\varphi(P^a_{\geq r-\frac{1}{k}}\alpha) \in T_{i+1}$. If $\beta = \alpha_j$ then $T_{\max\{i+1,j\}} \vdash \neg\varphi(P^a_{\geq r-\frac{1}{k}}\alpha)$. Let $\alpha_l = \varphi(P^a_{\geq r-\frac{1}{k}}\alpha)$. Then $T_{\max\{i+1,j,l+1\}} \vdash \neg\varphi(P^a_{\geq r-\frac{1}{k}}\alpha) \wedge \varphi(P^a_{\geq r-\frac{1}{k}}\alpha)$; a contradiction.

Now we will prove the closeness under R5. Suppose that $\{\beta \to \varphi([a]^n\alpha) \mid n \in \omega\} \subseteq T^*$ and $\beta \to \varphi([a^*]\alpha) \notin T^*$. By maximality of T^* we have $\neg(\beta \to \varphi([a^*]\alpha)) \in T^*$. By the construction of T^* there are i and m so that $\neg(\beta \to$

$\varphi([a^*]\alpha)), (\beta \to \neg\varphi([a]^m\alpha)) \in T_i$. Note that $T_i \vdash \beta$ and $T_i \vdash \neg\varphi([a]^m\alpha))$. If $\alpha_j = \beta \to \varphi([a]^m\alpha)$, then $T_{\max\{i,j+1\}} \vdash \varphi([a]^m\alpha) \wedge \neg\varphi([a]^m\alpha)$, a contradiction.

Thus, T^* is deductively closed. If it is inconsistent, then there is α such that $T^* \vdash \alpha \wedge \neg\alpha$. Then there is i such that $\alpha \wedge \neg\alpha \in T_i$, a contradiction. The step 2(d) of the construction guarantees that T^* is saturated. □

A canonical model $\mathcal{M}^* = \langle S, R, D, I, Pr\rangle$ is defined in the following way:

- S is the set of all saturated theories;
- For the atomic label a let $s\,R(a)\,t$ iff $\{\alpha \mid [a]\alpha \in s\} \subseteq t$;
- D is the set of all variable-free $PrDL^{fo}$-terms;
- $I(s)$ is an interpretation such that:
  - for every function symbol F_j^k, $I(s)(F_j^k)$ is a function from D^k to D such that for all variable-free $PrDL^{fo}$-terms $t_1, \ldots, t_k$, $I(s)(F_j^k) : \langle t_1, \ldots, t_k\rangle \mapsto F_j^k(t_1, \ldots, t_k)$;
  - for every relation symbol P_j^k, $I(s)(P_j^k) = \{\langle t_1, \ldots, t_k\rangle : t_1, \ldots, t_k$ are variable-free $PrDL^{fo}$-terms in $P_j^k(t_1, \ldots, t_k) \in s\}$;
- the probability space $Pr(s, a) = \langle H(s, a), \mu(s, a)\rangle$ is defined as follows:
  - $H(s, a) = \{[\alpha]_s^a \mid \alpha \in For(PrDL^{fo})\}$, $[\alpha]_s^a = \{s' \in W(s, a) \mid s' \vdash \alpha\}$;
  - $\mu(s, a)([\alpha]_s^a) = \sup\{r \in \mathcal{Q} \cap [0, 1] \mid s \vdash P_{\geq r}^a\alpha\}$.

The following theorem is a rather straightforward modification of the corresponding theorem presented in [8], so we will omit its proof.

Theorem 8. *$\mathcal{M}^*$ is a model.*

Theorem 9 (Strong completeness theorem). *Every consistent set T of sentences is satisfiable.*

Proof. Let $\mathcal{M}^*$ be the model constructed above We can show that for every sentence α, $(\mathcal{M}^*, s) \models \alpha$ iff $\alpha \in s$, using the induction on the complexity of α. If α is an atomic sentence,it follows from the definition of I. The cases when formulas are negations and conjunctions can be proved as usual. For the proof when α is of the forms $(\forall x)\beta$ or $P_{\geq r}^a\beta$, we refer the reader to [1] and [9].

Let $\alpha = [a]\beta$. Suppose that $[a]\beta \in s$. Then $\beta \in s'$ for each s' such that $sR(a)s'$. By the induction hypothesis we obtain $(\mathcal{M}^*, s') \models \beta$ for each s' such that $sR(a)s'$, so $(\mathcal{M}^*, s) \models [a]\beta$.

Conversely, let $(\mathcal{M}^*, s) \models [a]\beta$ and $[a]\beta \notin s$. Let $A = \{\alpha \mid [a]\alpha \in s\}$. Suppose that $A \cup \{\neg\beta\}$ is inconsistent. By Theorem 3, $A \vdash \beta$, and by Theorem 4 $[a]A \vdash [a]\beta$. Since $[a]A \subseteq s$, we have $s \vdash [a]\beta$, so $[a]\beta \in s$ (by maximality of s), a contradiction. Thus, $A \cup \{\neg\beta\}$ is consistent. By Theorem 7 there exist $t \in S$ such that $A \cup \{\neg\beta\} \subseteq t$. Note that $sR(a)t$. Moreover, $\neg\beta \in t$, and by the induction hypothesis we obtain $(\mathcal{M}^*, t) \models \beta$, which contradicts the assumption that $(\mathcal{M}^*, s) \models [a]\beta$.

By Theorem 7, the consistent set T can be extended to a saturated set $T^* \in S$. Since T^* is satisfied in $\mathcal{M}$, T is satisfiable. □

5 Summary and Conclusions

As it is the case for the classical first-order logic, our formalization of the first-order dynamic probability logic is undecidable. Investigation of decidable fragments can be of particular importance for various applications.

If we weaken the connection between the $[a^*]$ and $[a]$ by the statement "$R(a^*)$ is a reflexive and transitive relation that extends $R(a)$", then the corresponding pseudo-dynamic part would be finitely axiomatizable. In addition, if all probability function are restricted to some fixed finite range, than such logic can be completely axiomatized by finitary system.

Acknowledgements. This work is partially supported by Serbian Ministry of Education, Science and Technological Development through grants III044006, ON174026, III041013 and TR36001.

References

1. Doder, D., Ognjanović, Z., Marković, Z.: An Axiomatization of a First-order Branching Time Temporal Logic. Journal of Universal Computer Science 16(11), 1439–1451 (2010)
2. Feldman, Y.A., Harel, D.: A Probabilistic Dynamic Logic. Journal of Computer and System Sciences 28, 193–215 (1984)
3. Guelev, D.P.: A Propositional Dynamic Logic with Qualitative Probabilities. Journal of Philosophical Logic 28, 575–605 (1999)
4. Halpern, J.Y.: An Analysis of First-order Logics of Probability. Artificial Intelligence 46, 311–350 (1990)
5. Kooi, B.P.: Probabilistic Dynamic Epistemic Logic. Journal of Logic, Language and Information 12, 381–408 (2003)
6. Ognjanović, Z., Rašković, M.: Some First-order Probability Logics. Theoretical Computer Science 247(1-2), 191–212 (2000)
7. Ognjanović, Z.: Completeness Theorem for a First Order Linear-time Logic. Publications de L'institut Mathématique, Nouvelle Série 69(83), 1–7 (2001)
8. Ognjanović, Z.: Discrete Linear-time Probabilistic Logics: Completeness, Decidability and Complexity. Journal of Logic and Computation 16(2), 257–285 (2006)
9. Ognjanović, Z., Doder, D., Marković, Z.: A Branching Time Logic with Two Types of Probability Operators. In: Benferhat, S., Grant, J. (eds.) SUM 2011. LNCS, vol. 6929, pp. 219–232. Springer, Heidelberg (2011)
10. Renardel de Lavalette, G., Kooi, B., Verburgge, R.: Strong Completeness and Limited Canonicity for PDL and Similar Logics. Journal of Logic, Language and Information 17(1), 69–87 (2008)
11. Sack, J.: Extending Probabilistic Dynamic Epistemic Logic. Synthese 169(2), 241–257 (2009)
12. Stirling, C.: Modal and temporal logic. In: Handbook of Logic in Computer Science, vol. 2, pp. 477–563 (1992)

Selecting Source Behavior in Information Fusion on the Basis of Consistency and Specificity

Frédéric Pichon[1], Sébastien Destercke[2], and Thomas Burger[3]

[1] Thales Research and Technology, Campus Polytechnique,
1 avenue Augustin Fresnel, 91767 Palaiseau cedex, France
Frederic.Pichon@thalesgroup.com
[2] CNRS, UMR Heudiasyc, Centre de Recherches de Royallieu, Compiègne, France
Sebastien.Destercke@hds.utc.fr
[3] CNRS, iRTSV (FR3425), CEA / iRTSV / BGE, INSERM (U1038),
Université de Grenoble, France
Thomas.Burger@cea.fr

Abstract. Combining pieces of information provided by several sources without prior knowledge about the behavior of the sources is an old yet still important and rather open problem in belief function theory. In this paper, we propose a general approach to select the behavior of sources, based on two cornerstones of information fusion that are the notions of specificity and consistency. This approach is framed in a recently introduced and general fusion scheme that allows a wide range of assumptions on the sources. In the process, we are also led to generalize a recently introduced measure of conflict to all Boolean connectives. Eventually, we show that our approach generalizes some important existing information fusion strategies.

Keywords: Dempster-Shafer theory, Information fusion, Consistency, Specificity, Conflict.

1 Introduction

Determining the actual value taken by a variable of interest from information provided by several sources is a central problem in many information systems and has received much attention in belief function theory [16,19]. As argued in [18,14,3], such a task involves necessarily to make some (possibly uncertain) assumptions about the dependence and the behavior, *e.g.*, the relevance and truthfulness [14], of the sources of information. A main concern in information fusion is thus to find what assumption to make about the sources. In this paper, we focus on the problem of finding appropriate source behaviors and assume sources to be independent.

When some training data are available, one may resort to some learning procedures to estimate the behavior of the sources (see, *e.g.*, [7,11,6]). When there is no previous experience with the sources (the case in the present paper), then the selection of an appropriate assumption about source behaviors needs to be based on other considerations.

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 473–484, 2013.

A first interesting criterion for that choice is the consistency of the knowledge induced on the variable of interest by a given assumption, as suggested by the large body of literature on conflict management (see, *e.g.*, [18,9]). Indeed, it is common in the theory of belief functions to question the behavioral assumptions of the (unnormalized) Dempster's rule [1,16], *i.e.*, that the sources are truthful and relevant [14], when the conflict or inconsistency [2] resulting from its application is too high.

A second natural criterion is the specificity of the induced knowledge. Indeed, there exist assumptions on the sources that, despite their ensuring consistency, are not so often made because they yield poorly informative conclusions. This is the case for instance of the assumption of truthful sources, of which at least one is relevant (the disjunctive rule [4,17] corresponds to this assumption [14]).

There might be other relevant criteria to compare assumptions on sources, such as considering a kind of minimal change principle (see, for example [10]) where an assumption could be chosen on the basis of the closeness (in the sense of some distance [8]) of the induced knowledge with respect to the knowledge induced by some reference assumption (e.g., truthful and relevant).

In this paper, we propose an approach to select the behavior of sources based on the notions of specificity and consistency (as they are the most classical goals to be reached by a fusion process). This approach is framed in the scheme of Pichon *et al.* [14], a very general fusion framework that allows making a wide range of assumptions on the sources. In the process, we are led to extend some results presented by Destercke and Burger [2] on conflict measurement. We also show that our approach generalizes some important existing information fusion strategies. We follow a step-wise presentation, first expressing the notions of consistency and specificity in Pichon *et al.* framework in the case of a single source (Section 3), and then in the case of multiple sources (Section 4). We then describe our approach, and provide some important examples of its application (Section 5). Background material is presented in Section 2. Due to space limitation, proofs are omitted.

2 Preliminaries

In this section, we provide first the necessary concepts about belief function theory and then we recall the formal setting of Pichon *et al.* [14].

2.1 Necessary Concepts of Belief Function Theory

In this paper, we assume the beliefs held by an agent about the actual value taken by a given variable $\mathbf{x}$ defined on a finite domain $\mathcal{X}$, to be modeled using belief functions [16,19] and to be represented using associated mass functions. Formally, a mass function $m^{\mathcal{X}}$ on $\mathcal{X}$ is a probability distribution on the power set $2^{\mathcal{X}}$, hence $\sum_{A \subseteq \mathcal{X}} m^{\mathcal{X}}(A) = 1$. The probability allocation $m^{\mathcal{X}}(A)$ may be understood as the weight given to the assumption that the agent knows that the value of the variable of interest lies somewhere in set A, and nothing more

specific [5], or as the probability that the agent supplies information item $\mathbf{x} \in A$ [14]. Each $A \subseteq \mathcal{X}$ such that $m^{\mathcal{X}}(A) > 0$ is called a focal set of the mass function. $\mathcal{F}$ denotes the set of focal sets of $m^{\mathcal{X}}$.

From the mass function are usually defined two uncertainty measures, the belief and plausibility measures, which respectively reads for an event $A \subseteq \mathcal{X}$:

$$Bel(A) = \sum_{\emptyset \neq B \subseteq A} m^{\mathcal{X}}(B) \text{ and } Pl(A) = \sum_{B \cap A \neq \emptyset} m^{\mathcal{X}}(B).$$

That is, Bel is the sum of masses of sets that implies A, Pl the sum of masses of sets that are consistent with A. The contour function [16] $pl^{\mathcal{X}} : \mathcal{X} \to [0, 1]$ associated to a mass function $m^{\mathcal{X}}$ is defined by $pl^{\mathcal{X}}(x) = Pl^{\mathcal{X}}(\{x\})$.

There exist several ways to compare the informational contents of belief functions (see, e.g., [5]). In particular, the specialization ordering (the most natural extension of set inclusion) compares belief functions in terms of specificity: $m_1^{\mathcal{X}}$ is a specialization of $m_2^{\mathcal{X}}$, which we denote by $m_1^{\mathcal{X}} \sqsubseteq m_2^{\mathcal{X}}$, if and only if $m_1^{\mathcal{X}}$ can be obtained from $m_2^{\mathcal{X}}$ by transferring each mass $m_2^{\mathcal{X}}(A)$ to subsets of A.

Many combination rules have been proposed for belief functions [18]: the most common is the unnormalized Dempster's rule (or conjunctive rule), denoted by $\textcircled{\cap}$. The mass function $m_{1\textcircled{\cap}2}^{\mathcal{X}}$ resulting from its application on $m_1^{\mathcal{X}}$ and $m_2^{\mathcal{X}}$ is:

$$m_{1\textcircled{\cap}2}^{\mathcal{X}}(A) = \sum_{B \cap C = A} m_1^{\mathcal{X}}(B)\, m_2^{\mathcal{X}}(C), \quad \forall A \subseteq \mathcal{X}. \tag{1}$$

The disjunctive rule $\textcircled{\cup}$ [4,17] is obtained by simply replacing $\cap$ with $\cup$ in (1).

2.2 Source Behavioral States

The setting considered by Pichon *et al.* [14] is the following. Assume an agent wants to know the actual value taken by $\mathbf{x}$ based on testimonies provided by several sources of information identified as $\mathfrak{s}_i$, $1 \leq i \leq K$. These testimonies can be of several forms: a value $x_i \in \mathcal{X}$, a set $A_i \in \mathcal{X}$, a probability distribution p_i on $\mathcal{X}$, or in the most general form a mass function $m_i^{\mathcal{X}}$ on $\mathcal{X}$. In order to be able to interpret those testimonies, the agent must have some knowledge about the behavioral state (referred to as *meta-knowledge* in [14]) of the sources. In the approach of Pichon *et al.*, the possible elementary behavioral states of a source $\mathfrak{s}_i$ are formalized as a set $\mathcal{H}^i = \{h_1^i, \ldots, h_N^i\}$. The set of elementary joint states on sources is therefore the Cartesian product $\mathcal{H}^{1:K} := \times_{i=1}^{K} \mathcal{H}^i$. The state space $\mathcal{H}^i$ can be very general [14] and may include being unreliable, lying, being approximatively informed, etc. Two common assumptions for which we will use specific notations are the assumptions that a source $\mathfrak{s}_i$ is relevant (R^i) or not ($\neg R^i$), and truthful (T^i) or not ($\neg T^i$). Together, they form the space of possible states $\mathcal{H}^i = \{(T^i, R^i), (T^i, \neg R^i), (\neg T^i, R^i), (\neg T^i, \neg R^i)\}$. Like the testimonies provided by the sources, the meta-knowledge of the agent can be of several form, the most general one being a mass function defined over $\mathcal{H}^{1:K}$.

In the following, we detail how consistency and specificity can be characterized when using this setting, and how such characterizations can be used to select a particular piece of meta-knowledge.

3 Consistency and Specificity: Single Source

We start by characterizing consistency and specificity in the simple case where a single source provides information.

3.1 Crisp Testimony and Sure Meta-knowledge

The simplest situation is a source $\mathfrak{s}$ delivering a testimony of the form $\mathbf{x} \in A$ with $A \subseteq \mathcal{X}$, and being known to be in a state $h \in \mathcal{H}$, with $\mathcal{H}$ the state space of the source. The testimony $\mathbf{x} \in A$ should then be modified according to this state [14]. This transformation can be encoded by a multivalued mapping $\Gamma_A : \mathcal{H} \to \mathcal{X}$, where $\Gamma_A(h)$ indicates how to interpret the piece of information $\mathbf{x} \in A$ for each possible state h of the source. For instance, if $\mathcal{H} = \{(T,R),(T,\neg R),(\neg T,R),(\neg T,\neg R)\}$ are the possible states of the source, we have for all $A \subseteq \mathcal{X}$

$$\Gamma_A(R,T) = A, \Gamma_A(\neg R,T) = \mathcal{X}, \Gamma_A(R,\neg T) = A^c, \Gamma_A(\neg R,\neg T) = \mathcal{X}, \qquad (2)$$

with A^c the complement of A. Eqs. (2) translate that if $\mathfrak{s}$ is considered not relevant, it does not bring any information, while if it is considered not truthful, it declares the opposite of what it knows to be true – this corresponds to the crudest form of non-truthfulness, other forms are discussed in Pichon [12]. If the knowledge about the source state is imprecise and given by $H \subseteq \mathcal{H}$, then the transformation is the image $\Gamma_A(H) := \bigcup_{h \in H} \Gamma_A(h)$ of H by Γ_A.

Destercke and Burger [2] consider that any piece of knowledge $\mathbf{x} \in A$ about a variable $\mathbf{x}$ is consistent if $A \neq \emptyset$, and inconsistent otherwise. This extends easily to the current framework, a transformed testimony yielding a consistent piece of knowledge on $\mathcal{X}$ when $\Gamma_A(H) \neq \emptyset$, in which case $\mathbf{x} \in A$ is said H-consistent, and an inconsistent piece of knowledge when $\Gamma_A(H) = \emptyset$. We may then adapt the measure of consistency introduced in [2] to measure H-consistency as the degree $\phi_H : 2^{\mathcal{X}} \to \{0,1\}$ such that

$$\phi_H(A) = \begin{cases} 1 & \text{if } \Gamma_A(H) \neq \emptyset, \\ 0 & \text{if } \Gamma_A(H) = \emptyset. \end{cases}$$

In some way, this consistency measure evaluates whether H is a valid assumption on the source when it provides the testimony $\mathbf{x} \in A$. Consider, for instance, the assumption $h = (R,\neg T)$ corresponding to a relevant and lying source. This assumption will be considered invalid only when the source provides the testimony $\mathbf{x} \in \mathcal{X}$ as $\Gamma_{\mathcal{X}}(h) = \emptyset$ and $\phi_h(\mathcal{X}) = 0$.

Meta-knowledge can also be characterized in terms of specificity: namely a piece of meta-knowledge $H_1 \subseteq \mathcal{H}$ will be said *at least as meta-specific* as another piece of meta-knowledge $H_2 \subseteq \mathcal{H}$ when $\Gamma_A(H_1) \subseteq \Gamma_A(H_2)$ for any $A \subseteq \mathcal{X}$, and we will denote it $H_1 \sqsubseteq_{\mathcal{H}} H_2$. For example, the assumption (R,T) is at least as meta-specific as the assumption $(\neg R,T)$. Note that we have the relations $H_1 \sqsubseteq H_2 \Rightarrow H_1 \sqsubseteq_{\mathcal{H}} H_2$ and $H_1 \sqsubseteq_{\mathcal{H}} H_2 \Rightarrow \phi_{H_1}(A) \geq \phi_{H_2}(A)$, the latter relation being of particular interest in the context of this paper as it shows that reaching both consistency and specificity are somewhat opposite goals.

3.2 Uncertain Testimony and Meta-knowledge

More generally, both the testimony and the meta-knowledge of the agent may be uncertain. Let $m^{\mathcal{X}}$ be the uncertain testimony and $m^{\mathcal{H}}$ the uncertain meta-knowledge. The knowledge of the agent on $\mathcal{X}$ can then be represented by the mass function $m[m^{\mathcal{H}}]^{\mathcal{X}}$ defined for all $B \subseteq \mathcal{X}$ as [14]

$$m[m^{\mathcal{H}}]^{\mathcal{X}}(B) = \sum_{H \subseteq \mathcal{H}} m^{\mathcal{H}}(H) \sum_{A:\Gamma_A(H)=B} m^{\mathcal{X}}(A). \tag{3}$$

This definition is rather general. In particular, the discounting rule proposed by Shafer [16] is retrieved by $m^{\mathcal{H}}(R) = p$ and $m^{\mathcal{H}}(\neg R) = 1 - p$ [14].

The results of the previous section can be extended to this general setting: following [2], the mass function modeling the empty set ($m[m^{\mathcal{H}}]^{\mathcal{X}}(\emptyset) = 1$) can be associated to a complete inconsistent knowledge and a mass function $m[m^{\mathcal{H}}]^{\mathcal{X}}$ whose focal sets have a non-empty intersection can be associated to a totally consistent knowledge. That is, the testimony $m^{\mathcal{X}}$ is totally consistent under meta-knowledge $m^{\mathcal{H}}$ if and only if

$$\bigcap_{\substack{A \in \mathcal{F} \\ H \in \mathcal{F}_{\mathcal{H}}}} \Gamma_A(H) \neq \emptyset, \tag{4}$$

where $\mathcal{F}$ and $\mathcal{F}_{\mathcal{H}}$ denote the sets of focal sets of $m^{\mathcal{X}}$ and $m^{\mathcal{H}}$, respectively. A mass function $m^{\mathcal{X}}$ is then said $m^{\mathcal{H}}$*-consistent* if and only if (4) holds. Lemma 1 characterizes $m^{\mathcal{H}}$-consistent testimonies in terms of the contour function.

Lemma 1. $\bigcap_{\substack{A \in \mathcal{F} \\ H \in \mathcal{F}_{\mathcal{H}}}} \Gamma_A(H) \neq \emptyset \Leftrightarrow \exists x \in \mathcal{X}$ *such that* $pl[m^{\mathcal{H}}]^{\mathcal{X}}(x) = 1$, *where* $pl[m^{\mathcal{H}}]^{\mathcal{X}}$ *is the contour function associated to the mass function* $m[m^{\mathcal{H}}]^{\mathcal{X}}$ *obtained from* (3).

A source is thus $m^{\mathcal{H}}$-consistent if it allows us to conclude that at least one value of $\mathbf{x}$ is totally plausible under meta-knowledge $m^{\mathcal{H}}$. Following [2], this characterization of $m^{\mathcal{H}}$-consistency suggests the following definition:

Definition 1 ($m^{\mathcal{H}}$-consistency measure). *The measure* $\phi_{m^{\mathcal{H}}} : \mathcal{M}^{\mathcal{X}} \to [0, 1]$ *of* $m^{\mathcal{H}}$*-consistency, where* $\mathcal{M}^{\mathcal{X}}$ *denotes the set of all mass functions on* $\mathcal{X}$*, reads:*

$$\phi_{m^{\mathcal{H}}}(m^{\mathcal{X}}) = \max_{x \in \mathcal{X}} pl[m^{\mathcal{H}}]^{\mathcal{X}}(x).$$

The notion of meta-specificity may also be extended to this general setting.

Definition 2 (Meta-specificity). *An uncertain piece of meta-knowledge* $m_1^{\mathcal{H}}$ *is said to be* at least as meta-specific *as another uncertain piece* $m_2^{\mathcal{H}}$ *when* $m[m_1^{\mathcal{H}}]^{\mathcal{X}} \sqsubseteq m[m_2^{\mathcal{H}}]^{\mathcal{X}}$ *for any* $m^{\mathcal{X}} \in \mathcal{M}^{\mathcal{X}}$*. This is denoted by* $m_1^{\mathcal{H}} \sqsubseteq_{\mathcal{H}} m_2^{\mathcal{H}}$.

We may then show that in the general case, consistency and specificity are also at odds:

Proposition 1. *If* $m_1^{\mathcal{H}} \sqsubseteq_{\mathcal{H}} m_2^{\mathcal{H}}$, *then* $\phi_{m_1^{\mathcal{H}}}(m^{\mathcal{X}}) \leq \phi_{m_2^{\mathcal{H}}}(m^{\mathcal{X}}) \; \forall \; m^{\mathcal{X}} \in \mathcal{M}^{\mathcal{X}}$.

Example 1 (Inspired from Example 1 of [14]). Let $\mathcal{X} = \{x_1, x_2, x_3, x_4, x_5\}$ be an ordered space and consider the mass function such that $m^{\mathcal{X}}(\{x_1, x_2\}) = 0.3$, $m^{\mathcal{X}}(\{x_4, x_5\}) = 0.3$ and $m^{\mathcal{X}}(\{x_3\}) = 0.4$. Now consider the assumptions h_1 "informed" such that $\Gamma_A(h_1) = A$, h_2 "approximately informed" such that if $A = \{x_i, x_{i+1}, \ldots, x_j\}$ then $\Gamma_A(h_2) = \{x_{i-1}\} \cup A \cup \{x_{j+1}\}$ with $x_0 = x_6 = \emptyset$, and h_3 "unreliable" such that $\Gamma_A(h_3) = \mathcal{X}$. Then we have

$$\phi_{h_1}(m^{\mathcal{X}}) = 0.4, \quad \phi_{h_2}(m^{\mathcal{X}}) = 1, \quad \phi_{h_3}(m^{\mathcal{X}}) = 1,$$

$$h_1 \sqsubseteq_{\mathcal{H}} h_2 \sqsubseteq_{\mathcal{H}} h_3.$$

This example allows us to lay bare some preliminary ideas on the selection of source behavior based on consistency and specificity. As can be seen, assumptions h_2 and h_3 are the most desirable in terms of consistency, since they both yield a totally consistent state of knowledge on $\mathcal{X}$. However, the state of knowledge obtained under h_2 is more specific, or informative, than the one obtained under h_3, hence h_2 may appear preferable. Those ideas will be developed at length in Section 5.

4 Consistency and Specificity: Multiple Sources

We now consider multiple sources $\mathfrak{s}_i$, $i = 1, \ldots, K$ where each can be in states $\mathcal{H}^i = \{h_1^i, ..., h_N^i\}$ and deliver testimonies $m_i^{\mathcal{X}}$, $i = 1, \ldots, K$. We define for any state $\mathbf{h} = (h^1, \ldots, h^k) \in \mathcal{H}^{1:K}$ a mapping [14] for any $\mathbf{A} = (A_1, \ldots, A_K) \subseteq \mathcal{X}^K$ as $\Gamma_{\mathbf{A}}(\mathbf{h}) = \bigcap_{i=1}^K \Gamma_{A_i}(h^i)$. $\Gamma_{\mathbf{A}}(\mathbf{h})$ is the information on $\mathcal{X}$ deduced from testimonies $(A_1, \ldots, A_K)$ of sources $\mathfrak{s}_1, \ldots, \mathfrak{s}_K$ when they are in states $(h^1, \ldots, h^K)$. We keep the notation $\Gamma_{\mathbf{A}}(H) := \cup_{\mathbf{h} \in H} \Gamma_{\mathbf{A}}(\mathbf{h})$ for all $H \subseteq \mathcal{H}^{1:K}$ and all $\mathbf{A} \subseteq \mathcal{X}^K$.

4.1 General Case

If we have a joint meta-knowledge $m^{\mathcal{H}^{1:K}}$ over $\times_{i=1}^K \mathcal{H}_i$ and if sources $\mathfrak{s}_1, \ldots, \mathfrak{s}_K$ are independent, then the combined mass function $m[m^{\mathcal{H}^{1:K}}]^{\mathcal{X}}$ defined by (5) represents what can be inferred about $\mathbf{x}$ from $\mathbf{m}^{\mathcal{X}} = (m_1^{\mathcal{X}}, ..., m_K^{\mathcal{X}})$ [14]:

$$m[m^{\mathcal{H}^{1:K}}]^{\mathcal{X}}(B) = \sum_{H \subseteq \mathcal{H}^{1:K}} m^{\mathcal{H}^{1:K}}(H) \sum_{\substack{\mathbf{A} \subseteq \mathcal{X}^K \\ \Gamma_{\mathbf{A}}(H) = B}} \left[\prod_{i=1}^K m_i^{\mathcal{X}}(A_i) \right]. \tag{5}$$

We note that this approach has a computational complexity that increases exponentially in the number of sources.

Keeping the same definition of complete inconsistent and consistent knowledge as in Section 3.2, the counterpart of Lemma 1 suggests to use the following equation as a degree of $m^{\mathcal{H}^{1:K}}$-consistency for the collection $\mathbf{m}^{\mathcal{X}}$

$$\phi_{m^{\mathcal{H}^{1:K}}}(\mathbf{m}^{\mathcal{X}}) = \max_{x \in \mathcal{X}} pl[m^{\mathcal{H}^{1:K}}]^{\mathcal{X}}(x), \tag{6}$$

where $pl[m^{\mathcal{H}^{1:K}}]$ is the contour function of (5). Again, if $m_1^{\mathcal{H}^{1:K}} \sqsubseteq_{\mathcal{H}} m_2^{\mathcal{H}^{1:K}}$, then $\phi_{m_1^{\mathcal{H}^{1:K}}}(\mathbf{m}^{\mathcal{X}}) \leq \phi_{m_2^{\mathcal{H}^{1:K}}}(\mathbf{m}^{\mathcal{X}})$ for any $\mathbf{m}^{\mathcal{X}}$. We will make heavy use of this duality between specificity and consistency in Section 5.

An interesting feature of the approach [14] is that all Boolean operators on sets $\mathbf{A} = (A_1, \ldots, A_K) \subseteq \mathcal{X}^K$ can be obtained through particular assumptions on the behavior of the sources. As a result, Equation (5) covers all combination rules based on Boolean operators. For instance, consider the assumption H_r^K on $\mathcal{H}^{1:K}$ meaning the sources are truthful and "r-out-of-K" of them are relevant. This amounts to

$$\Gamma_{\mathbf{A}}(H_r^K) = \bigcup_{\mathcal{A} \subset \{A_1, \ldots, A_K\}, |\mathcal{A}| = r} (\cap_{A \in \mathcal{A}} A)\,, \tag{7}$$

and when applying H_r^K to Eq. (5), the conjunctive and disjunctive rules $\textcircled{\cap}$ and $\textcircled{\cup}$ are retrieved when $r = K$ and $r = 1$, respectively.

Remark 1 (Extension of conflict to all Boolean operators). This feature, once coupled with Equation (6), is fruitful: it provides a natural extension of the measure of conflict defined in [2] as the inconsistency resulting from the conjunctive combination, to all other combination rules based on Boolean operators.

4.2 Separable Meta-knowledge

Computing (5) can be resource demanding, however there are cases where it is easier. In particular, when all focal elements of $m^{\mathcal{H}^{1:K}}$ are *separable*.

Definition 3 (Separability). *A subset $H \subseteq \mathcal{H}^{1:K}$ is said* separable *if and only if $H = H^{\downarrow 1} \times \ldots \times H^{\downarrow K}$, where $H^{\downarrow i}$ denotes the projection of $H \subseteq \mathcal{H}^{1:K}$ on $\mathcal{H}^i$.*

Proposition 2. *When each focal set of $m^{\mathcal{H}^{1:K}}$ is separable*[1]*, Equation* (5) *can be rewritten as:*

$$m[m^{\mathcal{H}^{1:K}}]^{\mathcal{X}}(B) = \sum_{H \subseteq \mathcal{H}^{1:K}} m^{\mathcal{H}^{1:K}}(H) \cdot \left[\textcircled{\cap}_{i=1}^{K} m[H^{\downarrow i}]^{\mathcal{X}}\right](B), \tag{8}$$

where $m[H^{\downarrow i}]^{\mathcal{X}}$ denotes mass function $m_i^{\mathcal{X}}$ transformed according to $H^{\downarrow i}$.

That is, we first transform each $m_i^{\mathcal{X}}$ according to $H^{\downarrow i}$, apply unnormalized Dempster's rule to them and compute the weighted sum according to $m^{\mathcal{H}^{1:K}}$. We can therefore make use of efficient algorithms to compute Dempster's rule result [20].

This property also simplifies the computation of the consistency measure (6). Indeed, consider the meta-knowledge $m^{\mathcal{H}^{1:K}}(H) = 1$ with H separable and let $pl[H]^{\mathcal{X}}$ be the corresponding contour function. Then if $pl[H^{\downarrow i}]^{\mathcal{X}}$ is the contour

[1] This happens, *e.g.*, when $m^{\mathcal{H}^{1:K}}$ satisfies the property of meta-independence [14], which basically means that $m^{\mathcal{H}^{1:K}}$ is the result of independent pieces of meta-knowledge concerning each source.

function obtained by transforming $m_i^{\mathcal{X}}$ according to meta-knowledge $H^{\downarrow i}$, we have $pl[H]^{\mathcal{X}}(x) = \prod_{i=1}^{K} pl[H^{\downarrow i}]^{\mathcal{X}}(x)$. As Equation (8) is a convex mixture of such mass functions, and as the plausibility measure of a convex mixture is the convex mixture of plausibility measures, computing consistency measure (6) only requires to compute contour functions and to take their weighted averaged products (hence not necessitating any combination).

5 Source Behavior Selection Approach

Selecting which assumption to make on the sources when one has no previous experience with them, basically amounts to defining a set of candidate pieces of meta-knowledge, and a selection criterion allowing one to choose a particular element in this set. Based on the results of the previous sections, this section provides some guidelines to define such a set, as well as a selection criterion that can be used on any set satisfying those guidelines, leading to a general, yet practical and sensible, approach to select the behavior of the sources. Important examples of the application of this approach are also presented.

5.1 Initial Meta-knowledge

In absence of any particular information on the behavior of the sources, we propose to consider first an assumption $m_1^{\mathcal{H}^{1:K}}$ such that $m_1^{\mathcal{H}^{1:K}}(\mathbf{h}) = 1$, with $\mathbf{h} \in \mathcal{H}^{1:K}$ and $\Gamma_A(\mathbf{h}^{\downarrow i}) = A$, $\forall A \subseteq \mathcal{X}$, $i = 1, ..., K$, *i.e.*, an assumption that induces no transformation of the testimonies provided by the sources. This assumption corresponds to an agent that does not want to alter in any way the information he has received: it amounts to accepting the testimonies as they are. Most importantly, the assumption that the sources are all relevant and truthful, *i.e.*, the most classical assumption in information fusion in general and in belief function theory in particular, is formally an instance of $m_1^{\mathcal{H}^{1:K}}$. Our proposal corresponds indeed to combining the sources using the unnormalized Dempster's rule – the first rule usually considered to combine pieces of information. Hence, $m_1^{\mathcal{H}^{1:K}}$ is a natural default meta-knowledge.

Equation (6) provides us with an assessment of whether the assumption $m_1^{\mathcal{H}^{1:K}}$ applies to the current testimonies. In particular, and as is classically advocated in belief function theory, we propose that if the consistency induced by this assumption is high enough, that is if it is above some threshold τ, then this assumption should be used to combine the testimonies, and if the consistency is too low, *i.e.*, below τ, then the assumption $m_1^{\mathcal{H}^{1:K}}$ should not be used and other assumptions leading to higher consistency should be sought.

5.2 A Specificity Ordering Approach

To search for other assumptions with better consistency, the counterpart of Proposition 1 in the multiple source case can be instrumental: choosing a

meta-knowledge $m_2^{\mathcal{H}^{1:K}}$ such that $m_1^{\mathcal{H}^{1:K}} \sqsubseteq_{\mathcal{H}} m_2^{\mathcal{H}^{1:K}}$ will indeed ensure that the consistency increases. This leads us to propose the following strategy to select the meta-knowledge to be used:

- define a collection of meta-knowledge $\mathbf{m}^{\mathcal{H}^{1:\mathbf{K}}} = (m_1^{\mathcal{H}^{1:K}}, ..., m_M^{\mathcal{H}^{1:K}})$ such that for any $1 \leq j < M$, $m_j^{\mathcal{H}^{1:K}} \sqsubset_{\mathcal{H}} m_{j+1}^{\mathcal{H}^{1:K}}$, and with $m_1^{\mathcal{H}^{1:K}}$ as defined above;
- test each $m_j^{\mathcal{H}^{1:K}}$ iteratively with $j = 1, \ldots, M$, until $\phi_{m_j^{\mathcal{H}^{1:K}}}(\mathbf{m}^{\mathcal{X}}) \geq \tau$.

In other words, this strategy gradually decreases specificity until a satisfactory consistency level is reached. It comes down to considering a set of pieces of meta-knowledge that are comparable according to $\sqsubseteq_{\mathcal{H}}$, with $m_1^{\mathcal{H}^{1:K}}$ being the most meta-specific element of this set, and to select in this set the most meta-specific element $m_j^{\mathcal{H}^{1:K}}$ such that $\phi_{m_j^{\mathcal{H}^{1:K}}}(\mathbf{m}^{\mathcal{X}}) \geq \tau$.

Remark 2. The construction of $\mathbf{m}^{\mathcal{H}^{1:\mathbf{K}}}$ should also follow some sensible rules: pieces of meta-knowledge $m_j^{\mathcal{H}^{1:K}}$ should have a clear semantic and the spaces $\mathcal{H}^i$ should be of reduced size, *e.g.*, $\mathcal{H}^i = \{(T^i, R^i), (T^i, \neg R^i), (\neg T^i, R^i), (\neg T^i, \neg R^i)\}$.

5.3 Examples

As shown below, our approach subsumes important classical fusion strategies dedicated to conflict management in belief function theory. These strategies follow the same pattern: they first combine the testimonies using the unnormalized Dempster's rule, and if the consistency resulting from its application is too low, other assumptions on the sources yielding higher consistency are considered. Let us remark that the first fusion strategy discussed below is based on imprecise pieces of meta-knowledge, whereas the second one is based on probabilistic ones.

r-out-of-K Relevant Sources. We can implement the above methodology by choosing $m_j^{\mathcal{H}^{1:K}}(H^K_{K-j+1}) = 1$, with H^K_{K-j+1} the assumption that the sources are truthful and $r = K - j + 1$ out of them are relevant (see Eq. (7)), as the following proposition indicates:

Proposition 3. *If* $m_j^{\mathcal{H}^{1:K}}(H^K_{K-j+1}) = 1$, *then* $m_j^{\mathcal{H}^{1:K}} \sqsubseteq_{\mathcal{H}} m_{j+1}^{\mathcal{H}^{1:K}}$ *for* $1 \leq j < K$.

Example 2. Consider the mass functions $m_1^{\mathcal{X}}$, $m_2^{\mathcal{X}}$ and $m_3^{\mathcal{X}}$ on $\mathcal{X} = \{x_1, x_2, x_3\}$ in the left part of Table 1. Assume they were received from three independent sources. Let $\mathbf{m}^{\mathcal{H}^{1:\mathbf{K}}} = (m_1^{\mathcal{H}^{1:K}}, m_2^{\mathcal{H}^{1:K}}, m_3^{\mathcal{H}^{1:K}}) = (H_3^3, H_2^3, H_1^3)$ be three pieces of meta-knowledge we want to test on these sources. $m_1^{\mathcal{H}^{1:K}}$ corresponds to the use of the unnormalized Dempster's rule, while $m_3^{\mathcal{H}^{1:K}}$ corresponds to the use of the disjunctive rule. $m_2^{\mathcal{H}^{1:K}}$ corresponds to the assumption H_2^3 that the three sources are truthful and that two of them are relevant, but we do not know which ones, i.e., to the following subset of $\mathcal{H}^{1:K}$:

$$\{(R_1, T_1, R_2, T_2, \neg R_3, T_3), (R_1, T_1, \neg R_2, T_2, R_3, T_3), (\neg R_1, T_1, R_2, T_2, R_3, T_3)\}. \tag{9}$$

Table 1. Mass functions resulting from the three different assumptions

A	$m_1^{\mathcal{X}}$	$m_2^{\mathcal{X}}$	$m_3^{\mathcal{X}}$	$m[H_1^3]^{\mathcal{X}}$	$m[H_2^3]^{\mathcal{X}}$	$m[H_3^3]^{\mathcal{X}}$
$\emptyset$	0	0	0	0	0	0.36
$\{x_1\}$	0.5	0	0	0	0.06	0.2
$\{x_2\}$	0	0	0	0	0	0
$\{x_1, x_2\}$	0	0.2	0	0	0.04	0.04
$\{x_3\}$	0	0	0.6	0	0	0.24
$\{x_1, x_3\}$	0	0	0	0	0.24	0
$\{x_2, x_3\}$	0	0	0	0	0	0
X	0.5	0.8	0.4	1	0.66	0.16

The right part of Table 1 presents the mass functions on $\mathcal{X}$ resulting from the three different assumptions. We have $\phi_{H_1^3}(\mathbf{m}^{\mathcal{X}}) = 1, \phi_{H_2^3}(\mathbf{m}^{\mathcal{X}}) = 1$ and $\phi_{H_3^3}(\mathbf{m}^{\mathcal{X}}) = 0.4$, hence our approach suggests to use H_2^3 to combine the pieces of information in this example.

Note that the assumption "r-out-of-K" is not separable in general. For instance, the subset (9) is not the product of each of its projection. However, we may remark that this assumption treats all sources in the same way, which seems interesting in absence of meta-knowledge about each individual source.

Vectors of Reliabilities. Another interesting case is when we consider $\mathcal{H}^i = \{R_i, \neg R_i\}$ (relevant or not) and a vector $\mathbf{p} = (p_1, \ldots, p_K)$ such that $m^{\mathcal{H}^i}(R_i) = p_i$, $m^{\mathcal{H}^i}(\neg R_i) = 1 - p_i$ and where $m^{\mathcal{H}^{1:K}}$ is obtained by considering the stochastic product of probabilities $p_1, \ldots, p_k$. In such case, the assumption $m^{\mathcal{H}^{1:K}}$ amounts to discounting each source $\mathfrak{s}_i$ according to reliability rate $1 - p_i$ and then combining the discounted sources using unnormalized Dempster's rule [14]. If we define a set $\mathbf{p}^1, \ldots, \mathbf{p}^M$ of such vectors with $p_i^j > p_i^{j+1}$, we get corresponding meta-knowledges $m_1^{\mathcal{H}^{1:K}}, \ldots, m_M^{\mathcal{H}^{1:K}}$ with the following property.

Proposition 4. *Let* $m_j^{\mathcal{H}^{1:K}}$, $j = 1, \ldots, M$, *be the mass functions defined using* $\mathbf{p}^1, \ldots, \mathbf{p}^M$. *We have* $m_j^{\mathcal{H}^{1:K}} \sqsubseteq_{\mathcal{H}} m_{j+1}^{\mathcal{H}^{1:K}}$, *for* $1 \leq j < M$.

A useful feature of such $\mathbf{m}^{\mathcal{H}^{1:K}}$ is that each meta-knowledge $m_j^{\mathcal{H}^{1:K}}$ satisfies the meta-independence property [14] and therefore $\phi_{m_j^{\mathcal{H}^{1:K}}}(\mathbf{m}^{\mathcal{X}})$ can be computed efficiently using the results of Section 4.2.

Remark 3. If we associate p_i^j with the product of one minus the degrees of falsity of mass function i up to step j in Schubert's recent work on sequential discounting [15], then $m[m_j^{\mathcal{H}^{1:K}}]^{\mathcal{X}}$ is nothing else but the mass function on $\mathcal{X}$ obtained at step j in Schubert's scheme. Hence Schubert's method [15] is included in the present approach.

6 Conclusion

In this paper, we have proposed a practical and sensible methodology to select the behavior of sources in information fusion, based on the fundamental notions of specificity and consistency. Our approach is based on recent frameworks that measure inconsistency [2] and model source behaviors [14] in simple yet powerful ways. In particular, we have introduced measures of consistency and a partial ordering for the source behavior assumptions allowed by Pichon *et al.* framework [14], which are used in the behavior selection process. This also led notably to a natural extension of the measure of conflict defined in [2], to all combination rules based on Boolean operators. In addition, an interesting feature of our approach is that it subsumes important classical fusion strategies dedicated to conflict management in belief function theory.

We may mention a few research paths that were left unexplored in this paper:

- as in [2], it would be interesting to study the alternative consistency measure based on $m^{\mathcal{X}}(\emptyset)$, or what happens in the current framework when we relax the assumption of source independence;
- variations of our approach could be investigated, both from formal and practical point of views, and in particular using other criteria than specificity and consistency, for instance the idea of minimal change evoked in Section 1;
- besides the families of assumptions on the sources that are studied in Section 5.3, it may be interesting to identify other families of assumptions that are ordered according to the relation of meta-specificity and that include the unnormalized Dempster's rule as most meta-specific element;
- if several collections $\mathbf{m}^{\mathcal{X}}$ of testimonies are available, then one could try to learn the best meta-knowledge to be used in general to combine the testimonies. In particular, we may think of obtaining a probability distribution over $\mathbf{m}^{\mathcal{H}^{\mathbf{1:K}}}$ and exploit it for selecting the best meta-knowledge. This information could be coupled with other methods that learns reliability indices [7,11,6].
- one could try to integrate in the current framework some related works, such as Smets expert system [18] or Mercier *et al.* [11] contextual discounting;
- the idea of using consistency and specificity as rule selection methods could be extended to rules that have no clear interpretation in terms of meta-knowledge, such as weight-based ones [13].

Acknowledgements. This work was partially carried out in the framework of (1) the ANR funding ANR-11-IDEX-0004-02 (Labex MS2T, "Investissements d'Avenir" call) (2) the ANR funding ANR-10-INBS-08 (ProFI project, "Infrastructures Nationales en Biologie et Santé"; "Investissements d'Avenir" call) , and (3) the Prospectom project of the Mastodons 2012 challenge (CNRS).

References

1. Dempster, A.P.: Upper and lower probabilities induced by a multivalued mapping. Ann. of Math. Stat. 38, 325–339 (1967)
2. Destercke, S., Burger, T.: Toward an axiomatic definition of conflict between belief functions. IEEE Trans. on Syst., Man and Cyb. - B 43(2), 585–596 (2013)
3. Destercke, S., Dubois, D.: Idempotent conjunctive combination of belief functions: Extending the minimum rule of possibility theory. Inf. Sciences 181(18), 3925–3945 (2011)
4. Dubois, D., Prade, H.: A set-theoretic view of belief functions: logical operations and approximations by fuzzy sets. Int. J. of Gen. Syst. 12(3), 193–226 (1986)
5. Dubois, D., Prade, H., Smets, P.: A definition of subjective possibility. Int. J. Approx. Reasoning 48(2), 352–364 (2008)
6. Elouedi, Z., Lefevre, E., Mercier, D.: Discountings of a belief function using a confusion matrix. In: 22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI), pp. 287–294 (2010)
7. Elouedi, Z., Mellouli, K., Smets, P.: The evaluation of sensors' reliability and their tuning for multisensor data fusion within the transferable belief model. In: Benferhat, S., Besnard, P. (eds.) ECSQARU 2001. LNCS (LNAI), vol. 2143, pp. 350–361. Springer, Heidelberg (2001)
8. Jousselme, A.-L., Maupin, P.: Distances in evidence theory: Comprehensive survey and generalizations. Int. J. Approx. Reasoning 53(2), 118–145 (2012)
9. Liu, W.: Analyzing the degree of conflict among belief functions. Artif. Intell. 170(11), 909–924 (2006)
10. Ma, J., Liu, W., Dubois, D., Prade, H.: Bridging Jeffrey's rule, AGM revision and Dempster conditioning in the theory of evidence. Int. J. on Artif. Intell. Tools 20(4), 691–720 (2011)
11. Mercier, D., Quost, B., Denoeux, T.: Refined modeling of sensor reliability in the belief function framework using contextual discounting. Inf. Fusion 9, 246–258 (2008)
12. Pichon, F.: On the α-conjunctions for combining belief functions. In: Denœux, T., Masson, M.-H. (eds.) Belief Functions: Theory & Appl. AISC, vol. 164, pp. 285–292. Springer, Heidelberg (2012)
13. Pichon, F., Denœux, T.: The unnormalized Dempster's rule of combination: A new justification from the least commitment principle and some extensions. J. Autom. Reasoning 45(1), 61–87 (2010)
14. Pichon, F., Dubois, D., Denœux, T.: Relevance and truthfulness in information correction and fusion. Int. J. Approx. Reasoning 53(2), 159–175 (2012)
15. Schubert, J.: Conflict management in Dempster-Shafer theory using the degree of falsity. Int. J. Approx. Reasoning 52(3), 449–460 (2011)
16. Shafer, G.: A mathematical theory of evidence. Princeton University Press (1976)
17. Smets, P.: Belief functions: the disjunctive rule of combination and the generalized Bayesian theorem. Int. J. Approx. Reasoning 9(1), 1–35 (1993)
18. Smets, P.: Analyzing the combination of conflicting belief functions. Inf. Fusion 8(4), 387–412 (2007)
19. Smets, P., Kennes, R.: The transferable belief model. Artif. Intell. 66, 191–243 (1994)
20. Wilson, N.: Algorithms for Dempster-Shafer theory. In: Handbook of Defeasible Reasoning and Uncertainty Management Systems, vol. 5, pp. 421–475 (2000)

On the Problem of Reversing Relational Inductive Knowledge Representation

Nico Potyka[1], Christoph Beierle[1], and Gabriele Kern-Isberner[2]

[1] Dept. of Computer Science, FernUniversität in Hagen, 58084 Hagen, Germany
[2] Dept. of Computer Science, TU Dortmund, 44221 Dortmund, Germany

Abstract. By using the principle of maximum entropy incomplete probabilistic knowledge can be completed to a full joint distribution. This inductive knowledge representation method can be reversed to extract probabilistic rules from an empirical probability distribution. Based on this idea propositional learning approach has been developed. Recently, an extension to a relational language has been presented, where, however, a central aspect, finding and resolving algebraic equations needed for the solution, has been treated as a black box. Here, we investigate both problems in more detail. We explain how equations for relational knowledge bases can be resolved, and give a comprehensive example of computing a relational knowledge base from a probability distribution. Furthermore, we describe how propositional mechanisms for finding equations can be refined to focus on more interesting equations and to reduce the number of candidates.

1 Introduction

Given data collected in some domain, one is often interested in learning a knowledge base reflecting the most important dependencies in this dataset. Different fields like classical Data Mining [4] or Statistical Relational Learning [3] provide different learning techniques. In [5] an alternative way of learning is suggested. One supposes the empirical probability distribution that is induced by the observations in the dataset is originally generated by certain laws that can be represented by a conditional knowledge base. Applying the principle of maximum entropy (ME) the distribution will respect the laws, but will be as uniform as possible otherwise. This process of inductive knowledge completion by generating the ME-distribution from a set of conditionals can be reversed to construct a conditional knowledge base from the empirical probability distribution (cf. Fig. 1). To this end, one starts with a complete knowledge base reflecting each possible dependency and successively shortens it. Each shortening can be justified by an algebraic theory [5]. CondorCKD [6] is an algorithm that implements this idea, but it is restricted to propositional languages. Yet in some domains it is more appropriate to use relational languages to emphasize relations between individuals.

Example 1. Suppose we want to find out about the social behavior of a population of monkeys. At feeding time, we assume that hungry monkeys will eat their food, otherwise they might allow another monkey to eat it. For each monkey c we introduce a binary random variable $h(c)$ (hungry) and a variable $al(c, d)$ (allows eating) for each two different monkeys c, d. Even though the representation is effectively propositional,

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 485–496, 2013.

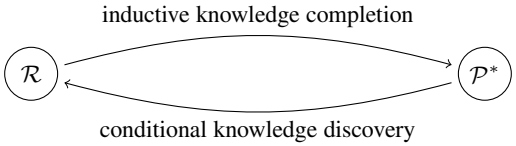


Fig. 1. Conditional knowledge discovery as inverse to inductive knowledge representation

we are interested in the underlying relational structure. We are interested in 'relational conditionals' like $(al(X,Y) \mid \overline{h(X)})[0.9]$ expressing that a monkey X that is not hungry probably allows another monkey Y to eat its food.

In [8] an approach is sketched how the approach followed in CondorCKD can be transferred to the relational language FO-PCL [2]. Starting with an extensive set of relational conditionals that most likely capture all possible dependencies, the set is successively shortened by deleting and combining conditionals appropriately. Again, each shortening operations is justified by algebraic equations. Finally the remaining conditionals are evaluated with respect to the empirical probability distribution to identify exceptional individuals. For example, the conditional $(al(c,X) \mid \overline{h(c)})[0.2]$ might identify an egoistic monkey that prefers hoarding its food rather than sharing it.

In [8] finding and resolving of interesting equations is treated like a black box. We close this gap, by explaining how fundamental ideas [5] underlying CondorCKD can be adapted to the relational structure of FO-PCL knowledge bases. Another issue is the way interesting equations are searched. Applying CondorCKD's search algorithm naively can result in many uninteresting equations. By taking the relational structure into account, we can concentrate on more promising equations and at the same time reduce the computational complexity of finding equations.

In Section 2 we give a quick overview of the FO-PCL and the maximum entropy framework. Afterwards we give a closer overview of the learning problem, the core concepts underlying CondorCKD and recapture the core ideas of extending it to FO-PCL from [8]. In Section 3 we deal with resolving relational equations. We explain the 'conditional structure' of FO-PCL knowledge bases. Subsequently we sketch how the propositional shortening operations transfer to FO-PCL and give a comprehensive computation example. In Section 4 we explain how the 'neighborhood graph' used in CondorCKD can be decomposed to focus on promising relational equations. Afterwards, we explain how the search in the obtained subgraphs can be simplified by exploiting similarity of their connected components.

2 Basics

FO-PCL: *FO-PCL* [2] is a restricted many-sorted first-order logic built up over signatures of the form $\Sigma = (S, Const, Pred)$. S is a set of sorts, $Const$ a set of sorted constants, and $Pred$ a set of sorted predicate symbols. Formulas are built up over a signature Σ and a set of sorted variables $\mathcal{V}$ in the usual way using conjunction, disjunction and negation, but no quantifiers. We abbreviate conjunctions $\psi \wedge \phi$ by $\psi\phi$ and negations $\neg\psi$ by $\overline{\psi}$. Variables are interpreted by means of a grounding operator gnd. For this

purpose additionally *constraint formulas* are considered. They are built over the same set of constants $Const$, variables $\mathcal{V}$ and a new sorted equality-symbol $=$. An *FO-PCL conditional* $R = \langle(\phi \mid \psi)[\xi], C\rangle$ consists of two formulas over Σ, the *consequence* ϕ and the *antecedence* ψ, a probability $\xi \in [0,1]$ and a constraint formula C. A set $\mathcal{R}$ of such conditionals is called an *FO-PCL knowledge base*. In the following, we will consider only *single-elementary conditionals*, i.e., conditionals whose consequence is a single atom and whose antecedence is a conjunction of positive or negated atoms.

An *instance* of an FO-PCL conditional R is obtained by applying a ground substitution over the variables appearing in R to the antecedence, consequence and constraint formula of R. An instance is called *admissible* if its constraint formula evaluates to true, where $=$ is interpreted by syntactical equality. The grounding operator gnd maps each FO-PCL conditional R to the set of its admissible ground instances $\mathrm{gnd}(R)$.

Example 2. Consider an FO-PCL signature $\Sigma = (S, Const, Pred)$ with a single sort $S = \{Monkey\}$, two constants $Const = \{a, b\}$ of sort $Monkey$ and two predicate symbols $Pred = \{h(Monkey), al(Monkey, Monkey)\}$ indicating a hungry monkey and a monkey allowing another monkey to eat its food.

Then $\langle(al(a,a) \mid \overline{h(a)})[0.9], a \neq a\rangle$ is an instance of $\langle(al(X,Y) \mid \overline{h(X)})[0.9], X \neq Y\rangle$. It is not admissible, since $a \neq a$ does not hold. $\langle(al(a,b) \mid \overline{h(a)})[0.9], a \neq b\rangle$ is an admissible ground instance. The constraint formula is usually left out for admissible ground instances. Hence the admissible instances are $(al(a,b) \mid \overline{h(a)})[0.9]$ and $(al(b,a) \mid \overline{h(b)})[0.9]$.

In the following, we will refer to the admissible ground instances of a conditional R by just instances of R. The Herbrand base $\mathcal{H}(\mathcal{R})$ of a knowledge base $\mathcal{R}$ includes all ground atoms appearing in instances of $\mathcal{R}$. For instance, in the example above we have $\mathcal{H}(\mathcal{R}) = \{al(a,b), al(b,a), h(a), h(b)\}$. To each $\mathcal{R}$ there is a corresponding set of *possible worlds*. A possible world ω is a truth function $\omega : \mathcal{H}(\mathcal{R}) \to \{0,1\}$ assigning a truth value to each ground atom in $\mathcal{H}(\mathcal{R})$. ω satisfies a ground atom $\mathfrak{a} \in \mathcal{H}(\mathcal{R})$, $\omega \models \mathfrak{a}$, iff $\omega(\mathfrak{a}) = 1$. The satisfaction relation $\models$ is extended to complex formulas in the usual way. $\models$ is undefined for formulas containing variables, but such formulas are never considered in our framework. For each ground formula ϕ its set of models is denoted by $\mathrm{Mod}(\phi) := \{\omega \in \Omega \mid \omega \models \phi\}$.

Conditionals are interpreted by probability distributions $\mathcal{P} : \Omega \to [0,1]$ assigning a degree of belief to worlds. For ground formulas ϕ we define $\mathcal{P}(\phi) := \sum_{\omega \in \mathrm{Mod}\,\phi} \mathcal{P}(\omega)$. We say a probability distribution $\mathcal{P}$ *satisfies* an FO-PCL conditional R iff for each instance $(\phi_{\mathrm{gnd}} \mid \psi_{\mathrm{gnd}})[\xi] \in \mathrm{gnd}(R)$ it holds $\mathcal{P}(\phi_{\mathrm{gnd}}\psi_{\mathrm{gnd}}) = \xi \cdot \mathcal{P}(\psi_{\mathrm{gnd}})$. A knowledge base $\mathcal{R}$ is satisfied by $\mathcal{P}$ iff each conditional in $\mathcal{R}$ is satisfied by $\mathcal{P}$. Usually, there are infinitely many satisfying distributions. Following the principle of maximum entropy, we select the unique distribution $\mathcal{P}^* = \mathrm{ME}(\mathcal{R}) := \arg\max_{\mathcal{P} \models \mathcal{R}} \mathrm{H}(\mathcal{P})$ having maximum entropy [7]. $\mathrm{H}(\mathcal{P}) := -\sum_{\omega \in \Omega} \mathcal{P}(\omega) \log \mathcal{P}(\omega)$ denotes the entropy of $\mathcal{P}$. This process is usually called *ME-inference*.

If we regard the ground atoms in $\mathcal{H}(\mathcal{R})$ as propositional variables, we can regard FO-PCL conditionals as templates for propositional conditionals. We will use this connection in the following, when using results for propositional languages for FO-PCL. In the following, we will abbreviate an FO-PCL conditional $R = \langle(\phi \mid \psi)[\xi], C\rangle$ by

$(\phi \mid \psi)[\xi]$ or $(\phi \mid \psi)$ if its constraint formula or its probability is not needed in the given context. The former is called a quantitative conditional, the latter a qualitative conditional.

Reversing Inductive Knowledge Representation: We briefly sketch the most important theoretical concepts underlying CondorCKD as they are needed in the following. Let $\mathcal{R}$ be a knowledge base consisting of n propositional conditionals $(\phi_i \mid \psi_i)[\xi_i]$. One can show that the entropy-maximal probability distribution satisfying $\mathcal{R}$ can be written as a product of $n+1$ non-negative real values a_i in the following way: $\mathcal{P}(\omega) = a_0 \prod_{i=1}^{n} (\prod_{\omega \models \phi_i\psi_i} a_i^{1-\xi_i} \prod_{\omega \models \overline{\phi_i}\psi_i} a_i^{-\xi_i})$ [5].

a_0 is a normalizing constant and for $i \geq 1$ the factor a_i corresponds to the i-th conditional. Taking into account the exponents, there are basically three effects a conditional can have on the probability of a world. If the antecedence is not satisfied the effect is $1 = a_i^0$. Otherwise, the effect is $a_i^{1-\xi_i}$ if the consequence is also satisfied, resp. $a_i^{-\xi_i}$ if it is not. The effects are called neutral, positive and negative [1]. In [5] for each conditional abstract symbols α^+, α^- are introduced to define the *conditional structure* of worlds with respect to the knowledge base $\mathcal{R}$: $\sigma_{\mathcal{R}}(\omega) = \prod_{i=1}^{n} (\prod_{\omega \models \phi_i\psi_i} \alpha_i^+ \prod_{\omega \models \overline{\phi_i}\psi_i} \alpha_i^-)$. This is just an abstraction of the numerical representation above. If two worlds ω_1, ω_2 have the same conditional structure, i.e. $\sigma_{\mathcal{R}}(\omega_1) = \sigma_{\mathcal{R}}(\omega_2)$, they necessarily have the same probability, i.e. $\mathcal{P}(\omega_1) = \mathcal{P}(\omega_2)$, since their probabilities are constituted by the same product of factors. If conversely the same probability implies the same structure, $\mathcal{P}$ is called a faithful representation of $\mathcal{R}$. For faithful representations, one can construct a knowledge base $\mathcal{R}'$ that represents most informative conditional relationships in $\mathcal{P}$ and can be looked upon as an approximation to $\mathcal{R}$. This is the fundamental idea of inverse knowledge representation [5] underlying CondorCKD [6].

CondorCKD is called with an empirical probability distribution $\mathcal{P}$. It starts with *single-elementary conditionals* of maximal length and shortens these conditionals by resolving algebraical equations over the effects of the current conditional set. Finally, it returns a knowledge base $\mathcal{R}'$ consisting of the remaining shortened conditionals.

RCondorCKD: In [8] an approach is sketched how CondorCKD can be transferred to FO-PCL. Whereas the basic idea of learning follows [5], CondorCKD is refined by taking the relational structure into account. Algorithm 1 from [8] shows the skeleton of our relational version. Instead of considering each single-elementary conditional of

Algorithm 1. RCondorCKD

```
1: procedure RCKD(P, Bias)
2:     R ← createBasicConditionals(Bias)
3:     E ← findEquations(P)
4:     while e ∈ E can be resolved do
5:         Resolve e and decrease R
6:     end while
7:     R ← postprocess(R)
8:     return R
9: end procedure
```

maximal length, we define our basic conditionals with respect to a language bias. Then equations are searched and resolved similar to CondorCKD but by taking the relational structure into account. In the postprocessing step probabilities for ground instances of the learned free conditionals are computed and outliers are identified by statistical means [8]. After splitting off exceptional rules the knowledge base is returned.

Our basic conditional set is defined by means of *template conditonals* inspired by the template language $\mathcal{D}$LAB [9]. In particular, we mainly consider free conditionals in the learning phase containing only variables and a wildcard symbol $*$ instead of a probability. In the postprocessing phase we compute the probabilities of their ground instances and split off exceptional rules [8]. A *template conditional* has the form $T = template((A|A_1A_2 \dots A_k), C)$. A is a single atom, the A_i are atoms different from A, and C is a constraint formula. $\mathcal{R}(T) := \{\langle (A \mid L_1L_2 \dots L_k)[*], C\rangle \mid L_i \in \{A_i, \overline{A_i}\}\}$ is the set of basic conditionals induced by T.

Example 3. For the template $template((al(X,Y)|h(X)\, h(Y)), X \neq Y)$ we obtain the following basic conditionals:

$$\langle (al(X,Y)|h(X)\, h(Y))[*], X \neq Y\rangle, \qquad \langle (al(X,Y)|\overline{h(X)}\, h(Y))[*], X \neq Y\rangle$$
$$\langle (al(X,Y)|h(X)\, \overline{h(Y)})[*], X \neq Y\rangle, \qquad \langle (al(X,Y)|\overline{h(X)}\, \overline{h(Y)})[*], X \neq Y\rangle.$$

Given a set of template conditionals $\mathcal{T}$, our initial basic conditional set is $\mathcal{R} := \bigcup_{T\in\mathcal{T}} \mathcal{R}(T)$. Compared to a complete basic conditional set, template conditionals can decrease the number of basic conditionals significantly. On the other hand, if the template conditionals do not include all conditionals of maximal length, it is not guaranteed that each possible dependency can be captured by the algorithm. Therefore, one has to trade efficiency off for completeness.

3 Resolving Equations

Conditional Structure of FO-PCL Knowledge Bases: As explained before, each FO-PCL conditional can be regarded as a template for several propositional conditionals. Hence, whereas in the propositional case there is one positive and negative effect corresponding to each conditional, FO-PCL conditionals induce several effects, one for each instance. Let $\mathcal{R}$ be an FO-PCL knowledge base consisting of n FO-PCL conditionals $(\phi_i \mid \psi_i)[\xi_i]$. Let the i-th conditional have k_i ground instances, and let $(\phi_{i,j} \mid \psi_{i,j})[\xi_i]$ denote the j-th ground instance of the i-th conditional, $1 \leq j \leq k_i$. Then the entropy-maximal probability distribution satisfying $\mathcal{R}$ factorizes as follows [2]: $\mathcal{P}(\omega) = a_0 \prod_{i=1}^{n} \prod_{j=1}^{k_i} (\prod_{\omega\models\phi_{i,j}\psi_{i,j}} a_{i,j}^{1-\xi_i} \prod_{\omega\models\overline{\phi_{i,j}}\psi_{i,j}} a_{i,j}^{-\xi_i})$. Hence for the i-th conditional there are k_i numerical effects. Now just like in the propositional case, we introduce abstract effects $\alpha_{i,j}^{+}$ for numerical effects $a_{i,j}^{1-\xi_i}$ and abstract effects $\alpha_{i,j}^{-}$ for numerical effects $a_{i,j}^{-\xi_i}$. Then we define the conditional structure of ω with respect to our FO-PCL knowledge base $\mathcal{R}$ to be:

$$\sigma_{\mathcal{R}}(\omega) = \prod_{i=1}^{n} \prod_{j=1}^{k_i} \Big(\prod_{\omega\models\phi_{i,j}\psi_{i,j}} \alpha_{i,j}^{+} \prod_{\omega\models\overline{\phi_{i,j}}\psi_{i,j}} \alpha_{i,j}^{-} \Big). \tag{1}$$

Again, if two worlds ω_1, ω_2 have the same conditional structure, i.e. $\sigma_{\mathcal{R}}(\omega_1) = \sigma_{\mathcal{R}}(\omega_2)$, they necessarily have the same probability, i.e. $\mathcal{P}(\omega_1) = \mathcal{P}(\omega_2)$, since their probabilities are constituted by the same product of factors. However, assuming that the same probability implies the same structure is not reasonable for FO-PCL knowledge bases in general. The reason is that different instances $(\phi_{i,j} \mid \psi_{i,j})[\xi_i]$ of the same conditional $(\phi_i \mid \psi_i)[\xi_i]$ often (but not always) have the same numerical factors [2]. This has to be taken into account in the relational case.

Resolving Equations: When facing a learning problem, we only know the empirical probability distribution $\mathcal{P}$, but do not know about its conditional structure that is induced by an unknown conditional knowledge base $\mathcal{R}^*$. However, we assume that the conditionals in $\mathcal{R}^*$, which we call *solution conditionals* in the following, are generalizations of our basic conditionals. Some basic conditionals might be specializations that have to be generalized. For instance, the qualitative conditionals $(al(X,Y) \mid \overline{h(X)}\,\overline{h(Y)})$ and $(al(X,Y) \mid \overline{h(X)}\,h(Y))$ are specializations of $(al(X,Y) \mid \overline{h(X)})$. Other conditionals might be redundant, i.e., they are not related to conditionals in $\mathcal{R}^*$ and do not affect $\mathcal{P}$. Then we want to delete these conditionals. To learn about the nature of our basic conditionals, we consider the conditional structure with respect to our current basic conditional set $\mathcal{R}$. We are interested in equations over their conditional effects that unveil dependencies between conditionals or redundancies. We give a thorough example later on, but describe the abstract idea here.

Given two equal world probabilities $\mathcal{P}(\omega_1) = \mathcal{P}(\omega_2)$, we can consider the corresponding conditional structure $\sigma_{\mathcal{R}}(\omega_1) = \sigma_{\mathcal{R}}(\omega_2)$ with respect to our current basic conditional set $\mathcal{R}$. The latter equation is an equation over conditional effects of our basic conditional set. We can resolve such equations into subequations over 'independent' effects [5]. To unveil more complex dependencies between conditionals, we consider products of probabilities like $\prod_{i=0}^{n-1} \mathcal{P}(\omega_{2i}) = \prod_{i=0}^{n-1} \mathcal{P}(\omega_{2i+1})$ yielding more complex equations $\prod_{i=0}^{n-1} \sigma_{\mathcal{R}}(\omega_{2i}) = \prod_{i=0}^{n-1} \sigma_{\mathcal{R}}(\omega_{2i+1})$. However, if two basic conditionals are specializations of the same solution conditional, their effects are equal and therefore they are not independent. But we can assume that effects of basic conditionals with different consequence literals are independent, as they cannot be specializations of one solution conditional. In this way we obtain two kinds of shortening operations.

We denote the effects of our basic conditionals by $\alpha_{L,i,j}$, where L denotes the consequence literal, i is an index over basic conditionals with consequence literal L, and j is an index over the ground instances of the i-th basic conditional with consequence literal L. In general we obtain equations of the following form:

$$\prod_L \prod_i \prod_j (\alpha^+_{L,i,j})^{r_{L,i,j}} (\alpha^-_{L,i,j})^{s_{L,i,j}} = \prod_L \prod_i \prod_j (\alpha^+_{L,i,j})^{r'_{L,i,j}} (\alpha^-_{L,i,j})^{s'_{L,i,j}},$$

where $r_{L,i,j}, s_{L,i,j}, r'_{L,i,j}, s'_{L,i,j} \in \mathbb{N}_0$. We can resolve these equations for independent effects of different consequence literals, yielding equations of the form

$$\prod_i \prod_j (\alpha^+_{L,i,j})^{r_{L,i,j}} (\alpha^-_{L,i,j})^{s_{L,i,j}} = \prod_i \prod_j (\alpha^+_{L,i,j})^{r'_{L,i,j}} (\alpha^-_{L,i,j})^{s'_{L,i,j}}$$

for each consequence literal L. Furthermore, we can resolve these equations for positive and negative effects. In this way we obtain for each L two equations

$$\prod_i \prod_j (\alpha^+_{L,i,j})^{r_{L,i,j}} = \prod_i \prod_j (\alpha^+_{L,i,j})^{r'_{L,i,j}} \text{ and}$$
$$\prod_i \prod_j (\alpha^-_{L,i,j})^{s_{L,i,j}} = \prod_i \prod_j (\alpha^-_{L,i,j})^{s'_{L,i,j}}.$$

Note that from a propositional perspective $al(a,b)$ and $al(b,a)$ are different consequence literals. However, we cannot resolve their effects, because, as we explained above, the corresponding numerical effects can be equal in general.

There are two types of resolved equations we are in particular interested in. The first type has the form $a_{i,j} = 1$. It states that the effect of the j-th ground instance of the i-th conditional is equal to the neutral element. Therefore, it has no effect at all and can be deleted. The second type has the form $a^+_{i,j} = a^+_{\iota,l}$, where the j-th ground instance of the i-th and the l-th ground instance of the ι-th conditional have the same consequence literal. The equation states that both conditionals have the same effect on the probability distribution. Therefore, these conditionals can be combined to a single conditional by connecting their antecedences by disjunction. When shortening conditionals their effects are shortened in the same way, i.e., they are deleted or combined to a single effect. In this way further equations can be resolved. We illustrate both cases in Example 4. For a more technical discussion and thorough proofs verifying that equations over independent effects can be resolved and correspondingly conditionals can be shortened as described above, we refer to [5], Chapter 8.

Example 4. The complete probability distribution $\mathcal{P}$ induced by the knowledge base $\mathcal{R}^* = \{\langle (al(X,Y) \mid \overline{h(X)})[0.9], X \neq Y\rangle\}$ is shown in Table 1. Assume we observed this distribution and want to learn a knowledge base without knowing about the conditionals that generated it. We could start with the basic conditional set $\mathcal{R}$ defined in Example 3. As we observed two monkeys a, b, there are eight instances of our basic conditionals. They are listed in Table 2 along with an identifier and their corresponding conditional effects.

Now we start searching for equations. We represent worlds by bit sequences like in Table 1. We find $\mathcal{P}(0010) = 0.0121 = \mathcal{P}(1010)$. By mapping to the conditional

Table 1. ME-optimal probability distribution

$al(a,b)$	$al(b,a)$	$h(a)$	$h(b)$	$\mathcal{P}^*(\omega)$	$al(a,b)$	$al(b,a)$	$h(a)$	$h(b)$	$\mathcal{P}^*(\omega)$
0	0	0	0	0.0017	1	0	0	0	0.0151
0	0	0	1	0.0121	1	0	0	1	0.1088
0	0	1	0	0.0121	1	0	1	0	0.0121
0	0	1	1	0.0873	1	0	1	1	0.0873
0	1	0	0	0.0151	1	1	0	0	0.1355
0	1	0	1	0.0121	1	1	0	1	0.1088
0	1	1	0	0.1088	1	1	1	0	0.1088
0	1	1	1	0.0873	1	1	1	1	0.0873

Table 2. Grounded basic conditionals and corresponding conditional effects

Identifier	Instance $X = a, Y = b$	Effects	Identifier	Instance $X = b, Y = a$	Effects
$c_{0,0}$	$(al(a,b) \mid \overline{h(a)}\,\overline{h(b)})$	$\alpha^+_{0,0}, \alpha^-_{0,0}$	$c_{0,1}$	$(al(b,a) \mid \overline{h(b)}\,\overline{h(a)})$	$\alpha^+_{0,1}, \alpha^-_{0,1}$
$c_{1,0}$	$(al(a,b) \mid \overline{h(a)}\,h(b))$	$\alpha^+_{1,0}, \alpha^-_{1,0}$	$c_{1,1}$	$(al(b,a) \mid \overline{h(b)}\,h(a))$	$\alpha^+_{1,1}, \alpha^-_{1,1}$
$c_{2,0}$	$(al(a,b) \mid h(a)\,\overline{h(b)})$	$\alpha^+_{2,0}, \alpha^-_{2,0}$	$c_{2,1}$	$(al(b,a) \mid h(b)\,\overline{h(a)})$	$\alpha^+_{2,1}, \alpha^-_{2,1}$
$c_{3,0}$	$(al(a,b) \mid h(a)\,h(b))$	$\alpha^+_{3,0}, \alpha^-_{3,0}$	$c_{3,1}$	$(al(b,a) \mid h(b)\,h(a))$	$\alpha^+_{3,1}, \alpha^-_{3,1}$

structure with respect to our basic conditionals we obtain $\alpha^-_{2,0}\alpha^-_{1,1} = \sigma_{\mathcal{R}}(0010) = \sigma_{\mathcal{R}}(1010) = \alpha^+_{2,0}\alpha^-_{1,1}$. As $\alpha^-_{1,1}$ appears on both sides of the equation it cancels out. We obtain $\alpha^-_{2,0} = \alpha^+_{2,0}$. As explained before, we can resolve such equations for positive and negative effects and hence obtain $\alpha^-_{2,0} = 1$ and $\alpha^+_{2,0} = 1$. Hence, the effect of the conditional $c_{2,0}$ is the neutral element and cannot affect the probability distribution. Therefore, it is redundant and can be deleted. We update $\mathcal{R}$ via $\mathcal{R} := \mathcal{R} \setminus \{c_{2,0}\}$. Hence in particular the effects $\alpha^-_{2,0}$ and $\alpha^+_{2,0}$ are not longer contained in the conditional structure $\sigma_{\mathcal{R}}$.

In a similar way the equations $\mathcal{P}(0001) = 0.0121 = \mathcal{P}(0101)$, $\mathcal{P}(0011) = 0.0873 = \mathcal{P}(1011)$ and $\mathcal{P}(0011) = 0.0873 = \mathcal{P}(0111)$ can be used to delete the conditionals $c_{2,1}$, $c_{3,0}$ and $c_{3,1}$ respectively. The corresponding effects are also deleted and do no longer appear in equations.

We also find the more complex equation $\mathcal{P}(0000)\mathcal{P}(1001) = 0.0017 \cdot 0.1088 \approx 0.00018 \approx 0.0121 \cdot 0.0151 = \mathcal{P}(0001)\mathcal{P}(1000)$. As the only remaining conditionals in $\mathcal{R}$ are $c_{0,0}, c_{1,0}, c_{0,1}, c_{1,1}$, we obtain $(\alpha^-_{0,0}\alpha^-_{0,1})(\alpha^+_{1,0}) = \sigma_{\mathcal{R}}(0000)\sigma_{\mathcal{R}}(1001) = \sigma_{\mathcal{R}}(0001)\sigma_{\mathcal{R}}(1000) = (\alpha^-_{1,0})(\alpha^+_{0,0}\alpha^-_{0,1})$. $\alpha^-_{0,1}$ appears on both sides of the equation and hence cancels out. We obtain $\alpha^-_{0,0}\alpha^+_{1,0} = \alpha^-_{1,0}\alpha^+_{0,0}$. We resolve for positive and negative effects and obtain $\alpha^-_{0,0} = \alpha^-_{1,0}$ and $\alpha^+_{1,0} = \alpha^+_{0,0}$. Hence the conditionals $c_{0,0}$ and $c_{1,0}$ have the same conditional effects. That is, they affect the probability distribution in exactly the same way and therefore can be combined to a single conditional by connecting their antecedences by disjunction . We obtain a new conditional $c_0 = (al(a,b) \mid \overline{h(a)}\,\overline{h(b)} \vee \overline{h(a)}\,h(b)) = (al(a,b) \mid \overline{h(a)})$. We update $\mathcal{R}$ via $\mathcal{R} := (\mathcal{R} \setminus \{c_{0,0}, c_{1,0}\}) \cup \{c_0\}$. In the conditional structure $\sigma_{\mathcal{R}}$ the corresponding effects $\alpha^{\pm}_{0,0}, \alpha^{\pm}_{0,1}$ are replaced by $\alpha^{\pm}_0$.

In the same way the equation $\mathcal{P}(0000)\mathcal{P}(0110) = 0.0017 \cdot 0.1088 \approx 0.00018 \approx 0.0121 \cdot 0.0151 = \mathcal{P}(0010)\mathcal{P}(0100)$ can be resolved to combine $c_{0,1}$ and $c_{1,1}$ to $c_1 = (al(b,a) \mid \overline{h(b)})$.

We find no more resolvable equations and end up with two conditionals $(al(a,b) \mid \overline{h(a)})$ and $(al(b,a) \mid \overline{h(b)})$. Except for the probabilities, these are indeed the instances of $\mathcal{R}^*$ that induced $\mathcal{P}$. As explained before, the probabilities of the conditionals are not needed to resolve equations and are computed in a postprocessing step.

Equations between effects of ground instances of the same FO-PCL conditional point to important connections between these ground instances and can justify consolidation of the knowledge base, as we demonstrate in the following example.

Example 5. For the two remaining conditionals from Example 4 we have factors α^+_0, α^-_0 for $(al(a,b) \mid \overline{h(a)})$ and factors α^+_1, α^-_1 for $(al(b,a) \mid \overline{h(b)})$. Furthermore $\mathcal{P}(1000) =$

$0.0151 = \mathcal{P}(0100)$ holds. Mapping to the conditional structure yields $\alpha_0^+\alpha_1^- = \alpha_1^+\alpha_0^-$. Resolving for positive and negative effects yields $\alpha_0^+ = \alpha_1^+$ and $\alpha_0^- = \alpha_1^-$. Indeed, both conditionals are ground instances of $(al(X,Y) \mid \overline{h(X)})$ and have equal numerical effects. This is reflected algebraically by finding $\alpha_0 = \alpha_1$, showing us that we can use $(al(X,Y) \mid \overline{h(X)})$ to represent both ground instances in the knowledge base.

In our algorithm, we do not shorten each ground instance separately. Instead, we save only free conditionals. In Example 4 these are the four conditionals listed in Example 3. When computing the conditional structure one effect for each ground instance has to be regarded just like in Example 4. But instead of shortening each ground instance separately, we regard each resolved equation for a ground instance as an indicator for the general effect of the conditional. That is, we define a treshold τ, say $\tau = 70\%$, and shorten a free conditional if $\tau\%$ of its ground instances can be shortened. Otherwise, due to incomplete or noisy data we might end up with a knowledge base where each ground instance of the original basic conditional is shortened in a different way, such that there is no generalization possible, see [8] for details.

4 Finding Equations

Considering each possible world product to find equations is infeasible and often unnecessary. A rule of thumb to generate interesting equations is to begin with an arbitrary world on the left-hand side of the equation. Then the interpretation of a single atom is 'flipped' to obtain another world on the right hand side. For instance, in Example 4 the equations $\mathcal{P}(0010) = \mathcal{P}(1010)$ and $\mathcal{P}(0011) = \mathcal{P}(1011)$ are obtained by flipping the first bit, which corresponds to the interpretation of $al(a,b)$. In this way only conditional effects of few conditionals are changed and in the corresponding equation of the conditional structure many effects just cancel out, so that often dependencies between the remaining conditional effects are unveiled. If there are many overlapping effects, it is necessary to consider several 'flips'. Such equations can be found in a neighborhood graph [6]. Two worlds are neighbors iff they differ in the interpretation of exactly one atom. In this way, equations correspond to cycles in the graph. If we regard worlds as bit sequences like in Example 4, each edge corresponds to a bit flip.

We compare two worlds using the Hamming distance that is defined by $\Delta(\omega_1,\omega_2) := \sum_{\mathfrak{a}\in\mathcal{H}(\mathcal{R})}(\omega_1(\mathfrak{a}) \oplus \omega_2(\mathfrak{a}))$, where $\oplus$ denotes addition modulo 2. $\Delta(\omega_1,\omega_2)$ corresponds to the number of ground atoms that are verified by one and falsified by the other world. Two worlds are neighbors iff $\Delta(\omega_1,\omega_2) = 1$. The nodes of the neighborhood graph $G = (V,E)$ are the worlds appearing in the dataset, i.e., $V := \{\omega \in \Omega \mid \mathcal{P}(\omega) > 0\}$, and its edges are $E := \{(\omega_1,\omega_2) \mid \Delta(\omega_1,\omega_2) = 1\}$. If $\mathcal{P}$ is strictly positive and there are g ground atoms, G is a g-regular graph, because for each ground atom there is an edge corresponding to a bit flip of the atom. When considering many individuals and many different predicates, there will be a huge number of circles in G and enumerating each circle becomes infeasible even for a restricted circle length. Therefore, it is reasonable to consider subsets of edges that induce subgraphs that still contain interesting circles.

To unveil dependencies for a certain ground instance of a conditional, it is reasonable to flip bits only that correspond to ground atoms included in the ground instance.

For convenience, suppose there is another unary predicate $i(X)$ indicating that a monkey was inactive all day, and there is a conditional $(h(X) \mid i(X))[0.3]$. When flipping the bit for $i(a)$, we change the conditional effect of the instance $(h(a) \mid i(a))[0.3]$. Effects of other instances like $(h(b) \mid i(b))[0.3]$ remain unchanged and therefore cancel out in the corresponding equation. Hence, in the best case only the effect of $(h(a) \mid i(a))[0.3]$ remains in a shortened equation and possibly unveils a dependency. To do not miss interesting equations, we consider edge sets of increasing size. Each edge set regarding only ground atoms containing certain constants. It is likely that we have to regard only small sets, because instances of FO-PCL conditionals usually contain only few constants, even though the set of all constants can be be quite large. For example, instances of $(al(X,Y) \mid \overline{h(X)})$ cannot contain more than two constants.

To begin with, we define a function H mapping sets of constants C to the set of ground atoms in $\mathcal{H}(\mathcal{R})$ that contain a constant from C. More strictly speaking, let $H : 2^{Const} \to 2^{\mathcal{H}(\mathcal{R})}$, $C \mapsto \{\mathfrak{a} \in \mathcal{H}(\mathcal{R}) \mid \mathfrak{a} \text{ contains a constant in } C\}$ for each $C \subseteq Const$. Let $\Delta|_C(\omega_1, \omega_2) := \sum_{\mathfrak{a} \in H(C)} (\omega_1(\mathfrak{a}) \oplus \omega_2(\mathfrak{a}))$ denote the Hamming distance restricted to atoms in $H(C)$. For each $C \subseteq Const$ we define a subset of neighborhood edges $E_C := \{(\omega_1, \omega_2) \in E \mid \Delta|_C(\omega_1, \omega_2) = 1\}$ containing only those edges that change the interpretation of a ground atom containing a constant in C. We start with subgraphs $G_{\{c\}} = (V, E_{\{c\}})$ for each $c \in Const$. The resulting graphs can be significantly smaller in terms of the number of edges and circles.

Example 6. Consider a signature like in Example 2 but with predicate symbols $Pred := \{h(Monkey), i(Monkey)\}$ as described before. Regarding the order $h(a), h(b), i(a), i(b)$ for bit sequences we obtain the graph $G_{\{a\}}$ sketched in Fig. 2. It decomposes into four isomorphic components, of those, however, we show only two. The third graph on the right-hand side can be regarded as a template for the components.

All equations used in Ex. 4 can indeed be found in the corresponding subgraphs $G_{\{a\}}$ and $G_{\{b\}}$. However, in general it might be necessary to consider more than one constant. We built up more complex graphs inductively. Starting with $G_{\{c\}} = (V, E_{\{c\}})$ for all $c \in Const$ we combine them to more complex graphs. Given the graph $G_{\{c_1,\ldots,c_k\}} = (V, E_{\{c_1,\ldots,c_k\}})$ for k constants $\{c_1, \ldots, c_k\}$, we can construct $G_{\{c_1,\ldots,c_k,c_{k+1}\}}$ for a new constant c_{k+1} by adding the edge set of $G_{c_{k+1}}$. That is, we have $G_{\{c_1,\ldots,c_k,c_{k+1}\}} = (V, E_{\{c_1,\ldots,c_k\}} \cup E_{c_{k+1}})$. As k grows, the graphs become more complex. Therefore we consider more complex graphs only if more equations are needed. That is, we start building up $G_{\{c\}}$ for all $c \in Const$, search for cycles and try to build up and resolve

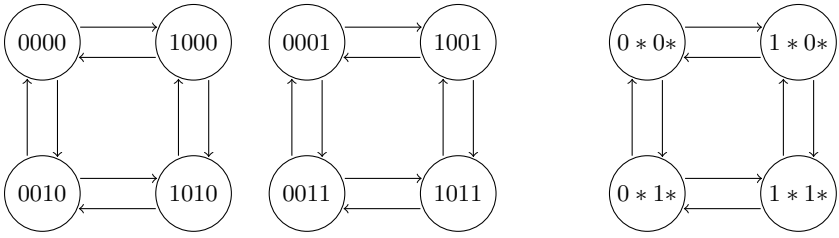


Fig. 2. Connected components of the neighborhood graph $G_{\{a\}}$

equations. If an abortion criterion is met, we stop. Otherwise, we proceed in the same way for the graphs containing one more constant. Appropriate abortion criteria are for example the number of conditionals in the learned knowledge base or the average number of literals in the antecedence.

In Example 6, $G_{\{a\}}$ decomposes into several isomorphic components. Generally, we can capture these components by a single graph over certain equivalence classes as we show in the following. For $C \subseteq \mathit{Const}$ we define $\omega_1 \equiv_C \omega_2$ iff $\omega_1|_C = \omega_2|_C$, where $\omega|_C : H(C) \to \{0,1\}$, $\mathfrak{a} \mapsto \omega(\mathfrak{a})$ for all $\mathfrak{a} \in H(C)$, denotes the restriction of ω to atoms in $H(C)$. Note that $\equiv_C$ is indeed an equivalence relation. The equivalence classes $[\omega]_C = \{\omega' \in V \mid \omega \equiv_C \omega'\}$ provide the nodes for what we call the *collapsed graph*. We can identify each equivalence class $[\omega]_C$ with the representative $\omega|_C$. We connect two equivalence classes $[\omega_1]_C$, $[\omega_2]_C$ by an edge in the graph iff their representatives $\omega_1|_C, \omega_2|_C$ differ in the interpretation of a single atom in $H(C)$. More strictly speaking, we define the collapsed graph with respect to G_C to be $G_{\equiv_C} = (V_{\equiv_C}, E_{\equiv_C}) := (\{[\omega]_C \mid \omega \in V\}, \{([\omega_1]_C, [\omega_2]_C) \mid \Delta|_C(\omega_1|_C, \omega_2|_C) = 1\})$.

The following proposition states that each edge in G_C is captured by an edge in $G_{\equiv_C}$. In particular, nodes in each connected component are equivalent with respect to $\overline{C} = \mathit{Const}\backslash C$. If $V = \Omega$, i.e., if all possible worlds are contained in the neighbordhood graph, the converse also holds. In this case, in particular, each component in G_C is isomorphic to $G_{\equiv_C}$ as observed in Example 6.

Proposition 1. *(Connection between G_C and $G_{\equiv_C}$)*

1. *If $(\omega_1, \omega_2) \in E_C$ then $([\omega_1]_C, [\omega_2]_C) \in E_{\equiv_C}$.*
2. *If $\omega, \omega' \in V$ are connected in G_C then $\omega' \equiv_{\overline{C}} \omega$.*
3. *If $V = \Omega$ and $\omega' \equiv_{\overline{C}} \omega$ then ω and ω' are connected in G_C*
4. *If $V = \Omega$ then each connected component in G_C is isomorphic to $G_{\equiv_C}$.*

Proof. 1. It holds $\Delta|_C(\omega_1, \omega_2) = 1$ by definition of E_C. As $\Delta|_C$ is restricted to C it also holds $\Delta|_C(\omega_1|_C, \omega_2|_C) = 1$. Hence $([\omega_1]_C, [\omega_2]_C) \in E_{\equiv_C}$.

2. The claim follows by contraposition. If $\omega' \not\equiv_{\overline{C}} \omega$ then ω and ω' differ in the interpretation of an atom $\mathfrak{a}$ that contains no constant in C. Hence each path between them contains an edge that flips the interpretation of $\mathfrak{a}$. But this edge is not contained in E_C. Hence ω and ω' are not connected in G_C.

3. As $\omega' \equiv_{\overline{C}} \omega$, they differ only in the interpretation of atoms *At* containing constants in C. As $V = \Omega$ we can construct a path in G_C between them by flipping successively the interpretation of the atoms in *At*.

4. Consider an arbitrary connected component Comp_ω in G_C containing a world $\omega \in V$. According to 2 and 3 Comp_ω contains exactly those $\omega' \in V$ that satisfy $\omega' \equiv_{\overline{C}} \omega$. We define a mapping from nodes in Comp_ω to nodes in $G_{\equiv_C}$ via $f(\omega') = [\omega']_C$. $f(\omega_1') = f(\omega_2')$ implies $\omega_1' \equiv_C \omega_2'$. According to 2 also $\omega_1' \equiv_{\overline{C}} \omega_2'$ holds, hence $\omega_1' = \omega_2'$, i.e., f is injective. A node $[\omega'']_C$ in $G_{\equiv_C}$ is image of the world ω' that is obtained from the representative $\omega''|_C$ by complementing it to a complete interpretation of $\mathcal{H}(\mathcal{R})$ by $\omega|_{\overline{C}}$. Hence f is surjective, i.e., bijective.

Consider two arbitrary worlds ω_1, ω_2 in Comp_ω. If $(\omega_1, \omega_2) \in E_C$ then $(f(\omega_1), f(\omega_2)) = ([\omega_1]_C, [\omega_2]_C) \in E_{\equiv_C}$ according to 1. Conversely, suppose $([\omega_1]_C, [\omega_2]_C) \in E_{\equiv_C}$. Then $\Delta|_C(\omega_1|_C, \omega_2|_C) = 1$. As $\Delta|_C$ is restricted to C it

also holds $\Delta|_C(\omega_1, \omega_2) = 1$. Hence $(\omega_1, \omega_2) \in E_C$. Hence f is a graph isomorphism between $Comp_\omega$ and $G_{\equiv_C}$, i.e., they are isomorphic.

Usually $V = \Omega$ does not hold, instead V is a proper subset of Ω. Then some components will contain less circles than we found in $G_{\equiv_C}$. However, as each edge in G_C is captured by an edge in $G_{\equiv_C}$, each circle in G_C is an 'instance' of a circle in $G_{\equiv_C}$. Therefore, we can search for circles in $G_{\equiv_C}$ and map these circles to the corresponding circles in the connected components in G_C.

5 Discussion

Based on ideas from [5], [6] and [8], we investigated the problem of reversing relational inductive knowledge representation in more detail and explained how finding and resolving of algebraic equations can be implemented. Based on the conditional structure of FO-PCL knowledge bases, we defined shortening operations that can be justified by the algebraic theory developed in [5]. To find equations, justifying these shortening operations, we often do not have to consider the whole neighborhood graph from [6], but can consider subgraphs of increasing complexity. As we saw, these subgraphs decompose into connected components that can be captured by a single 'collapsed graph'. In future work, we will investigate interactions between FO-PCL conditionals in more detail to provide more solid theoretical foundation for our heuristical search approach.

References

[1] DeFinetti, B.: Theory of Probability, vol. 1,2. John Wiley and Sons, New York (1974)
[2] Fisseler, J.: Learning and Modeling with Probabilistic Conditional Logic. Dissertations in Artificial Intelligence, vol. 328. IOS Press, Amsterdam (2010)
[3] Getoor, L., Taskar, B. (eds.): Introduction to Statistical Relational Learning. MIT Press (2007)
[4] Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques. Morgan Kaufmann (2011)
[5] Kern-Isberner, G.: Conditionals in Nonmonotonic Reasoning and Belief Revision. LNCS (LNAI), vol. 2087. Springer, Heidelberg (2001)
[6] Kern-Isberner, G., Fisseler, J.: Knowledge discovery by reversing inductive knowledge representation. In: Proceedings of the Ninth International Conference on the Principles of Knowledge Representation and Reasoning, KR 2004, pp. 34–44. AAAI Press (2004)
[7] Paris, J.B.: The uncertain reasoner's companion – A mathematical perspective. Cambridge University Press (1994)
[8] Potyka, N., Beierle, C.: An approach to learning relational probabilistic FO-PCL knowledge bases. In: Hüllermeier, E., Link, S., Fober, T., Seeger, B. (eds.) SUM 2012. LNCS (LNAI), vol. 7520, pp. 625–632. Springer, Heidelberg (2012)
[9] De Raedt, L., Blockeel, H., Dehaspe, L., Van Laer, W.: Three companions for data mining in first order logic. In: Relational Data Mining, pp. 105–139. Springer (2001)

Analogical Proportions and Multiple-Valued Logics

Henri Prade and Gilles Richard

IRIT University of Toulouse, 118 rte de Narbonne, Toulouse, France
{prade,richard}@irit.fr

Abstract. Recently, a propositional logic modeling of analogical proportions, i.e., statements of the form "A is to B as C is to D", has been proposed, and has then led to introduce new related proportions in a general setting. This framework is well-suited for analogical reasoning and classification tasks about situations described by means of Boolean properties. There is a clear need for extending this approach to deal with the cases where i) properties are gradual ; ii) properties may not apply to some situations ; iii) the truth status of a property is unknown. The paper investigates the appropriate extension in each of these three cases.

Keywords: analogical proportion, multiple-valued logic, three-valued logics.

1 Introduction

Analogy is not a mere question of similarity between two objects (or situations), but rather a matter of proportion or relation between objects. This view dates back to Aristotle and was enforced by Scholastic philosophy. An analogical proportion equates a relation between two objects with the relation between two other objects. These relations can be considered as a symbolic counterpart to the case where the ratio or the difference between two similar things is a matter of degree or number. As such, an analogical proportion of the form "A is to B as C is to D" poses an analogy of proportionality by (implicitly) stating that the way the two objects A and B, otherwise similar, differ is the same way as the two objects C and D, which are similar in some respects, differ.

A propositional logic modeling of analogical proportions viewed as a quaternary connective between the Boolean values of some property pertaining to A, B, C, and D has been proposed in [6]. This logical modeling amounts to precisely state that the difference between A and B is the same as the one between C and D, and that the difference between B and A is the same as the one between D and C. This view can then be proved to be equivalent to state that the considered Boolean property is true for A and D (resp. A or D) each time it is true for B and C (resp. B or C). This latter point shows that a counterpart of a characteristic behavior of numerical geometrical proportions ($\frac{a}{b} = \frac{c}{d}$), or of numerical arithmetic proportions ($a - b = c - d$), namely that the product, or in the second case that the sum, of the extremes is equal to the product (or, in the second case, the sum) of the means, is still observed here.

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 497–509, 2013.

The statement of the equality of numerical ratios, or of numerical differences, is useful for extrapolating a fourth value knowing three others that are linked by such a proportionality relation with it. Similarly, the solving of analogical proportion equations is at the basis of an analogical inference process which is of interest for solving non trivial reasoning tasks (e.g., such as IQ tests [2]), or for dealing with classification problems [5,10]. The underlying inference mechanism considers four Boolean vectors that describe four situations in terms of n binary properties. When an analogical proportion holds for a large number of properties between the four situations, then one makes the plausible inference that an analogical proportion should also hold for a $(n+1)$th property whose truth value is known for 3 of the situations, and unknown for the fourth one, which can thus be obtained as a solution of an analogical proportion equation. But, situations may be more generally described in terms of properties that are not always Boolean. This is the case if the properties are gradual, or if they are binary but may not apply. It may also happen that for some situations it is not known if a property holds or not. In these three types of cases (gradual property, property non applicable, and missing information about a property), it is thus of interest to be still able to evaluate in each case if one may consider that an analogical proportion holds. The paper investigates these three cases where different multiple-valued logical calculi are involved.

The paper is organized as follows. After a short background on Boolean analogical proportions (and two related proportions that play a role in the analysis of the problems encountered) in Section 2, the cases of gradual properties, of non-applicable properties and of unknown properties are successively discussed and contrasted in Sections 3, 4, and 5.

2 Background on Analogical and Related Proportions

A *logical proportion* [8] $T(a,b,c,d)$ is a particular type of Boolean expression involving 4 variables a,b,c,d, with truth values in $\mathbb{B}=\{0,1\}$. It is made of the conjunction of 2 distinct equivalences, involving a conjunction of variables a,b on one side, and a conjunction of variables c,d on the other side of $\equiv$, where each variable may be negated or not. Both $a \wedge \neg b$ and $\neg a \wedge b$ capture the idea of dissimilarity between a and b, while $a \wedge b$ and $\neg a \wedge \neg b$ capture the idea of similarity, positively and negatively. For instance, $(a\overline{b} \equiv c\overline{d}) \wedge (\overline{a}b \equiv \overline{c}d)$[1] is the expression of the *analogical proportion* [6]. As can be seen, analogical proportion uses only dissimilarities and could be informally read as *what is true for a and not for b is exactly what is true for c and not for d, and vice versa*. When a logical proportion does not mix similarities and dissimilarities in its definition, we call it *homogeneous*: For instance, analogical proportion is homogeneous. More generally, it has been proved that there are 120 semantically distinct logical proportions that can be built. Moreover, each logical proportion has exactly 6 lines leading to 1 in its truth table (and the 10 remaining lines lead to 0).

[1] For sake of brevity, $\overline{a}$ is a compact notation for $\neg a$ and $a\overline{b}$ for $a \wedge \neg b$, when useful.

Two properties seem essential for defining the logical proportions that could be considered as the best counterparts to numerical proportions:

- When all the items are identical, the logical proportion should hold true, i.e., the postulate $T(a, a, a, a)$ should be satisfied.
- The validity of a numerical proportion does not depend on the representation of the numbers in a particular basis. In the same spirit, logical proportions should satisfy the so-called *code independency* property: $T(a, b, c, d) \implies T(\overline{a}, \overline{b}, \overline{c}, \overline{d})$ insuring that the proportion T holds whether we encode falsity as 0 (resp. truth as 1) or vice versa.

Only 3 among the 120 proportions satisfy the two previous properties [9]. They are shown in Table 1. They are all homogeneous.

Table 1. 3 remarkable logical proportions: A, R, P

A	**R**	**P**
$a\overline{b} \equiv c\overline{d} \ \wedge \ \overline{a}b \equiv \overline{c}d$	$a\overline{b} \equiv \overline{c}d \ \wedge \ \overline{a}b \equiv c\overline{d}$	$ab \equiv cd \ \wedge \ \overline{a}\overline{b} \equiv \overline{c}\overline{d}$

Their truth tables (restricted to the 6 valuations leading to truth value 1), are derived from their Boolean expressions, and shown in Table 2.

Table 2. A, R, P: Boolean truth tables

A	**R**	**P**
0 0 0 0	0 0 0 0	0 0 0 0
1 1 1 1	1 1 1 1	1 1 1 1
0 0 1 1	0 0 1 1	1 0 0 1
1 1 0 0	1 1 0 0	0 1 1 0
0 1 0 1	0 1 1 0	0 1 0 1
1 0 1 0	1 0 0 1	1 0 1 0

$A(a, b, c, d)$ is the analogical proportion, which expresses that a (resp. b) differs from b (resp. a) as c (resp. d) differs from d (resp. c). $R(a, b, c, d)$ is the *reverse analogical proportion*, where $R(a, b, c, d) = A(a, b, d, c)$ (a is to b as d is to c). $P(a, b, c, d)$ has been named *paralogy* [8] and expresses that what a and b have in common, c and d have it also. Most of the semantical properties of these 3 proportions can be easily checked from their truth tables, and may be viewed as counterparts of properties of the numerical (geometrical) proportion $\frac{a}{b} = \frac{c}{d}$. For instance, the property $\frac{a}{b} = \frac{\frac{1}{b}}{\frac{1}{a}}$ parallels the property $T(a, b, \overline{b}, \overline{a})$ (called *exchange mirroring*) for a logical proportion T where the negation operator plays the role of the inverse. Table 3 summarizes the results: the third column enumerates the proportions among A, R, P satisfying the property respectively named and described in the 1st and 2nd columns. Note that A, R and P satisfy the symmetry property $T(a, b, c, d) = T(c, d, a, b)$: the pairs (a, b) and (c, d) play symmetrical roles. The 2 last lines of Table 3 highlight the strong link between A, R, P. Indeed, there also exists an *equivalent* expression for A that does not involve

Table 3. Boolean properties of A, R, P

Property name	Formal definition	Proportion
full identity	$T(a,a,a,a)$	A,R,P
1-full identity	$T(1,1,1,1) \wedge \neg T(0,0,0,0)$	none
0-full identity	$T(0,0,0,0) \wedge \neg T(1,1,1,1)$	none
reflexivity	$T(a,b,a,b)$	A,P
reverse reflexivity	$T(a,b,b,a)$	R,P
sameness	$T(a,a,b,b)$	A,R
symmetry	$T(a,b,c,d) \rightarrow T(c,d,a,b)$	A,R,P
permutation of means	$T(a,b,c,d) \rightarrow T(a,c,b,d)$	A
permutation of extremes	$T(a,b,c,d) \rightarrow T(d,b,c,a)$	A
all permutations of 2 terms	$\forall i,j, T(a,b,c,d) \rightarrow T(p_{i,j}(a,b,c,d))$	none
transitivity	$T(a,b,c,d) \wedge T(c,d,e,f) \rightarrow T(a,b,e,f)$	A,P
semi-mirroring	$T(a,b,\overline{a},\overline{b})$	R
exchange mirroring	$T(a,b,\overline{b},\overline{a})$	A
negation compatib.	$T(a,\overline{a},b,\overline{b})$	P
link A R	$A(a,b,c,d) \equiv R(a,b,d,c)$	
link A P	$A(a,b,c,d) \equiv P(a,d,c,b)$	

any negation, namely $A(a,b,c,d) = (a \wedge d \equiv b \wedge c) \ \wedge \ (a \vee d \equiv b \vee c)$. It looks like the counterpart of the equality of the product of the extremes and of the product of the means for geometrical numerical proportions. As can be seen from this table, the three proportions A, R, P, and in particular the analogical proportion A, enjoy properties that parallel properties of numerical proportions.

The idea of proportion is closely related to the idea of extrapolation, i.e. to guess / compute a new value on the ground of existing values. In other words, if for some reason, it is believed or known that a proportion should hold between 4 binary items, 3 of them being known, then one may try to infer the value of the 4th one, at least in the case this extrapolation leads to a unique value. For a proportion T, there are exactly 6 distinct valuations for (a,b,c,d) such that $T(a,b,c,d) = 1$[2]. In our context, the problem can be stated as follows. Given a logical proportion T and a 3-tuple (a,b,c), does it exist a Boolean value x such that $T(a,b,c,x) = 1$, and in that case, is this value unique? It is easy to see that there are always cases where the equation has no solution, since the triple a,b,c may take $2^3 = 8$ values, while any proportion T is true only for 6 distinct valuations. For instance, when we deal with analogy A, the equations $A(1,0,0,x)$ and $A(0,1,1,x)$ have no solution. And it can be checked that the analogical equation $A(a,b,c,x)$ is solvable iff $(a \equiv b) \vee (a \equiv c)$ holds. In that case, the unique solution is $x = a \equiv (b \equiv c)$. Similar results hold for R and P.

A, R, P proportions lead to successful applications when applied to reasoning and classification tasks. To cope with real world applications where objects cannot be simply encoded with a unique Boolean value, we need to extend to Boolean vectors what has been done for a single Boolean value. For a given proportion T, the extension to vectors in $\mathbb{B}^n$ is done componentwise as follows:

[2] By abuse of notation, we use the same symbol for a variable and its valuation.

$$T(\overrightarrow{a}, \overrightarrow{b}, \overrightarrow{c}, \overrightarrow{d}) \text{ iff } \forall i \in [1,n], T(a_i, b_i, c_i, d_i)$$

where $\overrightarrow{a} = (a_1, \cdots, a_n)$ and so on. All the previous properties still hold for A, R, P extensions and the equation solving process, when successful, provides a complete Boolean vector instead of a unique Boolean value. In practice, the analogical inference machinery is then based on the idea that if the same logical proportion holds for a number of components of $\overrightarrow{a}, \overrightarrow{b}, \overrightarrow{c}, \overrightarrow{d}$, then it may still hold for a new component known for $\overrightarrow{a}, \overrightarrow{b}, \overrightarrow{c}$, but not for $\overrightarrow{d}$, which can then be extrapolated (see e.g., [8]).

However, this vectorial extension may not still be enough for handling practical problems where we have to deal with missing information or properties whose satisfaction is a matter of levels. To cover such situations, extensions of the Boolean interpretation to multiple-valued logics (3-valued at least) is necessary. At this stage, two questions arise:

1) in a given model, what are the valuations that correspond to a "perfect" proportion of a given type (i.e., having 1 as truth value)? For instance, does $T(a,a,a,a)$ postulate still have to be satisfied by A, R, P or can we consider models where $A(u,u,u,u) = u$, u being a truth value distinct from 0 and 1?

2) are there valuations that could be regarded as "approximate" proportions (i.e. with a truth value distinct from 0 and 1) of a given type and in that case, what is their truth value?

In order to properly answer these two types of questions, we should carefully distinguish between three cases:

- when property satisfaction is a matter of levels or degrees instead of being binary, i.e. the truth value of a given property may be an intermediary value between 0 and 1.
- when property satisfaction does not make sense for a given item, i.e., the property is non applicable to it.
- when information about some properties is missing, i.e., we have no clue about the truth value of some properties for some items.

These are the questions we investigate in the following sections keeping in mind an essential principle: the Boolean model should be the limit case of our models when restricted to Boolean valuations.

3 Gradual Properties

When the satisfaction of properties may be a matter of degree, we have to consider that the truth values belong to a linearly ordered scale $\mathcal{L}$. The simplest case is when $\mathcal{L} = \{0, \alpha, 1\}$, with the ordering $0 < \alpha < 1$, which can be generalized into a finite chain $\mathcal{L} = \{\alpha_0 = 0, \alpha_1, \cdots, \alpha_n = 1\}$ or ordered grades $0 < \alpha_1 < \cdots < 1$, or to an infinite chain using the real interval $[0,1]$. A proposal for extending A in such cases has been advocated in [7]. It takes its source in the expression $A(a,b,c,d) = (a \wedge \neg b \equiv c \wedge \neg d) \wedge (\neg a \wedge b \equiv \neg c \wedge d)$, where now

- i) the central $\wedge$ is taken as equal to min;
- ii) $s \equiv t$ is taken as $\min(s \to_L t, t \to_L s)$ where $\to_L$ is Łukasiewicz implication, defined by $s \to_L t = \min(1, 1 - s + t)$, for $\mathcal{L} = [0, 1]$ (in the discrete cases, we take $\alpha = 1/2$ and $\alpha_i = i/n$), and thus $s \equiv t = 1 - |s - t|$;
- iii) $s \wedge \neg t = \max(0, s - t) = 1 - (s \to_L t)$, i.e. $\wedge\neg$ is understood as expressing a bounded difference.

The resulting expression for $A(a, b, c, d)$ is given in Table 4. Then, we understand the truth value of $A(a, b, c, d)$ as the extent to which the truth values a, b, c, d make an analogical proportion. For instance, in such a graded model, the truth value of $A(0.9, 1, 1, 1) = 0.9$, which fits the intuition. It can be checked that the semantics of $A(a, b, c, d)$ thus defined in the graded case, reduces to the previous definition when restricted to the Boolean case. It is interesting to study in what cases $A(a, b, c, d) = 1$ and in what cases $A(a, b, c, d) = 0$. Then it is clear that $A(a, b, c, d) = 1$ when $a - b = c - d$. When $a, b, c, d \in \{0, \alpha, 1\}$ with $\alpha = 1/2$, it yields the 19 following patterns $(1, 1, 1, 1)$; $(0, 0, 0, 0)$; $(\alpha, \alpha, \alpha, \alpha)$; $(1, 0, 1, 0)$; $(0, 1, 0, 1)$; $(1, \alpha, 1, \alpha)$; $(\alpha, 1, \alpha, 1)$; $(0, \alpha, 0, \alpha)$; $(\alpha, 0, \alpha, 0)$; $(1, 1, 0, 0)$; $(0, 0, 1, 1)$; $(1, 1, \alpha, \alpha)$; $(\alpha, \alpha, 1, 1)$; $(\alpha, \alpha, 0, 0)$; $(0, 0, \alpha, \alpha)$; $(1, \alpha, \alpha, 0)$; $(0, \alpha, \alpha, 1)$; $(\alpha, 1, 0, \alpha)$; $(\alpha, 0, 1, \alpha)$.

This means that $A(a, b, c, d) = 1$ when the change from a to b has the same direction and the same intensity as the change from c to d. However, the last 4 patterns show that there is no need to have $a = b$ and $a = c$ while these conditions hold for the 15 first patterns, which are all of the form (x, y, x, y), (x, x, y, y), or (x, x, x, x). In contrast, note that the last 4 patterns exhibit 3 distinct values.

Table 4. Graded definitions for A, R, P proposed in [7]

$A(a,b,c,d) =$
$1 - \mid (a-b) - (c-d) \mid$ if $a \geq b$ and $c \geq d$, or $a \leq b$ and $c \leq d$
$1 - max(\mid a-b \mid, \mid c-d \mid)$ if $a \leq b$ and $c \geq d$, or $a \geq b$ and $c \leq d$
$R(a,b,c,d) = A(a,b,d,c)$
$P(a,b,c,d) =$
$min(1 - \lvert max(a,b) - max(c,d) \rvert, 1 - \lvert min(a,b) - min(c,d) \rvert)$

$A(a, b, c, d) = 0$ when $a - b = 1$ and $c \leq d$, or $b - a = 1$ and $d \leq c$, or $a \leq b$ and $c - d = 1$, or $b \leq a$ and $d - c = 1$. It means the 22 following patterns in the 3-valued case: $(1, 1, 1, 0)$; $(1, 1, 0, 1)$; $(1, 0, 1, 1)$; $(0, 1, 1, 1)$; $(0, 0, 0, 1)$; $(0, 0, 1, 0)$; $(0, 1, 0, 0)$; $(1, 0, 0, 0)$; $(1, 0, 0, 1)$; $(0, 1, 1, 0)$; $(1, 0, \alpha, \alpha)$; $(0, 1, \alpha, \alpha)$; $(\alpha, \alpha, 1, 0)$; $(\alpha, \alpha, 0, 1)$; $(1, 0, 0, \alpha)$; $(0, 1, 1, \alpha)$; $(1, 0, \alpha, 1)$; $(\alpha, 0, 0, 1)$; $(0, \alpha, 1, 0)$; $(1, \alpha, 0, 1)$; $(0, 1, \alpha, 0)$; $(\alpha, 1, 1, 0)$. Thus, $A(a, b, c, d) = 0$ when the change inside the pairs (a, b) and (c, d) is maximal, while the other pair shows no change or a change in the opposite direction. Thus, $A(a, b, c, d) = \alpha$ for 81 - 19 - 22 = 40 distinct patterns when we use $\mathcal{L} = \{0, \alpha, 1\}$.

In [7], $R(a, b, c, d)$ is defined by permuting c and d in the definition of A, but P is no longer obtained by permuting b and d in the definition of A. In fact, $P(a, b, c, d)$ is defined directly from its definition given in Table 1, changing

$\neg a \wedge \neg b \equiv \neg c \wedge \neg d$ into $a \vee b \equiv c \vee d$, and taking $\wedge = \min$, $\vee = \max$, and $s \equiv t = 1 - |s - t|$, we obtain the definition in Table 4. If we exchange b and d in this definition, we obtain an alternative definition for the graded analogical proportion, namely

$A^*(a, b, c, d) = min(1 - |max(a, d) - max(b, c)|, 1 - |min(a, d) - min(b, c)|)$

This is the direct counterpart of the definition without negation of the analogical proportion in the Boolean case. It can checked that $A^*(a, b, c, d) = 1$ only for the 15 patterns with at most two distinct values for which $A(a, b, c, d) = 1$, while $A^*(a, b, c, d) = \alpha$ for the 4 other patterns for which $A(a, b, c, d) = 1$, namely for $(1, \alpha, \alpha, 0)$; $(0, \alpha, \alpha, 1)$; $(\alpha, 1, 0, \alpha)$; $(\alpha, 0, 1, \alpha)$. Besides, $A^*(a, b, c, d) = 0$ for only 18 among the 22 patterns that make $A(a, b, c, d) = 0$. The 4 patterns for which $A^*(a, b, c, d) = \alpha$ (instead of 0) are $(1, 0, \alpha, \alpha)$; $(0, 1, \alpha, \alpha)$; $(\alpha, \alpha, 1, 0)$; $(\alpha, \alpha, 0, 1)$. Thus, $A^*(a, b, c, d) = \alpha$ for 81 - 15 - 18 = 48 distinct patterns when we use $\mathcal{L} = \{0, \alpha, 1\}$.

Thus, it appears that $A^*(a, b, c, d)$ does not acknowledge as perfect the analogical proportion patterns where the amount of change between a and b is the same as between c and d and has the same direction, but where this change applies in different areas of the truth scale. Still, $A^*(a, b, c, d)$ remains half-true in these cases, for $\mathcal{L} = \{0, \alpha, 1\}$. When $\mathcal{L} = [0, 1]$, it can be checked that $A^*(a, b, c, d) \geq 1/2$ when $a - b = c - d$; in particular, $\forall a, b, A^*(a, b, a, b) = 1$, which corresponds to the case where $a = c$ and $b = d$. In the same spirit, if $\mathcal{L} = \{0, \alpha, 1\}$ as well as for $\mathcal{L} = [0, 1]$, $A^*(a, b, c, d) = 0$ when a change inside the pairs (a,b) and (c,d) is maximal, while the other pair shows a change in the opposite direction starting from 0 or 1. However, $A^*(1, 0, c, c) = min(c, 1 - c)$ and A^* takes the same value for the 7 other permutations of $(1, 0, c, c)$ obtained by applying symmetry and/or central permutation.

As can be seen in Table 5, A^* and A also coincide on some patterns having intermediary truth values, but diverge on others. Generally speaking, A^* is smoother than A in the sense that more patterns have intermediary truth values with A^* than with A. A^* also maintains the link with P, which is no longer true with A. However, it would be possible to define another, maybe less natural, graded paralogy as $P^*(a, b, c, d) = A(a, d, c, b)$. In practice, the graded version A has been used, apparently in a rather successful way, for classification [10], while A^*, which is considered here for the first time, has not been experienced yet. It is still unclear if A^* may be more suitable for classification purposes.

Table 5. The two graded definitions of the analogical proportion in [0, 1]

A	A^*
$A(1, 1, u, v) = 1 - \|u - v\|$	$A^*(1, 1, u, v) = 1 - \|u - v\|$
$A(1, 0, u, v) = u - v$ if $u \geq v$	$A^*(1, 0, u, v) = \min(u, 1 - v)$
$= 0$ if $u \leq v$	
$A(0, 1, u, v) = v - u$ if $u \leq v$	$A^*(0, 1, u, v) = min(v, 1 - u)$
$= 0$ if $u \geq v$	
$A(0, 0, u, v) = A(1, 1, u, v)$	$A^*(0, 0, u, v) = A^*(1, 1, u, v)$

Both A and A^* continue to satisfy the *symmetry property* (as P, R, and P^*, R^* with $R^*(a,b,c,d) = A^*(a,b,d,c) = P^*(a,c,d,b)$). However, only A^* still enjoys the *means permutation* properties and the *extremes permutation* properties. *This is no longer the case with* A, as shown by the following counter-example. $A(0.8, 0.6, 1, 0.3) = 1 - \mid (0.8 - 0.6) - (1 - 0.3) \mid = 1 - \mid 0.2 - 0.7 \mid = 0.5$ since $0.8 \geq 0.6$ and $1 \geq 0.3$, and $A(0.8, 1, 0.6, 0.3) = 1 - max(\mid 0.8 - 1 \mid, \mid 0.6 - 0.3 \mid) = 1 - max(0.2, 0.3) = 0.7$ since $0.8 \leq 1$ and $0.6 \geq 0.3$.

But, *both* A *and* A^* continue to satisfy the *code independency* property with respect to $\overline{a} = 1 - a$. Some more Boolean properties which remain valid in the multiple-valued case are summarized in Table 6.

Table 6. Graded properties of A, A^*, R, P

Property name	Formal definition	Proportion
full identity	$T(a,a,a,a)$	A^*, A, R, P
reflexivity	$T(a,b,a,b)$	A^*, A, P
reverse reflexivity	$T(a,b,b,a)$	R,P
sameness	$T(a,a,b,b)$	A^*, A, R
symmetry	$T(a,b,c,d) \rightarrow T(c,d,a,b)$	A^*, A, R, P
permutation of means	$T(a,b,c,d) \rightarrow T(a,c,b,d)$	A^*
permutation of extremes	$T(a,b,c,d) \rightarrow T(d,b,c,a)$	A^*
all permutations	$\forall i,j, T(a,b,c,d) \rightarrow T(p_{i,j}(a,b,c,d))$	none
semi-mirroring	$T(a,b,\overline{a},\overline{b})$	R
exchange mirroring	$T(a,b,\overline{b},\overline{a})$	A
negation compatib.	$T(a,\overline{a},b,\overline{b})$	none
link A R	$A(a,b,c,d) \equiv R(a,b,d,c)$	
link A P	$A(a,b,c,d) \not\equiv P(a,d,c,b)$	
link A^* P	$A^*(a,b,c,d) \equiv P(a,d,c,b)$	

4 Non-applicable Properties

The abbreviation 'n/a' is currently used in data tables when an attribute does not apply, when a property does not make sense or is *not applicable* for a particular item. However, the extensive use of 'n/a' may be often ambiguous when it also appears in the same tables when information is *not available* for some attribute values of some items. Indeed one has to carefully distinguish the case where the property does apply to the item, but it is not known if the property is true or is false for the item, from the case where the property is neither true nor false for the item since the property does not apply to it. The case of unknown truth values is discussed in the next section, while we now address the problem of dealing with genuinely non applicable properties.

The idea of introducing a third truth value for 'not applicable' (*na* for short in the following) in the context of analogy can be already found in the pioneering work of Sheldon Klein [3,4] who was the first to propose to solve analogical proportion equations $A(a,b,c,x) = 1$, where x is unknown, as $x = c \equiv (a \equiv b)$ (without providing an explicit definition for $A(a,b,c,d)$). However, his handling

of na is based on $(na \equiv na) = na$, which suggests that the evaluation of an analogical proportion where na appears may receive the truth value na, which seems to be more in the spirit of understanding na as 'not available', or 'unknown'.

Indeed, although a property may be 'true', 'false', or 'not applicable' for an item, it seems natural to expect that $A(a, b, c, d)$ can only be 'true' or 'false', since $(1, na, 1, na)$ looks intuitively satisfactory as an analogical proportion, while $(1, na, 0, 0)$ is not. More precisely, the acceptable 4-tuples of valuations that make an analogical proportion true are of the form (x, x, x, x), (x, y, x, y), and (x, x, y, y), where $x, y \in \{0, 1, na\}$, where any other 4-tuple should make it false, since $0, 1$ and na play the same role. This leads to acknowledge as true the 15 following patterns $(1, 1, 1, 1)$; $(0, 0, 0, 0)$; (na, na, na, na); $(1, 0, 1, 0)$; $(0, 1, 0, 1)$; $(1, na, 1, na)$; $(na, 1, na, 1)$; $(0, na, 0, na)$; $(na, 0, na, 0)$; $(1, 1, 0, 0)$; $(0, 0, 1, 1)$; $(1, 1, na, na)$; $(na, na, 1, 1)$; $(na, na, 0, 0)$; $(0, 0, na, na)$, all the others being false.

In other words, we are in a situation somewhat similar to the one encountered in the previous section in the case of a unique intermediary truth-value α between true and false, meaning 'half-true' (or equivalently 'half-false'), when we refuse the four patterns $(1, \alpha, \alpha, 0)$, $(0, \alpha, \alpha, 1)$, $(\alpha, 0, 1, \alpha)$ and $(\alpha, 1, 0, \alpha)$ as being true, *except that* now no pattern has the third truth value. It is possible to find logical definitions of the analogical proportion having the expected behavior for the truth values $\{0, 1, na\}$. First, it can be checked that this is obtained with the following expression

$$A(a, b, c, d) = (a \wedge d \equiv b \wedge c) \wedge (a \vee d \equiv b \vee c)$$

where the $\{0, 1, na\}$ are ordered as the chain $1 > na > 0$ (i.e. $\wedge$ is Kleene conjunction, see, e.g., [1], and $x \equiv y = 1$ if and only if $x = y$, and $x \equiv y = 0$ otherwise.

A counterpart to $A(a, b, c, d) = (a \setminus b \equiv c \setminus d) \wedge (b \setminus a \equiv d \setminus c)$ where $\setminus$ here denotes the Boolean logical connective corresponding to set difference, can also be found. However, since we do not want to have $(1, na, na, 0)$ true, the difference between 1 and na and the difference between na and 0 should not be the same, neither the same as between 1 and 0, nor 1 between 1 for sure. Thus we need 4 distinct values for the difference. This is impossible with 3 truth values! This contrasts with the Boolean case where there are only two possible difference values needed. The solution is then to use *2* connectives for differences:
$x \setminus_1 y = 1$ if $x = 1$ and $y = 0$; $x \setminus_1 y = na$ if $x = 1$ and $y = na$; $x \setminus_1 y = 0$ otherwise;
$x \setminus_2 y = 1$ if $x = 1$ and $y = 0$; $x \setminus_2 y = na$ if $x = na$ and $y = 0$; $x \setminus_2 y = 0$ otherwise.
Then the definition of $A(a, b, c, d)$ becomes

$$(a \setminus_1 b \equiv c \setminus_1 d) \wedge (b \setminus_2 a \equiv d \setminus_2 c) \wedge (a \setminus_2 b \equiv c \setminus_2 d) \wedge (b \setminus_1 a \equiv d \setminus_1 c)$$

where $x \equiv y = 1$ iff $x = y$; $x \equiv y = 0$ otherwise; and $\wedge$ is any conjunction connective that coincides with classical conjunction on $\{0, 1\}$. This definition yields 1 for the 15 expected patterns and is 0 otherwise for the $81 - 15 = 66$ remaining patterns.

It is even possible to find an expression for $A(a, b, c, d)$ where $\setminus_1$ and $\setminus_2$ are expressed in terms of a conjunction and negations, i.e. where $x \setminus_1 y$ is replaced by $x \wedge^* \neg_1(y)$ and $x \setminus_2 y$ is replaced by $x \wedge^* \neg_2(y)$. We obtain a definition for $A(a, b, c, d)$ under the form

$(a \barwedge^* \neg_1 b \equiv c \barwedge^* \neg_1 d) \barwedge^* (b \barwedge^* \neg_2 a \equiv d \barwedge^* \neg_2 c) \barwedge^* (a \barwedge^* \neg_2 b \equiv c \barwedge^* \neg_2 d) \barwedge^* (b \barwedge^* \neg_1 a \equiv d \barwedge^* \neg_1 c)$

where the two negations are Post-like negations defined through a circular ordering of the three truth-values, where the negation of a value is the successor value in the ordering, namely $\neg_1(0) = na; \neg_1(na) = 1; \neg_1(1) = 0$ and $\neg_2(0) = 1; \neg_2(na) = 0; \neg_2(1) = na$. This acknowledges the fact that in some sense these three truth-values play similar roles. The non-standard three-valued conjunction $\wedge^*$, which is defined by

$x \wedge^* y = 1$ if $x = 1$ and $y = 1$
$x \wedge^* y = na$ if $x = na$ and $y = na$
$x \wedge^* y = 0$ otherwise

also agrees with this view, while coinciding with classical conjunction in the binary case. As in the previous section, we summarize in Table 7 the properties of the Boolean case that remain valid in this 3-valued model where na, standing for non applicable, is the third truth value.

Table 7. Properties of A, R, P with truth value na (as non applicable)

Property name	Formal definition	Proportion
full identity	$T(a,a,a,a)$	A,R,P
reflexivity	$T(a,b,a,b)$	A,P
reverse reflexivity	$T(a,b,b,a)$	R,P
sameness	$T(a,a,b,b)$	A,R
symmetry	$T(a,b,c,d) \to T(c,d,a,b)$	A,R,P
permutation of means	$T(a,b,c,d) \to T(a,c,b,d)$	A
permutation of extremes	$T(a,b,c,d) \to T(d,b,c,a)$	A
all permutations	$\forall i,j, T(a,b,c,d) \to T(p_{i,j}(a,b,c,d))$	none
link A R	$A(a,b,c,d) \equiv R(a,b,d,c)$	
link A P	$A(a,b,c,d) \equiv P(a,d,c,b)$	

5 Unknown Properties

In this section, we briefly consider a situation that is quite different from the ones studied in the two previous sections. We assume now that the features used for describing situations are all binary (i.e., they can be only true or false), but their truth value may be unknown. Thus, the possible states of information regarding a Boolean variable x pertaining to a given feature may be represented by one of the 3 truth value subsets $\{0\}, \{1\}$ or $\{0,1\}$, corresponding respectively to the case where the truth value of x is false, true or unknown. We denote this state of information by $\tilde{x}$, which is a subset of $\{0,1\}$. The evaluation of a logical proportion $T(a,b,c,d)$ amounts to compute the state of information denoted $\mathcal{T}(\tilde{a},\tilde{b},\tilde{c},\tilde{d})$ about its truth value, knowing $\tilde{a},\tilde{b},\tilde{c},\tilde{d}$. It is given by the standard set extension where v denotes a Boolean valuation:

$$\mathcal{T}(\tilde{a},\tilde{b},\tilde{c},\tilde{d}) = \{v(T(a,b,c,d)) \mid v(a) \in \tilde{a}, v(b) \in \tilde{b}, v(c) \in \tilde{c}, v(d) \in \tilde{d}\}$$

From now on, we focus on analogical proportion A only, but R, P and I could be handled in a similar manner. For instance, let us take the example $A(a, b, c, d)$ where $\tilde{a} = \{1\}, \tilde{b} = \{0\}, \tilde{c} = \tilde{d} = \{0, 1\}$. Applying the previous formula leads to

$$\mathcal{A}(\tilde{a}, \tilde{b}, \tilde{c}, \tilde{d}) = \{0, 1\}$$

since the truth value of $A(a, b, c, d)$ may be 0 for the valuations 1001, 1000, 1011, and 1 for 1010. If we consider the following expression $A(a, b, a, b)$ when $\tilde{a} = \tilde{b} = \{0, 1\}$, a similar computation leads to

$$\mathcal{A}(\tilde{a}, \tilde{b}, \tilde{a}, \tilde{b}) = \{1\}$$

since the truth value of $A(a, b, a, b)$ is 1 for any of the valuations 1010, 1111, 0101, or 0000. Similarly, the truth value of $A(a, a, a, a)$ is 1, even when $\tilde{a} = \{0, 1\}$. But, the set of possible truth values for $A(a, b, c, d)$ is $\{0, 1\}$ when $\tilde{a} = \{0, 1\}, \tilde{b} = \{0, 1\}, \tilde{c} = \{0, 1\}, \tilde{d} = \{0, 1\}$, i.e. we have the same state of information for all of them. This expresses that the full identity property does not hold any longer at the information level for analogical proportion. And this illustrates the fact that the logic of uncertainty is no longer truth functional, since the state of information about the truth value of $A(a, b, c, d)$ does not only depend on the state of information about the truth values of a, b, c, and d, but is also constrained by the existence of possible logical dependencies between these variables.

Nevertheless, some key properties of homogeneous proportions remain valid at the information level such as symmetry, or central and extreme permutations. Indeed it can be checked that, for instance, for symmetry:

$$\mathcal{A}(\tilde{a}, \tilde{b}, \tilde{c}, \tilde{d}) = \mathcal{A}(\tilde{c}, \tilde{d}, \tilde{a}, \tilde{b})$$

Using the set extension evaluation of logical proportions in presence of incomplete information, we can compute the set of possible truth values of the analogical proportion for the different 4-tuples of states of information. We now denote by u the state $\{0, 1\}$, and respectively by 0 and 1, the states of information $\{0\}$ and $\{1\}$. A 4-tuple of states of information will be called *information pattern*, or pattern for short, and denoted by a 4-tuple of elements of $\{0, 1, u\}$ without blank space. For instance, $01u1$ is such a pattern and should be understood as the 4-tuple of states of information $(\{0\}, \{1\}, \{0, 1\}, \{1\})$.

Then, the 6 patterns $0000, 1111, 0011, 1100, 1010, 0101$ that makes A true in the Boolean case, and where u does not appear, are the only ones that are still true with the above view (for which we get the singleton $\{1\}$ as information state for $A(a, b, c, d)$). As soon as at least one state of information is u in the pattern, the state of information for $A(a, b, c, d)$ is u or 0. Indeed, for instance, $01u0$ leads to 0 since whatever the truth value of the 3rd variable, the analogical proportion does not hold. Thus, despite the lack of knowledge regarding the 3rd variable, we know the exact truth value of the proportion in this case, namely it is false. It appears that there are 18 patterns that lead to 0. They are the 10 patterns of the Boolean case and the 8 following ones: $01u0$, $0u10$, $u001$, $100u$, $10u1$, $1u01$, $u110$, $011u$. Thus, in the $81 - 6 - 18 = 57$ remaining cases, the state of information for $A(a, b, c, d)$ is u.

It can be checked that these results can be retrieved both with the initial definition of A or with A^* where complete ignorance u is handled with $^-$, $\wedge, \vee$ as the strong Kleene connectives (see [1]) and $\equiv$ as Bochvar connective, where u is an absorbing element. The corresponding truth tables are recalled in Table 8. This provides a way to extend the definition of the analogical proportion in

Table 8. Truth tables for u as lack of knowledge

$^-$	
0	1
1	0
u	u

$\wedge$	0	1	u
0	0	0	0
1	0	1	u
u	0	u	u

$\vee$	0	1	u
0	0	1	u
1	1	1	1
u	u	1	u

$\equiv$	0	1	u
0	1	0	u
1	0	1	u
u	u	u	u

case of lack of knowledge when no dependencies between the variables exist. As in the Boolean case, the definitions A (resp. R, P, I) and A^* (resp. R^*, P^*, I^*) are equivalent. Nevertheless, this truth-functional calculus provides only a description of the evaluation of the patterns at the information level. Namely, it enables us to retrieve the tri-partition of the patterns in respectively 6, 18 and 57 patterns leading respectively to 1, 0 and u, but it does not account for the full calculus of the extended definition of logical proportions in presence of incomplete information, when dependencies take place between variables, for instance it can be checked that $A(a, b, a, b)$ and $A^*(a, b, a, b)$, when a and b are unknown, does not yield 1 as expected, but u (this is just due to the fact that constraints $a = c$ and $b = d$ are ignored).

6 Concluding Remarks

This paper has discussed three extensions of the notion of analogical proportions (and related logical proportions) by carefully distinguishing the problems of handling graded truth values, of dealing with non applicable properties, and of coping with unknown truth values. In each case, a different modeling has been obtained with a different repartition of the patterns found to be true, false, or having another value, and where the set of properties preserved for the analogical proportion is not the same. More generally, it would be of interest of developing an approach where the three types of problem can be handled together.

References

1. Ciucci, D., Dubois, D.: Relationships between connectives in three-valued logics. In: Greco, S., Bouchon-Meunier, B., Coletti, G., Fedrizzi, M., Matarazzo, B., Yager, R.R. (eds.) IPMU 2012, Part I. CCIS, vol. 297, pp. 633–642. Springer, Heidelberg (2012)
2. Correa, W., Prade, H., Richard, G.: When intelligence is just a matter of copying. In: De Raedt, L., et al. (eds.) Proc. 20th Europ. Conf. on Artificial Intelligence, Montpellier, August 27-31, pp. 276–281. IOS Press (2012)

3. Klein, S.: Culture, mysticism & social structure and the calculation of behavior. In: Proc. 5th Eur. Conf. in AI (ECAI 1982), Paris, pp. 141–146 (1982)
4. Klein, S.: Analogy and mysticism and the structure of culture (and Comments & Reply). Current Anthropology 24(2), 151–180 (1983)
5. Miclet, L., Bayoudh, S., Delhay, A.: Analogical dissimilarity: definition, algorithms and two experiments in machine learning. JAIR 32, 793–824 (2008)
6. Miclet, L., Prade, H.: Handling analogical proportions in classical logic and fuzzy logics settings. In: Sossai, C., Chemello, G. (eds.) ECSQARU 2009. LNCS, vol. 5590, pp. 638–650. Springer, Heidelberg (2009)
7. Prade, H., Richard, G.: Multiple-valued logic interpretations of analogical, reverse analogical, and paralogical proportions. In: Proc. 40th IEEE Inter. Symp. on Multiple-Valued Logic (ISMVL 2010), Barcelona, pp. 258–263 (May 2010)
8. Prade, H., Richard, G.: Reasoning with logical proportions. In: Lin, F.Z., Sattler, U., Truszczynski, M. (eds.) Proc. 12th Inter. Conf. on Principles of Knowledge Representation and Reasoning (KR 2010), Toronto, Ontario, Canada, May 9-13, pp. 545–555. AAAI Press (2010)
9. Prade, H., Richard, G.: Homogeneous logical proportions: Their uniqueness and their role in similarity-based prediction. In: Brewka, G., Eiter, T., McIlraith, S.A. (eds.) Proc. 13th Inter. Conf. on Principles of Knowledge Representation and Reasoning (KR 2012), Roma, June 10-14, pp. 402–412. AAAI Press (2012)
10. Prade, H., Richard, G., Yao, B.: Enforcing regularity by means of analogy-related proportions-a new approach to classification. International Journal of Computer Information Systems and Industrial Management Applications 4, 648–658 (2012)

Chain Graph Interpretations and Their Relations

Dag Sonntag and Jose M. Peña

ADIT, IDA, Linköping University, Sweden
{dag.sonntag,jose.m.pena}@liu.se

Abstract. This paper deals with different chain graph interpretations and the relations between them in terms of representable independence models. Specifically, we study the Lauritzen-Wermuth-Frydenberg, Andersson-Madigan-Pearlman and multivariate regression interpretations and present the necessary and sufficient conditions for when a chain graph of one interpretation can be perfectly translated into a chain graph of another interpretation. Moreover we also present a feasible split for the Andersson-Madigan-Pearlman interpretation with similar features as the feasible splits presented for the other two interpretations.

Keywords: Chain Graphs, Lauritzen-Wermuth-Frydenberg interpretation, Andersson-Madigan-Pearlman interpretation, multivariate regression interpretation.

1 Introduction

Today there exist mainly three interpretations of chain graphs (CGs). These are the Lauritzen-Wermuth-Frydenberg (LWF) interpretation presented by Lauritzen, Wermuth and Frydenberg in the late eighties [6,7], the Andersson-Madigan-Pearlman (AMP) interpretation presented by Anderson, Madigan and Pearlman in 2001 [2] and the multivariate regression (MVR) interpretation presented by Cox and Wermuth in the nineties [3,4]. A fourth interpretation of CGs can also be found in a study by Drton [5] but this interpretation has not been further studied and will not be discussed in this paper.

Each interpretation has a different separation criterion and do therefore represent different independence models. So far most papers have studied the different interpretations independently with a few exceptions such as the study of discrete CG models by Drton [5] and the study of CGs representing Gaussian distributions by Wermuth et al. [12]. Therefore it has not really been studied what differences and similarities that exist between the different interpretations in terms of representable independence models. Andersson et al. made a small study of this when they presented their new (AMP) interpretation and managed to show when the independence model of a CG of the AMP interpretation could be represented perfectly by a CG of the LWF interpretation. They did however not show when the opposite held and did no comparison with CGs of the MVR interpretation. Wermuth and Sadeghi did on the other hand present conditions for when a CG of the MVR interpretation could be translated into a CG of the LWF or AMP interpretation when they introduced regression graphs [11].

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 510–521, 2013.

The conditions were however only necessary and sufficient if the two CGs contained the same connectivity components and not the more general case where the CGs could take any form.

In this paper we hope to fill this gap and hence the main contribution of this paper is a table where we show the necessary and sufficient conditions for when a CG of one interpretation can be perfectly translated into a CG of another interpretation. First we do however define a feasible split for the AMP interpretation, with similar features as the feasible splits shown for the LWF [10] and MVR [9] interpretation, that are used in these conditions. Hence this is our second contribution. Finally we also show that there for all three CG interpretations exists a minimal set of non-directed edges for each Markov equivalence class and that the CG containing these, and only these, non-directed edges can be reached through repeated feasible splits from any member of the class.

The remainder of the article is organized as follows. In the next section we present the notation we will use throughout the article. This is followed by the definitions of the feasible splits for each interpretation as well as the proof that the feasible split for CGs of the AMP interpretation is sound. In section 4 we start by presenting the conditions of when a CG of one interpretation can be perfectly represented by a CG of another interpretation. This is then followed by the proofs that these conditions are sound.

2 Notation

All graphs are defined over a finite set of variables V.

If a graph G contains an edge between two nodes V_1 and V_2, we denote with $V_1 \rightarrow V_2$ a *directed edge*, with $V_1 \leftrightarrow V_2$ a *bidirected edge* and with $V_1 - V_2$ an *undirected edge*. By $V_1 \circ\!\!\rightarrow V_2$ we mean that either $V_1 \rightarrow V_2$ or $V_1 \leftrightarrow V_2$ is in G. By $V_1 -\!\!\circ V_2$ we mean that either $V_1 \rightarrow V_2$ or $V_1 - V_2$ is in G. By $V_1 \circ\!\!-\!\!\circ V_2$ we mean that there exists an edge between V_1 and V_2 in G while we with $V_1 \cdots V_2$ mean that there might or might not exist an edge between V_1 and V_2. By a *non-directed* edge we mean either a bidirected edge or an undirected edge. A set of nodes is said to be *complete* if there exist edges between all pairs of nodes in the set.

The *parents* of a set of nodes X of G is the set $pa_G(X) = \{V_1 | V_1 \rightarrow V_2$ is in G, $V_1 \notin X$ and $V_2 \in X\}$. The *children* of X is the set $ch_G(X) = \{V_1 | V_2 \rightarrow V_1$ is in G, $V_1 \notin X$ and $V_2 \in X\}$. The *spouses* of X is the set $sp_G(X) = \{V_1 | V_1 \leftrightarrow V_2$ is in G, $V_1 \notin X$ and $V_2 \in X\}$. The *neighbours* of X is the set $nb_G(X) = \{V_1 | V_1 - V_2$ is in G, $V_1 \notin X$ and $V_2 \in X\}$. The *boundary* of X is the set $bd_G(X) = pa_G(X) \cup nb_G(X) \cup sp_G(X)$. The *adjacents* of X is the set $ad_G(X) = \{V_1 | V_1 \rightarrow V_2, V_1 \leftarrow V_2, V_1 \leftrightarrow V_2$ or $V_1 - V_2$ is in G, $V_1 \notin X$ and $V_2 \in X\}$.

A *route* from a node V_1 to a node V_n in G is a sequence of nodes $V_1, \ldots, V_n$ such that $V_i \in ad_G(V_{i+1})$ for all $1 \leq i < n$. A *path* is a route containing only distinct nodes. The length of a path is the number of edges in the path. A path is called a *cycle* if $V_n = V_1$. A path is *descending* if $V_i \in pa_G(V_{i+1}) \cup sp_G(V_{i+1}) \cup nb_G(V_{i+1})$ for all $1 \leq i < n$. A path $\pi = V_1, \ldots, V_n$ is *minimal* if there exists no other path π_2 between V_1 and V_n st $\pi_2 \subset \pi$ holds. The *descendants* of a set of nodes X of G is the set $de_G(X) = \{V_n |$ there is a descending path from V_1 to V_n

in G, $V_1 \in X$ and $V_n \notin X\}$. A path is *strictly descending* if $V_i \in pa_G(V_{i+1})$ for all $1 \leq i < n$. The *strict descendants* of a set of nodes X of G is the set $sde_G(X) = \{V_n|$ there is a strict descending path from V_1 to V_n in G, $V_1 \in X$ and $V_n \notin X\}$. The *ancestors* (resp. *strict ancestors*) of X is the set $an_G(X) = \{V_1|V_n \in de_G(V_1), V_1 \notin X, V_n \in X\}$ (resp. $san_G(X) = \{V_1|V_n \in sde_G(V_1), V_1 \notin X, V_n \in X\}$). A cycle is called a *semi-directed cycle* if it is descending and $V_i \rightarrow V_{i+1}$ is in G for some $1 \leq i < n$. A CG under the Lauritzen-Wermuth-Frydenberg (LWF) interpretation, denoted LWF CG, contains only directed and undirected edges but no semi-directed cycles. Likewise a CG under the Andersson-Madigan-Perlman (AMP) interpretation, denoted AMP CG, is a graph containing only directed and undirected edges but no semi-directed cycles. A CG under the multivariate regression (MVR) interpretation, denoted MVR CG, is a graph containing only directed and bidirected edges but no semi-directed cycles. A *connectivity component* C of a LWF CG or an AMP CG (resp. MVR CG) is a maximal (wrt set inclusion) set of nodes such that there exists a path between every pair of nodes in C containing only undirected edges (resp. bidirected edges). We denote the set of all connectivity components in a CG G by $cc(G)$ and the component to which a set of nodes X belong in G by $co_G(X)$. A *subgraph* of G is a subset of nodes and edges in G. A subgraph of G induced by a set of its nodes X is the graph over X that has all and only the edges in G whose both ends are in X. A *bidirected flag* is an induced subgraph of the form $X \leftrightarrow Y \leftrightarrow Z$ in a MVR CG. With the moral closure graph of a component C in a LWF CG G, denoted $(G_{cl(C)})^m$, we mean the subgraph of G induced by $C \cup pa_G(C)$ where every edge have been made undirected and every pair of nodes in $pa_G(C)$ have been made adjacent with undirected edges.

Let X, Y and Z denote three disjoint subsets of V. We say that X *separated* from Y given Z denoted as $X \perp_G Y|Z$ if the following criteria is met: If G is a LWF CG then X and Y are separated given Z iff there exists no route between X and Y such that every node in a non-collider section on the route is not in Z and some node in every collider section on the route is in Z. A *section* of a route is a maximal non-empty set of nodes $B_1 ... B_n$ such that the route contains the subpath $B_1 - B_2 - \ldots - B_n$. It is called a *collider section* if $B_1 \ldots B_n$ together with the two neighbouring nodes in the route, A and C, form the subpath $A \rightarrow B_1 - B_2 - \ldots - B_n \leftarrow C$. For any other configuration the section is a non-collider section. If G is an AMP CG then X and Y is separated given Z iff there exists no S-open path between X and Y. A path is said to be *S-open* iff every non-head-no-tail node on the path is not in Z and every head-no-tail node on the path is in Z or $san_G(Z)$. A node B is said to be a *head-no-tail* in an AMP CG G between two nodes A and C on a path if one of the following configurations exists in G: $A \rightarrow B \leftarrow C$, $A \rightarrow B - C$ or $A - B \leftarrow C$. Moreover G is also said to contain a triplex $(\{A, C\}, B)$ iff one such configuration exists in G and A and C are not adjacent in G. For any other configuration the node B is a non-collider. If G is a MVR CG then X and Y are separated given Z iff there exists no d-connecting path between X and Y. A path is said to be *d-connecting* iff every non-collider on the path is not in Z and every collider on the path is

in Z or $san_G(Z)$. A node B is said to be a *collider* in a MVR CG G between two nodes A and C on a path if one of the following configurations exists in G: $A \rightarrow B \leftarrow C$, $A \rightarrow B \leftrightarrow C$,$A \leftrightarrow B \leftarrow C$ or $A \leftrightarrow B \leftrightarrow C$. For any other configuration the node B is a non-collider.

The *independence model* M induced by a graph G, denoted as $I(G)$ or $I_{PGM-class}(G)$, is the set of separation statements $X \perp_G Y|Z$ that hold in G according to the interpretation to which G belongs or the subscripted PGM-class. We say that two graphs G and H are *Markov equivalent* (under the same interpretation) or that they are in the same *Markov equivalence class* iff $I(G) = I(H)$.

3 Feasible Splits

For the LWF and MVR interpretation, operations for altering a CG structure without changing its Markov equivalence class have been presented [9,10]. One such operation is called feasible split and is in this article used to prove certain theorems. Hence we repeat the definitions here. Moreover, we also present the corresponding operation, called feasible split for AMP CGs, for the AMP CG interpretation and prove that it is sound. Note that this is not the inverse operation to a legal merging presented in the deflagging procedure for AMP CGs by Roverto and Studený [8]. Their operation was applied to so called strong equivalence classes, not the more general Markov equivalence classes used here.

Definition 1. *Feasible split for LWF CGs [10]*
A connectivity component C of CG G under the LWF interpretation can be feasibly split into two disjoint sets U and L st $U \cup L = C$ by replacing every undirected edge between U and L with a directed edge orientated towards L iff:
1. $\forall A \in ne_G(L) \cap U, pa_G(L) \subseteq pa_G(A)$
2. $ne_G(L) \cap U$ *is complete*

Definition 2. *Feasible split for AMP CGs*
A connectivity component C of CG G under the AMP interpretation can be feasibly split into two disjoint sets U and L st $U \cup L = C$ by replacing every undirected edge between U and L with a directed edge orientated towards L iff:
1. $\forall A \in ne_G(L) \cap U, L \subseteq ne_G(A)$
2. $ne_G(L) \cap U$ *is complete*
3. $\forall B \in L, pa_G(ne_G(L) \cap U) \subseteq pa_G(B)$

Definition 3. *Feasible split for MVR CGs [9]*
A connectivity component C of CG G under the MVR interpretation can be feasible split into two disjoint sets U and L st $U \cup L = C$ by replacing every bidirected edge between U and L with a directed edge orientated towards L iff:
1. $\forall A \in sp_G(U) \cap L$, $U \subseteq sp_G(A)$ *holds*
2. $\forall A \in sp_G(U) \cap L$, $pa_G(U) \subseteq pa_G(A)$ *holds*
3. $\forall B \in sp_G(L) \cap U$, $sp_G(B) \cap L$ *is a complete set*

Definition 4. *Maximally orientated CG*
A CG G (under any interpretation) is maximally orientated iff no feasible splits can be performed on G.

Lemma 1. *A CG G of the AMP interpretation is in the same Markov equivalence class before and after a feasible split.*

Proof. Assume the contrary. Let G be a CG under the AMP interpretations and G' a graph st G' is G with a feasible split performed upon it. G and G' are in different Markov equivalence classes or G' is not a CG under the AMP interpretation iff (1) G and G' does not have the same adjacencies, (2) G and G' does not have the same triplexes or (3) G' contains semi-directed cycles.

First it is clear that G and G' contains the same adjacencies since a feasible split does not change the adjacencies of any node in G. Secondly let us assume G and G' does not have the same triplexes. First let us assume that G' contains a triplex $(\{X, Y\}, Z$ that does not exist in G. It is clear that such a triplex can only occur if $Z \in L$ since the only difference between G and G' is that G' contains some directed edges orientated towards L where G contains undirected edges. It is clear that if the triplex is a flag then the one of the node X or Y, let's say X, must be in U and the other one, let's say Y, must be in L. However, according to condition 1 Y must be adjacent to X which causes a contradiction. If the triplex is not a flag both X and Y must be in U. They also have to be in $ne_G(L)$, which, together with condition 2, contradicts that they are not adjacent. Hence we have a contradiction for that G' contains a triplex that does not exist in G.

Secondly assume G contains a triplex $(\{X, Y\}, Z)$ that does not exist in G'. It is clear that this new triplex can not be over a node in L since these nodes only have edges orientated towards them. Instead assume $Z \in U$. This gives that one of the nodes X or Y, let's say X, must be a parent of Z and the other, let's say Y, must be in L. This does however contradict condition 3, since every parent of Z also must be a parent of Y, and hence X and Y must be adjacent. This gives us a contradiction.

Finally assume G' contain a semi-directed cycle. This means there exists two nodes X and Y st $X \in pa_{G'}(Y)$ but $X \in de_{G'}(Y) \cup co_{G'}(Y)$. It is clear that $\forall A \in V\ de_{G'}(A) \subseteq de_G(A)$ and $co_{G'}(A) \subseteq co_G(A)$ hold. Hence we must have that $X \in de_G(Y) \cup co_G(Y)$ also hold which, together with $\forall B \in V \setminus L\ pa_{G'}(B) = pa_G(B)$, means that Y is in L and since $\forall D \in L\ pa_{G'}(D) = pa_G(D) \cup U$ holds X must be in U. However, at the same time $co_{G'}(Y) = co_G Y \setminus U$ and $de_{G'}(Y) \subseteq de_G Y$ must hold and hence we have a contradiction.

A maximally orientated CG can be obtained from any member of its Markov equivalence class by performing feasible splits until no more feasible splits can be performed.

Theorem 1. *A CG (under any interpretation) has the minimal set of non-directed edges for its Markov equivalence class if no feasible split is possible.*

The following theorem shows that there may exist several maximally orientated CGs in a given Markov equivalence class but all of them share the same non-directed edges.

Theorem 2. *For any Markov equivalence class of CGs (under any interpretation), there exists a unique minimal (wrt inclusion) set of non-directed edges that is shared by all members of the class.*

The proofs of the Theorem 1 and 2 for the MVR interpretation can be found in the article by Sonntag and Peña [9]. These proofs can easily be adapted for the LWF and AMP interpretations.

4 Translations between Interpretations

In this section the main result of this paper is presented, namely what the conditions are for a CG of one interpretation to be possible to translate into a CG of another interpretation. With translate we mean that the induced independence model of a CG of one interpretation can be represented perfectly by a CG of another interpretation. A summary of these results is presented in Table 1.

Table 1. Given a CG G of the interpretation denoted in the row, and a maximally oriented CG G' in the Markov equivalence class of G, there exists a CG H of the interpretation denoted in the column st G and H are Markov equivalent iff the condition in the intersecting cell is fulfilled

	LWF	AMP	MVR
LWF	-	Unidentified	$(G_{cl(K)})^m$ is chordal for all $K \in cc(G)$.
AMP	G contains no k-biflag where $k \geq 2$ [2]	-	G' does not contain any induced subgraph of the form $X-Y-Z$
MVR	G' contains no bidirected edge	G' contains no bidirected flag	-

From the table two things can be noted. First that the conditions given in the table may include a maximally oriented CG G' in the same equivalence class as G. This is done for several reasons. First, such a graph is easy and computationally simple to find. Secondly this allows the proofs to be based on the idea that no feasible split is possible for the interpretation in mind. Third and last the search space of CGs is smaller and more assumptions can be made on the CG. This in turn allows for more efficient algorithms when calculating if the condition holds for some CG. The second note that can be made is that there still does not exist any necessary and sufficient condition for when a perfect translation of a LWF CG G into an AMP CG H is possible. Andersson et al. gave a necessary condition but also showed that this condition was not sufficient [2]. We have managed to prove the necessity of more elaborate conditions but still been unable to prove sufficiency for these. Hence this condition is left for future work.

The rest of this section contains the theorems stating the conditions shown in Table 1 together with their proofs. Some of the proofs (Lemmas 6 and 7) are rather technical and we omit these due to page limitations. These proofs can be found at *www.ida.liu.se/∼jospe/ecsqaru13extended.pdf*.

4.1 Translation of MVR CGs to AMP CGs

Theorem 3. *Given a MVR CG G, and a maximally oriented MVR CG G' in the Markov equivalence class of G, there exists an AMP CG H st $I_{MVR}(G) = I_{AMP}(H)$ iff G' contains no bidirected flag.*

Proof. Sufficiency follows from from Lemmas 4 and 5 and necessity follows from Lemma 2.

Lemma 2. *A MVR CG G and an AMP CG H with the same structure, except that every bidirected edge in G is replaced by a undirected edge in H and where G contains no bidirected flag, represent the same independence model.*

Proof. Assume to contrary that there exists two CGs, G under the MVR interpretation and H under the AMP interpretation, st G does not contain any bidirected flag, i.e induced subgraph of the form $X \leftrightarrow Y \leftrightarrow Z$, G and H contain the same directed edges, and for every bidirected edge in G H has an undirected edge instead (and only contains those undirected edges) but $I_{MVR}(G) \neq I_{AMP}(H)$. Clearly we must have $V_G = V_H$ and that $adj_G(X) = adj_H(X), pa_G(X) = pa_H(X)$ and $co_G(X) = co_H(X)$ holds for all $X \in V_G$. Given the definition of strict descendants $san_G(X) = san_H(X)$ must also hold. Moreover note that H can not contain any induced subgraph of the form $X - Y - Z$. Finally note that both G and H contains the same paths between X and Y.

For $I(G) \neq I(H)$ to hold there has to exist a path π in G (resp. H) that is d-connecting (resp. S-open) st there exist no path in H (resp. G) that is S-open (resp. d-connecting). Let π be a minimal d-connecting (resp. S-open) path in G (resp. H). Note that π can not contain any contain any subpath of the form $V_1 \leftrightarrow V_2 \leftrightarrow V_3$ (resp. $V_1 - V_2 - V_3$) since the edge $V_1 \leftrightarrow V_3$ (resp. $V_1 - V_3$) must exist in G (resp. H) or G contains a bidirected flag or semi-directed cycle. This in turn would mean that π is not minimal since the path $\pi \setminus V_2$ also must be d-connecting and shorter than π. For π to be both d-connecting and S-open for any set of nodes Z it must contain the same colliders and head-no-tail nodes. A node $W \in \pi$ is a collider if it is part of the following configurations of edges in π (1) $\rightarrow W \leftarrow$, (2) $\leftrightarrow W \leftarrow$, (3) $\rightarrow W \leftrightarrow$ and (4) $\leftrightarrow W \leftrightarrow$. Clearly the fourth case can not occur. Case 1-3 would be translated into (1) $\rightarrow W \leftarrow$, (2) $- W \leftarrow$, (3)$\rightarrow W -$ in H which are all (and the only) head-no-tail configurations. Hence π must be d-connecting in G iff π is S-open in H which contradicts the assumption.

Lemma 3. *If a maximally oriented CG G of the MVR interpretation contains a bidirected flag $X \leftrightarrow Y \leftrightarrow Z$ then G also contains an induced subgraph of the form shown in (1) Figure 1a or (2) 1b or (3) $P \circ\!\!\rightarrow Q \leftrightarrow Y \leftrightarrow Z$ or (4) $P \circ\!\!\rightarrow Q \leftrightarrow W \leftrightarrow Z$ st $bd_G(Q) \subseteq bd_G(Y) \cup Y$ and $Y \in sp_G(Q)$ hold.*

Proof. Assume the contrary, that no such induced subgraph exists in G even though G contains a bidirected flag and G is maximally orientated. Let C be the component of which X, Y and Z belongs. Let A be the set of nodes A_k st $A_k \in sp_G(Y)$ but $A_k \notin sp_G(Z)$. We know that X fulfills these criteria and hence $|A| \geq 1$.

First note that if there exists a node $A_k \in A$ st $bd_G(A_k) \not\subseteq bd_G(Y) \cup Y$ then there exists an induced subgraph $P \circ\!\!\rightarrow A_k \leftrightarrow Y \leftrightarrow Z \cdots P$ in G for some node $P \in bd_G(A_k) \setminus bd_G(Y) \setminus Y$. Hence we have a contradiction since G either contains an induced subgraph of the form shown in Figure 1b ($P \in bd_G(Z)$) or of the form $P \circ\!\!\rightarrow Q \leftrightarrow Y \leftrightarrow Z$ ($P \notin bd_G(Z)$). Therefore we must have that $bd_G(A_k) \subseteq Y \cup bd_G(Y)$ holds for all $A_k \in A$, i.e. that $bd_G(A) \subseteq Y \cup bd_G(Y)$ holds.

Secondly note that we can let B be a subset of A st B consists of the nodes in one connected subgraph in the subgraph of G induced by A (any connected subgraph will do). Let D be the set of nodes st $D = sp_G(Y) \cap sp_G(Z) \cap sp_G(B)$. With these sets we know that the spouses of Y can be either adjacent of Z or not, hence $sp_G(Y) = D \cup A$ must hold. This in turn gives that $sp_G(A) = D \cup Y$ and $bd_G(A) \subseteq D \cup Y \cup pa_G(Y)$ since $\forall A_k \in A$ $bd_G(A_k) \subseteq Y \cup bd_G(Y)$ holds. Moreover $sp_G(B) = D \cup Y$ and $bd_G(B) \subseteq D \cup Y \cup pa_G(Y)$ must also hold. Hence, if D is empty then $sp_G(B) = \{Y\}$ and $bd_G(B) \subseteq Y \cup pa_G(Y)$ must hold. This does however lead to a contradiction because a split then is possible st U consists of B and L consists of $C \setminus U$. Hence there has to exists at least one node in D.

Thirdly note that $D \cup Y$ must be complete or the induced subpath $B_k \leftrightarrow DY_i \leftrightarrow Z \leftrightarrow DY_j \leftrightarrow B_1 \leftrightarrow ... \leftrightarrow B_l \leftrightarrow B_k, l \geq 0$, exists in G for some nodes $B_k, B_1, ..., B_l \in B$ and $DY_i, DY_j \in D \cup Y$. This means that G contains an induced subgraph of the form shown in either Figure 1a ($l > 0$) or 1b ($l = 0$).

Fourth and finally note that there must exist a node P st $P \in bd_G(B) \cup B$ but $P \notin bd_G(D_j)$ for some $D_j \in D \cup Y$ or a split is feasible where U consists of B and L consists of $C \setminus U$. Note that $D_j \neq Y$ must hold since $bd_G(B) \cup B \subseteq bd_G(Y) \cup Y$. This means that there must exist 2 nodes B_i, D_j st $P \in bd_G(B_i), P \notin bd_G(D_j), B_i \in B$, $B_i \in sp(D_j)$ and $D_j \in D$ st the induced subgraph $P \circ\!\!\rightarrow B_i \leftrightarrow D_j \leftrightarrow Z \cdots P$ exist in G. This is a contradiction either because G contains an induced subgraph of the form shown in Figure 1b ($P \in bd_G(Z)$) or $P \circ\!\!\rightarrow B_i \leftrightarrow D_j \leftrightarrow Z$ ($P \notin bd_G(Z)$) where $bd_G(B_i) \subseteq bd_G(Y) \cup Y$ and $Y \in sp_G(B_i)$ holds.

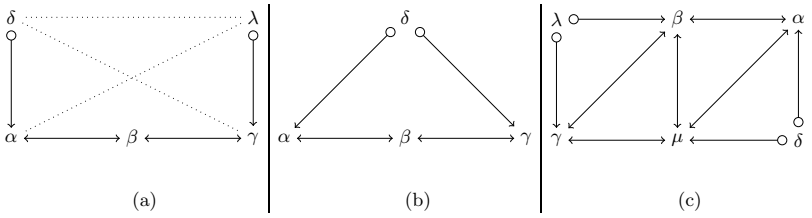


Fig. 1. MVR subgraph forms

Lemma 4. *If a maximally oriented CG G of the MVR interpretation contains a bidirected flag then G at least one of the induced subgraphs shown in Figure 1 exists in G.*

Proof. Assume the contrary, that no such induced subgraph exists in G even though G contains a bidirected flag and G is maximally orientated. Since G contains a bidirected flag we do with Lemma 3 get that G must contain an induced subgraph $X \leftrightarrow Y \leftrightarrow Z \leftarrow\!\circ W$ or a contradiction directly follows. If we now apply Lemma 3 to $X \leftrightarrow Y \leftrightarrow Z$ we get that, since for G to contain any induced subgraph of the form shown in Figure 1a or 1b is a contradiction, there exist a set of nodes (that can be renamed to) c_1, c_2, c_3 st the induced subgraph $c_1 \circ\!\rightarrow c_2 \leftrightarrow c_3 \leftrightarrow Z$ exists in G and $c_3 = Y$ holds or $bd_G(c_2) \subseteq bd_G(Y) \cup Y$ and $Y \in sp_G(c_2)$ hold. If $c_3 = Y$, G must contain the subgraph $c_1 \circ\!\rightarrow c_2 \leftrightarrow Y \leftrightarrow Z \leftarrow\!\circ W$ where $c_1 \notin adj_G(Y)$ and $W \notin adj_G(Y)$ must hold and $c_1 = W$ might hold. Clearly this subgraph takes the form of either Figure 1a ($c_1 \neq W$) or 1b ($c_1 = W$) which is a contradiction. Hence $c_3 \neq Y$, $bd_G(c_2) \subseteq bd_G(Y) \cup Y$ and $Y \in sp_G(c_2)$ must hold.

Since $W \notin adj_G(Y)$ holds and $bd_G(c_2) \subseteq bd_G(Y) \cup Y$ it is clear that $c_1, c_3 \in bd_G(Y)$ must hold. Hence $W \neq c_2$ holds since $W \notin adj_G(Y) \cup Y$. This in turn means that $W \notin bd_G(c_2)$ holds since $bd_G(c_2) \subseteq bd_G(Y) \cup Y$ and $W \notin bd_G(Y) \cup Y$. Finally we can see that $W \in bd_G(c_3)$ holds or the induced subgraph $c_1 \circ\!\rightarrow c_2 \leftrightarrow c_3 \leftrightarrow Z \leftarrow\!\circ W$ takes the form shown in Figure 1a ($c_1 \neq W$) or 1b ($c_1 = W$). However, if $W \in bd_G(c_3)$ holds G contains an induced subgraph of the form shown in Figure 1c (where $\delta = W$, $\lambda = c_1$, $\mu = c_3$, $\gamma = c_2$, $\beta = Y$ and $\alpha = Z$) and we have a a contradiction.

Lemma 5. *The independence model of a CG G of the MVR interpretation which contains an induced subgraph of one of the forms shown in Figure 1 cannot be perfectly represented as a CG H of the AMP interpretation.*

Proof. Assume the contrary, that there exists a CG H under the AMP interpretation that can represent these independence models.

First assume that the independence model of the graph shown in Figure 1a can be represented in a CG H of the AMP interpretation. It is clear that H must have the same skeleton, or clearly some separations or non-separations that hold in G would not hold in H. The following independence statements holds in G: $\delta \perp_G \beta | pa_G(\beta)$, $\alpha \perp_G \gamma | pa_G(\alpha)$ and $\beta \perp_G \lambda | pa_G(\beta)$. $\delta \perp_G \beta | pa_G(\beta)$ gives us that a triplex $(\{\delta, \beta\}, \alpha)$ must exist in H, since $\alpha \notin pa_G(\beta)$ i.e. that (1) $\delta \rightarrow \alpha - \beta$, (2) $\delta - \alpha \leftarrow \beta$ or (3) $\delta \rightarrow \alpha \leftarrow \beta$ exists in H. $\alpha \perp_G \gamma | pa_G(\alpha)$ does however also state that a triplex $(\{\alpha, \gamma\}, \beta)$ must exist in H, since $\beta \notin pa_G(\alpha)$. For this to happen the edge between α and β can not be orientated towards α hence the subgraph $\delta \rightarrow \alpha - \beta \leftarrow \gamma$ must exist in H. The orientation of the edge between β and γ does however contradict the third independence statement $\beta \perp_G \lambda | pa_G(\beta)$ which implies that the triplex $(\{\beta, \lambda\}, \gamma)$ must exist in H, since $\gamma \notin pa_G(\beta)$. Hence we have a contradiction if G contains the induced subgraph shown in Figure 1a.

Secondly assume that the independence model of the graph shown in Figure 1b can be represented in a CG H of the AMP interpretation. It is clear that H must

have the same skeleton, or clearly some separations or non-separations that hold in G would not hold in H. The following independence statements must then hold in G: $\delta \perp_G \beta | pa_G(\beta)$ and $\alpha \perp_G \gamma | pa_G(\alpha)$. $\delta \perp_G \beta | pa_G(\beta)$ gives us that two triplexes must exist in H, first $(\{\delta, \beta\}, \alpha)$ and secondly $(\{\delta, \beta\}, \gamma)$, since $\alpha, \gamma \notin pa_G(\beta)$. $(\{\delta, \beta\}, \alpha)$ gives that one of the following configurations must occure in H: (1) $\delta - \alpha \leftarrow \beta$, (2) $\delta \rightarrow \alpha - \beta$ or (3) $\delta \rightarrow \alpha \leftarrow \beta$. However, the independence statement $\alpha \perp_G \gamma | pa_G(\alpha)$ implies that the triplex $(\{\alpha, \gamma\}, \beta)$ must exist in H since $\beta \notin pa_G(\alpha)$. If the triplex $(\{\alpha, \gamma\}, \beta)$ should hold in H the edge between α and β can not be orientated towards α hence the subgraph $\delta \rightarrow \alpha - \beta \leftarrow \gamma$ must exist in H. The orientation of the edge between β and γ does however contradict the triplex $(\{\delta, \beta\}, \gamma)$ and hence we have a contradiction for the G shown in Figure 1b.

Third and last assume that the independence model of the graph shown in Figure 1c can be represented in a CG H of the AMP interpretation. From the Figure we can read the following independence statements: $\lambda \perp_G \mu | pa_G(\mu)$, $\alpha \perp_G \gamma | pa_G(\alpha), \beta \perp_G \delta | pa_G(\beta)$. It is clear that H must have the same skeleton, or clearly some separations or non-separations that hold in G would not hold in H. $\lambda \perp_G \mu | pa_G(\mu)$ and $\alpha \perp_G \gamma | pa_G(\alpha)$ gives that the triplexes $(\{\lambda, \mu\}, \beta)$ and $(\{\alpha, \gamma\}, \mu)$ must exists in H since $\beta \notin pa_G(\mu)$ and $\mu \notin pa_G(\alpha)$. As seen above this gives that $\lambda \rightarrow \gamma - \mu \leftarrow \alpha$ must exist in H. Similarly $\beta \perp_G \delta | pa_G(\beta)$ and $\lambda \perp_G \mu | pa_G(\mu)$ gives that $\lambda \rightarrow \beta - \mu \leftarrow \delta$ must exist in H. Finally $\alpha \perp_G \gamma | pa_G(\alpha)$ and $\beta \perp_G \delta | pa_G(\beta)$ gives that the triplexes $(\{\alpha, \gamma\}, \beta)$ and $(\{\beta, \delta\}, \alpha)$ must hold in H, since $\beta \notin pa_G(\alpha)$ and $\alpha \notin pa_G(\beta)$, which in turn gives that $\gamma \rightarrow \beta - \alpha \leftarrow \delta$ must exist in H. This does however contradict that H is a CG since the semi-directed cycle $\gamma \rightarrow \beta - \mu - \gamma$ exists in H. Hence we have a contradiction.

4.2 Translation of AMP CGs to MVR CGs

Theorem 4. *Given an AMP CG G, and a maximally oriented AMP CG G' in the Markov equivalence class of G, there exists a CG H st $I_{AMP}(G) = I_{MVR}(H)$ iff G' does not contain any induced subgraph of the form $X - Y - Z$.*

Proof. Sufficiency follows from Lemma 2 while necessity follows from 6.

Lemma 6. *If a maximally orientated CG G of the AMP interpretation contains an induced subgraph of the form $X - Y - Z$ then G there exists no CG H of the MVR interpretation st $I_{AMP}(G) = I_{MVR}(H)$.*

4.3 Translation of MVR CGs to LWF CGs

Theorem 5. *Given a MVR CG G, and a maximally oriented MVR CG G' that is in the same Markov equivalence class as G, there exist a LWF CG H st $I_{MVR}(G) = I_{LWF}(H)$ iff G' contains no bidirected edge, i.e. can be represented as a BN.*

Proof. From Lemma 7 it follows that a maximally oriented CG G' of the MVR interpretation with a bidirected edge must have a subgraph of the form shown

in Figure 2. If it does not contain any bidirected edge in the maximally oriented model it trivially follows that it is a BN (and hence it can be represented as a CG of the LWF interpretation). From Lemma 8 it then follows that no CG G of the MVR interpretation which contains a subgraph of the form shown in Figure 2 can be represented as a CG of the LWF interpretation.

Lemma 7. *If a bidirected edge exists in a maximally oriented CG G of the MVR interpretation then G must contain an induced subgraph of the form shown in Figure 2.*

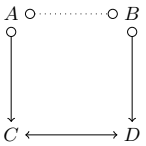


Fig. 2. Included subgraph in Lemma 7 and 8

Lemma 8. *If a CG G of the MVR interpretation contains an induced subgraph of the form shown in Figure 2 then G can not be translated into a CG H of the LWF interpretation.*

Proof. Assume to the contrary that there exists a CG H, of the LWF interpretation, with the same independence model as G while G contains an induced subgraph of the form shown in Figure 2. Clearly H and G must contain the same nodes and adjacencies or some separations or non-separations must exist in G but not in H.

From Figure 2 we can read that $A \perp_G D | pa_G(D)$ and $C \perp_G B | pa_G(C)$ hold. For $A \perp_G D | pa_G(D)$ to hold in H C must be a collider between A and D and hence H must contain the induced subgraph $A \rightarrow C \leftarrow D$. Similarly $C \perp_G B | pa_G(C)$ gives that H must contain the induced subgraph $C \rightarrow D \leftarrow B$ and hence we have a contradiction.

4.4 Translation of LWF CGs to MVR CGs

Theorem 6. *Given a LWF CG G there exists a CG H st $I_{LWF}(G) = I_{MVR}(H)$ iff $(G_{cl(K)})^m$ is chordal for all $K \in cc(G)$.*

Proof. To prove the "if" part, note that if $(G_{cl(K)})^m$ is chordal for all $K \in cc(G)$, then there is a DAG D st $I_{LWF}(G) = I_{BN}(D)$ [1, Proposition 4.2] and, thus, it suffices to take $H = D$.

To prove the "only if" part, assume to the contrary that $V_1 - \ldots - V_n$ is a chordless undirected cycle in $(G_{cl(K)})^m$ for some $K \in cc(G)$. Note that H has the same adjacencies as G. Therefore, $V_{i-1} \leftarrow V_i$ and/or $V_i \rightarrow V_{i+1}$ must be in H because, otherwise, $V_{i-1} \perp_G V_{i+1} | Z \in I_{LWF}(G)$ for some Z st $V_i \in Z$ whereas

$V_{i-1} \perp_H V_{i+1} | Z \notin I_{MVR}(H)$, which contradicts that $I_{LWF}(G) = I_{MVR}(H)$. Assume without loss of generality that $V_i \rightarrow V_{i+1}$ is in H. Then, $V_{i+1} \rightarrow V_{i+2}$ must be in H too, by an argument similar to the previous one. Repeated application of this reasoning implies that H has a semi-directed cycle, which contradicts the definition of CG.

Acknowledgments. This work is funded by the Center for Industrial Information Technology (CENIIT) and a so-called career contract at Linköping University, by the Swedish Research Council (ref. 2010-4808), and by FEDER funds and the Spanish Government (MICINN) through the project TIN2010-20900-C04-03.

References

1. Andersson, S.A., Madigan, D., Pearlman, M.D.: A Characterization of Markov Equivalence Classes for Acyclic Digraphs. Annals of Statistics 25(2), 505–541 (1997)
2. Andersson, S.A., Madigan, D., Pearlman, M.D.: An Alternative Markov Property for Chain Graphs. Scandianavian Journal of Statistics 28, 33–85 (2001)
3. Cox, D.R., Wermuth, N.: Linear Dependencies Represented by Chain Graphs. Statistical Science 8, 204–283 (1993)
4. Cox, D.R., Wermuth, N.: Multivariate Dependencies: Models, Analysis and Interpretation. Chapman and Hall (1996)
5. Drton, M.: Discrete Chain Graph Models. Bernoulli 15(3), 736–753 (2009)
6. Frydenberg, M.: The Chain Graph Markov Property. Scandinavian Journal of Statistics 17, 333–353 (1990)
7. Lauritzen, S.L., Wermuth, N.: Graphical Models for Association Between Variables, Some of Which are Qualitative and Some Quantitative. Annals of Statistics 17, 31–57 (1989)
8. Roverto, A., Studený, M.: A Graphical Representation of Equivalence Classes of AMP Chain Graphs. Journal of Machine Learning Research 7(6), 1045–1078 (2006)
9. Sonntag, D., Peña, J.M.: Learning Multivariate Regression Chain Graphs under Faithfulness. In: Proceedings of the 6th European Workshop on Probabilistic Graphical Models, pp. 299–306 (2012)
10. Studený, M., Roverto, A., Štěpánová, Š.: Two Operations of Merging and Splitting Components in a Chain Graph. Kybernetika 45, 208–248 (1997)
11. Wermuth, N., Sadeghi, K.: Sequences of Regression and their Indepdendences. TEST 21(2), 215–252 (2012)
12. Wermuth, N., Wiedenbeck, M., Cox, D.: Partial Inversion for Linear Systems and Partial Closure of Independence Graphs. BIT Numerical Mathematics 46, 883–901 (2006)

On the Plausibility of Abstract Arguments

Emil Weydert

ILIAS-CSC, University of Luxembourg

Abstract. We propose and investigate a plausibility-based extension semantics for abstract argumentation frameworks based on their generic instantiation by default knowledge bases and the ranking construction paradigm for default reasoning.

1 Prologue

The past decade has seen a flourishing of abstract argumentation theory, a coarse-grained high-level form of defeasible reasoning introduced by Dung [4]. It is characterized by a top-down perspective which ignores the logical fine structure of arguments and focuses instead on logical or extra-logical relations - like conflicts or preferences - between given arguments to identify reasonable argumentative positions. The complexity of enriched argument structures with interacting relations, the competing proposals for evaluating even Dung's simple attack frameworks, all this calls for unifying semantic foundations to compare, judge, and improve existing approaches.

A major issue is whether an abstract account adequately models concrete argumentative reasoning in the context of a sufficiently expressive, often defeasible logic. The instantiation of abstract frameworks by more fine-grained logic-based argument configurations is therefore an important tool for justifying or criticising abstract argumentation theories. Most of this work is however based on the first generation of default formalisms, like Reiter's default logic or logic programming. While these are closer to classical logic and the original spirit of Dung's approach, it is well known that they are haunted by counterintuitive behaviour and fail to satisfy major desiderata for default reasoning encoded in benchmark examples and rationality postulates [9].

The goal of the present work is therefore to supplement existing interpretation efforts with a simple ranking-based semantic instantiation model which interprets arguments and attacks with conditional knowledge bases. The well-behaved ranking construction semantics for default reasoning [15,16] is exploited to specify a new extension semantics for Dung frameworks which allows to directly evaluate the plausibility of argument collections. Its partly unorthodox behaviour sheds a new light on basic argumentation-theoretic concepts.

We start with an introduction to default reasoning based on the ranking construction paradigm. After a short look at abstract argumentation theory, we indicate how to interpret argumentation frameworks semantically and instantiate abstract arguments and the attacks between them with sets of conditionals.

L.C. van der Gaag (Ed.): ECSQARU 2013, LNAI 7958, pp. 522–533, 2013.

Based on the concept of generic instantiations, we then specify a ranking-based extension semantics. We conclude with a simple algorithm, some instructive examples, and an analysis of important properties.

2 Ranking-Based Default Reasoning

First, we assume a basic language L closed under the usual propositional connectives, with a classical satisfaction relation $\models$ inducing a monotonic entailment relation $\vdash$. Its model sets are denoted by $[\![\varphi]\!] = \{m \mid m \models \varphi\}$, resp. $[\![\Sigma]\!] = \cap_{\varphi \in \Sigma}[\![\varphi]\!]$ for $\Sigma \subseteq L$. $\mathcal{B}_L$ is the boolean propositional algebra over $\mathbb{B}_L = \{[\![\varphi]\!] \mid \varphi \in L\}$.

Default inference is an important instance of nonmonotonic reasoning concerned with drawing reasonable but potentially defeasible conclusions from knowledge bases of the form $\Sigma \cup \Delta$, where Σ is a set of assumptions or facts, e.g. describing a specific state of affairs in some domain language, and Δ is a collection of conditionals expressing strict or exception-tolerant implicational information and guiding the defeasible inference process. In the following we will focus on finite $\Sigma \subseteq L$ and finite $\Delta \subseteq L(\twoheadrightarrow, \rightsquigarrow)$. $L(\twoheadrightarrow, \rightsquigarrow)$ is a flat conditional language on top of L with $L(\twoheadrightarrow, \rightsquigarrow) = \{\varphi \twoheadrightarrow \psi \mid \varphi, \psi \in L\} \cup \{\varphi \rightsquigarrow \psi \mid \varphi, \psi \in L\}$.

The strict implication $\varphi \twoheadrightarrow \psi$ states that φ necessarily implies ψ, forcing us to accept ψ given φ. The default implication $\varphi \rightsquigarrow \psi$ tells us that φ plausibly/by default implies ψ, only recommending the acceptance of ψ. The actual impact of a default depends of course on the whole context $\Sigma \cup \Delta$ and the chosen nonmonotonic inference concept $\mid\!\sim$, which we will discuss later on.

We can distinguish two perspectives in default reasoning: the autoepistemic, context-based, and the plausibilistic, quasi-probabilistic one. The former is exemplified by Reiter's default logic, where defaults can be modeled by normal default rules $\varphi : \psi/\psi$. The alternative is to use default conditionals interpreted by some preferential or valuational semantics, like System Z [10], or ME-accounts [5] (ME = maximum-entropy). For historical reasons and technical convenience, the first approach has received most attention, in particular in the context of argumentation. However, this ignores the fact that the conditional semantic paradigm has a much better record when it comes to the natural handling of benchmark examples and the satisfaction of rationality postulates. It therefore seems promising to investigate whether semantic-based accounts can help to instantiate and evaluate abstract argumentation frameworks.

Our default conditional semantics for interpreting argumentation frameworks is based on the simplest plausibility measure concept able to reasonably handle independence and conditionalization, namely ranking measures [11,12]. These are quasi-probabilistic belief valuations expressing the order of magnitude (intuitively: $R(A) = r \sim P(A) = \varepsilon^r$) or degree of surprise of propositions. They generalize Spohn's original integer-valued κ-ranking functions introduced to model the iterated revision of graded plain belief, and also standard possibility measures (log-link) [3].

Definition 2.1 (Ranking measures). *A map $R : \mathcal{B}_L \to ([0, \infty], 0, \infty, +, \geq)$ is called a rational/real-valued ranking measure iff $R(\top) = 0$, $R(\bot) = R(\emptyset) = \infty$,*

and for all $A, B \in \mathbb{B}_L$, $R(A \cup B) = \min_{\leq}\{R(A), R(B)\}$. $R(.|.)$ *is the associated conditional ranking measure defined by* $R(B|A) = R(A \cap B) - R(A)$ *if* $R(A) \neq \infty$, *else* $R(B|A) = \infty$. *We use the abbreviation* $R(\varphi) := R([\![\varphi]\!])$.

For several major default formalisms, e.g. rk-ME, JZ (see below), rational ranking values are actually necessary (and sufficient in finite contexts). Note that lower values indicate less surprise/more plausibility. R_0 is the uniform ranking measure, i.e. $R_0(A) = 0$ for $A \neq \emptyset$. A suitable truth condition for defaults is

$$R \models_{rk} \varphi \rightsquigarrow \psi \;\text{ iff }\; R(\varphi \wedge \psi) + 1 \leq R(\varphi \wedge \neg\psi).$$

Because all the $r \in]0, \infty[$ can be exchanged by automorphisms, w.l.o.g., we may focus on the threshold 1. We use $\leq$ because this guarantees the existence of minima for relevant ranking construction procedures, and there is also a priori no reason to privilege the weakest truth condition (... < ...) for interpreting defaults. $\varphi \twoheadrightarrow \psi$ can be expressed by $\varphi \wedge \neg\psi \rightsquigarrow F$ ($\varphi \wedge \neg\psi$ is doxastically impossible). For $\Delta \cup \{\delta\} \subseteq L(\twoheadrightarrow, \rightsquigarrow)$, we set $[\![\Delta]\!]_{rk} = \{R \mid R \models_{rk} \Delta\}$, and $\Delta \vdash_{rk} \delta$ iff $[\![\Delta]\!]_{rk} \subseteq [\![\delta]\!]_{rk}$. $\vdash_{rk}$ is monotonic and verifies the axioms and rules of preferential conditional logic and disjunctive rationality (the threshold reading blocks rational monotony) for $\rightsquigarrow$ [8].

It is important to understand that the central concept in default reasoning is not some monotonic conditional logic for $L(\twoheadrightarrow, \rightsquigarrow)$, but a nonmonotonic meta-level inference relation $\mid\!\sim$ over $L \cup L(\twoheadrightarrow, \rightsquigarrow)$ specifying which conclusions $\psi \in L$ can be plausibly inferred from usually finite $\Sigma \cup \Delta \subseteq L \cup L(\twoheadrightarrow, \rightsquigarrow)$. We write $\Sigma \cup \Delta \mid\!\sim \psi$, or alternatively $\Sigma \mid\!\sim_\Delta \psi$, and set $C^{\mid\sim}_\Delta(\Sigma) = \{\psi \mid \Sigma \mid\!\sim_\Delta \psi\}$.

The ranking semantics for plausibilistic default reasoning is based on non-monotonic ranking choice operators $\mathcal{I}$ which map each finite $\Delta \subseteq L(\twoheadrightarrow, \rightsquigarrow)$ to a collection $\mathcal{I}(\Delta) \subseteq [\![\Delta]\!]_{rk}$ of preferred ranking models of Δ over $\mathcal{B}_L$. The corresponding rational default inference notion $\mid\!\sim^{\mathcal{I}}$ is then specified by

$$\Sigma \mid\!\sim^{\mathcal{I}}_\Delta \psi \;\text{ iff }\; \text{for all } R \in \mathcal{I}(\Delta),\;\; R(\neg\psi \mid \wedge\Sigma) > 0.$$

For instance, if we compare the rankings pointwisely, $\mathcal{I}(\Delta) = \{Min_{\leq_{pt}}[\![\Delta]\!]_{rk}\}$ essentially characterizes System Z [10]. The construction paradigm for default reasoning developed in [14,15,16] is another well-motivated strategy for getting reasonable $\mathcal{I}$ based on Spohn's Jeffrey-conditionalization for ranking measures. Here defaults do not only specify ranking constraints, but also admissible ranking construction steps. It offers powerful default inference notions with nice inheritance features.

Definition 2.2 (Constructibility). *Let* $\Delta = \{\varphi_i \rightsquigarrow \psi_i \mid i \leq n\} \subseteq L(\rightsquigarrow)$. *A ranking measure* R' *is constructible from* R *over* Δ, *written* $R' \in Constr(\Delta, R)$, *iff there are ranking values* $r_0, \ldots, r_n \in [0, \infty]$ *s.t.* $R' = R + \Sigma_{i \leq n} r_i[\varphi_i \wedge \neg\psi_i]$, *with* $(R + r[\varphi])(\psi) = min\{R(\psi \wedge \varphi) + r, R(\psi \wedge \neg\varphi)\}$ *(uniformly shifting* φ *by* r*).*

For instance, we can obtain a well-behaved robust default inference relation, System J [14], just by setting $\mathcal{I}_J(\Delta) = Constr(\Delta, R_0) \cap [\![\Delta]\!]_{rk}$. It can be strengthened to System JJ [14] by focusing on what we call justifiably constructible ranking models. Here proper shifting is only allowed to realize ranking constraints

interpreting defaults as equalities, so as to prevent oversatisfaction. $R \models_{rk} \Delta$ is called a justifiably constructible model of Δ iff $R = R_0 + \Sigma_{i \leq n} a_i[\varphi_i \wedge \neg\psi_i]$ and for each $a_j > 0$, $R(\varphi_i \wedge \psi_i) + 1 = R(\varphi_i \wedge \neg\psi_i)$. If $\Delta \not\vdash_{rk} \mathbf{F}$, $\mathcal{I}_{jj}(\Delta) \neq \emptyset$.[1]

For minimal core default sets [5], i.e. where no $[\![\varphi_i \wedge \neg\psi_i]\!] \neq \emptyset$ is covered by $\cup_{j \neq i}[\![\varphi_j \wedge \neg\psi_j]\!]$, $\mid\!\sim^{jj}$ offers the same results as maximum-entropy-based approaches. The asymptotic order-of-magnitude translation of entropy maximization to the ranking level (rk-ME) [13,16] always produces a unique justifiably constructible ranking model, i.e. $\mid\!\sim_{jj} \subset \mid\!\sim_{me}$. Similarly for the well-behaved System JZ, which is based on a natural canonical hierarchical ranking construction in the tradition of System Z [15,16] and also implements the minimal information philosophy. But $\mid\!\sim_{jj}, \mid\!\sim_{jz}, \mid\!\sim_{me}$ are equivalent for the generic default sets we will use to interpret abstract argumentation frameworks.

3 Abstract Argumentation

The idea of abstract argumentation theory has been to replace the traditional bottom-up strategy, which models and exploits the logical fine structure of arguments, by a top-down perspective, where arguments become black boxes evaluated only according to specific logical or extra-logical relationships linking them. Such a coarse-grained relational analysis may often be enough to determine which collections of arguments are reasonable. In addition to possible conceptual and computational gains, the abstract approach also provides a powerful methodological tool for general argumentation-theoretic investigations. Launched by Dung [4], abstract argumentation theory has evolved in the last ten years into a powerhouse of nonmonotonic reasoning light.

An abstract argumentation framework in the original sense of Dung is a structure of the form $\mathcal{A} = (\mathbb{A}, \rhd)$, where $\mathbb{A}$ is a collection of abstract entities representing arguments, and $\rhd$ is a binary, possibly asymmetric attack relation modeling conflicts between arguments. To grasp the complexity of real-world argumentation, many authors have extended this basic account to include further inferential/epistemic relations, like support, preferences, valuations, or collective attacks. On the most general level, an abstract argumentation framework is just a structure $\mathcal{A} = (\mathbb{A}, \mathcal{C}, (\mathcal{P}_i)_{i \in I})$ where $\mathbb{A}$ is the domain of possible arguments, $\mathcal{C}$ is the collection of conflict sets (closed under supersets), and the $\mathcal{P}_i$ are relevant relations over $\mathbb{A}$ (or even $2^{\mathbb{A}}$). A Dung framework $(\mathbb{A}, \rhd)$ can be rewritten as $(\mathbb{A}, \mathcal{C}, \rhd)$ where $\mathcal{C} = \{B \subseteq \mathbb{A} \mid \rhd \cap B \times B \neq \emptyset\}$.

A general inferential task in abstract argumentation is to find reasonable evaluations of the arguments described by $\mathcal{A}$, e.g. which sets of arguments to adopt (extensions). In Dung's scenario, the extensions are just conflict-free $E \subseteq \mathbb{A}$ satisfying suitable acceptability conditions in the context of $\mathcal{A}$. For instance, assuming conflict-freeness, E is admissible iff each attacker of an $a \in E$ is attacked by some $b \in E$. E is grounded iff it is minimally admissible, it is stage iff $E \cup \rhd''E$ is maximal, and stable iff $\mathbb{A} - E \subseteq \rhd''E$.

[1] For other constructibility-flavoured accounts: [3,6].

In concrete decision contexts, we may however also seek finer-grained assessments of arguments, like labelings or prioritizations. This suggests more general semantics which associate with each $\mathcal{A}$ a set $\mathcal{E}(\mathcal{A})$ of so-called hyperextensions [17], which are distinguished evaluation structures $(\mathbb{A}, In^{\mathcal{A}}, (\mathcal{Q}_j)_{j\in J})$ over $\mathbb{A}$. $In^{\mathcal{A}}$ $(\notin \mathcal{C})$ is here just a classical extension, whereas the $\mathcal{Q}_j$ $(j \in J)$ are relations expressing more sophisticated evaluations of arguments (e.g. a posteriori plausibility). If $J = \emptyset$, we are back to Dung. So far for the general landscape.

4 Concrete Instantiations

To evaluate and apply abstract argumentation techniques, the argumentation frameworks have to be instantiated. That is, we have to interpret their abstract elements by concrete logical entities representing actual arguments, and their abstract structure by specific inferential or epistemic relationships fitting the conceptual intentions at the abstract level. The minimal requirement for an instantiation I is to identify from each instantiated argument $I(a)$ the corresponding explicit claim or conclusion, expressed by a sentence ψ_a from the chosen base logic $(L, \vdash)$. If two arguments a and b generate inconsistent claims, i.e. if $\psi_a, \psi_b \vdash \mathbf{F}$, then they are clearly not jointly acceptable. On the other hand, in argumentation the incompatibility of claims is not a necessary prerequisite for the existence of an attack. The claims and their logical connections are just the tip of the iceberg and far from characterizing all the relevant relationships between arguments. We will now investigate how to instantiate a pure Dung framework $\mathcal{A} = (\mathbb{A}, \rhd)$ over a nonmonotonic logic $\mathcal{L} = (L \cup L(\twoheadrightarrow, \rightsquigarrow), \mathrel{|\!\sim})$ with base logic $(L, \vdash)$. Let us first interpret the abstract arguments.

On the syntactic level, because we are primarily interested in the logical content, it may be enough to consider flat – not tree – instantiations I_{syn}, which associate with each $a \in \mathbb{A}$ a correct finite inference pair $I_{syn}(a) = (\Sigma_a \cup \Delta_a, \psi_a)$, with $\Sigma_a \subseteq L$, $\Delta_a \subseteq L(\twoheadrightarrow, \rightsquigarrow)$, $\psi_a \in L$, and $\Sigma_a \cup \Delta_a \mathrel{|\!\sim} \psi_a$. We emphasize that here we do neither impose premise consistency, nor premise minimality. While minimality is a standard assumption in monotonic argumentation, it is questionable in the context of nonmonotonic reasoning. In fact, by adding premises, a conclusion may successively get accepted, rejected, and accepted again, so that the character of the inferential support may change between different levels of specifity, calling for a discrimination between the corresponding inference pairs.

On the semantic level, lack of space prevents us to provide a semantic account of the full inferential relationship expressed by $I_{syn}(a)$, so we will focus on the inferential link between L-formulas. First, we may observe that the L-content of $\Sigma_a \cup \Delta_a$ is determined not only by Σ_a but also by the necessities Δ_a supports:

$$\Delta_a^{\Box} = \{\varphi \in L \mid \{\neg\varphi\} \cup \Delta_a \mathrel{|\!\sim} F\}.$$

The strict propositional content of a, determined by the premises, is therefore $[\![\Sigma_a]\!] \cap [\![\Delta_a^{\Box}]\!]$. Its defeasible propositional content, fixed by $\mathrel{|\!\sim}$, is $[\![C^{\mathrel{|\!\sim}}_{\Delta_a}(\Sigma_a)]\!] \subseteq [\![\Sigma_a]\!] \cap [\![\Delta_a^{\Box}]\!] \cap [\![\psi_a]\!]$ (by inferential correctness). The shallow semantic instantiation of a w.r.t. I_{syn} is then given by

$$I_{sem}(a) = ([\![\Sigma_a]\!] \cap [\![\Delta_a^{\Box}]\!], [\![C_{\Delta_a}^{\sim}(\Sigma_a)]\!]).$$

In a finitary context, the strict resp. defeasible content of a can be represented by the L-propositions $[\![\varphi_a]\!] = [\![\Sigma_a]\!] \cap [\![\Delta_a^{\Box}]\!]$ resp. $[\![\psi_a]\!] = [\![C_{\Delta_a}^{\sim}(\Sigma_a)]\!]$. We sloppily use (φ_a, ψ_a) to denote $I_{sem}(a) = ([\![\varphi_a]\!], [\![\psi_a]\!])$. Note that $[\![\psi_a]\!] \subseteq [\![\varphi_a]\!]$. We emphasize that these semantic profiles of arguments are not intended to grasp their full nature, but only to reflect certain characteristics exploitable by suitable argumentation semantics. We observe that each proposition pair (φ, ψ) with $\psi \vdash \varphi$ and $\psi \not\vdash \mathbf{F}$ can become a shallow instantiation. In fact, because the defeasible modus ponens is valid, we may set $I_{syn}(a) = (\{\varphi, \varphi \rightsquigarrow \psi\}, \psi)$ and obtain $I_{sem}(a) = (\varphi, \psi)$. The above handling of necessities implies that the only other possibility for $I_{sem}(a)$ is here $(\mathbf{F}, \mathbf{F})$.

The next step is to interpret the abstract framework structure $\mathcal{A}$ by suitable relations over the instantiated arguments. While our semantic tools have a much broader scope, here we will consider only attack links $a \rhd b$. To simplify the discussion, we focus on minimal syntactic instantiations of the form $I(a) = I_{syn}(a) = (\{\varphi_a, \varphi_a \rightsquigarrow \psi_a\}, \psi_a)$ with $\psi_a \vdash \varphi_a$, hence $I_{sem}(a) = (\varphi_a, \psi_a)$. Let $\Delta^I = \{\varphi_a \rightsquigarrow \psi_a \mid a \in \mathbb{A}\}$. Each $a \in \mathbb{A}$ therefore induces a ranking constraint $R(\varphi_a \wedge \psi_a) + 1 \leq R(\varphi_a \wedge \neg\psi_a)$. To create a uniform semantic perspective we may therefore try to exploit the ranking semantics also to interpret the attack graph. That is, given I, the idea is to map the full attack structure $\mathcal{A}$ to a suitable collection of ranking measures verifying Δ^I.

So, let a, b be two arguments with $a \rhd b$ and $I_{sem}(a) = (\varphi_a, \psi_a)$, $I_{sem} = (\varphi_b, \psi_b)$. On the semantic level, if there are no explicit priorities, the attack should indicate an actual conflict between the defeasible contents of a and b. Hence we have to impose at least $R(\psi_a \wedge \psi_b) = \infty$. Now there are two possibilities. If the strict contents are incompatible as well, i.e. if $R(\varphi_a \wedge \varphi_b) = \infty$, the conflict becomes symmetric. If $R(\varphi_a \wedge \varphi_b) \neq \infty$, it follows from $R(\psi_a \wedge \psi_b) = \infty$ that $R(\psi_a \wedge \neg\psi_b | \varphi_a \wedge \varphi_b) = R(\psi_a | \varphi_a \wedge \varphi_b)$ and $R(\neg\psi_a \wedge \psi_b | \varphi_a \wedge \varphi_b) = R(\psi_b | \varphi_a \wedge \varphi_b)$. These two conditional ranking values state the degree of surprise, relative to the common context $\varphi_a \wedge \varphi_b$, of exclusively concluding ψ_a, resp. ψ_b. If ψ_a is here less surprising than ψ_b, we may interpret this as a one-sided attack from a on b, similarly for the converse. If on the other hand their ranks turn out to be equal, we get a balanced mutual attack.

Definition 4.1 (Ranking instantiation models). *Let $\mathcal{A} = (\mathbb{A}, \rhd)$ be a Dung framework, I a shallow semantic instantiation over $\mathbb{A}$ with $I(a) = (\varphi_a, \psi_a)$, and R a ranking measure over $\mathcal{B}_L$. For $a, b \in \mathbb{A}$, we set $a \rhd_I^R b$ iff $R(\psi_a \wedge \psi_b) = \infty$ and $R(\psi_a | \varphi_a \wedge \varphi_b) \leq R(\psi_b | \varphi_a \wedge \varphi_b)$. Then we call (R, I) a ranking (instantiation) model for $\mathcal{A}$ iff $R \models_{rk} \Delta^I = \{\varphi_a \rightsquigarrow \psi_a \mid a \in \mathbb{A}\}$ and for all $a, b \in \mathbb{A}$, $a \rhd b$ iff $a \rhd_I^R b$. Let $\mathcal{R}^{\mathcal{A}}$ be the collection of ranking models for $\mathcal{A}$.*

That is, the semantic-based attack relation $\rhd_I^R$ specified by (R, I) has to correspond exactly to the abstract attack relation $\rhd$. Each $\mathcal{A} = (\mathbb{A}, \rhd)$ obviously admits many ranking models (R, I), obtained by varying the ranking values or the proposition pairs associated with the abstract arguments. Note that for each

1-loop $a \rhd a$, $R(\psi_a \wedge \psi_a) = \infty = R(\varphi_a \wedge \psi_a) + 1 \leq R(\varphi_a \wedge \neg\psi_a) = \infty$, hence $R(\varphi_a) = \infty$. That is, the collection of ranking models doesn't change if we add or drop attack links between a self-reflective and another argument because the details are absorbed by the impossible joint context. If $\mathcal{A}$ and $\mathcal{A}'$ share the same 1-loops and the same attack structure over the other arguments, $\mathcal{R}^{\mathcal{A}} = \mathcal{R}^{\mathcal{A}'}$.

What about classical types of attack? If we focus on the actual semantic content, rebuttal is characterized by incompatible defeasible consequents, and undermining by a defeasible consequent conflicting with a strict antecedent. In the ranking context, these two types of attacks may be modeled by constraints expressing necessities. The straightforward definitions are as follows. Recall that $\psi_a \vdash \varphi_a$, $\psi_b \vdash \varphi_b$.

a **rebuts** b: $R(\psi_a \wedge \psi_b) = \infty$, e.g. if $\psi_a \vdash \neg\psi_b$.
a **undermines** b: $R(\psi_a \wedge \varphi_b) = \infty$, e.g. if $\psi_a \vdash \neg\varphi_b$.

In our simple semantic reading, undermining entails rebuttal because $\psi_b \vdash \varphi_b$. There are four qualitative attack configurations involving two arguments: $\varphi_a \wedge \varphi_b$ being compatible with neither, one, or both of ψ_a, ψ_b. If a asymmetrically undermines b, we have $R(\psi_a \wedge \varphi_b) = \infty$ and $R(\psi_b \wedge \varphi_a), R(\varphi_a \wedge \varphi_b) \neq \infty$. This implies $R(\psi_b|\varphi_a \wedge \varphi_b) < R(\psi_a|\varphi_a \wedge \varphi_b) = \infty$, i.e. $b \rhd_I^R a$ and $a \not\rhd_I^R b$ according to our attack semantics. It follows that undermining has no obvious ranking semantic justification if the defeasible claim entails the antecedent. Also note that rebuttal is compatible with, and entailed by, symmetric and asymmetric attacks.

5 Ranking Extensions

Ranking (instantiation) models offer new possibilities to identify reasonable argumentative positions. Let (R, I) be a ranking model for the framework $\mathcal{A} = (\mathbb{A}, \rhd)$. In the context of (R, I), a minimal requirement for aceptable argument sets $S \subseteq \mathbb{A}$ are coherent premises, i.e. the doxastic possibility of the joint antecedents $\varphi_S = \wedge_{a \in S}\varphi_a$ w.r.t. R, or $R(\varphi_S) \neq \infty$. This excludes self-attacks, but not conflicts within S. Because evidence should not be rejected without good reasons, the maximal coherent $S \subseteq \mathbb{A}$ are of particular interest and constitute suitable background contexts when looking for extensions. Each $E \subseteq S$ then specifies a proposition given by

$$\psi_{S,E} := \varphi_S \ \wedge \ \wedge_{a \in E}\psi_a \ \wedge \ \wedge_{a \in \mathbb{A}-E}\neg\psi_a.$$

$\psi_{S,E}$ characterizes those worlds verifying the strict content of the $a \in S$ and exactly the defeasible content of the $a \in E$. Because $a \rhd_I^R b$ implies $R(\psi_a \wedge \psi_b) = \infty$, any conflict $a \rhd b$ in E makes $\psi_{S,E}$ impossible. Note however that $R(\psi_{S,E}) = \infty$ may also result from non-binary conflicts, or a specific choice of logically dependent φ_a, ψ_a.

What are the most reasonable extension candidates $E \subseteq S \subseteq \mathbb{A}$ according to (R, I)? One idea is to focus on those E which induce the most plausible $\psi_{S,E}$ for all their maximal coherent supersets S.

Definition 5.1 (Ranking extensions). *Let (R, I) be a ranking model for $\mathcal{A} = (\mathbb{A}, \rhd)$. $E \subseteq \mathbb{A}$ is called a ranking-extension of $\mathcal{A}$ w.r.t. (R, I) iff there are maximal coherent $S \subseteq \mathbb{A}$ with $E \subseteq S$, and for all such S, and for all $E' \subseteq S$, $R(\psi_{S,E}) \leq R(\psi_{S,E'})$.*

While this specification looks rather decent, a cause of concern may be the great diversity of ranking models (R, I) available for any given $\mathcal{A}$. Consider for instance $\mathcal{A} = (\{p, q, r\}, \{(p, q), (q, r)\})$, i.e. $p \rhd q \rhd r$. $\mathcal{A}$ together with a shallow instantiation I then induces ranking constraints described by the conditionals in

$$\Delta^{\mathcal{A},I} = \{\psi_p \wedge \psi_q \rightsquigarrow \mathbf{F},\ \psi_q \wedge \psi_r \rightsquigarrow \mathbf{F},\ \varphi_p \wedge \varphi_q \rightsquigarrow \psi_p,\\ \varphi_q \wedge \varphi_r \rightsquigarrow \psi_q, \varphi_p \rightsquigarrow \psi_p, \varphi_q \rightsquigarrow \psi_q, \varphi_r \rightsquigarrow \psi_r\}.$$

If we assume that the φ_x, ψ_x are logically independent, $\Delta^{\mathcal{A},I}$ admits a unique justifiably constructible model, which is also the JZ and ME-model: $R^{\mathcal{A},I}_{jz}$

$$R^{\mathcal{A},I}_{jz} = R_0 + \infty[\psi_p \wedge \psi_q] + \infty[\psi_q \wedge \psi_r] + 1[\varphi_p \wedge \varphi_q \wedge \neg\psi_p] + 1[\varphi_q \wedge \varphi_r \wedge \neg\psi_q] +\\ 1[\varphi_p \wedge \neg\psi_p] + 1[\varphi_q \wedge \neg\psi_q] + 1[\varphi_r \wedge \neg\psi_r].$$

Because $S = \mathbb{A}$ is coherent, there are eight extension candidates and we have $R^{\mathcal{A},I}_{jz}(\psi_{\mathbb{A},\{p,r\}}) = 2 < 3 = R^{\mathcal{A},I}_{jz}(\psi_{\mathbb{A},\{p\}}) = R^{\mathcal{A},I}_{jz}(\psi_{\mathbb{A},\{q\}}) < 4 = R^{\mathcal{A},I}_{jz}(\psi_{\mathbb{A},\{r\}}) < 5 = R^{\mathcal{A},I}_{jz}(\psi_{\mathbb{A},\emptyset}) < \infty$ for the doxastically possible alternatives. Hence, the resulting ranking extension is $\{p, r\}$, which is also the standard Dung solution.

However, if the choice of the extension generating ranking model (R, I) only presupposed the validation of $\Delta^{\mathcal{A},I}$, we could pick up $R = R^{\mathcal{A},I}_{jz} + \infty[\psi_p \wedge \psi_r \wedge \varphi_q]$ such that $R(\psi_p \wedge \psi_r \wedge \varphi_q) = \infty$, resp. I so that $\psi_p \wedge \psi_r \wedge \varphi_q \vdash F$. But under both conditions, the minima would then become $R(\psi_{\mathbb{A},\{p\}}) = R(\psi_{\mathbb{A},\{q\}}) = 3$, imposing the ranking extensions $\{p\}, \{q\}$. Because of $R(\psi_{\mathbb{A},\{p,r\}}) = \infty$, the standard extension $\{p, r\}$ would necessarily be rejected. But this violates a hallmark of abstract argumentation, namely the support of unattacked arguments like p. From this it follows that we have to prioritize the choice of ranking models to implement a reasonable ranking extension semantics.

The idea is now to choose on one hand a well-justified canonical ranking measure model of the default base $\Delta^{\mathcal{A},I}$ as our doxastic background, e.g. the JZ-model $R^{\mathcal{A},I}_{jz}$, and to focus on the other hand on the most generic instantiations of a given framework $\mathcal{A}$. In particular, we stipulate by default that the syntactic instantiations of individual arguments are logically independent, modulo possible constraints imposed by the framework structure (e.g. loops). Furthermore, Ockham's razor suggests to choose the simplest possible instantiations. We can implement this by using disjoint vocabularies for instantiating different abstract arguments, and by relying on elementary instances of the defeasible modus ponens for the corresponding inference pairs. That is, we introduce for each $a \in \mathbb{A}$ independent propositional atoms X_a, Y_a and set $I_{syn}(a) = (\{X_a\} \cup \{X_a \rightsquigarrow Y_a\}, Y_a)$. The corresponding generic semantic instantiation is then $I(a) = (\varphi_a, \psi_a) = (X_a, X_a \wedge Y_a)$. Note that up to boolean isomorphy, such a generic I is completely characterized by the cardinality of $\mathbb{A}$.

Thus, if we fix a generic instantiation I, $\mathcal{A}$ specifies a default base $\Delta^{\mathcal{A}}$ describing the relevant ranking constraints.

$$\Delta^{\mathcal{A}} = \{\varphi_a \rightsquigarrow \psi_a \mid a \in \mathbb{A}\} \cup \{\psi_a \wedge \psi_b \rightsquigarrow F \mid a \rhd b \text{ or } b \rhd a\}$$
$$\cup\{\varphi_a \wedge \varphi_b \rightsquigarrow \psi_a \mid a \rhd b, b \not\rhd a\}.$$

If the indices $a \rhd b$ and $a \lhd/\rhd b$ indicate one-sided resp. any-sided attacks, the unique justifiably constructible ranking measure model of $\Delta^{\mathcal{A}}$ is

$$R^{\mathcal{A}}_{jz} = R_0 + \Sigma_{a\not\rhd a} 1[\varphi_a \wedge \neg\psi_a] + \Sigma_{a\rhd a}\infty[\varphi_a \wedge \neg\psi_a] + \Sigma_{a\rhd b}1[\varphi_a \wedge \varphi_b \wedge \neg\psi_a] +$$
$$\Sigma_{a\lhd/\rhd b}\infty[\psi_a \wedge \psi_b].$$

Because the $\{X_a, Y_a\}$ are logically independent for distinct a, and the defaults expressing an attack $a \rhd b$ just concern $\varphi_a \wedge \varphi_b$, only those φ_a with $a \rhd a$ become impossible. In fact, $\{\varphi_a \rightsquigarrow \psi_a, \psi_a \wedge \psi_a \rightsquigarrow F\} \vdash_{rk} \varphi_a \rightsquigarrow F$. Hence, in line with intuition, $(R^{\mathcal{A}}_{jz}, I)$ trivializes exactly the self-defeating arguments. Assuming genericity, $\mathbb{A}^- = \{a \in \mathbb{A} \mid a \not\rhd a\}$ is therefore the only maximal coherent subset of $\mathbb{A}$. Note that $(R^{\mathcal{A}}_{jz}, I)$ doesn't necessarily characterize $\mathcal{A}$. For instance, $\mathcal{A} = (\{a, b\}, \{(b, a), (a, a)\})$ and $\mathcal{A}' = (\{a, b\}, \{(b, a), (a, b), (a, a)\})$ both produce the same $R^{\mathcal{A}}_{jz} = R^{\mathcal{A}'}_{jz} = R_0 + \infty[\varphi_a] + 1[\varphi_b \wedge \neg\psi_b]$. We are now ready to specify our JZ-evaluation semantics. Note that all the generic I are equivalent.

JZ-evaluation semantics:
$\mathcal{E}_{jz} = \{E \subseteq \mathbb{A} \mid E$ is a ranking extension w.r.t. $(R^{\mathcal{A},I}_{jz}, I)$ for any/all generic $I\}$.

There is a simple way to identify the JZ-extensions through extension weights.

Definition 5.2 (Extension weight). *For each argumentation framework $\mathcal{A} = (\mathbb{A}, \rhd)$, the extension weight function $r_{\mathcal{A}} : 2^{\mathbb{A}} \to [0, \infty]$ is defined as follows:*
If E is conflict-free, $r_{\mathcal{A}}(E) = |\mathbb{A}^- - E| + |\{a \in \mathbb{A}^- - E \mid \exists b \in \mathbb{A}^-(a \rhd b \wedge b \not\rhd a)\}|$, if not, $r_{\mathcal{A}}(E) = \infty$.

It is not too difficult to see that $r_{\mathcal{A}}(E) = R^{\mathcal{A},I}_{jz}(\psi_{\mathbb{A}^-,E})$. Hence, $E \in \mathcal{E}_{jz}(\mathcal{A})$ iff $r_{\mathcal{A}}(E) = min\{r_{\mathcal{A}}(X) \mid X \subseteq \mathbb{A}\}$. That is, the JZ-extensions are those where the sum of the number of non-reflective non-extension arguments and the number of one-sided attacks starting from them is minimal.

6 Examples and Properties

To get a better understanding of the ranking extension semantics and its position in the space of extension concepts, let us first take a look at how it handles some basic examples. Because of its uncommon semantic perspective and its partly quantitative character, we will see some unorthodox behaviour. Under instantiation genericity, it is enough to compare $R^{\mathcal{A}}(\psi_{\mathbb{A}^-,E})$ for $E \subseteq \mathbb{A}^-$, or to focus on 1-loop-free frameworks. For each instance, we specify the domain $\mathbb{A}$ and the full attack relation $\rhd$. $\psi_{\mathbb{A}^-,\{x_1 \ldots x_n\}}$ is abbreviated by $\psi_{x_1,\ldots,x_n}$ resp. $\psi_\emptyset$.

Simple reinstatement: $\{a, b, c\}$ with $a \rhd b \rhd c$.

The grounded extension $\{a, c\}$ is the canonical result put forward by any standard acceptability semantics. The unique JJ-model, i.e. the JZ-model R of $\Delta^{\mathcal{A},I}$,

satisfies $R(\psi_a) = R(\psi_b) = 3, R(\psi_c) = 4, R(\psi_{a,c}) = 2$, and $R(\psi_\emptyset) = 5$. The other candidates all get rank ∞. Because $R(\psi_{a,c})$ is minimal, $\{a, c\}$ is the only JZ-ranking extension, i.e. $\mathcal{E}_{jz}(\mathcal{A}) = \{\{a, c\}\}$.

3-loop: $\{a, b, c\}$ with $a \rhd b \rhd c \rhd a$.

Semantics under the admissibility dogm reject $\{a\}, \{b\}, \{c\}$, only $\emptyset$ is admissible. But the JZ-model R verifies $R(\psi_a) = R(\psi_b) = R(\psi_c) = 4 < 5 = R(\psi_\emptyset)$. Because all the alternatives are set to ∞, our ranking extensions are the maximal conflict-free sets $\{a\}, \{b\}, \{c\}$, i.e., $\mathcal{E}_{jz}$ clearly violates admissibility.

Attack on 2-loop: $\{a, b, c\}$ with $a \rhd b \rhd c \rhd b$.

We have $R(\psi_\emptyset) = 4, R(\psi_a) = 2, R(\psi_b) = R(\psi_c) = 3, R(\psi_{a,c}) = 1$, but ∞ for the others. Here $\mathcal{E}_{jz}(\mathcal{A}) = \{\{a, c\}\}$ picks up the canonical stable extension.

Attack from 2-loop: $\{a, b, c\}$ with $b \rhd a \rhd b \rhd c$.

We get $R(\psi_\emptyset) = 4, R(\psi_a) = 3, R(\psi_b) = 2, R(\psi_c) = 3$, $R(\psi_{a,b}) = R(\psi_{b,c}) = \infty$, and $R(\psi_{a,c}) = 2$. Thus, $\mathcal{E}_{jz}(\mathcal{A}) = \{\{b\}, \{a, c\}\}$ collects the stable extensions.

3,1-loop: $\{a, b, c\}$ with $a \rhd a \rhd b \rhd c \rhd a$.

$E = \emptyset$ is here the only admissible extension. The maximal coherent set is $\mathbb{A}^- = \{b, c\}$, and we get $R(\psi_b) = 1, R(\psi_c) = 2$, as well as $R(\psi_\emptyset) = 3$. It follows that $\mathcal{E}_{jz}(\mathcal{A}) = \{\{b\}\}$, rejecting the stage extension $\{c\}$.

3,2-loop: $\{a, b, c\}$ with $b \rhd a \rhd b \rhd c \rhd a$.

We have $R(\psi_\emptyset) = 5, R(\psi_a) = 4, R(\psi_b) = 3$, and $R(\psi_c) = 3$, i.e. $\mathcal{E}_{jz}(\mathcal{A}) = \{\{b\}, \{c\}\}$. The stable extension $\{b\}$ is the only admissible ranking extension.

The previous examples show that the ranking extension semantics $\mathcal{E}_{jz}$ diverges from all the other major proposals found in the literature. It may look as if the main difference is its more liberal attitude towards some non-admissible, but still justifiable extensions. However, the semantics is more exotic than this. Consider the following examples, where we indicate the minimal extension weights $r_{\mathcal{A}}(E)$.

2-loop chain: $\{a, b, c\}$, $b \rhd a \rhd b \rhd c \rhd b$: $r(\{a, c\}) = 1 < 2 = r(\{b\})$.

Splitted 3-chain: $\{a, b, c, d\}$, $a \rhd b \rhd c$, $a \rhd d \rhd c$: $r(\{a, c\}) = r(\{b, d\}) = 4$.

Spoon: $\{a, b, c, d\}$, $a \rhd b \rhd c \rhd d \rhd c$: $r(\{a, d\}) = r(\{a, c\}) = r(\{b, d\}) = 3$.

The first example documents the rejection of a stable extension, namely $\{b\}$. The second one shows the impact of quantitative considerations when dealing with a splitted variant of simple reinstatement. The third instances illustrates the coexistence of two stable extension with a non-admissible one. This shows that even attack-free a can be questioned. Thus, the above ranking semantic interpretation of argumentation frameworks deviates considerably from standard

accounts and expectations. Let us now see how $\mathcal{E}_{jz}$ handles some common principles for extension semantics.

Isomorphy. $f : \mathcal{A} \cong \mathcal{A}'$ implies $\mathcal{E}(\mathcal{A}') = f''\mathcal{E}(\mathcal{A})$.

Conflict-freedom. If $a, b \in E \in \mathcal{E}(\mathcal{A})$, then $a \not\rhd b$.

CF-maximality. If $E \in \mathcal{E}(\mathcal{A})$, then E is a maximal conflict-free subset of $\mathbb{A}$.

Inclusion-maximality. If $E, E' \in \mathcal{E}(\mathcal{A})$ and $E \subseteq E'$, then $E = E'$.

Reinstatement. If $E \in \mathcal{E}(\mathcal{A})$, $a \in \mathbb{A}$, and for each $b \rhd a$, there is an $a' \in E$ with $a' \rhd b$, then $a \in E$.

Directionality. Let $\mathcal{A}_1 = (\mathbb{A}_1, \rhd_1), \mathcal{A}_2 = (\mathbb{A}_2, \rhd_2)$ be such that $\mathbb{A}_1 \cap \mathbb{A}_2 = \emptyset$, $\rhd_0 \subseteq A_1 \times A_2$, $\mathcal{A} = (\mathbb{A}_1 \cup \mathbb{A}_2, \rhd_1 \cup \rhd_0 \cup \rhd_2)$. Then $\mathcal{E}(\mathcal{A}_1) = \{E \cap \mathbb{A}_1 \mid E \in \mathcal{E}(\mathcal{A})\}$.

Theorem 6.1 (Basic properties).
$\mathcal{E}_{jz} = \mathcal{E}_{jj}$ verifies isomorphy, conflict-freedom, inclusion maximality, and CF-maximality. It falsifies reinstatement and directionality.

The first four features are easy consequences of the $\mathcal{E}_{jz}$-specification. The violation of reinstatement directly follows from how the semantics handles 3-loops. The spoon example documents the failure of directionality if we set $\mathbb{A}_1 = \{a, b\}$. But this property also fails for other prominent approaches, like the semi-stable semantics. Note however that it can be indirectly enforced by using $\mathcal{E}_{jz}$ as the base function for an SCC-recursive semantics [2].

The following properties are inspired by the cumulativity principle for non-monotonic reasoning. They state that if we drop an argument rejected by every extension, then this shouldn't add or erase skeptically supported arguments.

Rejection cumulativity: ($\mathcal{A}|B$ here means $\mathcal{A}$ restricted to B.)
– **Rej-CUT :** If $a \notin \cup\mathcal{E}(\mathcal{A})$, then $\cap\mathcal{E}(\mathcal{A}|\mathbb{A} - \{a\}) \subseteq \cap\mathcal{E}(\mathcal{A})$.
– **Rej-CM :** If $a \notin \cup\mathcal{E}(\mathcal{A})$, then $\cap\mathcal{E}(\mathcal{A}) \subseteq \cap\mathcal{E}(\mathcal{A}|\mathbb{A} - \{a\})$.

Although our semantics relies on default inference notions verifying cumulativity at the level of L, it nevertheless fails to validate these postulates.

Theorem 6.2 (No rejection cumulativity). *$\mathcal{E}_{jz}$ violates Rej-CUT, Rej-CM.*

The counterexample for Rej-CUT is provided by $b \rhd c \rhd a \rhd b \rhd a$, because $\{b\} \not\subseteq \{b\} \cap \{c\}$. The one for Rej-CM is obtained by adding $c \rhd b$. Here $\{c\} \not\subseteq \{b\} \cap \{c\}$.

Another idea for combining plausibilistic default reasoning and argumentation theory has been presented in [7]. It combines defeasible logic programming with a prioritization criterion based on System Z. While it handles some benchmarks better than the individual systems do, its heterogeneous character makes it hard to assess. It doesn't share our goal to seek a plausibilistic semantics for abstract argumentation and also seems to produce different results even in the generic context. It is unclear whether replacing system Z could help.

We have shown how the ranking construction paradigm for default reasoning can be exploited to interpret abstract argumentation frameworks and to specify

corresponding extension semantics if we focus on generic R and I. We have investigated the simplest semantic instantiations, where arguments are essentially interpreted as pairs of strict and defeasible content. Our basic ranking extension semantics $\mathcal{E}_{jz}$ has interesting properties, but it also exhibits a non-orthodox behaviour which needs further exploration. However, our new semantic perspective appears to be a good starting point for more sophisticated proposals, able to meet further demands.

References

1. Baroni, P., Caminada, M., Giacomin, M.: An introduction to argumentation semantics. The Knowledge Engineering Review 26(04), 365–410 (2011)
2. Baroni, P., Giacomin, M., Guida, G.: SCC-recursiveness: a general schema for argumentation semantics. AIJ 168, 163–210 (2005)
3. Benferhat, S., Saffiotti, A., Smets, P.: Belief functions and default reasoning. Artificial Intelligence 122(1-2), 1–69 (2000)
4. Dung, P.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic progr. and n-person games. AIJ 77, 321–357 (1995)
5. Goldszmidt, M., Morris, P., Pearl, J.: A maximum entropy approach to nonmonotonic reasoning. IEEE Transact. Patt. Anal. and Mach. Int. 15, 220–232 (1993)
6. Kern-Isberner, G.: Conditionals in Nonmonotonic Reasoning and Belief Revision. LNCS (LNAI), vol. 2087. Springer, Heidelberg (2001)
7. Kern-Isberner, G., Simari, G.R.: A Default Logical Semantics for Defeasible Argumentation. In: Proc. of FLAIRS 2011. AAAI Press (2011)
8. Kraus, S., Lehmann, D., Magidor, M.: Nonmonotonic reasoning, preferential models and cumulative logics. Artificial Intelligence 44, 167–207 (1990)
9. Makinson, D.: General patterns of nonmonotonic reasoning. In: Gabbay, et al. (eds.) Handbook of Logic in AI and LP, vol. 3, pp. 35–110. Oxford University Press (1994)
10. Pearl, J.: System Z: a natural ordering of defaults with tractable applications to nonmonotonic reasoning. In: TARK, vol. 3, pp. 121–135. Morgan Kaufmann (1990)
11. Spohn, W.: Ordinal conditional functions: a dynamic theory of epistemic states. In: Harper, W.L., Skyrms, B. (eds.) Causation in Decision, Belief Change, and Statistics, pp. 105–134. Kluwer (1988)
12. Weydert, E.: General belief measures. In: UAI 1994. Morgan Kaufmann (1994)
13. Weydert, E.: Defaults and infinitesimals. Defeasible inference by non-archimdean entropy maximization. In: UAI 1995, pp. 540–547. Morgan Kaufmann (1995)
14. Weydert, E.: System J - revision entailment – Default reasoning through ranking measure updates. In: Gabbay, D.M., Ohlbach, H.J. (eds.) FAPR 1996. LNCS, vol. 1085, pp. 637–649. Springer, Heidelberg (1996)
15. Weydert, E.: System JZ - How to build a canonical ranking model of a default knowledge base. In: KR 1998, pp. 190–201. Morgan Kaufmann (1998)
16. Weydert, E.: System JLZ - Rational default reasoning by minimal ranking constructions. Journal of Applied Logic 1(3-4), 273–308 (2003)
17. Weydert, E.: Semi-stable extensions for infinite frameworks. In: Proc. BNAIC 2012, pp. 336–343 (2012)

Author Index