Working Papers

of the

IJCAI-2017 Workshop on

Logical Foundations for Uncertainty and Machine Learning

LFU-2017

August 20, 2017
Melbourne (Australia)

Preface

The purpose of this workshop is to promote logical foundations for reasoning and learning under uncertainty. Uncertainty is inherent in many AI applications, and coping with this uncertainty, in terms of preferences, probabilities and weights, is essential for the system to operate purposefully. In the same vein, expecting a domain modeler to completely characterize a system is often unrealistic, and so enabling mechanisms by means of which the system can infer and learn about the environment is needed. While probabilistic reasoning and Bayesian learning has enjoyed many successes and is central to our current understanding of the data revolution, a deeper investigation on the underlying semantical issues as well as principled ways of extending the frameworks to richer settings is what this workshop strives for. Broadly speaking, we aim to bring together the many communities focused on uncertainty reasoning and learning – including knowledge representation, machine learning, logic programming and databases – by focusing on the logical underpinnings of the approaches and techniques.

This IJCAI 2017 workshop, LFU-2017, is an evolution of a series of three workshops called "Weigthed Logics for Artificial Intelligence" (WL4AI) that were successfully held in 2012 in collocation with ECAI-2012 in Montpellier (France), in 2013 in collocation with IJCAI-2013 in Beijing (China) and the third one in collocation with IJCAI-2015 in Buenos Aires (Argentina).

We are very happy to gather in this proceedings volume a very interesting set of contributions on different uncertainty formalisms and approaches that we believe are representative of the richness of the area.

Finally, we would like to express our gratitude to:

- Adnan Darwiche and Vanina Martínez for having accepted to give invited talks at this workshop.
- The program committee members for their commitment to the success of this event and for their work.
- The authors of LFU-2017 for the quality of their contributions.

Vaishak Belle, James Cussens, Marcelo Finger, Lluis Godo, Henri Prade and Guilin Qi

Workshop Chairs

Vaishak Belle	University of Edinburgh, UK
James Cussens	University of York, UK;
Marcelo Finger	University of São Paulo, Brazil
Lluis Godo	IIIA - CSIC, Spain
Henri Prade	IRIT - University of Toulouse, France
Guilin Qi	Southeast University, China

Program Committee

Fabio Cozman	University of Sao Paulo, Brazil
Jesse Davies	KU Leuven, Belgium
Adnan Darwiche,	University of California, Los Angeles, USA
Didier Dubois	IRIT, France
Esra Erdem	Sabanci University, Turkey
Linda van der Gaag	Universiteit Utrecht, The Netherlands
Tommaso Flaminio	IIIA-CSIC, Spain
Vibhav Gogate	University of Texas at Dallas, USA
Joe Halpern	Cornell University, USA
Manfred Jaeger	Aalborg University, Denmark
Souhila Kaci	University Montpellier, France
Gabriele Kern-Isberner	Dortmund University, Germany
Gerhard Lakemeyer	RWTH Aachen University, Germany
Churn-Jung Liau	Academia Sinica,Taiwan
Emiliano Lorini	IRIT, France
Thomas Lukasiewicz	Oxford University, UK
Carsten Lutz	University of Bremen, Germany
Denis Maua	University of São Paulo, Brazil
Vanina Martinez	Universidad Nacional del Sur, Argentina
Zoran Ognjanović	Mathematical Institute SANU, Serbia
Ron Petrick	University of Edinburgh, UK
Rodrigo De Salvo Braz	SRI, USA
Giuseppe Sanfilippo	University of Palermo, Italy
Steven Schockaert	Cardiff University, UK
Guillermo Simari	Universidad Nacional del Sur, Argentine
Umberto Straccia	ISTI-CNR, Italy

LFU-2017 Workshop Programme

August 20, 2017, Melbourne, Australia

8.30 - 8.35	**Welcome and Introduction**
Session 1:	**Logic and Uncertainty - I**
8.35 - 9.00	*An overview: Axiomatizations of probabilities with non-standard ranges* Z. Ognjanović, M. Raškovic, Z. Marković, A. Ilic-Stepić, N. Ikodinović, A. Perović and D. Doder
9.00 - 10.00	**Invited talk:** *On the Role of Logic in Probabilistic Inference and Machine Learning* Adnan Darwiche
	Coffee Break 10.00 – 10.30
Session 2:	**Logic and Uncertainty - II**
10.30 - 11.00	*From First-Order Logic to Assertional Logic* Y. Zhou
11.00 - 11.30	*Strength Factors: An Uncertainty System for a Quantified Modal Logic* N. S. Govindarajulu and S. Bringsjord
11.30 - 12.00	*A Model of Multi-Agent Consensus for Compound Sentences* M. Crosscombe and J. Lawry
12.00 - 12.30	*Optimal Feature Selection for Decision Robustness in Bayesian Networks* Y. Choi, A. Darwiche and G. Van den Broeck
	Lunch Break 12.30 – 14.00
14.00 - 15.00	**Invited talk:** *Probabilistic Reasoning with Preferences for the Social Semantic Web* María Vanina Martínez
Session 3:	**Uncertain DBs and Ontologies**
15.00 - 15.30	*Ontology-Mediated Queries for Probabilistic Databases* S. Borgwardt, I. Ilkan Ceylan and T. Lukasiewicz
15.30 - 16.00	*Schema Induction From Incomplete Semantic Data* H. Gao, G. Qi and Q. Ji
	Coffee Break 16.00 – 16.30
Session 4:	**Argumentation and Learning**
16.30 - 17.00	*Towards Argumentation-based Classification* M. Thimm and K. Kersting Authors
17.00 - 17.30	*A probabilistic author-centered model for Twitter discussions* T. Alsinet, J. Argelich, R. Béjar, L. Godo and F. Esteva
17.30 - 18.00	*Learning Possibilistic Logic Theories from Default Rules* O. Kuželka, J. Davis and S. Schockaert
18.00	**Closing**

Table of Contents

On the Role of Logic in Probabilistic Inference and Machine Learning

Adnan Darwiche

University of California
Los Angeles, USA

Abstract

I will discuss in this talk some fundamental roles of logic in probabilistic inference and machine learning. In particular, I will discuss four key probabilistic queries that are complete for the complexity classes NP, PP, NP^{PP} and PP^{PP}, showing how each of these queries can be reduced to logical manipulations. The treatment of these queries will be systematic and based on compiling corresponding problems into Boolean circuits with increasingly strong properties that guarantee linear-time probabilistic inference. I will also link the proposed approach to the recent "Beyond NP" initiative that calls for a similar systematic treatment when handling computational problems that are harder than NP. On the machine learning front, I will show how the proposed approach can be used to: (a) learn statistical models from a combination of data and background knowledge (expressed using logical constraints); (b) learn from a new class of more expressive datasets; and (c) estimate parameters in closed-form in some cases. The talk will include a number of case studies and experimental results, showing the broad scope and competitiveness of the proposed approach, while also highlighting situations where it is now the state of the art.

Probabilistic Reasoning with Preferences for the Social Semantic Web

María Vanina Martínez

Universidad Nacional del Sur
Bahía Blanca, Argentina

Abstract

Reasoning about an entity's preferences (be it a user of an application, an individual targeted for marketing, or a group of people whose choices are of interest) has a long history in different areas of study. In this talk, we adopt the point of view that grows out of the intersection of databases and knowledge representation, where preferences are usually represented as strict partial orders over the set of tuples in a database or the consequences of a knowledge base. We describe how probability theory can be used to model uncertainty in preferences in these domains in two complementary ways. First, we introduce order-based Markov models (OMMs), which flexibly combine preferences with probabilistic uncertainty; unfortunately, the complexity of basic reasoning tasks over these models is intractable, involving exponential factors in the number of statements. To ameliorate this problem, we show how we can exploit the structure of the models to do exact inference much more efficiently when the model is comprised of atomic formulas and the query/evidence are conjunctions of atomic preference formulas. Since the potential application of this kind of models is clear in domains such as the Social Semantic Web (where users often express preferences in an incomplete manner and through different means, often in contradiction with each other), in the second part of the talk we study an extension of the Datalog+/- family of ontology languages with two models: one representing user preferences and one representing the (probabilistic) uncertainty with which inferences are made. Assuming that more probable answers are in general more preferable, the main problem that needs to be solved is how to rank answers to a user's queries, since the preference model may be in conflict with the preferences induced by the probabilistic model.

A probabilistic author-centered model for Twitter discussions*

Teresa Alsinet and **Josep Argelich** and **Ramón Béjar**
INSPIRES Research Center, University of Lleida, Spain
{tracy,jargelich,ramon}@diei.udl.cat

Francesc Esteva and **Lluís Godo**
AI Research Institute, IIIA-CSIC, Bellaterra, Spain
{godo,esteva}@iiia.csic.es

Abstract

In a recent work some of the authors have developed an argumentative approach for discovering relevant opinions in Twitter discussions with probabilistic valued relationships. Given a Twitter discussion, the system builds an argument graph where each node denotes a tweet and each edge denotes a criticism relationship between a pair of tweets of the discussion. Relationships between tweets are associated with a probability value, indicating the uncertainty that the relationships hold. In this work we introduce and investigate a natural extension of the representation model, referred as probabilistic author-centered model, in which tweets within a discussion are grouped by authors, in such a way that tweets of a same author describe his/her opinion in the discussion and are represented with a single node in the graph, and criticism relationships denote controversies between opinions of Twitter users in the discussion. In this new model, the interactions between authors can give rise to circular criticism relationships, and the probability of one opinion criticizing another has to be evaluated from the probabilities of criticism among the tweets that compose both opinions.

1 Introduction

In a recent work [Alsinet *et al.*, 2017a], an argumentative approach has been proposed for discovering relevant opinions in Twitter with probabilistic valued relationships.

Argumentation-based reasoning models aim at reflecting how humans make use of conflicting information to construct and analyze arguments. An argument is an entity that represents some grounds to believe in a certain statement and that can be in conflict with arguments establishing contradictory claims. The most commonly used framework to talk about general issues of argumentation is that of abstract argumentation [Dung, 1995].

In abstract argumentation, a graph is used to represent a set of arguments and counterarguments. Each node is an argument and each edge denotes an attack between arguments. Several different kinds of semantics for abstract argumentation frameworks have been proposed that highlight different aspects of argumentation (for reviews see e.g. [Bench-Capon and Dunne, 2007; Besnard and Hunter, 2001; Rahwan and Simari, 2009]). Usually, semantics for abstract argumentation frameworks are given in terms of sets of extensions. For a specific extension, an argument is either accepted, rejected, or undecided and, usually, there is a set of extensions that is consistent with the semantic context.

Given a Twitter discussion, the system presented in [Alsinet *et al.*, 2017a] builds an argument graph where each node denotes a tweet and each edge denotes a criticism relationship between a pair of tweets of the discussion. In Twitter, a tweet always answers or refers to previous tweets in the discussion, so the obtained underlying argument graph is acyclic. Moreover, when constructing relationships between tweets from informal descriptions expressed in natural language with other attributes such as emoticons, jargon, onomatopoeia and abbreviations, it is often evident that there is uncertainty about whether some of the criticism relationships actually hold. So, in order to deal with such uncertainty, each edge of an argument graph is associated with a probability value, quantifying the uncertainty on criticism relationships between pairs of tweets.

In this work we introduce and investigate a natural extension of this representation model, referred as probabilistic author-centered model. In this new model, tweets within a discussion are grouped by authors, such that tweets of a same author describe his/her opinion in the discussion that is represented by a single node in the graph, and criticism relationships denote controversies between the opinions of Twitter users in the discussion. In this model, the interactions between authors can give rise to circular criticism relationships, and the probability of one opinion criticizing another has to be evaluated from the individual probabilities of criticism among the tweets that compose both opinions. So, the underlying argument graph can contain cycles and a model for the aggregation of probabilities has to be proposed. The representation model can be of special relevance for assessing Twitter discussions in fields where identifying groups of authors whose opinions are globally compatible or consistent is of particular interest.

The rest of the paper is organized as follows. In Section 2, we recall from [Alsinet *et al.*, 2017a] the formal graph struc-

*This work was partially funded by the Spanish MICINN Projects TIN2015-71799-C2-1-P and TIN2015-71799-C2-2-P.

ture to model Twitter discussions. Then, in Section 3, we describe the author-centered model for representing discussions in Twitter and, in Section 4, we formalize the probabilistic weighting scheme of criticism relationships between authors' opinions. Finally, in Section 5 we define the reasoning system to compute the sets of accepted and rejected opinions and, in Section 6, we conclude.

2 Twitter discussion graph

Following the approach proposed in [Alsinet *et al.*, 2017a], in this section, we introduce a computational structure (a probabilistic weighted graph) to represent a Twitter discussion with probabilistic valued relationships. Each node of the graph denotes a tweet, each edge denotes an answer relationship between a pair of tweets of the discussion, and each edge is associated with a probability value, indicating the probability that a criticism relationships between the pair of tweets holds.

Definition 1 *(Twitter Discussion) A Twitter discussion Γ is a non-empty set of tweets. A tweet $t \in \Gamma$ is a triple $t = (m, a, f)$, where m is the up to 140 characters long message of the tweet, a is the author's identifier of the tweet and $f \in \mathbb{N}$ is the number of followers of the author, according to its temporal instant generation during the discussion. Let $t_1 = (m_1, a_1)$ and $t_2 = (m_2, a_2)$ be tweets from different authors; i.e. $a_1 \neq a_2$. We say that t_1 answers t_2 iff t_1 is a reply to the tweet t_2 or t_1 mentions (refers to) tweet t_2.*

Definition 2 *(Discussion Graph) The Discussion Graph* (DisG) *for a Twitter discussion Γ is the directed graph (T, E) such that for every tweet in Γ there is a node in T and if tweet t_1 answers tweet t_2 there is a directed edge (t_1, t_2) in E. Only the nodes and edges obtained by applying this process belong to T and E, respectively.*

Definition 3 *(Probabilistic Discussion Graph) A probabilistic discussion graph* (PDisG) *for a Twitter discussion Γ is a triple $\langle T, E, P \rangle$, where*

- *(T, E) is the* DisG *graph for Γ and*
- *P is a labeling function $P : E \to [0, 1]$ for edges in E. The labeling function P maps an edge (t_1, t_2) to a probability value $p \in [0, 1]$, which expresses the degree of belief that the message of tweet t_1 is a criticism to the message of tweet t_2. Criticism means that the message of tweet t_1 does not agree with the claim expressed in the message of tweet t_2. So, $p = 1$ means that we fully believe that tweet t_1 disagrees with the claim expressed in tweet t_2, while $p = 0$ means that we believe that tweet t_1 agrees with the claim expressed in tweet t_2.*

Given a PDisG $\langle T, E, P \rangle$ for a Twitter discussion Γ, we say that a tweet $t_1 \in \Gamma$ *criticizes* a tweet $t_2 \in \Gamma$, written $t_1 \rightsquigarrow t_2$, iff t_1 answers t_2 and the degree of belief that the message of tweet t_1 is a criticism to the message of tweet t_2 is greater than zero; i.e. $t_1 \rightsquigarrow t_2$ iff $(t_1, t_2) \in E$ and $P(t_1, t_2) > 0$.

Since the social network we are considering in this work is Twitter, every tweet of a discussion can reply at most one tweet, but can mention many tweets , and all of them are prior in the discussion. So, every tweet can answer and, in turn, can criticize many prior tweets, from a same author or from different authors, of the discussion. Given a tweet t_1, we consider the set of tweets $\{t_{1,a_1}, \ldots, t_{1,a_n}\}$ that t_1 is answering to, as the set of tweets that includes the tweet that t_1 is replying to plus all the other previous tweets in the discussion that are from authors mentioned by t_1.

To check whether a tweet t_1 does not agree with the claim expressed in one of its answered tweets t_{1,a_i}, the system uses an automatic labeling system based on Support Vector Machines (SVM). Our current SVM model is built from a set of 582 pairs of tweets (answers) obtained from a discussion set on Spanish politics, and manually labeled with the most probable label: criticism or not criticism. To build the SVM model, for each pair of tweets (t_1, t_{1,a_i}) we consider different attributes from the tweets of the pair: attributes that count the number of occurrences of relevant words in the tweets and attributes that have to be computed from the message. In particular, for each tweet, we have considered regular words and stop-words, the number of images, the number of URLs mentioned in the tweet, the number of positive and negative emoticons and the sentiment expressed by the tweet. We use LibSVM ([Chang and Lin, 2011]) to train a probabilistic SVM model, that is, a labeling function that assigns a probability value p for each possible label to each answer (t_1, t_{1,a_i}). The probability estimates can be obtained using the likelihood methods of [Platt, 1999]. LibSVM uses the same Platt method but algorithmically improved ([Lin *et al.*, 2007]). With our SVM model for Spanish politics discussions, we obtain an accuracy of 75% over our training set of tweet pairs. This SVM model, obtained from such small data set, may not be good enough to be used in a final system, but one can always consider training a SVM model with a larger data set.

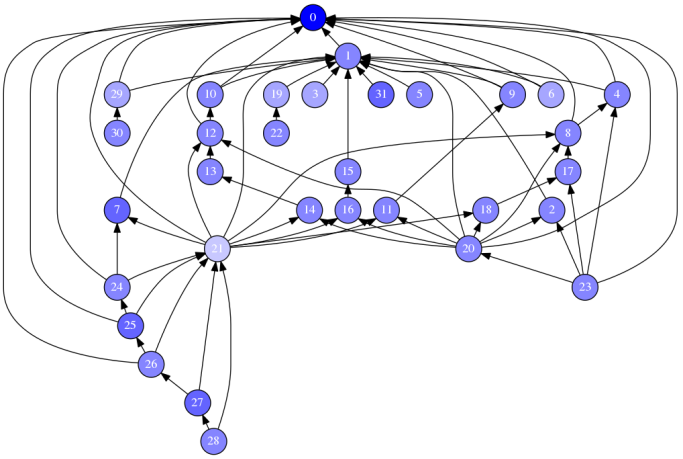


Figure 1: Tweet-based model for a Twitter discussion.

In Figure 1 we show the PDisG for a Twitter discussion [1] from the political domain obtained by our discussion retrieval system. Each tweet is represented as a node and each criticism relationship between tweets is represented as an edge

[1] The discussion URL is `https://twitter.com/jordievole/status/574324656905281538`

(answers with probability values greater than zero). The root tweet of the discussion is labeled with 0 and the other tweets are labeled with consecutive identifiers according to their generation order. The discussion has a simple structure. The root tweet starts the discussion (node 0), the reply (node 1) criticizes the root tweet and the rest of tweets within the discussion criticize mainly node 0 and node 1. The discussion contains 32 tweets of 14 different authors, and 81 criticizes relations between tweets. Nodes are colored in *blue scale*, where the darkness of the color is directly proportional to the number of followers of the authors of the tweets with respect to the maximum value in the discussion. Notice that the graph does not contain cycles, since a tweet only answers previous tweets in the discussion.

3 Author-centered model

As we have already pointed out, our goal is to introduce and investigate an author-centered model of Twitter discussions with probabilistic valued relationships. To this end, tweets are grouped by authors and criticism relationships between tweets denote controversies at the level of authors.

In this work we consider discussions in which every author's opinion is consistent. That is, we consider discussions such that an author is not self-referenced and we assume that every author does not contradict himself; i.e. for each author a_i and each pair of tweets $t_1 = (m_1, a_i, f_1)$ and $t_2 = (m_2, a_i, f_2)$, we assume that messages m_1 and m_2 do not express neither conflicting nor inconsistent information. Next we formalize the notion of author's opinion and number of followers of an author in a discussion.

Let Γ be a Twitter discussion with authors' identifiers $\{a_1, \ldots, a_n\}$. The opinion of an author $a_i \in \{a_1, \ldots, a_n\}$ in the discussion, denoted T_{a_i}, is the set of tweets of a_i in Γ; i.e. $T_{a_i} = \{(m, a, f) \in \Gamma : a = a_i\}$.

The number of followers of an author $a_i \in \{a_1, \ldots, a_n\}$ in the discussion, denoted $f_{a_i} \in \mathbb{N}$, is the mode of the set $\{f \mid (m, a_i, f) \in \Gamma\}$, which provides us the most frequent number of followers of the author during the discussion.

Given a Twitter discussion, in what follows, we assume that every author a_i can be represented by his/her opinion T_{a_i}, since for each discussion there is a unique set of authors and, for each author, there is a unique opinion in the discussion. So, we shall refer to both terms indistinctly.

Definition 4 *(Probabilistic Author Graph) Let Γ be a Twitter discussion with authors' identifiers $\{a_1, \ldots, a_n\}$ and let $\langle T, E, P\rangle$ be the* PDisG *for Γ. The probabilistic author graph (*ADisG*) for Γ is a triple $\langle \mathcal{T}, \mathcal{E}, \mathcal{P}\rangle$, where*

- *the set of nodes $\mathcal{T}$ is the set of authors' opinions $\{T_{a_1}, \ldots, T_{a_n}\}$; i.e. there is a node T_{a_i} in $\mathcal{T}$ iff there is an author with identifier a_i in $\{a_1, \ldots, a_n\}$.*
- *the set of edges $\mathcal{E}$ is the set of answers between different authors of the discussion; i.e. there is an edge (T_{a_i}, T_{a_j}) in $\mathcal{E}$, with $a_i \neq a_j$, iff for some $(t_1, t_2) \in E$, $t_1 \in T_{a_i}$ and $t_2 \in T_{a_j}$.*
- *$\mathcal{P}$ is a probabilistic weighting scheme $\mathcal{P} : \mathcal{E} \to [0, 1]$ for edges in $\mathcal{E}$. The probabilistic weighting scheme $\mathcal{P}$ maps an edge $(T_{a_i}, T_{a_j}) \in \mathcal{E}$ to a probability value in $[0, 1]$ which expresses the probability or degree of belief that the author a_i criticizes the author a_j. For each edge $(T_{a_i}, T_{a_j}) \in \mathcal{E}$, $\mathcal{P}(T_{a_i}, T_{a_j})$ is evaluated from the set of beliefs that the tweets in T_{a_i} criticizes the tweets in T_{a_j}; i.e. $\mathcal{P}(T_{a_i}, T_{a_j})$ is to be computed from the following set of probabilities values:*

$$\{P(t_1, t_2) \mid (t_1, t_2) \in E \text{ and } t_1 \in T_{a_i} \text{ and } t_2 \in T_{a_j}\}.$$

Notice that an author can answer several authors in a discussion, and thus, can criticize several authors. However, if an author criticizes the opinion of another author through several tweets, the set of discrepancies is represented with a single edge in $\mathcal{E}$ and with a single probability value, which is meant to denote the belief that one opinion criticizes the other.

The ADisG graph shows discrepancies between authors only if there is some (explicit) criticism relationship between the tweets of the authors, and thus, indirect criticism relations between authors have not been considered yet in our model. For instance, consider a Twitter discussion with tweets $t_1 = (m_1, a_1, f_1)$, $t_2 = (m_2, a_2, f_2)$ and $t_3 = (m_3, a_3, f_3)$, with $a_1 \neq a_2 \neq a_3$. Suppose that t_1 criticizes t_2 and t_3 criticizes t_1; i.e. $\{(a_1, a_2), (a_3, a_1)\} \subseteq E$, $P(a_1, a_2) > 0$ and $P(a_3, a_1) > 0$. In our current approach, we restrict ourselves to consider that t_3 criticizes t_2 iff t_3 answers t_2. The reason is that the information contained in a typical tweet, written with natural language and with possibly other attributes, almost never allows us to consider a sound way to assess an indirect criticism relation between two tweets t and t' if t' does not directly reply or mention t.

Next section is devoted to the formalization of three different probabilistic weighting schemes, depending on the semantics assumed for the criticism relation between two authors' opinions.

4 Probabilistic weighting schemes

In our approach, each node of a ADisG graph denotes an author's opinion, since it represents the set of tweets of the author within a discussion, and relationships between nodes are mined from the prevailing sentiment between the aggregated tweets of the opinions.

In the following, let Γ be a Twitter discussion and let $\langle \mathcal{T}, \mathcal{E}, \mathcal{P}\rangle$ be the probabilistic author graph (ADisG) for Γ. Suppose further we have two authors' opinions or sets of authors' tweets $T_a, T_b \in \mathcal{T}$, with $(T_a, T_b) \in \mathcal{E}$. Our aim is to define a probabilistic weighting scheme $\mathcal{P} : \mathcal{E} \to [0, 1]$ for edges in $\mathcal{E}$, by combining in an appropriate form the individual probabilities values $\{P(t_1, t_2) \mid t_1 \in T_a \text{ and } t_2 \in T_b\}$, where we consider $P(t_1, t_2) = 0$ for pairs of tweets such that $(t_1, t_2) \notin E$. As we will see, the addition of zero values to this set will be harmless.

Next we define three possible probabilistic weighting schemes $\mathcal{P}$ for answers between authors of a discussion, which depend on the semantics assumed for criticism relationships between two authors' opinions T_a and T_b.

4.1 Skeptical scheme

A skeptical notion of criticism between T_a and T_b can be defined as follows: T_a criticizes T_b, written $T_a \rightsquigarrow T_b$, when for all $t \in T_b$, there is $t' \in T_1$ such that $t' \rightsquigarrow t$.

In logical terms, we can express $T_a \rightsquigarrow T_b$ as equivalent to the following clause:

$$T_a \rightsquigarrow T_b := \bigwedge_{t \in T_b} \left(\bigvee_{t' \in T_a} t' \rightsquigarrow t \right)$$

Assuming independence of all the $t' \rightsquigarrow t$'s, which is a reasonable assumption in our context [2], we can easily compute the probability of $T_a \rightsquigarrow T_b$ as

$$\mathcal{P}(T_a \rightsquigarrow T_b) = \prod_{t \in T_b} \left(\bigoplus_{t' \in T_a} P(t', t) \right),$$

where $\oplus$ corresponds to the probabilistic sum operation $x \oplus y = x + y - x \cdot y$. Observe that 0 is a neutral element for $\oplus$ (i.e. $x \oplus 0 = x$), and so having probability values such that $P(t', t) = 0$ does not affect the computation of $\mathcal{P}(T_a \rightsquigarrow T_b)$. Analogously for the next schemes.

4.2 Credulous scheme

A credulous notion of criticism between T_a and T_b can be defined as follows: T_a criticizes T_b, written $T_a \rightsquigarrow^c T_b$, when for at least some $t \in T_b$, there is $t' \in T_1$ such that $t' \rightsquigarrow t$.

In logical terms, $T_a \rightsquigarrow^c T_b$ can be now equivalently expressed as the following clause:

$$T_a \rightsquigarrow^c T_b := \bigvee_{t \in T_b} \left(\bigvee_{t' \in T_a} t' \rightsquigarrow t \right).$$

Again, assuming independence of all the $t' \rightsquigarrow t$'s, we can easily compute the probability of $T_a \rightsquigarrow T_b$ as

$$\mathcal{P}(T_a \rightsquigarrow^c T_b) = \bigoplus_{t' \in T_a, t \in T_b} P(t', t).$$

4.3 Intermediate scheme

A more flexible definition of when T_1 criticizes T_2 is to stipulate that this holds when for *most* of the tweets $t \in T_b$ there is a tweet $t' \in T_a$ such that $a \rightsquigarrow b$. We denote this notion of attack as $T_1 \rightsquigarrow_{most} T_2$.

The question is how we interpret the quantifier *most*. A first option is to understand *most* as a proportion of at least r, for some $r \geq 0.5$ to be chosen. For any set X, let us define $most(X) = \{S \subseteq X \mid \frac{|S|}{|X|} \geq r\}$. Then we can express $T_1 \rightsquigarrow_{most} T_2$ as follows:

$$T_a \rightsquigarrow_{most} T_b := \bigvee_{S \in most(T_b)} T_a \rightsquigarrow S.$$

But we can simplify a bit this expression. Indeed, since if $S \subset R$ then $(T_1 \rightsquigarrow S) \vee (T_1 \rightsquigarrow R) = T_1 \rightsquigarrow S$, we can write

$$T_a \rightsquigarrow_{most} T_b := \bigvee_{S \in Min(most(T_b))} T_a \rightsquigarrow S,$$

where $Min(most(X))$ denotes the minimal subsets of X with a proportion of at least r. Then, we can compute:

$$\mathcal{P}(T_a \rightsquigarrow_{most} T_b) = \mathcal{P}(\bigvee \{T_a \rightsquigarrow S : S \in Min(most(T_b))\}).$$

This can be computationally expensive. But we can provide a lower approximation as follows: taking into account that for any probability we have $P(A \cup B) \geq \max(P(A), P(B))$, a lower approximation can be computed as:

$$\mathcal{P}_*(T_a \rightsquigarrow_{most} T_b) = \max\{\mathcal{P}(T_a \rightsquigarrow S) : S \in Min(most(T_b))\}.$$

Actually there is a simple procedure to compute P_*:

(i) compute, for all $t \in T_b$, the probabilities $\mathcal{P}(T_a \rightsquigarrow t) = \bigoplus_{t' \in T_a} P(t', t)$;

(ii) rank them, from higher to lower: $P(T_a \rightsquigarrow t_1) \geq P(T_a \rightsquigarrow t_2) \geq \ldots$;

(iii) let k be the smallest index such that $\frac{k}{|T_b|} \geq r$.

Then $\mathcal{P}_*(T_a \rightsquigarrow_{most} T_b) = \prod_{i=1}^{k} P(T_a \rightsquigarrow t_i)$

5 Mining the set of consistent opinions

Once we have introduced the author-centered model of discussions in Twitter, the next key component is the definition of the reasoning system to compute the set of accepted authors' opinions. To this end, we have extended the reasoning system developed in [Alsinet *et al.*, 2017b] to deal with PDisG graphs. The reasoning system maps a PDisG graph, with a particular probabilistic weighting scheme, to a valued abstract argumentation framework (VAF) and considers the ideal semantics [Dung *et al.*, 2007] for computing the set of consistent authors' opinions of the discussion. The ideal semantics guarantees that all opinions in the solution are consistent and that the solution is maximal in the sense that the solution contains all acceptable arguments.

Valued abstract argumentation is based on the extension of abstract argumentation with a valuation function *Val* on a set of values R for arguments and a (possible partial) preference relation *Valpref* between values in R. In our approach, we use the valued argumentation framework introduced by Bench-Capon in [Bench-Capon, 2002], and we consider an uncertainty threshold α which characterizes how much uncertainty on probability values we are ready to tolerate.

Definition 5 *(VAF for a PDisG) Let Γ be a Twitter discussion with authors identifiers $\{a_1, \ldots, a_n\}$ and let $\alpha \in [0, 1]$ be a threshold on the probability values. If $G = \langle \mathcal{T}, \mathcal{E}, \mathcal{P} \rangle$ is the ADisG for Γ with probabilistic weighting scheme $\mathcal{P}$, the Valued Argumentation Framework (VAF) for G relative to the threshold α is a tuple $\langle \mathcal{T}, attacks, R, Val, Valpref \rangle$, where*

- *each node (or author's opinion) T_{a_i} in $\mathcal{T}$ results in an argument,*
- *attacks is an irreflexive binary relation on $\mathcal{T}$ and it is defined according to the threshold α as follows:*

$$attacks = \{(T_{a_i}, T_{a_j}) \in \mathcal{E} \quad | \quad \mathcal{P}(T_{a_i}, T_{a_j}) \geq \alpha\},$$

- *R is a non-empty set of ordered values that models the authors' relevance in Twitter,*

[2] This is because in our probabilistic model the label $P(t_1, t_2)$ assigned to an edge (t_1, t_2) is based only on the information inside the tweets t_1 and t_2 and not on other answers from the same authors.

- *Val* : $\mathcal{T} \rightarrow R$ *is a valuation function for arguments that maps the author's opinion to the authors' relevance in Twitter, and*
- *Valpref* $\subseteq R \times R$ *is the ordering relation (transitive, irreflexive and asymmetric) over the authors' relevance values* R*.*

An important element of our approach is the use of an uncertainty threshold, which characterizes how much uncertainty on probability values we are prepared to tolerate: given an uncertainty threshold α, we would be prepared to disregard criticism relationships between authors' opinions up to α. So, the *attacks* relation is interpreted as the fact that the opinion of the author a_i is in disagreement with the opinion of the author a_j with at least a probability value α according to the probabilistic weighting scheme $\mathcal{P}$.

Given a Twitter discussion, in our approach we build a VAF where arguments denote the authors' opinions and attacks between arguments denote discrepancies between the authors' opinions according to a probabilistic weighting scheme and an uncertainty threshold. Then, given such a VAF $\langle \mathcal{T}, \textit{attacks}, R, \textit{Val}, \textit{Valpref} \rangle$, a *defeat* relation (or effective attack relation) between arguments (authors' opinions) is defined according to a valuation function *Val* and a preference relation *Valpref* as follows: $\textit{defeats} = \{(Ta_i, T_{a_j}) \in \textit{attacks} \mid (\textit{Val}(Ta_j), \textit{Val}(Ta_i)) \notin \textit{Valpref}\}$.

As we have already pointed out, we consider the ideal semantics for computing the set of consistent authors' opinions of a discussion. The ideal semantics for valued argumentation guarantees that the set of tweets in the solution is the maximal set of tweets that satisfies that it is consistent, in the sense that there are no defeaters among them, and that all of the tweets outside the solution are defeated by a tweet within the solution. That is, if a tweet outside the solution defeats a tweet within the solution, it is, in turn, defeated by another tweet within the solution. In other words, the solution is the biggest consistent set of tweets that defeats any defeater outside the solution.

Formally, given a VAF $\langle \mathcal{T}, \textit{attacks}, R, \textit{Val}, \textit{Valpref} \rangle$, a set of arguments $S \subseteq \mathcal{T}$ is *conflict-free* iff for all $T_{a_i}, T_{a_j} \in S, (T_{a_i}, T_{a_j}) \notin \textit{defeats}$. Given a conflict-free set of arguments $S \subseteq \mathcal{T}$, S is *maximally admissible* iff

(i) for all $T_{a_1} \notin S$, $S \cup \{T_{a_1}\}$ is not conflict-free and

(ii) for all $T_{a_1} \notin S$ and $T_{a_2} \in S$, if $(T_{a_1}, T_{a_2}) \in \textit{defeats}$, there exists $T_{a_3} \in S$ such that $(T_{a_3}, T_{a_1}) \in \textit{defeats}$.

Finally, given an uncertainty threshold α and a probabilistic weighting scheme $\mathcal{P}$, the *set of accepted authors' opinions* of a discussion Γ, referred as the *solution* of Γ, is defined from the VAF $\mathcal{F}$=$\langle \mathcal{T}, \textit{attacks}, R, \textit{Val}, \textit{Valpref} \rangle$ and the *defeat* relation between the authors' opinions in $\mathcal{T}$, and it is computed as the largest admissible conflict-free set of authors' opinions $S \subseteq \{T_{a_1}, \ldots, T_{a_n}\}$ in the intersection of all maximally admissible conflict-free sets.

As for the implementation purposes, we have instantiated the set of ordered values R to the natural numbers $\mathbb{N}$ and the preference relation *Valpref* to the natural order relation on $\mathbb{N}$. We have also considered a valuation function $\texttt{followers} : \mathcal{T} \rightarrow \mathbb{N}$, that allows us to quantify authors' relevance from the orders of magnitude of authors' followers by defining:

$$\texttt{followers}(T_{a_i}) = \lfloor \log_{10}(f_{a_i} + 1) \rfloor,$$

where $f_{a_i} \in \mathbb{N}$ is the number of followers of the author a_i computed as the mode of the set $\{f \mid (m, a_i, f) \in T_{a_i}\}$, which provides us with the most frequent number of followers of the author during the discussion.

To implement the reasoning system, we have used an approach based on Answer Set Programming (ASP) described in [Egly *et al.*, 2008], and available in the argumentation system ASPARTIX. We have extended ASPARTIX to deal with VAFs, as the current implementation only works with non-valued arguments. To develop such extension we have modified the manifold ASP program described in [Faber and Woltran, 2009] to incorporate the valuation function for arguments and the preference relation.

The author-centered approach allows us to perform an analysis of results different from the tweet-based approach proposed in [Alsinet *et al.*, 2017b]. Aggregating the information by author allows us to identify the set of authors whose opinions are consistent or in agreement in the discussion, the authors involved in a circular argumentative discussion, and the most controversial authors. That is, for instance, we can look for the authors who receive the greatest number of criticisms, the authors who participate in the greatest number of cycles, or the authors that generate the longest argumentative chains.

Figure 2 shows the solution for a ADisG graph instance for the discussion of Figure 1. To build the ADisG graph, we have used the intermediate probabilistic weighting scheme $\mathcal{P}_*(T_{a_i} \rightsquigarrow_{most} T_{a_j})$ with the proportion parameter $r = 0.6$ [3]. To find the solution for the ADisG graph (the set of accepted opinions of the discussion), we have used the uncertainty threshold $\alpha = 0.6$ and the `followers` valuation function for estimating the authors' relevance in Twitter. The nodes colored in blue are the accepted authors (authors' opinions in the solution) and the nodes colored in gray are the rejected ones, where the darkness of the color is directly proportional to the `followers` valuation function of each author in the discussion with respect to the maximum value. The edges colored in black are the answers between authors that can not be classified as attacks, since the criticism probabilities are below the threshold $\alpha = 0.6$, while the edges colored in red are attacks between authors; i.e. answers with a criticism probability of at least the threshold $\alpha = 0.6$. For attack edges, the darkness of the color is directly proportional to the criticism probability with respect to the maximum value. With $r = 0.6$ and $\alpha = 0.6$, 11 answers between authors do not give rise to attacks. The ADisG has 13 cycles considering all answers among authors and authors 8 and 2 seem to be the most controversial authors. The solution contains 11 of the 14 authors and only 3 are rejected. According to the `followers` valuation function, the authors of the discussion are stratified in five levels denoting their relevance. They are level 0 (lowest level): $\{11\}$, level 1: $\{5, 6, 7, 13\}$, level 2: $\{0, 1, 3, 4, 8, 9, 10\}$, level 3: $\{12\}$ and level 4: $\{2\}$.

[3] We plan to implement the other weighting schemes in the near future.

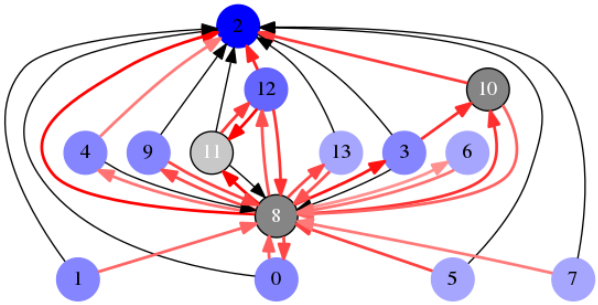


Figure 2: Author-centered model and its solution.

On the one hand, Author 2 is the owner of the root tweet of the conversation (node 0 in the tweet-based model of Figure 1), and a total of four other authors (4, 9, 10 and 12) attack him, but he in turn does not reply later to the rest of tweets of the conversation. So, Author 2 is not involved in any cycle in the ADisG graph. Because the weight of Author 2 is greater than the one of any of his attacking authors, Author 2 belongs to the solution of the graph. With respect to the four attackers of Author 2, two of them (4 and 12) are also in the solution, since Author 12 does not defeat Author 2 and his weight is greater than the one of any of his attacking authors. Moreover, Author 12 defeats Author 8 and this allows Authors 3, 4 and 9 to be in the solution, and in turn accepting Author 3 allows Author 10 to be rejected.

On the other hand, Author 8 is involved in a total of 8 cycles, considering only attacks answers among authors, and almost all authors involved in cycles with Author 8 are in the solution (0, 6, 13 , 9 and 12). Thus, Author 8 produces a lot of circular discussions, but the weight of Author 12 is high enough to make Author 8 lose the discussion. Observe that in the ideal semantics, authors with a same weight that form a cycle are not accepted if none of the authors in the cycle is attacked by other authors outside of the cycle and accepted in the solution. Hence, in this discussion with high controversy around Author 8 (with a high number of cycles), we end up accepting many of these authors' opinions. Finally, as Authors 1, 5 and 7 only attack Author 8, all of them are also in the solution, while Author 11 is rejected, since it is defeated by Author 12.

6 Conclusions and future work

In this paper we have introduced first ideas on a probabilistic author-centered approach to analyze the set of accepted authors' opinions in Twitter discussions. We model discussions with a graph, where nodes represent whole sets of tweets of a single author, and thus representing his opinion, and edges between nodes represent criticism relationships between authors. Then, using valued abstract argumentation and ideal semantics, we compute the set of winning authors in the discussion. By comparing the set of accepted opinions with the rejected ones, we can detect the degree of polarization between both sets.

As future work, we plan to extend the author-centered model to also consider support relationships between tweets and also to explore more credulous acceptability semantics.

References

[Alsinet *et al.*, 2017a] Teresa Alsinet, Josep Argelich, Ramón Béjar, Cèsar Fernández, Carles Mateu, and Jordi Planes. An argumentative approach for discovering relevant opinions in Twitter with probabilistic valued relationships. *Pattern Recognition Letters*, 2017. Submitted.

[Alsinet *et al.*, 2017b] Teresa Alsinet, Josep Argelich, Ramón Béjar, Cèsar Fernández, Carles Mateu, and Jordi Planes. Weighted argumentation for analysis of discussions in Twitter. *Int. J. Approx. Reasoning*, 85:21–35, 2017.

[Bench-Capon and Dunne, 2007] Trevor J. M. Bench-Capon and Paul E. Dunne. Argumentation in artificial intelligence. *Artif. Intell.*, 171(10-15):619–641, 2007.

[Bench-Capon, 2002] Trevor J. M. Bench-Capon. Value-based argumentation frameworks. In *Proceedings of 9th International Workshop on Non-Monotonic Reasoning, NMR 2002*, pages 443–454, 2002.

[Besnard and Hunter, 2001] Philippe Besnard and Anthony Hunter. A logic-based theory of deductive arguments. *Artif. Intell.*, 128(1-2):203–235, 2001.

[Chang and Lin, 2011] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM TIST*, 2(3):27:1–27:27, 2011.

[Dung *et al.*, 2007] Phan Minh Dung, Paolo Mancarella, and Francesca Toni. Computing ideal sceptical argumentation. *Artif. Intell.*, 171(10-15):642–674, 2007.

[Dung, 1995] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321 – 357, 1995.

[Egly *et al.*, 2008] Uwe Egly, Sarah Alice Gaggl, and Stefan Woltran. Aspartix: Implementing argumentation frameworks using answer-set programming. In *Proceedings of the 24th International Conference on Logic Programming, ICLP 2008*, pages 734–738, 2008.

[Faber and Woltran, 2009] Wolfgang Faber and Stefan Woltran. Manifold answer-set programs for meta-reasoning. In *Proceedings of Logic Programming and Nonmonotonic Reasoning, LPNMR 2009*, pages 115–128, 2009.

[Lin *et al.*, 2007] Hsuan-Tien Lin, Chih-Jen Lin, and Ruby C. Weng. A note on platt's probabilistic outputs for support vector machines. *Machine Learning*, 68(3):267–276, 2007.

[Platt, 1999] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 1999.

[Rahwan and Simari, 2009] Iyad Rahwan and Guillermo R. Simari. *Argumentation in Artificial Intelligence*. Springer Publishing Company, 1st edition, 2009.

Ontology-Mediated Queries for Probabilistic Databases *

Stefan Borgwardt and **İsmail İlkan Ceylan**
Faculty of Computer Science
Technische Universität Dresden, Germany
firstname.lastname@tu-dresden.de

Thomas Lukasiewicz
Department of Computer Science
University of Oxford, UK
thomas.lukasiewicz@cs.ox.ac.uk

Abstract

Probabilistic databases (PDBs) are usually incomplete, e.g., contain only the facts that have been extracted from the Web with high confidence. However, missing facts are often treated as being false, which leads to unintuitive results when querying PDBs. Open-world probabilistic databases (OpenPDBs) were proposed to address this issue by allowing probabilities of unknown facts to take any value from a fixed probability interval. We extend OpenPDBs by Datalog$^{\pm}$ ontologies, under which both upper and lower probabilities of queries become even more informative, enabling us to distinguish queries that were indistinguishable before. We show that the dichotomy between P and PP in (Open)PDBs can be lifted to the case of first-order rewritable positive programs (without negative constraints); and that the problem can become NP$^{\text{PP}}$-complete, once negative constraints are allowed.

1 Introduction

The effort for building *large-scale knowledge bases* from data in an automated manner has resulted in a number of systems including NELL [Mitchell *et al.*, 2015], DeepDive [Shin *et al.*, 2015] and Knowledge Vault [Dong *et al.*, 2014]. They combine methods from information extraction, natural language processing, relational learning, and databases to process large volumes of uncertain data. The state of the art to store and process such data is founded on *probabilistic databases* (PDBs) [Suciu *et al.*, 2011].

Each of the above systems encodes only a portion of the real world, and this description is necessarily *incomplete*. Thus, a meaningful querying semantics must provide a way to deal with missing information. Recently, an effort in this direction was made by introducing *open-world probabilistic databases* (OpenPDBs) [Ceylan *et al.*, 2016a], which generalize PDBs to be able to deal with incompleteness. More precisely, in OpenPDBs the probabilities of facts that are not in the database, called *open tuples*, are relaxed to a default probability interval, which is very different from the *closed-world assumption* of PDBs, which requires the probabilities of such facts to be zero. In the resulting framework of OpenPDBs, query probabilities are given in terms of *upper* and *lower* probability values, which is more in line with an incomplete view of the world.

While forming a natural and flexible basis for querying incomplete data sources, OpenPDBs are limited in the following sense: All open tuples can take on probability values from a single *fixed* interval $[0, \lambda]$, which results in the *same* upper and lower probabilities for many queries. Consider, for instance, the PDB containing the probabilistic tuples $\langle \texttt{Author}(a) : 0.8 \rangle$, $\langle \texttt{Pub}(a, b) : 0.6 \rangle$, $\langle \texttt{Pub}(c, d) : 0.9 \rangle$, $\langle \texttt{Novel}(d) : 1 \rangle$. In OpenPDBs, $\texttt{Author}(c)$ and $\texttt{Author}(d)$ evaluate to the *same lower and upper probabilities* (0 and λ, respectively), since both tuples are open. Intuition, however, tells us that c is more likely to be an author, as we already know (with high confidence) that c has published a novel. On the other hand, $\texttt{Author}(d)$ is unlikely to hold, since we know (almost surely) that d is a novel. Essentially, we lack the common-sense knowledge that (i) anyone who has published a novel is an author, and (ii) authors and novels are disjoint entities, which helps us to distinguish such queries. Observe that (i) is a positive axiom and would lead to higher probabilities, whereas (ii) is a negative (constraining) axiom and would entail lower probabilities for some queries.

This problem has been widely studied in the context of classical databases under the name of *ontology-based data access* (OBDA) [Poggi *et al.*, 2008], a popular paradigm that encodes the domain knowledge through an ontology, thus being able to deduce facts not explicitly specified in the database. Following this, we encode the domain knowledge using a Datalog$^{\pm}$ ontology [Calì *et al.*, 2012a], which helps to break down the symmetries between open tuples, letting us distinguish more queries.

We study the semantic and computational properties of OpenPDBs under Datalog$^{\pm}$ programs. The main distinction between a PDB and an OpenPDB is that the latter represents a set of probability distributions instead of a single one, and introduces the difficulty of choosing the distribution that will maximize (or minimize) the probability of a query. It is known that the data complexity of probabilistic UCQ evaluation in OpenPDBs exhibits the same dichotomy between P and PP as in PDBs for unions of conjunctive queries [Dalvi

*This is an abridged version of the paper [Borgwardt *et al.*, 2017]

and Suciu, 2012; Ceylan *et al.*, 2016a]. We lift this dichotomy to first-order rewritable (positive) Datalog$^{\pm}$ programs using standard techniques. We then show that, once negative constraints are allowed, reasoning can become NP$^{\text{PP}}$-hard. We conclude with complexity results beyond the data complexity for ontology-mediated query evaluation relative to (tuple-independent) PDBs and OpenPDBs. All proofs can be found in the extended version of this paper (see https://lat.inf.tu-dresden.de/research/papers.html).

2 Background and Motivation

We consider a relational vocabulary γ consisting of *finite* sets $\mathbf{R}$ of *predicates*, $\mathbf{C}$ of *constants*, and $\mathbf{V}$ of *variables*. A *γ-term* is a constant or a variable. A *γ-atom* is of the form $\mathtt{P}(s_1, \ldots, s_n)$, where $\mathtt{P}$ is an n-ary predicate, and $s_1, \ldots, s_n$ are γ-terms. A *γ-tuple* is a γ-atom without variables.

Queries and Databases.
A *conjunctive query (CQ)* over γ is an existentially quantified formula $\exists \mathbf{x}\, \phi$, where ϕ is a conjunction of γ-atoms, written as a comma-separated list. A *union of conjunctive queries (UCQ)* is a disjunction of CQs. A query is *Boolean* if it has no free variables. A database $\mathcal{D}$ over γ is a finite set of γ-tuples. The central problem studied for databases is *query evaluation*: Finding all *answers* to a query Q over a database $\mathcal{D}$, which are assignments of the free variables in Q to constants such that the resulting first-order formula is satisfied in $\mathcal{D}$ in the usual sense, i.e., there is a homomorphism from the atoms in Q to the tuples in $\mathcal{D}$. In the following, we consider only Boolean queries Q, and focus on the associated decision problem, i.e., deciding whether Q is satisfied in $\mathcal{D}$, denoted as usual by $\mathcal{D} \models \mathsf{Q}$.

Probabilistic Databases.
The most elementary probabilistic database model is based on the tuple-independence assumption. We adopt this model and refer to [Suciu *et al.*, 2011] for details on this model and alternatives. A probabilistic database induces a set of classical databases (called *worlds*), each of which is associated with a probability value.

Formally, a *probabilistic database (PDB)* $\mathcal{P}$ over γ is a finite set of *(probabilistic) tuples* of the form $\langle t : p \rangle$, where t is a γ-tuple and $p \in [0, 1]$, and, whenever $\langle t : p \rangle, \langle t : q \rangle \in \mathcal{P}$, then $p = q$. A PDB $\mathcal{P}$ assigns, to every γ-tuple t, the probability p, if $\langle t : p \rangle \in \mathcal{P}$, and the probability 0, otherwise.

Under the *tuple-independence* assumption, any such probability assignment P induces the following *unique joint probability distribution* over classical databases $\mathcal{D}$:

$$\mathrm{P}(\mathcal{D}) := \prod_{t \in \mathcal{D}} \mathrm{P}(t) \prod_{t \notin \mathcal{D}} (1 - \mathrm{P}(t)).$$

Accordingly, query evaluation is enriched to also consider the probabilistic information. More formally, the *probability of a Boolean query* Q w.r.t. P is $\mathrm{P}(\mathsf{Q}) := \sum_{\mathcal{D} \models \mathsf{Q}} \mathrm{P}(\mathcal{D})$. Here, we do not need to consider worlds with probability 0; e.g., if $\mathrm{P}(t) = 0$, then the worlds containing t do not affect $\mathrm{P}(\mathsf{Q})$.

Example 1. Consider the PDB $\mathcal{P}_{ex}$ from the introduction and the query $\mathsf{Q}_1 := \exists x_1, x_2\, \mathtt{Author}(x_1), \mathtt{Pub}(x_1, x_2)$. The probability of Q_1 on $\mathcal{P}_{ex}$ is obtained by summing the probabilities of the worlds that satisfy Q_1, i.e., all worlds containing the first two tuples, resulting in the probability 0.48. In contrast, the natural query

$$\mathsf{Q}_2 := \exists x_1, x_2\, \mathtt{Author}(x_1), \mathtt{Pub}(x_1, x_2), \mathtt{Novel}(x_2)$$

evaluates to 0 on $\mathcal{P}_{ex}$, since all worlds that satisfy this query have probability 0.

Open-World Probabilistic Databases.
An *open-world probabilistic database (OpenPDB)* over γ is a pair $\mathcal{G} = (\mathcal{P}, \lambda)$, where $\lambda \in [0, 1]$ and $\mathcal{P}$ is a PDB. A *λ-completion* of $\mathcal{G}$ is a PDB that is obtained by introducing, for each γ-tuple t that does not occur in $\mathcal{P}$ (called an *open tuple*), a probabilistic tuple $\langle t : p \rangle$ with $p \in [0, \lambda]$. For a fixed value $\alpha \in [0, \lambda]$, we define a special λ-completion, denoted $\mathcal{P}_\alpha$, in which the probabilities of all open tuples are equal to α. Note that $\mathcal{P}_0$ is equivalent to $\mathcal{P}$.

Example 2. Consider the OpenPDB $\mathcal{G}_{ex} := (\mathcal{P}_{ex}, 0.5)$. The set $\mathcal{P}_{ex} \cup \{\langle \mathtt{Novel}(b) : 0.2 \rangle\}$ is a λ-completion of $\mathcal{G}_{ex}$ (tuples with probability 0 are omitted).

An OpenPDB $\mathcal{G} = (\mathcal{P}, \lambda)$ defines the set $\mathrm{K}_\mathcal{G}$ of all probability distributions P induced by the λ-completions of $\mathcal{G}$. $\mathrm{K}_\mathcal{G}$ constitutes a so-called *credal set*, which means that it is closed, convex, and has a finite number of extremal points [Cozman, 2000]. The range of probabilities of a query under such a set can be expressed as a probability interval. Formally, the *probability interval* of a Boolean query Q w.r.t. $\mathcal{G}$ is $\mathrm{K}_\mathcal{G}(\mathsf{Q}) := [\underline{\mathrm{P}}_\mathcal{G}(\mathsf{Q}), \overline{\mathrm{P}}_\mathcal{G}(\mathsf{Q})]$, where

$$\underline{\mathrm{P}}_\mathcal{G}(\mathsf{Q}) := \min_{\mathrm{P} \in \mathrm{K}_\mathcal{G}} \mathrm{P}(\mathsf{Q}) \quad \text{and} \quad \overline{\mathrm{P}}_\mathcal{G}(\mathsf{Q}) := \max_{\mathrm{P} \in \mathrm{K}_\mathcal{G}} \mathrm{P}(\mathsf{Q}).$$

Example 3. Consider again the OpenPDB $\mathcal{G}_{ex}$. While the lower probability $\underline{\mathrm{P}}_\mathcal{G}(\mathsf{Q}_2)$ remains 0, the upper probability evaluates to $\overline{\mathrm{P}}_\mathcal{G}(\mathsf{Q}_2) > 0$ due to the λ-completion

$$\mathcal{P}_{0.5} = \mathcal{P}_{ex} \cup \{\langle \mathtt{Author}(b) : 0.5 \rangle, \langle \mathtt{Author}(c) : 0.5 \rangle, \ldots\},$$

which contains all open tuples with probability $\lambda = 0.5$.

This example shows that OpenPDBs improve our view of the domain compared to PDBs. However, we have already illustrated in the introduction that OpenPDBs can further benefit from an axiomatic encoding of the domain knowledge, since many queries involving open tuples will yield the same lower and upper probabilities, although according to common-sense knowledge, they should differ. This motivates our introduction of a logical theory, in the form of Datalog$^{\pm}$.

Datalog$^{\pm}$ Programs.
We now extend the vocabulary γ by a (potentially infinite) set $\mathbf{N}$ of *nulls*. An *instance* I over γ is a (possibly infinite) set of γ-tuples that may additionally contain nulls.

A *tuple-generating dependency (TGD)* σ is a first-order formula $\forall \mathbf{x}\, \varphi(\mathbf{x}) \rightarrow \exists \mathbf{y}\, \mathtt{P}(\mathbf{x}, \mathbf{y})$, where $\varphi(\mathbf{x})$ is a conjunction of γ-atoms, called the *body* of σ, and $\mathtt{P}(\mathbf{x}, \mathbf{y})$ is a γ-atom, called the *head* of σ. A *negative constraint (NC)* ν is a first-order formula $\forall \mathbf{x}\, \varphi(\mathbf{x}) \rightarrow \bot$, where $\varphi(\mathbf{x})$ is a conjunction of γ-atoms, called the *body* of ν, and $\bot$ is the truth constant *false*. A *(Datalog$^{\pm}$) program* Σ is a finite set of TGDs and

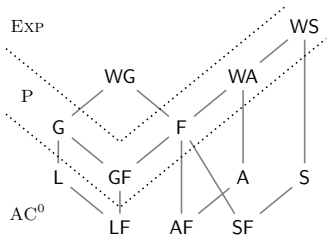


Figure 1: Inclusion relations and data complexity of UCQ entailment for Datalog$^{\pm}$ languages

NCs. An *ontology-mediated query (OMQ)* is a pair (Q, Σ), where Σ is a program, and Q is a Boolean query. For brevity, we omit the universal quantifiers in TGDs and NCs, and use commas for conjoining atoms. We only focus on single-atom-head TGDs; however, our results can be easily extended to TGDs with conjunctions of atoms in the head.

An instance I satisfies a TGD or NC σ, if $I \models \sigma$, where $\models$ denotes the standard first-order entailment relation. I satisfies a program Σ, written $I \models \Sigma$, if I satisfies each formula in Σ. The set of *models* of a program Σ relative to a database $\mathcal{D}$, denoted $mods(\mathcal{D}, \Sigma)$, is $\{I \mid I \supseteq \mathcal{D} \text{ and } I \models \Sigma\}$. $\mathcal{D}$ is *consistent* w.r.t. Σ, if $mods(\mathcal{D}, \Sigma)$ is non-empty. The OMQ (Q, Σ) is *entailed* by $\mathcal{D}$, denoted $\mathcal{D} \models (\mathsf{Q}, \Sigma)$, if $I \models \mathsf{Q}$ holds for all $I \in mods(\mathcal{D}, \Sigma)$.

In general, the entailment problem is undecidable [Beeri and Vardi, 1981]. For this reason, many different restrictions on the TGDs have been proposed. We consider here *guarded* (G), *linear* (L), *sticky* (S), *acyclic* (A), *weakly guarded* (WG), *weakly sticky* (WS), and *weakly acyclic* (WA) sets of TGDs [Calì *et al.*, 2013; 2012b]. Other important classes are given by *full* TGDs (F), *full and guarded* TGDs (GF), and similarly for LF, SF, and AF. Figure 1 illustrates the inclusion relations between these classes; for a more detailed description, see the extended version of this paper. We extend all these notions to programs Σ in the obvious way; for instance, Σ is guarded if all the TGDs in Σ are guarded. In the following, we use $\mathcal{L}$ to denote the set of Datalog$^{\pm}$ languages introduced above.

A key paradigm in OBDA is the *FO-rewritability* of queries; an OMQ (Q, Σ) is *FO-rewritable*, if there exists a Boolean UCQ Q_Σ such that, for all databases $\mathcal{D}$ that are consistent w.r.t. Σ, we have $\mathcal{D} \models (\mathsf{Q}, \Sigma)$ iff $\mathcal{D} \models \mathsf{Q}_\Sigma$. In this case, Q_Σ is called a *FO-rewriting* of (Q, Σ). A class of programs X is *FO-rewritable*, if it admits an FO-rewriting for any UCQ and program in X; these classes are characterized by a data complexity of AC^0 (see Figure 1).

3 Ontology-Mediated Queries for OpenPDBs

We now introduce the basics of OMQ evaluation relative to OpenPDBs. We assume that the input PDB $\mathcal{P}$ induces a consistent distribution w.r.t. the program. Formally, a probability distribution P is *consistent* w.r.t. Σ, if the database $\{t \mid \mathrm{P}(t) > 0\}$ is consistent w.r.t. Σ. Note that this assumption does not change the nature of the problem. The semantics of OMQs is again based on λ-completions. The difference appears in the deductive power provided by the Datalog$^{\pm}$ program, which is taken into consideration in the semantics.

Definition 4 (Semantics)**.** The *probability of an OMQ* (Q, Σ) *relative to a probability distribution* P is

$$\mathrm{P}(\mathsf{Q}, \Sigma) = \sum_{\mathcal{D} \models (\mathsf{Q}, \Sigma)} \mathrm{P}(\mathcal{D}),$$

where $\mathcal{D}$ ranges over all databases over γ. The *probability interval of* (Q, Σ) *relative to an OpenPDB* $\mathcal{G}$ is then given by $\mathrm{K}_\mathcal{G}(\mathsf{Q}, \Sigma) := [\underline{\mathrm{P}}_\mathcal{G}(\mathsf{Q}, \Sigma), \overline{\mathrm{P}}_\mathcal{G}(\mathsf{Q}, \Sigma)]$, where

$$\underline{\mathrm{P}}_\mathcal{G}(\mathsf{Q}, \Sigma) := \min_{\mathrm{P} \in \mathrm{K}_\mathcal{G}} \{\mathrm{P}(\mathsf{Q}, \Sigma) \mid \mathrm{P} \text{ is consistent w.r.t. } \Sigma\},$$

$$\overline{\mathrm{P}}_\mathcal{G}(\mathsf{Q}, \Sigma) := \max_{\mathrm{P} \in \mathrm{K}_\mathcal{G}} \{\mathrm{P}(\mathsf{Q}, \Sigma) \mid \mathrm{P} \text{ is consistent w.r.t. } \Sigma\}.$$

The special case of $\lambda = 0$ corresponds to having a single (closed-world) PDB $\mathcal{P}$. In this case, we simply speak of the *probability of* (Q, Σ) *relative to a PDB* $\mathcal{P}$.

This semantics defers the decision of whether a world satisfies a query to an entailment test and we maximize only over consistent λ-completions.

3.1 Semantic Considerations

Let us evaluate our semantics w.r.t. the goals identified in the motivation of this paper, and discuss our choices.

Distinguishing Queries.
We argued that OpenPDBs can benefit from an axiomatic encoding of the knowledge of the domain. Consider again our running example, which is now enriched with a program.

Example 5. Consider the OpenPDB $\mathcal{G}_{ex}$ given before and the program $\Sigma_{ex} := \{\texttt{Author}(x), \texttt{Novel}(x) \rightarrow \bot, \texttt{Pub}(x, y), \texttt{Novel}(y) \rightarrow \texttt{Author}(x)\}$ which states that authors and novels are disjoint entities, and that anyone who has published a novel is an author. The lower probability of $\texttt{Author}(d)$ remains 0, while the upper probability is now reduced to 0 with the help of the program Σ_{ex}. In contrast, the lower probability of $\texttt{Author}(c)$ increases to 0.9, while the upper probability increases to 0.95. These intervals are much more informative than the default interval $[0, 0.5]$.

Restricting to Consistent Distributions.
The most subtle aspect of choosing the *best* distribution is the question of how to deal with inconsistent worlds. Ignoring inconsistencies (and optimizing over *all* completions) leads to a drowning effect: since inconsistent worlds entail everything, this semantics would be biased towards choosing inconsistent λ-completions. This does not satisfy our goals, as even an unsatisfiable query could evaluate to a positive probability.

An alternative approach, which is standard for (closed-world) PDBs, and is quite intuitive at first glance, would be to choose the distribution which maximizes the conditional probability $\mathrm{P}((\mathsf{Q}, \Sigma) \mid (\mathcal{D}, \Sigma) \not\models \bot)$, i.e., the probability of the query on the set of all consistent worlds. A careful inspection shows that this semantics also favors inconsistent distributions over consistent ones. To illustrate this, consider our running example, and suppose that we want to compute the upper probability of Q_2 (mediated by Σ_{ex}). The semantics based on the conditional probability would favor the λ-completion $\mathcal{P}_{0.5}$, even though this PDB is highly inconsistent. This is mainly due to the normalization process internal

to the computation. As part of this normalization, the probability mass of inconsistent worlds is distributed to consistent worlds. As a consequence, it is often possible to increase the query probability by simply increasing the probability of inconsistent worlds. This is not a desired effect, since we are interested in finding the most suitable λ-completion from the open world, and not the one that increases the query probability by increasing the probability mass of inconsistent worlds.

To avoid such drowning effects, our proposal considers only consistent distributions. That is, we do not want to introduce inconsistencies when completing our knowledge over the domain by choosing a λ-completion. One drawback of our approach is the fact that inconsistencies are not tolerated even if the inconsistency degree is very small. However, it would be easy to introduce a threshold value, say 0.1, to tolerate the inconsistent completions where the probability of the inconsistent worlds does not exceed this threshold.

4 Data Complexity Results

We now formulate the task of probabilistic query evaluation as a decision problem.

Definition 6 (Decision Problems)**.** Let (Q, Σ) be an OMQ, $\mathcal{G}$ an OpenPDB and $p \in [0, 1]$. The problem of *upper (resp., lower) probabilistic query entailment* is to decide whether $\overline{\mathrm{P}}_{\mathcal{G}}(\mathsf{Q}, \Sigma) > p$ (resp., $\underline{\mathrm{P}}_{\mathcal{G}}(\mathsf{Q}, \Sigma) < p$) holds. *Probabilistic query entailment relative to PDBs* is a special case, where $\lambda = 0$.

Note that this definition is rather general, but in the scope of this paper, we are concerned with UCQs, and thus we use the term *probabilistic UCQ entailment* instead. Moreover, we are mainly concerned with the *data complexity*, which is calculated based on the size of the OpenPDB; i.e., the schema $\mathbf{R}$, the query Q, and the program Σ are assumed to be fixed [Vardi, 1982]. The relevant data complexity results for UCQ entailment in Datalog$^{\pm}$ are summarized in Figure 1.

Most of our complexity results are related to the complexity class PP [Gill, 1977], which comprises the languages recognized by a polynomial-time non-deterministic Turing machine that accepts an input if and only if *more than half* of the computation paths are accepting. Intuitively, PP is the decision counterpart of #P [Valiant, 1979]. It has been shown in [Dalvi and Suciu, 2012] that probabilistic UCQ entailment for PDBs exhibits a dichotomy between P and PP. Queries that admit a P algorithm are called *safe* and the remaining ones *unsafe*. This result has been lifted to OpenPDBs in [Ceylan *et al.*, 2016a]. We borrow this notion, and say that an OMQ (Q, Σ) is *safe*, if there exist polynomial-time algorithms for lower and upper probabilistic entailment of (Q, Σ) relative to any OpenPDB (resp., PDB).

4.1 Positive Programs

We first consider *positive* Datalog$^{\pm}$ programs, which do not contain NCs. Under this restriction, there are no inconsistent distributions, and Definition 4 simplifies. We later show that this distinction is important, since the complexity increases in the presence of NCs. This is surprising, as in the classical case NCs are usually not problematic.

Recall that OpenPDBs induce an infinite set of probability distributions that form a credal set, which has the following useful property [Cozman, 2000]: To determine the upper or lower probability of an event, it suffices to consider the *extremal* probability distributions, which are obtained by setting the probability values of all elementary events to one of the extreme points. In the context of OpenPDBs, this means that each of the open tuples may have probability λ or 0, but no intermediate choices need to be examined. For UCQs, this implies an even stronger result.

Lemma 7. *Let (Q, Σ) be an OMQ, where Q is a UCQ and Σ is a positive Datalog$^{\pm}$ program. Then, it holds that* $\mathrm{K}_{\mathcal{G}}(\mathsf{Q}, \Sigma) = [\mathrm{P}_{\mathcal{P}_0}(\mathsf{Q}, \Sigma), \mathrm{P}_{\mathcal{P}_\lambda}(\mathsf{Q}, \Sigma)]$.

Thus, it suffices to consider a single λ-completion (either $\mathcal{P}_0$ or $\mathcal{P}_\lambda$) and the particular distribution it induces. As a result, probabilistic UCQ entailment can be solved by standard methods; i.e., summing up the probabilities of all worlds that pass the entailment test. This naïve approach yields tight complexity bounds for the considered problems.

Theorem 8. *Probabilistic UCQ entailment is* PP*-complete for the languages in $\mathcal{L} \setminus \{\mathsf{WG}\}$; it is* EXP*-complete in* WG.

This result is of no surprise given the PP-hardness of inference in OpenPDBs. However, all our PP-hardness results are based on the result of [Dalvi and Suciu, 2012], and hence are valid only with respect to Turing reductions. All other complexity results in this paper also hold under standard many-one reductions. The striving question is now whether it is possible to lift the dichotomy result from OpenPDBs. For this purpose, we elaborate on query rewritability.

Lemma 9. *Let (Q, Σ) be an OMQ, P be a tuple-independent probability distribution over worlds such that $\mathrm{P}(\mathcal{D}) = 0$ whenever $\mathcal{D}$ is inconsistent w.r.t. Σ, and Q_Σ be an FO-rewriting of (Q, Σ). Then, we have $\mathrm{P}(\mathsf{Q}, \Sigma) = \mathrm{P}(\mathsf{Q}_\Sigma)$.*

Since all worlds are consistent under positive programs, Lemmas 7 and 9 imply that we can reduce probabilistic UCQ entailment under positive programs to the case of OpenPDBs via query rewriting.

Corollary 10. *Let (Q, Σ) be an OMQ, where Q is a UCQ, and Σ is a positive program, and Q_Σ be an FO-rewriting of (Q, Σ). Then, for any OpenPDB $\mathcal{G}$, it holds that $\overline{\mathrm{P}}_{\mathcal{G}}(\mathsf{Q}, \Sigma) = \overline{\mathrm{P}}_{\mathcal{G}}(\mathsf{Q}_\Sigma)$ and $\underline{\mathrm{P}}_{\mathcal{G}}(\mathsf{Q}, \Sigma) = \underline{\mathrm{P}}_{\mathcal{G}}(\mathsf{Q}_\Sigma)$.*

We now obtain a dichotomy from the results in [Dalvi and Suciu, 2012; Ceylan *et al.*, 2016a].

Theorem 11. *Let (Q, Σ) be an OMQ , where Q is a UCQ, and Σ is a positive program, and Q_Σ be a rewriting of (Q, Σ). Then, (Q, Σ) is safe iff Q_Σ is safe (over OpenPDBs). If (Q, Σ) is not safe, then it is* PP*-hard.*

In particular, either all rewritings of (Q, Σ) are safe, or none of them are. Hence, in FO-rewritable languages, we can take an *arbitrary* rewriting and check safety using the characterization of [Dalvi and Suciu, 2012]. Such a rewriting can be obtained by well-known algorithms, e.g., using backward chaining of TGDs [Gottlob *et al.*, 2011].

To conclude this section, we illustrate some effects that simple positive programs can have on the complexity of probabilistic query entailment.

Example 12. The query $\exists x, y\, \mathtt{C}(x) \wedge \mathtt{M}(x, y)$ is safe for OpenPDBs. It becomes unsafe under the TGD $\mathtt{R}(x, y), \mathtt{T}(y) \rightarrow \mathtt{M}(x, y)$, since then it rewrites to the query $(\exists x, y\, \mathtt{C}(x), \mathtt{M}(x, y)) \vee (\exists x, y\, \mathtt{C}(x), \mathtt{R}(x, y), \mathtt{T}(y))$. Conversely, the CQ $\exists x, y\, \mathtt{C}(x) \wedge \mathtt{L}(x, y) \wedge \mathtt{S}(y)$ is not safe for OpenPDBs, but becomes safe under $\mathtt{L}(x, y) \rightarrow \mathtt{S}(y)$, as it rewrites to $\exists x, y\, \mathtt{C}(x) \wedge \mathtt{L}(x, y)$. Note that these are very simple TGDs, which are full, acyclic, guarded, and sticky.

4.2 Programs with Negative Constraints

In the presence of NCs, it still suffices to consider the extremal λ-completions: once the correct completion is known, the probabilistic UCQ entailment problem can still be reduced to probabilistic inference (in FO-rewritable languages). The key difference is that we have to make sure that this completion is consistent. Simply choosing the completion $\mathcal{P}_\lambda$ that sets all open tuples to λ is not feasible, as this will very likely lead to inconsistencies. However, observe that the *lower* probability can still be obtained from the completion $\mathcal{P}_0$ (consistent by assumption), and hence the previous results still hold for lower probabilistic UCQ entailment.

A naïve way of solving the upper probabilistic UCQ entailment problem is to *guess* a λ-completion and then check whether it is consistent and compare the resulting probability to the threshold. This yields an $\mathrm{NP}^{\mathrm{PP}}$ upper bound for our decision problem. Our next result shows a matching lower bound for the class GF, and so for all considered Datalog$^{\pm}$ languages with data complexity above AC^0 (see Figure 1).

Theorem 13. *Upper probabilistic UCQ entailment is* $\mathrm{NP}^{\mathrm{PP}}$*-complete in full, guarded programs. It is* PP*-complete for all languages with polynomial data complexity once restricted to PDBs.*

On the one hand, this result is surprising, as NCs are not problematic for PDBs, even with normalization semantics; on the other hand, this is not so surprising, as non-monotonicity is also a source of additional hardness in OpenPDBs: query evaluation becomes $\mathrm{NP}^{\mathrm{PP}}$-complete in OpenPDBs if negated atoms are allowed in UCQs [Ceylan *et al.*, 2016a]. In contrast, our result applies to UCQs without negated atoms.

Before concluding this section, we illustrate the effects of NCs on some examples, which also show the difficulties in lifting the dichotomy of Theorem 11 to NCs.

Example 14. Consider the query $(\exists x, y\, \mathtt{C}(x) \wedge \mathtt{S}(y)) \vee (\exists x, y\, \mathtt{C}(x) \wedge \mathtt{L}(x, y))$, which is not safe for OpenPDBs, but becomes safe relative to the NC $\mathtt{S}(y), \mathtt{L}(x, y) \rightarrow \bot$. The reason is that the algorithm of [Dalvi and Suciu, 2012] that decides safety will produce the unsafe query $\exists x, y\, \mathtt{C}(x) \wedge \mathtt{S}(y) \wedge \mathtt{L}(x, y)$ through a sequence of reduction rules; however, this query automatically has probability 0 under the given NC, and hence becomes trivially safe.

5 Beyond Data Complexity

For the sake of completeness, we also provide results beyond the data complexity. We consider *fixed-program combined (fp-combined) complexity*, which is calculated in the size of the database and the query, while the program and schema remain fixed. Additionally, we remove the assumption that the program is fixed, and study *fixed-schema combined (fs-combined) complexity*. Our results are summarized in Table 1; all results except one are completeness results. The results are given relative to both PDBs and OpenPDBs to emphasize the computational differences.

Datalog$^{\pm}$ Languages	PDBs		OpenPDBs	
	fs-c.	*fp*-c.	*fs*-c.	*fp*-c.
L, LF, AF	$\mathrm{PP}^{\mathrm{NP}}$	$\mathrm{PP}^{\mathrm{NP}}$	$\mathrm{NP}^{\mathrm{PP}}$	$\mathrm{NP}^{\mathrm{PP}}$
G	EXP	$\mathrm{PP}^{\mathrm{NP}}$	EXP	$\mathrm{NP}^{\mathrm{PP}}$
WG	EXP	EXP	EXP	EXP
S, F, SF, GF	$\mathrm{PP}^{\mathrm{NP}}$	$\mathrm{PP}^{\mathrm{NP}}$	$\mathrm{NP}^{\mathrm{PP}}$	$\mathrm{NP}^{\mathrm{PP}}$
A	NEXP	$\mathrm{PP}^{\mathrm{NP}}$	in P^{NE}	$\mathrm{NP}^{\mathrm{PP}}$
WS, WA	2EXP	$\mathrm{PP}^{\mathrm{NP}}$	2EXP	$\mathrm{NP}^{\mathrm{PP}}$

Table 1: (fs/fp)-combined complexity of probabilistic UCQ entailment relative to OpenPDBs and PDBs.

Theorem 15. *Let* X *be a class of programs, and UCQ entailment in* X *be* $\mathbf{C}$*-complete in (fs/fp)-combined complexity. Then, probabilistic UCQ entailment in* X *is* $\mathbf{C}$*-hard and in* $\mathrm{PSPACE}^{\mathbf{C}}$ *in (fs/fp)-combined complexity. If* $\mathbf{C} = \mathrm{NEXP}$*, it is in* P^{NE}*, and* NEXP*-complete when restricted to PDBs.*

Hence, if $\mathbf{C} = \mathrm{EXP}$ or $\mathbf{C} = 2\mathrm{EXP}$, the complexity is not affected by adding OpenPDBs, since the complexity of UCQ entailment dominates the problem. We now consider the special case of NP-complete classes.

Theorem 16. *Let* X *be a class of programs. If UCQ entailment in* X *is* NP*-complete in (fs/fp)-combined complexity, then probabilistic UCQ entailment in* X *is complete for* $\mathrm{NP}^{\mathrm{PP}}$ *in (fs/fp)-combined complexity; it is complete for* $\mathrm{PP}^{\mathrm{NP}}$ *when restricted to a PDB.*

The hardness proof uses no TGDs and only one NC. This implies that the additional hardness in probabilistic UCQ entailment relative to OpenPDBs is caused solely by the interaction between NCs and the open-world semantics. This provides more evidence that OpenPDBs with NCs are more powerful than PDBs with NCs.

6 Related Work

Our work builds on the research on tuple-independent probabilistic databases [Suciu *et al.*, 2011], with an emphasis on the dichotomy result of Dalvi and Suciu (2012). The most closely related work is by Jung and Lutz (2012), where the authors lift the dichotomy result of PDBs to the light-weight description logics $\mathcal{EL}$ and *DL-Lite* over PDBs. We consider the more expressive languages of the Datalog$^{\pm}$ family and provide results both relative to PDBs and OpenPDBs. We show that the semantic differences between these formalisms lead to different results (even in the data complexity).

Most of the recent work on probabilistic query answering using ontologies is based on lightweight ontology languages. Some [Ceylan and Peñaloza, 2015; Gottlob *et al.*, 2013] result from a combination of ontologies with probabilistic graphical models such as Bayesian networks [Pearl, 1988]. Both the semantics and the assumptions used in these works are very different than ours. More closely related is the work by Ceylan

et al. (2016b), where the computational complexity of query answering in probabilistic Datalog$^{\pm}$ under the possible world semantics is investigated. Differently, the authors consider PDBs, and thus a unique probability distribution.

Possible world semantics is common in relational probabilistic models [Poole, 1997]. OpenPDBs extend this semantics to a (finite) open universe, and allow imprecise probabilities [Levi, 1980] for tuples in this universe. The latter can be seen as analogous to extending Bayesian networks [Pearl, 1988] to credal networks [Cozman, 2000]. Our framework enriches OpenPDBs further by mediating the query with an ontology, where the query evaluation problem over a database is replaced with a logical entailment problem.

7 Summary and Outlook

We introduced a refinement of the recently proposed OpenPDBs, using Datalog$^{\pm}$ ontologies to express additional background knowledge. We lifted the dichotomy from [Dalvi and Suciu, 2012; Ceylan *et al.*, 2016a] to all FO-rewritable languages for positive programs, showed that NCs can increase the complexity, and provided other complexity results.

In future work, we want to determine whether it is possible to obtain a dichotomy result for programs with NCs for FO-rewritable Datalog$^{\pm}$ languages. Similarly, the question whether the P-complete languages admit a dichotomy when restricting to positive programs is open. Note that we assume a finite set of constants, but allow infinitely many unknown individuals (nulls). Dealing with distributions over infinitely many objects as in BLOG [Milch *et al.*, 2005] is an important task, and a crucial part of future work.

Acknowledgments

This work is supported by the DFG within the Collaborative Research Center SFB 912 HAEC and the Graduiertenkolleg RoSI (GRK 1907), and by the EPSRC grants EP/J008346/1, EP/L012138/1, EP/M025268/1, and EP/N510129/1.

References

[Beeri and Vardi, 1981] Catriel Beeri and Moshe Y Vardi. The implication problem for data dependencies. In *Proc. of ICALP*, 1981.

[Borgwardt *et al.*, 2017] Stefan Borgwardt, İsmail İlkan Ceylan, and Thomas Lukasiewicz. Ontology-mediated queries for probabilistic databases. In *Proc. of AAAI*, 2017.

[Calì *et al.*, 2012a] Andrea Calì, Georg Gottlob, and Thomas Lukasiewicz. A general Datalog-based framework for tractable query answering over ontologies. *J. Web Sem.*, 14:57–83, 2012.

[Calì *et al.*, 2012b] Andrea Calì, Georg Gottlob, and Andreas Pieris. Towards more expressive ontology languages: The query answering problem. *AIJ*, 193:87–128, 2012.

[Calì *et al.*, 2013] Andrea Calì, Georg Gottlob, and Michael Kifer. Taming the infinite chase: Query answering under expressive relational constraints. *JAIR*, 48:115–174, 2013.

[Ceylan and Peñaloza, 2015] İsmail İlkan Ceylan and Rafael Peñaloza. Probabilistic query answering in the Bayesian description logic BEL. In *Proc. of SUM*, 2015.

[Ceylan *et al.*, 2016a] İsmail İlkan Ceylan, Adnan Darwiche, and Guy Van den Broeck. Open-world probabilistic databases. In *Proc. of KR*, 2016.

[Ceylan *et al.*, 2016b] İsmail İlkan Ceylan, Rafael Peñaloza, and Thomas Lukasiewicz. Complexity results for probabilistic Datalog+/–. In *Proc. of ECAI*. IOS Press, 2016.

[Cozman, 2000] Fabio Gagliardi Cozman. Credal networks. *AIJ*, 120(2):199–233, 2000.

[Dalvi and Suciu, 2012] Nilesh Dalvi and Dan Suciu. The dichotomy of probabilistic inference for unions of conjunctive queries. *J. ACM*, 59(6):1–87, 2012.

[Dong *et al.*, 2014] Xin Luna Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Patrick Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge Vault: A web-scale approach to probabilistic knowledge fusion. In *Proc. of SIGKDD*, pages 601–610. ACM, 2014.

[Gill, 1977] John T. Gill. Computatonal complexity of probabilistic Turing machines. *SIAM J. on Computing*, 6(4):675–695, 1977.

[Gottlob *et al.*, 2011] Georg Gottlob, Giorgio Orsi, and Andreas Pieris. Ontological queries: Rewriting and optimization. In *Proc. of ICDE*, pages 2–13. IEEE Press, 2011.

[Gottlob *et al.*, 2013] Georg Gottlob, Thomas Lukasiewicz, Maria Vanina Martinez, and Gerardo I. Simari. Query answering under probabilistic uncertainty in Datalog+/– ontologies. *Ann. Math. Artif. Intell.*, 69(1):37–72, 2013.

[Jung and Lutz, 2012] Jean Christoph Jung and Carsten Lutz. Ontology-based access to probabilistic data with OWL QL. In *Proc. of ISWC*, 2012.

[Levi, 1980] Isaac Levi. *The Enterprise of Knowledge*. MIT Press, 1980.

[Milch *et al.*, 2005] Brian Milch, Bhaskara Marthi, Stuart Russell, David Sontag, Daniel L. Ong, and Andrey Kolobov. BLOG: Probabilistic models with unknown objects. In *Proc. of IJCAI*, 2005.

[Mitchell *et al.*, 2015] T Mitchell, W Cohen, E Hruschka, P Talukdar, J Betteridge, A Carlson, B Dalvi, and M Gardner. Never-ending learning. In *Proc. of AAAI*, 2015.

[Pearl, 1988] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[Poggi *et al.*, 2008] Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Linking data to ontologies. *J. Data Sem.*, 10:133–173, 2008.

[Poole, 1997] David Poole. The independent choice logic for modelling multiple agents under uncertainty. *AIJ*, 94(1-2):7–56, 1997.

[Shin *et al.*, 2015] Jaeho Shin, Feiran Wang, Christopher De Sa, Ce Zhang, and Sen Wu. Incremental knowledge base construction using DeepDive. In *Proc. of VLDB*, 2015.

[Suciu *et al.*, 2011] Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch. *Probabilistic Databases*. Morgan & Claypool, 2011.

[Valiant, 1979] Leslie Gabriel Valiant. The complexity of computing the permanent. *TCS*, 8(2):189–201, 1979.

[Vardi, 1982] Moshe Y Vardi. The complexity of relational query languages. In *Proc. of STOC*, pages 137–146, 1982.

Optimal Feature Selection for Decision Robustness in Bayesian Networks*

YooJung Choi, Adnan Darwiche, and **Guy Van den Broeck**
Computer Science Department
University of California, Los Angeles
{yjchoi, darwiche, guyvdb}@cs.ucla.edu

Abstract

In many applications, one can define a large set of features to support the classification task at hand. At test time, however, these become prohibitively expensive to evaluate, and only a small subset of features is used, often selected for their information-theoretic value. For threshold-based, Naive Bayes classifiers, recent work has suggested selecting features that maximize the expected robustness of the classifier, that is, the expected probability it maintains its decision after seeing more features. We propose the first algorithm to compute this expected same-decision probability for general Bayesian network classifiers, based on compiling the network into a tractable circuit representation. Moreover, we develop a search algorithm for optimal feature selection that utilizes efficient incremental circuit modifications. Experiments on Naive Bayes, as well as more general networks, show the efficacy and distinct behavior of this decision-making approach.

1 Introduction

Classification and Bayesian decision making are complicated by the fact that features – the input to our decision making process – are expensive to evaluate in many application domains. In medical diagnosis, cost prohibits the doctor from running all possible tests, necessitating selective and active sensing [Yu *et al.*, 2009]. Similar issues arise in sensor networks [Krause and Guestrin, 2009], adaptive testing [Millán and Pérez-De-La-Cruz, 2002; Munie and Shoham, 2008], network diagnosis [Bellala *et al.*, 2013], and seismic risk monitoring [Malings and Pozzi, 2016].

Traditionally, such problems have been tackled by selecting features that optimize the decision-theoretic value of information [Heckerman *et al.*, 1993; Bilgic and Getoor, 2011]. These approaches seek to maximize the expected reward of observing the features. In one common instance, this means observing features that maximize the information gain, or equivalently, minimize the conditional entropy of the variable of interest (e.g., the medical diagnosis) [Zhang and Ji, 2010; Gao and Koller, 2011]. We refer to Krause and Guestrin [2009] for a more detailed discussion.

Another criterion called same-decision probability (SDP) came to the front more recently [Choi *et al.*, 2012; Chen *et al.*, 2014]. Given a current set of observed features, and the corresponding threshold-based decision, the SDP measures the *robustness* of this decision against further observations. It is the probability that our classification will be unchanged after all remaining features are revealed. SDP was successfully used to evaluate mammography-based diagnosis [Gimenez *et al.*, 2014] and adaptive testing [Chen *et al.*, 2015a].

Chen *et al.* [2015b] propose to use decision robustness for *feature selection*. In this context, we need to evaluate the expected robustness of a future decision based on the selected subset of features. It is compared to the hypothetical decision based on all features. The probability that these two classifiers agree is the *expected* SDP.

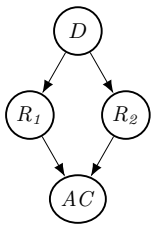
As our first contribution, we present the first algorithm for the expected SDP query on general Bayesian networks, which is PP^{PP}-complete [Choi *et al.*, 2012].[1] Previous expected SDP algorithms were restricted to Naive Bayes networks, where computing the SDP is (only) NP-hard [Chen *et al.*, 2013], and is amenable to heuristic search. Our expected SDP algorithm is instead based on the knowledge compilation approach of Oztok *et al.* [2016] for solving PP^{PP}-complete problems using sentential decision diagrams (SDDs) [Darwiche, 2011].

Our second contribution is an optimal feature selection algorithm. It searches for the subset of features that fits in our budget and is maximally robust according to expected SDP. The algorithm incrementally modifies an SDD representation of the Bayesian network during search, in order to efficiently re-evaluate the expected SDP of a large number of feature sets. It is the first optimal feature selection algorithm for general Bayesian network classifiers that optimizes robustness.

As a third contribution, we illustrate how decision robustness leads to different feature selection behavior on general Bayesian networks, compared to value of information. Moreover, the feature selection behavior depends strongly on the

*This work was originally published in the Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI), Melbourne, Australia, August 2017. [Choi *et al.*, 2017]

[1] Note that $\text{NP} \subseteq \text{PP} \subseteq \text{NP}^{\text{PP}} \subseteq \text{PP}^{\text{PP}} \subseteq \text{PSPACE} \subseteq \text{EXPTIME}$. MPE queries (NP), marginal probabilities (PP), and marginal MAP $\left(\text{NP}^{\text{PP}}\right)$ are all easier than (expected) SDP queries [Darwiche, 2009].



$\Pr(D=+)$
0.2

D	$\Pr(R_1=+\mid D)$
+	0.7
−	0.2

D	$\Pr(R_2=+\mid D)$
+	0.6
−	0.1

R_1	R_2	$\Pr(AC=+\mid R_1\,R_2)$
+	+	0.9
+	−	0.2
−	+	0.3
−	−	0

Figure 1: Bayesian network for review decisions

threshold used – a distinct property of the expected SDP criterion. We demonstrate our feature selection algorithm on both naive Bayes and general Bayesian network classifiers.

2 Same-Decision Feature Selection

We write uppercase letters for random variables and lowercase letters for their instantiation. Sets of variables are written in bold uppercase, and their joint instantiation in bold lowercase. Concatenations of sets denote their union (e.g. $\mathbf{XY}$).

2.1 Motivation and Intuition

Imagine a scenario where a program chair (PC) wants to determine the true quality D of a submitted paper. Two reviewers, R_1 and R_2, independently evaluate the paper, and their assessments are summarized by area chair AC. Figure 1 depicts this scenario as a Bayesian network. The PC wants to ascertain that accepted papers are high quality with at least 60% probability. This means that papers with two positive reviews get accepted regardless of the AC's evaluation, as

$$\Pr(D=+|R_1=+, R_2=+) = 0.84 \geq 0.60,$$

and all other papers get rejected. We refer to this ideal classifier as $C^\star$. In practice, however, the PC only has time to observe a single evaluation. The question then is: which feature should the decision be based on; R_1, R_2, or AC?

This scenario gives rise to three possible classifiers, C_{R_1}, C_{R_2}, and C_{AC}, depending on which feature is selected. For our threshold of 60%, these make the following decisions.

$$C_{R_1} = \left\{ \begin{array}{ll} \Pr(D=+|R_1=+) = 0.47 & \rightarrow \text{reject} \\ \Pr(D=+|R_1=-) = 0.09 & \rightarrow \text{reject} \end{array} \right\}$$

$$C_{R_2} = \left\{ \begin{array}{ll} \Pr(D=+|R_2=+) = 0.60 & \rightarrow \text{accept} \\ \Pr(D=+|R_2=-) = 0.10 & \rightarrow \text{reject} \end{array} \right\}$$

$$C_{AC} = \left\{ \begin{array}{ll} \Pr(D=+|AC=+) = 0.61 & \rightarrow \text{accept} \\ \Pr(D=+|AC=-) = 0.12 & \rightarrow \text{reject} \end{array} \right\}$$

It is customary to evaluate these classifiers with information-theoretic measures, such as information gain, or equivalently, the conditional entropy $\mathrm{H}(D|F)$ of D given feature F:

$$\mathrm{H}(D|R_1) = 0.30 \cdot \mathrm{h}(0.47) + 0.70 \cdot \mathrm{h}(0.09) = 0.59 \text{ bits}$$
$$\mathrm{H}(D|R_2) = 0.20 \cdot \mathrm{h}(0.60) + 0.80 \cdot \mathrm{h}(0.10) = 0.57 \text{ bits}$$
$$\mathrm{H}(D|AC) = 0.16 \cdot \mathrm{h}(0.61) + 0.84 \cdot \mathrm{h}(0.12) = 0.60 \text{ bits},$$

where $\mathrm{h}(p)$ is the entropy of a Bernoulli with probability p. This tells us that C_{R_2} is the best classifier, yielding most certainty about D, and the area chair's review C_{AC} is the worst, providing the least amount of information (i.e., high entropy).

Nevertheless, our PC may not want to maximize information content, and may simply strive to make the same *decisions* as the ones made by the ideal classifier $C^\star$. The most informative classifier C_{R_2} agrees with the optimal classifier on 90% of the papers, in all cases except when $R_1 = -$ and $R_2 = +$. Classifier C_{R_1} also makes the same decision in 90% of the cases. Our least informative classifier C_{AC}, however, outperforms both. When it rejects a paper because $AC = -$, the ideal classifier $C^\star$ agrees 99% of the time, namely in those cases where the AC did not overrule the reviewers. When C_{AC} accepts a paper because $AC = +$, the ideal classifier $C^\star$ agrees 56% of the time, in those cases where the reviewers both voted accept. These quantities are called *same-decision probabilities*. Overall, we expect C_{AC} to make the same decisions as $C^\star$ on 92% of the papers, which is its *expected same-decision probability*.

In conclusion, to optimize the decision, the PC should follow the AC evaluation, not an individual reviewer, even though their evaluations contain more information.

2.2 Problem Statement

Next, we formalize the robustness of a current decision against future observations as the same-decision probability.

Definition 1. *Let d and $\mathbf{e}$ be instantiations of the decision variable D and evidence variables $\mathbf{E}$. Let T be a threshold. Let $\mathbf{X}$ be a set of variables distinct from D and $\mathbf{E}$. The* same-decision probability *(SDP) in distribution* $\Pr$ *is*

$$\mathrm{SDP}_{d,T}(\mathbf{X} \mid \mathbf{e}) = \sum_{\mathbf{x}} [\Pr(d \mid \mathbf{xe}) =_T \Pr(d \mid \mathbf{e})] \cdot \Pr(\mathbf{x} \mid \mathbf{e}).$$

Here, the equality $=_T$ holds if both sides evaluate to a probability on the same side of threshold T, and $[\alpha]$ is 1 when α is true and 0 otherwise.

Expected SDP measures the redundancy of a feature set $\mathbf{X}$ if we were to first observe another feature set $\mathbf{Y}$.

Definition 2. *Let d and $\mathbf{e}$ be instantiations of the decision variable D and evidence variables $\mathbf{E}$. Let T be a threshold. Let $\mathbf{X}$ and $\mathbf{Y}$ be disjoint sets of variables. The* expected same-decision probability *(E-SDP) in distribution* $\Pr$ *is*

$$\begin{aligned}\mathrm{SDP}_{d,T}(\mathbf{X} \mid \mathbf{Y}, \mathbf{e}) &= \sum_{\mathbf{y}} \mathrm{SDP}_{d,T}(\mathbf{X} \mid \mathbf{ye}) \cdot \Pr(\mathbf{y} \mid \mathbf{e}) \\ &= \sum_{\mathbf{xy}} [\Pr(d \mid \mathbf{xye}) =_T \Pr(d \mid \mathbf{ye})] \cdot \Pr(\mathbf{xy} \mid \mathbf{e}).\end{aligned}$$

We will drop subscripts d and T when clear from context.

In *same-decision feature selection*, we are given a set of candidate features $\mathbf{F}$, a positive cost function $c(.)$, and budget B. The goal is to find features $\mathbf{Y} \subseteq \mathbf{F}$ maximizing $\mathrm{SDP}_{d,T}((\mathbf{F} \setminus \mathbf{Y}) \mid \mathbf{Y}, \mathbf{e})$, subject to a cost constraint $\sum_{Y \in \mathbf{Y}} c(Y) \leq B$. That is, we select those features that fit in budget and maximize the decision robustness, measured by E-SDP against the remaining features that were not selected.

3 A Tractable Circuit Representation

This section describes the logical foundation of our algorithms. Modern approaches to discrete probabilistic inference often reduce the problem to a logical one, encoding the distribution in weighted propositional logic [Chavira and Darwiche, 2005; 2008; Sang *et al.*, 2005; Dechter and Mateescu, 2007; Fierens *et al.*, 2015]. This technique naturally exploits structure in the distribution such as determinism and context-specific independence, and attains state-of-the-art performance [Darwiche *et al.*, 2008; Choi *et al.*, 2013]. In particular, we follow the knowledge compilation approach, where one compiles the logical description of the inference problem into a tractable (circuit) representation [Selman and Kautz, 1996; Darwiche and Marquis, 2002]. Knowledge compilation is particularly useful to solve some of the harder reasoning problems in AI, referred to as problems "beyond NP".[2] These include problems that are PP-hard, NP^{PP}-hard, or even PP^{PP}-hard, while still being of significant practical interest [Huang *et al.*, 2006; Oztok *et al.*, 2016]. SDP and E-SDP queries belong to this family.

We employ the same notation for propositional logic variables and random variables, as well as for their instantiations. A literal is a variable or its negation. Abusing notation, an instantiation can denote its corresponding set or conjunction of literals. A model $\mathbf{x}$ of a sentence α over variables $\mathbf{X}$ is a complete instantiation of $\mathbf{X}$ that satisfies α, denoted $\mathbf{x} \models \alpha$. Instantiations $\mathbf{x}$ and $\mathbf{y}$ are compatible, written $\mathbf{x} \sim \mathbf{y}$, iff $\mathbf{x} \wedge \mathbf{y}$ is satisfiable. Conditioning $\alpha|\mathbf{x}$ substitutes all $\mathbf{X}$ in α by their values in $\mathbf{x}$.

3.1 Encoding in Weighted Propositional Logic

Several encodings have been proposed to reduce Bayesian networks into (i) a propositional sentence α, and (ii) a function $w(.)$ that maps literals to weights. Variables $\mathbf{Z}$ in α come from two disjoint sets: indicators $\mathbf{I}$ and parameters $\mathbf{P}$, corresponding to values of network variables and network parameters, respectively. Literals ℓ constructed from $\mathbf{I}$ are assumed to have $w(\ell) = 1$. We refer to Chavira and Darwiche [2008] for the technical details and alternatives. Here, we instead show a simple encoding of the CPT for $\Pr(R_1|D)$ in Figure 1, which reduces to the sentence $\alpha_{R_1|D}$ consisting of

$$\begin{aligned} P_1 &\Leftrightarrow D \wedge R_1 & P_2 &\Leftrightarrow D \wedge \neg R_1 \\ P_3 &\Leftrightarrow \neg D \wedge R_1 & P_4 &\Leftrightarrow \neg D \wedge \neg R_1, \end{aligned} \quad (1)$$

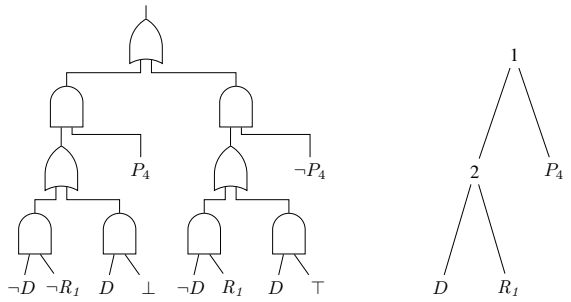
and its associated weight function sets $w(P_1)=0.7$, $w(P_2)=0.3$, $w(P_3)=0.2$, $w(P_4)=0.8$, and all other weights to 1. The full encoding α is the conjunction of sentences for each CPT.

The logical task called *weighted model counting* (WMC) sums the weight of all models of α that we are interested in:

$$\phi^w_\alpha(\mathbf{x}) = \sum_{\mathbf{z} \models \alpha \wedge \mathbf{x}} \prod_{\ell \in \mathbf{z}} w(\ell)\,, \text{ and}$$
$$\phi^w_\alpha(\mathbf{x}|\mathbf{e}) = \phi^w_\alpha(\mathbf{x} \wedge \mathbf{e}) / \phi^w_\alpha(\mathbf{e}).$$

We omit w when clear from context and write ϕ_α for $\phi_\alpha(\top)$. Given a weighted propositional logic encoding (α, w) of Pr, inference of conditional probabilities reduces to WMC. Indeed, $\Pr(\mathbf{x}|\mathbf{e}) = \phi^w_\alpha(\mathbf{x}|\mathbf{e})$, where $\mathbf{x}$ and $\mathbf{e}$ are variable instantiations, encoded using indicators from $\mathbf{I}$.

[2] http://beyondnp.org/



(a) Sentential decision diagram (SDD) (b) Variable tree (vtree)

Figure 2: Tractable SDD circuit representation for Sentence 1

3.2 Sentential Decision Diagrams

Sentential decision diagrams (SDDs) [Darwiche, 2011] are a knowledge compilation target language that allows for efficient WMC inference and incremental modifications (conjunction, disjunction, and conditioning of circuits) [Van den Broeck and Darwiche, 2015], which our algorithms will make use of. SDDs are also the most compact representation known to have these properties, exponentially smaller than ordered binary decision diagrams (OBDDs) [Bova, 2016].

Partitions SDDs are based on a new type of decomposition, called $(\mathbf{X}, \mathbf{Y})$-*partitions*. Consider a sentence α and suppose that we split its variables into two disjoint sets, $\mathbf{X}$ and $\mathbf{Y}$. It is always possible to decompose the sentence α as

$$\alpha = (p_1(\mathbf{X}) \wedge s_1(\mathbf{Y})) \vee \cdots \vee (p_n(\mathbf{X}) \wedge s_n(\mathbf{Y}))\,.$$

Sentences p_i are called *primes*, and are mutually exclusive, exhaustive, and consistent. Sentences s_i are called *subs*.

For example, consider Sentence 1 in our encoding of $\Pr(R_1|D)$. By splitting the variables into $\mathbf{X} = \{D, R_1\}$ and $\mathbf{Y} = \{P_4\}$, we obtain the $(\mathbf{X}, \mathbf{Y})$-partition

$$(\underbrace{(\neg D \wedge \neg R_1)}_{\text{prime}} \wedge \underbrace{P_4}_{\text{sub}}) \vee (\underbrace{(D \vee R_1)}_{\text{prime}} \wedge \underbrace{\neg P_4}_{\text{sub}}).$$

The primes are indeed mutually exclusive, exhaustive and non-false. This partition is represented in terms of logical gates by the top two layers of the SDD circuit in Figure 2a. In the graphical depiction of SDDs, primes and subs are either a constant, a literal or an input wire from another gate.

Vtrees SDDs represent a sequence of recursive $(\mathbf{X}, \mathbf{Y})$-partitions. Determining which $\mathbf{X}$ and $\mathbf{Y}$ to use in every partition in the SDD is governed by a variable tree (vtree): a full, binary tree, whose leaves are labeled with variables; see Figure 2b. The root v of the vtree partitions variables into those appearing in the left subtree ($\mathbf{X}$) and those in the right subtree ($\mathbf{Y}$). This implies an $(\mathbf{X}, \mathbf{Y})$-partition of sentence α, leading to the top two layers in Figure 2a. We say that the SDD's root node is *normalized* for vtree node v. The primes and subs of

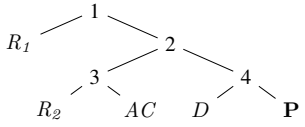


Figure 3: Constrained vtree where $\mathbf{X} = \{R_2, AC\}$, $\mathbf{Y} = \{R_1\}$, $\mathbf{Y}$-constr. node is 2 and $\mathbf{XY}$-constr. node is 4.

Algorithm 1 $\mathrm{SDP}_{d,T}(\mathbf{X} \mid \mathbf{Y}, \mathbf{e})$

Input: d : hypothesis; T : threshold
$\mathbf{X}$,$\mathbf{Y}$: disjoint sets of features; $\mathbf{e}$: evidence
S : $\mathbf{Y}$-constrained and $\mathbf{XY}$-constrained SDD
Output: computes $\mathrm{SDP}_{d,T}(\mathbf{X} \mid \mathbf{Y}, \mathbf{e})$

1: **for** each SDD node α in S (children before parents) **do**
2: **if** α is a terminal **then**
3: $vr_1(\alpha) \leftarrow \phi_\alpha$ **if** $\alpha \sim \mathbf{e}$; **else** 0
4: $vr_2(\alpha) \leftarrow \phi_\alpha$ **if** $\alpha \sim d\mathbf{e}$; **else** 0
5: **else**
6: $vr_1(\alpha) \leftarrow \sum_{(p_i,s_i)\in\alpha} vr_1(p_i) \times vr_1(s_i)$
7: $vr_2(\alpha) \leftarrow \sum_{(p_i,s_i)\in\alpha} vr_2(p_i) \times vr_2(s_i)$
8: **for** each SDD node α in S (children before parents) **do**
9: **if** α is a terminal **then**
10: $vr_3(\alpha) \leftarrow \phi_\alpha$
11: **else if** α is $\mathbf{XY}$-constrained **then**
12: $\beta \leftarrow \mathbf{Y}$-constrained ancestor of α
13: $vr_3(\alpha) \leftarrow vr_1(\alpha)$ **if** $\frac{vr_2(\alpha)}{vr_1(\alpha)} =_T \frac{vr_2(\beta)}{vr_1(\beta)}$; **else** 0
14: **else**
15: $vr_3(\alpha) \leftarrow \sum_{(p_i,s_i)\in\alpha} vr_3(p_i) \times vr_3(s_i)$
16: $\phi(\mathbf{e}) \leftarrow vr_1(S)$; $Q \leftarrow vr_3(S)$
17: **return** $Q/\phi(\mathbf{e})$

this partition are turned into SDDs, recursively, normalized for vtree nodes from the left and right subtrees. The process continues until we reach variables or constants. The vtree used to construct an SDD can have a dramatic impact on its size, sometimes leading to an exponential difference.

4 Computing the Expected SDP with SDDs

We now introduce our approach to compute E-SDP by compilation into SDDs. As described in Section 3, we can encode a Bayesian network into weighted propositional logic (α, w) and further compile it into an SDD. Then, computing the E-SDP over a set of features translates to computing the E-SDP over a set of variables of its circuit encoding.

Definition 3. *Let d and $\mathbf{e}$ be instantiations of the decision variable D and evidence variables $\mathbf{E}$. Let T be a threshold. Let $\mathbf{X}$ and $\mathbf{Y}$ be disjoint sets of variables. The* expected same-decision probability *on a WMC encoding (α, w) of* Pr *is*

$$\mathrm{SDP}_{d,T}(\mathbf{X} \mid \mathbf{Y}, \mathbf{e}) = \sum_{\mathbf{xy}} [\phi^w_\alpha(d \mid \mathbf{xye}) =_T \phi^w_\alpha(d \mid \mathbf{ye})] \cdot \phi^w_\alpha(\mathbf{xy} \mid \mathbf{e}).$$

Our approach to compute the E-SDP using SDDs is shown in Algorithm 1. It requires a special type of SDDs that are normalized for constrained vtrees [Oztok *et al.*, 2016].

Definition 4. *A vtree node v is $\mathbf{X}$-constrained, denoted $v_\mathbf{X}$, iff it appears on the right-most path of the vtree and $\mathbf{X}$ is exactly the set of variables outside v. A vtree is $\mathbf{X}$-constrained iff it has an $\mathbf{X}$-constrained node. An SDD is $\mathbf{X}$-constrained iff it is normalized for an $\mathbf{X}$-constrained vtree. An SDD node is $\mathbf{X}$-constrained iff it is normalized for $v_\mathbf{X}$.*

In order to compute $\mathrm{SDP}_{d,T}(\mathbf{X} \mid \mathbf{Y}, \mathbf{e})$, we require an SDD that is normalized for a vtree with a $\mathbf{Y}$-constrained node and an $\mathbf{XY}$-constrained node. Figure 3 depicts an example of such a vtree, which would allow us to compute $\mathrm{SDP}(R_2, AC \mid R_1)$. Note that the $\mathbf{Y}$-constrained node $v_\mathbf{Y}$ is always an ancestor of the $\mathbf{XY}$-constrained node $v_\mathbf{XY}$.

The algorithm performs two bottom-up passes over the SDD and keeps three value registers per node. In the first pass (Lines 1–7), it computes the WMC of each SDD node with respect to instantiations $\mathbf{e}$ and $d\mathbf{e}$, similar to the SDP algorithm of Oztok *et al.* [2016]. Intuitively, each SDD node normalized for $v_\mathbf{XY}$ or its ancestor represents some instantiation of variables outside that vtree node (which must be variables in $\mathbf{XY}$). For such node, vr_1, or vr_2, stores the probability of $\mathbf{e}$, or $d\mathbf{e}$, joint with the instantiation represented by that node.

Lemma 1. *Let α be an SDD node normalized for vtree v. Then, $vr_1(\alpha) = \phi_\alpha(\mathbf{e}_v)$ and $vr_2(\alpha) = \phi_\alpha(d_v\mathbf{e}_v)$, where $\mathbf{e}_v$ and d_v denote the subset of instantiation $\mathbf{e}$ and d, respectively, that pertains to the variables of vtree v.*

During the second bottom-up pass (Lines 8–15), the algorithm simply computes the WMC for an SDD node α that is neither an $\mathbf{XY}$-constrained node nor its ancestor. Next, if α is $\mathbf{XY}$-constrained and represents some instantiation $\mathbf{xy}$, then its $\mathbf{Y}$-constrained ancestor β must represent $\mathbf{y}$. The if-condition of Line 13 is then analogous to the indicator function in Definition 2. More precisely, Line 13 computes $[\phi_\alpha(d \mid \mathbf{e}) =_T \phi_\beta(d \mid \mathbf{e})]\, \phi_\alpha(\mathbf{e})$. We can use induction on the distance of v to $v_\mathbf{XY}$ to show that the following holds.

Lemma 2. *Let α be an SDD node normalized for vtree v, where v is $v_\mathbf{XY}$ or one of its ancestors, but v is not an ancestor of $v_\mathbf{Y}$. Let β be the $\mathbf{Y}$-constrained ancestor of α (that is, β is normalized for $v_\mathbf{Y}$). Let $\mathbf{W} = vars(v) \cap \mathbf{X}$. Then,*

$$vr_3(\alpha) = \sum_{\mathbf{w}} [\phi_\alpha(d \mid \mathbf{we}) =_T \phi_\beta(d \mid \mathbf{e})]\, \phi_\alpha(\mathbf{we}).$$

It follows from above lemma that at the $\mathbf{Y}$-constrained SDD node α, our algorithm computes the following quantity:

$$vr_3(\alpha) = \sum_{\mathbf{x}} [\phi_\alpha(d \mid \mathbf{xe}) =_T \phi_\alpha(d \mid \mathbf{e})]\, \phi_\alpha(\mathbf{xe}).$$

We can again use induction on the distance of v to $v_\mathbf{Y}$ to show the following.

Lemma 3. *Let α be an SDD node normalized for vtree v that is $v_\mathbf{Y}$ or one of its ancestors. Let $\mathbf{Z} = vars(v) \cap \mathbf{Y}$. Then,*

$$vr_3(\alpha) = \sum_{\mathbf{x},\mathbf{z}} [\phi_\alpha(d \mid \mathbf{xze}) =_T \phi_\alpha(d \mid \mathbf{ze})]\, \phi_\alpha(\mathbf{xze}).f$$

The complete proofs of above lemmas are found in the appendix. Line 16 now computes the quantity

$$Q = \sum_{\mathbf{x},\mathbf{y}} [\phi_S(d \mid \mathbf{xye}) =_T \phi_S(d \mid \mathbf{ye})] \cdot \phi_S(\mathbf{xye}).$$

Algorithm 2 FS-SDD($\mathbf{Q}, d, b$)

Input:
d : hypothesis; T : threshold; $\mathbf{e}$: evidence;
B : budget; $\mathbf{F}$: $\{F_1, \ldots, F_n\}$, set of features
c : cost function; S : $\mathbf{F}$-constrained SDD
Data:
$\mathbf{Q} \leftarrow \{\}$: features selected; $b \leftarrow B$: budget left
$k \leftarrow 0$: depth; $\mathbf{Y}$: best subset; p : best E-SDP
Output: Subset $\mathbf{Y} \subseteq \mathbf{F}$ with best E-SDP within budget B

1: **if** $b < 0$ or $k > n$ **then return**
2: **else if** $c(F_k) \leq b$ **then**
3: $\quad \mathbf{Z} \leftarrow \mathbf{Q} \cup \{F_k\}$
4: $\quad$ move variable in S to make it $\mathbf{Z}$-constrained
5: $\quad$ **if** $\mathrm{SDP}((\mathbf{F} \setminus \mathbf{Z}) \mid \mathbf{Z}, \mathbf{e}) > p$ **then**
6: $\quad\quad \mathbf{Y} \leftarrow \mathbf{Z}$
7: $\quad\quad p \leftarrow \mathrm{SDP}((\mathbf{F} \setminus \mathbf{Z}) \mid \mathbf{Z}, \mathbf{e})$
8: $\quad$ FS-SDD($\mathbf{Z}, k+1, b - c(F_k)$)
9: FS-SDD($\mathbf{Q}, k+1, b$)

Dividing Q by $vr_1(S) = \phi_S(\mathbf{e})$ yields the expected SDP.

Proposition 1. *Alg. 1 computes* $\mathrm{SDP}_{d,T}(\mathbf{X} \mid \mathbf{Y}, \mathbf{e})$.

Note that the algorithm performs a constant amount of work at each SDD node. Thus, we have the following.

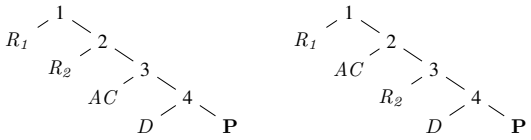
Proposition 2. *Alg. 1 runs in time linear in the size of SDD* S.

5 Feature Selection using SDDs

A naive approach to feature selection computes E-SDP for each possible subset of features that respects the budget and chooses the best one. However, this requires exponentially many compilations of SDDs since our E-SDP algorithm expects a different $\mathbf{Y}$-constrained SDD for each subset $\mathbf{Y}$.

We can improve upon this approach by introducing an operation to move a variable within a vtree and adjusting the SDD accordingly. Suppose we want to move a variable X of an SDD α, normalized for vtree v. If we condition α on X being true, the resulting SDD $\beta = \alpha|X$ no longer contains the variable X. Similarly, we can obtain an SDD for $\gamma = \alpha|\neg X$. Since X is not used for β and γ, we can move it to a new location to obtain a new vtree v', and β and γ will still be normalized for vtree v'. Finally, we join them to get a new SDD $\alpha' = (X \wedge \beta) \vee (\neg X \wedge \gamma)$, using an efficient APPLY function [Darwiche, 2011; Van den Broeck and Darwiche, 2015], which is normalized for vtree v' and still represents the same Boolean formula as α. Thus, once we compile an SDD, we can move variables around to make a $\mathbf{Y}$-constrained node for each subset $\mathbf{Y}$, instead of recompiling the SDD.

Our proposed algorithm FS-SDD, shown in Algorithm 2, generates candidate subsets by depth-first searching an inclusion-exclusion tree, as described in Korf [2009] and Chen *et al.* [2015b]. At each depth of the tree, we either include or exclude the variable pertaining to that depth in the subset. We backtrack if the cost so far exceeds the budget. Each time we include a variable in the subset, the expected SDP over that subset is computed using Algorithm 1, updating the optimal subset and its E-SDP as necessary.



(a) $\mathbf{Y} = \{R_1\}$ (b) $\mathbf{Y} = \{R_1, AC\}$

Figure 4: Moving AC after R_1. Vtree is right-linear outside of $\mathbf{F}$-constrained node where $\mathbf{F} = \{R_1, R_2, AC\}$.

Note that we are interested in computing the exact value of E-SDP only if it exceeds the highest E-SDP computed for some previous subset. Thus, we can early terminate the computation and continue our search to the next candidate, if an upper bound for current E-SDP falls below the highest value so far. We can do this by adding the following after line 13 of Algorithm 1: if $vr_3(\alpha) = 0$ then $ub \leftarrow ub - \frac{vr_1(\alpha)}{vr_1(S)}$; if $ub < best\_esdp$ then return. At the start of each E-SDP computation, ub is initialized to 1, and $best\_esdp$ is set to the highest E-SDP value until that point in search.

Analysis We illustrate that moving a variable each search iteration of FS-SDD is efficient, if the input SDD S is normalized for a vtree that is right-linear outside of its $\mathbf{F}$-constrained node.[3] For example, for $\mathbf{F} = \{R_1, R_2, AC\}$, Figure 4a satisfies this requirement, but Figure 3 does not. Each time we compute E-SDP over $\mathbf{Z} = \mathbf{Q} \cup \{F_k\}$, we already have a $\mathbf{Q}$-constrained node from the last recursive step, and variable F_k should appear inside the $\mathbf{Q}$-constrained vtree but outside the $\mathbf{F}$-constrained vtree. We can simply move F_k, using the operation defined previously, right after the nth variable in the vtree where n is the size of $\mathbf{Q}$. Figure 4 gives an example of such operation where $\mathbf{Q} = \{R_1\}$ and $F_k = AC$. In other words, the $\mathbf{F}$ variables appear in the same order in the vtree as in the path to the current search point in the inclusion-exclusion tree. This also maintains the right-linear structure outside the $\mathbf{F}$-constrained node, so we only need to move one variable in each search step.

6 Experimental Evaluation

We now empirically evaluate our SDD-based approach for decision-robust feature selection.

Naive Bayes We evaluated our system on Naive Bayes networks learned from datasets provided by the UCI repository [Bache and Lichman, 2013], BFC (`http://www.berkeleyfreeclinic.org/`), and CRESST (`http://www.cse.ucla.edu/`). We performed experiments using three variations of our SDD-based algorithm. We compare to MAXDR which, to our knowledge, is the only exact algorithm for feature selection based on E-SDP [Chen *et al.*, 2015b]. Since both algorithms find exact solutions, we compare their running times.

For each network, we find the optimal subset for E-SDP with the budget set to 1/3 the number of features. In all exper-

[3] A vtree is right-linear if each left child is a leaf.

Network	$\|\mathbf{F}\|$	MaxDR	FS-SDD1	FS-SDD2	FS-SDD3
bupa	6	0.021	0.184	0.035	0.044
pima	8	0.033	0.372	0.058	0.056
ident	9	0.105	1.548	0.127	0.128
anatomy	12	2.393	35.720	2.951	2.252
heart	13	18.649	122.822	9.907	6.321
voting	16	682.396	timeout	1110.96	810.042

Table 5: Running time (s) on Naive Bayes networks.

Network	source	# nodes	naive	FS-SDD
alarm	UAI	37	143.920	19.061
win95pts	UAI	76	23.581	14.732
tcc4e	HRL	98	48.508	2.384
emdec6g	HRL	168	28.072	3.688
diagnose	UAI	203	105.660	6.667

Table 6: Running time (s) on general Bayesian networks.

iments, the cost of each feature is 1, timeout is 1 hour, and a 2.6GHz Intel Xeon E5-2670 CPU with 4GB RAM was used. FS-SDD1 refers to the naive approach that compiles a constrained SDD for every candidate subset. FS-SDD2 directly compiles the network into an $\mathbf{F}$-constrained SDD which is then passed into our FS-SDD algorithm. Lastly, FS-SDD3 first compiles an unconstrained SDD and, after compilation, moves variables to make it $\mathbf{F}$-constrained. Table 5 shows that the naive approach performs worst. This highlights that repeated SDD compilation is very expensive. Moreover, FS-SDD3 outperforms FS-SDD2 as network size increases, illustrating that directly compiling a constrained SDD can be challenging for large networks and that moving variables is more effective. Moreover, even though MAXDR outperforms SDD-based approaches in most of the benchmarks, the running times of MAXDR and FS-SDD3 are comparable. Thus, utilizing efficient SDD operations enables our general-purpose algorithm to perform as well as MAXDR which is designed specifically for Naive Bayes networks.

General Bayesian Networks We now evaluate our algorithm on general Bayesian networks provided by HRL Laboratories and benchmarks from the UAI 2008 evaluation. For each network, a decision variable was chosen at random from root nodes, and 10 variables were randomly selected to be our set of features $\mathbf{F}$. We used FS-SDD to select an optimal subset of size at most 3. As this is the first algorithm for feature selection using E-SDP in general Bayesian networks, we compare our algorithm to a naive, brute-force approach which enumerates all possible instantiations of features to compute the E-SDP. Table 6 shows that our algorithm runs significantly faster than the naive approach, and that it performs as well on larger general Bayesian networks as it does on Naive Bayes networks. To calculate the marginals for the brute-force approach, we used jointree inference as implemented in SAMIAM.[4] Note that the runtime of the jointree algorithm is exponential in the treewidth of the network, whereas the SDD approach can sometimes run efficiently on high-treewidth networks. [Choi *et al.*, 2013] We also want to stress that the running time includes initial SDD compilation time, which in practice would be performed in an offline phase. Once we compile a constrained SDD, we can make observations and find the next optimal subset in an online fashion, thereby making decisions in a sequential manner.

[4] SAMIAM is available at `http://reasoning.cs.ucla.edu/samiam`.

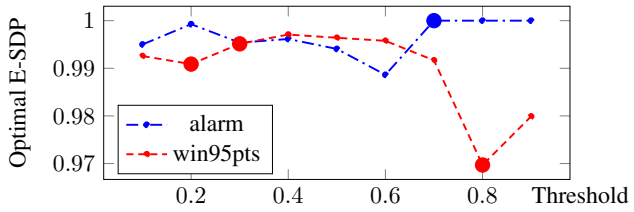


Figure 7: Expected SDP by threshold. Large markers indicate a change in the optimal selected subset.

Threshold-Based Decisions Lastly, we demonstrate that decision robustness is not only a function of the probability distribution and features but also of the threshold, unlike other measures of value of information. For networks `alarm` and `win95pts`, we used FS-SDD to select features from $\mathbf{F}$, which the set of all leaf nodes in each network. No evidence was asserted, the decision variable was chosen randomly from root nodes, and the budget was set to 1/3 the size of $\mathbf{F}$. Using the same decision variable, we repeatedly evaluated our algorithm with decision thresholds in $\{0.1, 0.2, \ldots, 0.9\}$. Figure 7 shows different E-SDP for different thresholds. In fact, FS-SDD selects different features as the threshold changes. For example, three leaf nodes are chosen as an optimal subset for `alarm` for $T \in [0.1, 0.6]$, whereas one leaf node can achieve expected SDP of 1.0 for $T \in [0.7, 1.0]$. Intuitively, the E-SDP measures redundancy of remaining features given a selected set of features, taking into account the decision procedure defined by the threshold. On the other hand, other measures such as information gain are unaware of the decision procedure and choose the same features regardless of changes in threshold.

7 Conclusion

We presented the first algorithm to compute the expected same-decision probability on general Bayesian network, as well as the first algorithm to use this measure of decision robustness for feature selection on general networks. This approach yields distinct results from other selection criteria. Our algorithms exploit the properties of sentential decision diagrams to evaluate and search feature sets efficiently.

Acknowledgments

This work is partially supported by NSF grants #IIS-1514253, #IIS-1657613, and #IIS-1633857, ONR grant #N00014-15-1-2339 and DARPA XAI grant #N66001-17-2-4032. The authors thank Arthur Choi for helpful discussions.

References

[Bache and Lichman, 2013] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[Bellala *et al.*, 2013] Gowtham Bellala, Jason Stanley, Suresh K. Bhavnani, and Clayton Scott. A rank-based approach to active diagnosis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(9):2078–2090, 2013.

[Bilgic and Getoor, 2011] Mustafa Bilgic and Lise Getoor. Value of information lattice: Exploiting probabilistic independence for effective feature subset acquisition. *Journal of Artificial Intelligence Research (JAIR)*, 41:69–95, 2011.

[Bova, 2016] Simone Bova. SDDs are exponentially more succinct than OBDDs. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 929–935. AAAI Press, 2016.

[Chavira and Darwiche, 2005] M. Chavira and A. Darwiche. Compiling bayesian networks with local structure. In *Proceedings of IJCAI*, volume 5, pages 1306–1312, 2005.

[Chavira and Darwiche, 2008] M. Chavira and A. Darwiche. On probabilistic inference by weighted model counting. *AIJ*, 172(6–7):772–799, 2008.

[Chen *et al.*, 2013] Suming Chen, Arthur Choi, and Adnan Darwiche. An exact algorithm for computing the Same-Decision Probability. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pages 2525–2531, 2013.

[Chen *et al.*, 2014] Suming Chen, Arthur Choi, and Adnan Darwiche. Algorithms and applications for the Same-Decision Probability. *Journal of Artificial Intelligence Research (JAIR)*, 49:601–633, 2014.

[Chen *et al.*, 2015a] Suming Chen, Arthur Choi, and Adnan Darwiche. Computer adaptive testing using the same-decision probability. In *Proceedings of the Twelfth UAI Conference on Bayesian Modeling Applications Workshop*, pages 34–43, 2015.

[Chen *et al.*, 2015b] Suming Chen, Arthur Choi, and Adnan Darwiche. Value of information based on Decision Robustness. In *Proceedings of the 29th Conference on Artificial Intelligence (AAAI)*, 2015.

[Choi *et al.*, 2012] Arthur Choi, Yexiang Xue, and Adnan Darwiche. Same-Decision Probability: A confidence measure for threshold-based decisions. *International Journal of Approximate Reasoning (IJAR)*, 2:1415–1428, 2012.

[Choi *et al.*, 2013] A. Choi, D. Kisa, and A. Darwiche. Compiling probabilistic graphical models using sentential decision diagrams. In *ECSQARU*, pages 121–132, 2013.

[Choi *et al.*, 2017] YooJung Choi, Adnan Darwiche, and Guy Van den Broeck. Optimal feature selection for decision robustness in Bayesian networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, August 2017.

[Darwiche and Marquis, 2002] A. Darwiche and P. Marquis. A knowledge compilation map. *JAIR*, 17:229–264, 2002.

[Darwiche *et al.*, 2008] Adnan Darwiche, Rina Dechter, Arthur Choi, Vibhav Gogate, and Lars Otten. Results from the probablistic inference evaluation of UAI-08. `http://graphmod.ics.uci.edu/uai08/Evaluation/Report`, 2008.

[Darwiche, 2009] Adnan Darwiche. *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press, 2009.

[Darwiche, 2011] A. Darwiche. SDD: A new canonical representation of propositional knowledge bases. In *Proceedings of IJCAI*, pages 819–826, 2011.

[Dechter and Mateescu, 2007] Rina Dechter and Robert Mateescu. And/or search spaces for graphical models. *Artificial intelligence*, 171(2-3):73–106, 2007.

[Fierens *et al.*, 2015] Daan Fierens, Guy Van den Broeck, Joris Renkens, Dimitar Shterionov, Bernd Gutmann, Ingo Thon, Gerda Janssens, and Luc De Raedt. Inference and learning in probabilistic logic programs using weighted Boolean formulas. *Theory and Practice of Logic Programming*, 15:358–401, 5 2015.

[Gao and Koller, 2011] T. Gao and D. Koller. Active classification based on value of classifier. In *Advances in Neural Information Processing Systems (NIPS 2011)*, 2011.

[Gimenez *et al.*, 2014] Francisco J Gimenez, Yirong Wu, Elizabeth S Burnside, and Daniel L Rubin. A novel method to assess incompleteness of mammography reports. In *AMIA Annual Symposium Proceedings*, volume 2014, page 1758. American Medical Informatics Association, 2014.

[Heckerman *et al.*, 1993] David Heckerman, Eric Horvitz, and Blackford Middleton. An approximate nonmyopic computation for value of information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(3):292–298, 1993.

[Huang *et al.*, 2006] Jinbo Huang, Mark Chavira, Adnan Darwiche, et al. Solving map exactly by searching on compiled arithmetic circuits. In *AAAI*, volume 6, pages 3–7, 2006.

[Korf, 2009] Richard E Korf. Multi-way number partitioning. In *IJCAI*, pages 538–543. Citeseer, 2009.

[Krause and Guestrin, 2009] Andreas Krause and Carlos Guestrin. Optimal value of information in graphical models. *Journal of Artificial Intelligence Research (JAIR)*, 35:557–591, 2009.

[Malings and Pozzi, 2016] Carl Malings and Matteo Pozzi. Conditional entropy and value of information metrics for optimal sensing in infrastructure systems. *Structural Safety*, 60:77–90, 2016.

[Millán and Pérez-De-La-Cruz, 2002] Eva Millán and José Luis Pérez-De-La-Cruz. A Bayesian diagnostic algorithm for student modeling and its evaluation. *User Modeling and User-Adapted Interaction*, 12(2-3):281–330, 2002.

[Munie and Shoham, 2008] Michael Munie and Yoav Shoham. Optimal testing of structured knowledge. In *Proceedings of the 23rd National Conference on Artificial intelligence*, pages 1069–1074, 2008.

[Oztok *et al.*, 2016] Umut Oztok, Arthur Choi, and Adnan Darwiche. Solving PP^{PP}-complete problems using knowledge compilation. In *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 94–103, 2016.

[Sang *et al.*, 2005] Tian Sang, Paul Beame, and Henry A Kautz. Performing bayesian inference by weighted model counting. In *AAAI*, volume 5, pages 475–481, 2005.

[Selman and Kautz, 1996] Bart Selman and Henry Kautz. Knowledge compilation and theory approximation. *Journal of the ACM (JACM)*, 43(2):193–224, 1996.

[Van den Broeck and Darwiche, 2015] G. Van den Broeck and A. Darwiche. On the role of canonicity in knowledge compilation. In *Proceedings of the 29th Conference on Artificial Intelligence (AAAI)*, 2015.

[Yu *et al.*, 2009] Shipeng Yu, Balaji Krishnapuram, Romer Rosales, and R Bharat Rao. Active sensing. In *International Conference on Artificial Intelligence and Statistics*, pages 639–646, 2009.

[Zhang and Ji, 2010] Yongmian Zhang and Qiang Ji. Efficient sensor selection for active information fusion. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 40(3):719–728, June 2010.

A Model of Multi-Agent Consensus for Compound Sentences

Michael Crosscombe and **Jonathan Lawry**

Department of Engineering Mathematics,
University of Bristol,
BS8 1UB, United Kingdom
m.crosscombe@bristol.ac.uk · j.lawry@bristol.ac.uk

Abstract

We present a model for multi-agent consensus in which agents attempt to reach an agreement about a shared set of compound sentences. Using Kleene's three-valued logic as a representation of borderline cases resulting from the inherently vague concepts of natural language, we investigate consensus formation at the sentence level where the vagueness of the underlying propositions can be exploited. A number of simulation experiments are conducted in which agents iteratively apply a *consensus operator* in random, pairwise interactions for a sufficient length of time or until convergence to a single truth assignment on the sentences has occurred. Preliminary results show that, given different sets of compound sentences about which agents attempt to form consensus, the population indeed converges on a single truth assignment at the sentence level. Furthermore, we highlight some interesting properties of the model regarding the kind of consensus achieved, particularly in relation to the underlying propositions.

1 Introduction

Negotiation and distributed decision-making are common challenges for multi-agent and robotics systems [Brambilla *et al.*, 2013; Parker and Zhang, 2009]. In this context, a population of agents must first reach an agreement about a shared set of relevant propositions by iteratively combining their beliefs until a consensus has been reached. Once the agents have converged on a share position or viewpoint, the population is then able to make a decision reflecting the beliefs of the population as a whole, rather than basing decisions on the beliefs of individuals or subgroups of individuals. However, inconsistencies between beliefs are common in a logical setting where propositions are restricted to Boolean truth values. Furthermore it is often unclear how best to resolve direct conflicts where one agent believes a sentence to be *absolutely true* and another absolutely *absolutely false*. Vague concepts, by definition, admit *borderline* cases which neither absolutely satisfy the concept nor its negation [Keefe and Smith, 1997]. Thus by allowing agents to adopt a more vague interpretation of the underlying propositions so as to soften directly conflicting beliefs, the level of inconsistency across the population decreases, allowing agents to reach an agreement while maintaining internal consistency. We therefore model borderline cases using Kleene's three-valued logic, where the third truth value $\frac{1}{2}$ is interpreted here as meaning *borderline true/false*. In recent studies [Balenzuela *et al.*, 2015; Crosscombe and Lawry, 2017; 2016; de la Lama, M. S. *et al.*, 2006; Perron *et al.*, 2009; Vazquez and Redner, 2004] it has been shown that an intermediate truth state does indeed improve convergence in opinion dynamics by facilitating compromise amongst agents.

While propositional beliefs may be sufficient to express simple statements about the state of the world, the ability to express beliefs about compound sentences is fundamental in allowing agents to reason about more complex relationships inherent in real-world scenarios. To this end, we propose a model of consensus formation for compound sentences where an agent's belief in the sentences is represented by a truth assignment over the sentences, and is a direct consequence of their underlying belief in the truth values of the propositions. That is, agents only communicate their beliefs at the *sentence level* as truth assignments over a given set of compound sentences in order to form consensus about which truth assignment they believe to be correct. Individual beliefs, however, remain hidden from other agents as these simply reflect an internal belief at the underlying *propositional level*. Depending on the set of compound sentences about which agents are trying to reach an agreement, the truth assignment may result from one or more underlying valuations at the propositional level. However, here an agent is adopting a belief at the propositional level in conjunction with a truth assignment at the sentence level, and the adoption of such a belief leaves no ambiguity about *which* valuation an agent believes to be true for a given a truth assignment. This is in contrast to the model proposed in [Cholvy, 2016] where an agent's belief is given only by a Boolean formula that corresponds to a set of possible truth states of the underlying propositions, such that agents are able to express a form of uncertainty.

The outline of this paper is as follows: Section 2 briefly surveys the existing literature on consensus modelling and opinion dynamics. Section 3 describes the proposed model where we introduce Kleene's three-valued logic in conjunction with a consensus operator for combining truth assign-

ments before describing our approach to modelling consensus of compound sentences in a multi-agent setting. We then describe the experiments in section 4 and present early results results for sets of compound sentences, before ending with some discussions and conclusions in section 5.

2 Background and Related Work

While a large body of work relating to opinion dynamics exists in the literature, a significant number of approaches are limited to a single underlying parameter, such as a real value or a propositional variable. For example, work on 'opinion pooling' dates back to [Degroot, 1974] and [Stone, 1961] where beliefs take the form of a probability distribution over an underlying parameter. The process of 'pooling' then refers to an aggregation of a set of beliefs, often population-wide and typically through a weighted linear combination of the probabilist distributions. However, alternative opinion pooling functions and their convergence properties have been studied by [Hegselmann and Krause, 2005] while the axiomatic characterisations of different operators are given by [Dietrich and List, 2017]. Other models for consensus in which opinions are based on a single real value include [Krause, 2000; Hegselmann and Krause, 2002] which introduced the idea of bounded confidence (an extension of which was proposed in [Deffuant *et al.*, 2002], referred to as relative agreement). The idea of bounded confidence is the imposition of limits on agent interactions whereby agents only combine their opinions with other individuals holding sufficiently similar opinions to their own. In the proposed model we implement our own version of bounded confidence where agents measure the relative inconsistency of their own opinions with those of others, only combining opinions with agents whose inconsistency measure is below a certain threshold. Furthermore, the models discussed here move from population-wide belief aggregations to pairwise interactions between individual agents.

Perhaps the most relevant area of research is that of opinion diffusion in the context of belief revision games [Schwind *et al.*, 2015], including the work of [Grandi *et al.*, 2015]. The general idea is that, at each iteration, a population of agents share their beliefs over a set of propositional variables before each agent aggregates the beliefs of others based on some predetermined merging operator. In [Cholvy, 2016] beliefs are represented as propositional formulas, allowing for a level of uncertainty to be conveyed as to which truth assignment on the propositions an agent believes to be correct. Agents combine their beliefs according to a preference ordering on the population of agents, with agents ordered from most to least influential. Unlike in the proposed model, all agents aggregate the beliefs of their influencers at each iteration, and depending on their personal preference ordering, agents may completely disregard their own beliefs when aggregating the beliefs of others. We prefer instead to view consensus as a pairwise operation between two agents with sufficiently similar beliefs such that their resulting consensus is reflective of the propositions on which they agree, and a compromise about those which they disagree as is captured by the consensus operator in table 2.

An important part of the consensus process described in this paper is the ability for two conflicting agents to be able to reach an agreement, which we achieve by incorporating an intermediate truth state representing 'borderline true/false'. In this context, there are a number of related studies of three-valued models in the opinion dynamics literature. One such model is [Balenzuela *et al.*, 2015] in which a partitioning threshold to the underlying real value to achieve a third truth state. Through iterative pairwise interactions, updating takes place on the real value where the magnitude and sign of the increments are relative to the agents' current truth states. An alternative updating operator was introduced in [Perron *et al.*, 2009] and was extended to incorporate feedback in [Crosscombe and Lawry, 2017] as well as evidence in [Crosscombe and Lawry, 2016]. This operator is applied directly to truth states of propositions to form a compromise between two opinions with strictly opposing truth values. We will adapt this operator to be applied at the sentence level, though the properties remain much the same.

3 A Combination Operator for Compound Sentences

We consider a simple language $\mathcal{L}$ based on Kleene's three-valued logic, with propositional variables $\mathcal{P} = \{p_1, ..., p_n\}$ and connectives $\neg$, $\vee$, $\wedge$ and $\rightarrow$. Let $S\mathcal{L}$ denote the sentences of $\mathcal{L}$ formed by recursive application of the logical connectives to the propositions of $\mathcal{L}$ in the usual manner. A Kleene valuation is then the allocation of truth values 0 (false), $\frac{1}{2}$ (borderline) and 1 (true) to the sentences of $\mathcal{L}$ as follows:

Definition 1. Kleene Valuations

A Kleene valuation v on $S\mathcal{L}$ is a function $v : S\mathcal{L} \rightarrow \{0, \frac{1}{2}, 1\}$ such that $\forall \theta, \varphi \in S\mathcal{L}$ the following hold:

- $v(\neg\theta) = 1 - v(\theta)$
- $v(\theta \wedge \varphi) = \min(v(\theta), v(\varphi))$
- $v(\theta \vee \varphi) = \max(v(\theta), v(\varphi))$

The truth table for Kleene valuations are shown in table 1. Note that given definition 1, a Kleene valuation v on $S\mathcal{L}$ is completely characterised by its values on $\mathcal{P}$.

It can also be convenient to represent a Kleene valuation v by its associated *orthopair* [Lawry and Dubois, 2012], (P, N), where $P = \{p_i \in \mathcal{P} : v(p_i) = 1\}$ and $N = \{p_i \in \mathcal{P} : v(p_i) = 0\}$. Notice that $P \cap N = \emptyset$ and that $(P \cup N)^c$ corresponds to the set of borderline propositional variables. Orthopairs are particularly helpful when discussing valuations as they relate to the underlying propositional variables. Consider the following example: Given two propositional variables p and q, an orthopair $(\{p\}, \emptyset)$ represents the valuation v such that $v(p) = 1$ and $v(q) = \frac{1}{2}$, or where p and q are true and borderline respectively, with q contained in neither P nor N. The use of orthopairs is discussed in detail in [Ciucci *et al.*, 2014] including difference of interpretation of the third truth value i.e. borderline or unknown.

We define a consensus operator $\odot$ in table 2 introduced by Perron et al. [2009] and Lawry and Dubois [2012] for combining a pair of truth values $t_1, t_2 \in \{0, \frac{1}{2}, 1\}$. The idea behind the consensus operator is to allow agents to reach an

$\neg$	
1	0
$\frac{1}{2}$	$\frac{1}{2}$
0	1

$\wedge$	1	$\frac{1}{2}$	0
1	1	$\frac{1}{2}$	0
$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0
0	0	0	0

$\vee$	1	$\frac{1}{2}$	0
1	1	1	1
$\frac{1}{2}$	1	$\frac{1}{2}$	$\frac{1}{2}$
0	1	$\frac{1}{2}$	0

Table 1: Kleene truth tables.

$\odot$	1	$\frac{1}{2}$	0
1	1	1	$\frac{1}{2}$
$\frac{1}{2}$	1	$\frac{1}{2}$	0
0	$\frac{1}{2}$	0	0

Table 2: Truth table for the consensus operator.

agreement by weakening conflicting truth values, while making vague beliefs more precise by preserving non-borderline truth values. For example, if one agent has the truth value 1 and the other 0, then the resulting consensus value $\frac{1}{2}$ is adopted by both agents. Meanwhile, a borderline truth value is replaced by a more precise value (i.e. 1 or 0) during consensus.

We now extend the consensus operator to vectors of truth values on the sentences of $\mathcal{L}$. Given a set of sentences $\Theta = \{\theta_1, ..., \theta_k\}$ for $\Theta \subseteq S\mathcal{L}$, then a truth-assignment on Θ is denoted by $\vec{t} \in \{0, \frac{1}{2}, 1\}^k$ where the i^{th} element of $\vec{t}$ is the truth assignment on θ_i for $i = 1, ..., k$. For a pair of truth assignments $\vec{t_1}, \vec{t_2}$ on Θ, we can then apply the consensus operator such that $\vec{t_1} \odot \vec{t_2}$ is given by

$$(t_{1,1}, ..., t_{1,k}) \odot (t_{2,1}, ..., t_{2,k}) = (t_{1,1} \odot t_{2,1}, ..., t_{1,k} \odot t_{2,k}).$$

Also, let $\mathbb{V}$ be the set of all Kleene valuations on $\mathcal{L}$. Then $\mathbb{V}_{\vec{t}} = \{v \in \mathbb{V} : v(\theta_i) = t_i, i = 1, ..., k\}$ is the set of Kleene valuations satisfying the truth assignment $\vec{t}$ on Θ.

Given that consensus then takes place between the truth assignments on the sentences Θ, for each of which it is possible that multiple Kleene valuations $v \in \mathbb{V}$ produce the same truth assignment, we decided that agents would then adopt the corresponding underlying Kleene valuation that was the most similar to their currently held belief. To this end, we now introduce a similarity measure.

Definition 2. A measure of similarity

A similarity measure between two Kleene valuations is a function $S : \mathbb{V}^2 \to [0, 1]$ such that $\forall v_1, v \in \mathbb{V}$:

$$S(v_1, v) = \frac{1}{n} \sum_{i=1}^{n} 1 - |v_1(p_i) - v(p_i)|$$

Then, for a pair of agents with initial valuations v_1, v_2 and a newly formed truth assignment $\vec{t_1} \odot \vec{t_2}$ adopted by the pair, v_1 and v_2 are replaced with new valuations $v_1', v_2' \in \mathbb{V}_{\vec{t_1} \odot \vec{t_2}}$ given by

$$v_1' = arg\ max\ \{S(v_1, v) : v \in \mathbb{V}_{\vec{t_1} \odot \vec{t_2}}\}$$

and

$$v_2' = arg\ max\ \{S(v_2, v) : v \in \mathbb{V}_{\vec{t_1} \odot \vec{t_2}}\}.$$

We now introduce a measure of inconsistency quantifying direct conflict between two truth assignments $\vec{t_1}$ and $\vec{t_2}$ as follows:

Definition 3. A measure of inconsistency

The degree of inconsistency between two truth assignments $\vec{t_1}, \vec{t_2}$ on the set of sentences $\Theta = \{\theta_1, ..., \theta_k\}$ is the proportion of truth values in direct conflict between the two truth assignments, expressed as a function $I(\vec{t_1}, \vec{t_2}) \to [0, 1]$, and is given by

$$I(\vec{t_1}, \vec{t_2}) = \frac{1}{k} \sum_{i=1}^{k} |t_{1,i} - t_{2,i}|.$$

We will employ this measure to study the resulting convergence properties of the model under varying restrictions on the level of inconsistency between pairs of agents. If, for a pair of agents, $I(\vec{t_1}, \vec{t_2}) > \gamma$ for an inconsistency threshold $\gamma \in [0, 1]$, then the consensus operator is not applied and both agents retain their current beliefs. If, however, $I(\vec{t_1}, \vec{t_2}) \leq \gamma$ then the consensus operator is applied and both agents adopt the resulting truth assignment $\vec{t_1} \odot \vec{t_2}$ as well as updating their underlying beliefs. Notice that an inconsistency threshold $\gamma = 1$ means that every pair of agents chosen from the population will combine, given that by definition the inconsistency measure cannot exceed 1. Conversely, an inconsistency threshold $\gamma = 0$ would therefore allow only the most consistent pairs of agents to form consensus for $I(\vec{t_1}, \vec{t_2}) \leq \gamma$. In other words, only when the truth assignments on the sentences are either exactly the same, or one of the truth assignments assigns a borderline truth value $\frac{1}{2}$ while the other assigns either 1 or 0.

4 Simulation Experiments

We now illustrate this approach by running a number of simulation experiments in which agents aim to reach consensus on a set of sentences Θ. We set a fixed limit of 1000[1] iterations for each experiment and average results over 100 independent runs. Initial beliefs of agents are distributed uniformly at random across the Kleene valuations on $\mathcal{L}$; we realise that this naturally generates a bias in favour of truth assignments with a greater number of associated valuations, however we felt that the most natural approach to belief initialisation was to initialise agents' internal beliefs, which consequently lead to truth assignments on the sentences.

While agents openly broadcast their truth assignments on Θ during the consensus process, their valuations on the propositions remain private. Given the consensus set of valuations $\mathbb{V}_{\vec{t_1} \odot \vec{t_2}}$, agents adopt the most similar valuation to their currently held beliefs, with both agents remaining unaware of which valuation $v' \in \mathbb{V}_{\vec{t_1} \odot \vec{t_2}}$ has been adopted by the other.

4.1 $p \to q$ and $p \to \neg q$

Figures 1 and 2 show the number of agents with truth assignments on the sentences $p \to q$ and $p \to \neg q$, and valuations on p and q respectively at steady state against incon-

[1] Preliminary experiments had shown this was more than sufficient to allow a population of 100 agents to reach consensus.

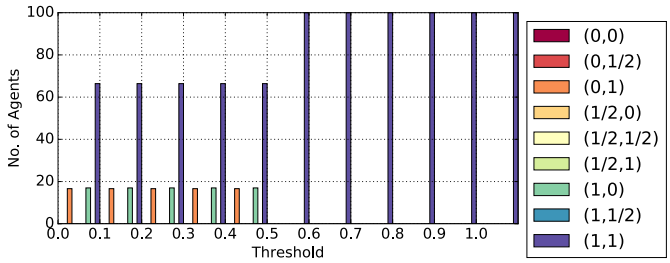


Figure 1: Number of agents with truth assignments on the sentences $p \to q$ and $p \to \neg q$.

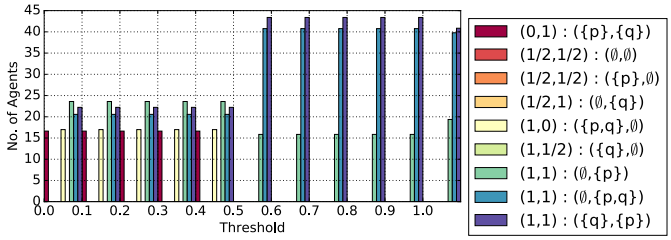


Figure 2: Number of agents with valuations on the propositions p, q for sentences $p \to q$ and $p \to \neg q$.

sistency threshold γ plotted as histograms. Results are averaged across the 100 independent runs, such that if all 100 agents have the truth assignment $(1, 1)$ averaged across all runs, then the population always converges on this truth assignment. From this, it is clear from figure 1 that the population does converge on a single truth assignment on Θ, with the population forming consensus on the truth assignment $(1, 1)$ for $\gamma \geq 0.5$. However, figure 2 provides a more detailed insight of the resulting consensus. In particular, we see that on average the population has not in fact converged towards a single underlying belief on the propositions, despite having converged on a single truth assignment for $\gamma \geq 0.5$. The largest majorities of the population are effectively split between the two most *crisp* (i.e. admitting no borderline cases.) valuations on the propositions, with a smaller minority of the population believing that q is in fact borderline. Even for an inconsistency threshold $\gamma = 1.0$ where all randomly selected pairs of agents combine form consensus, the population fails to converge to a single valuation.

Figure 3, showing the number of distinct truth assignments as a trajectory against iterations for $\gamma = 0.5$, provides a more detailed picture of the system's convergence. We see that the population converges on a single truth assignment after just 600 iterations. There are three valuations that result in the truth assignment $(1, 1)$ where every other possible truth assignment is associated with a much smaller set of Kleene valuations $\mathbb{V}_{\vec{t}}$. Therefore to identify if convergence at the sentence level corresponds to convergence at the propositional level, figure 4 shows the number of distinct valuations as a trajectory against iterations. From this, we can confirm that the population does not converge to a single valuation, despite convergence on a single truth assignment, and instead converges to the three valuations in $\mathbb{V}_{(1,1)}$, confirming that

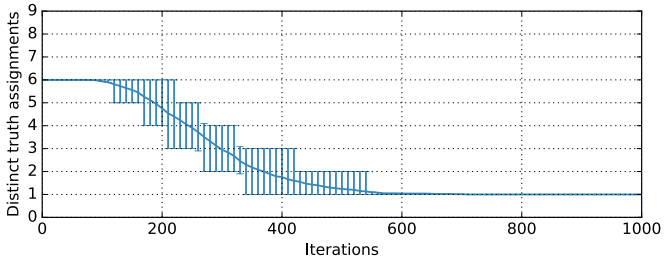


Figure 3: Number of distinct truth assignments on the sentences $p \to q$ and $p \to \neg q$ against iterations for an inconsistency threshold $\gamma = 0.5$.

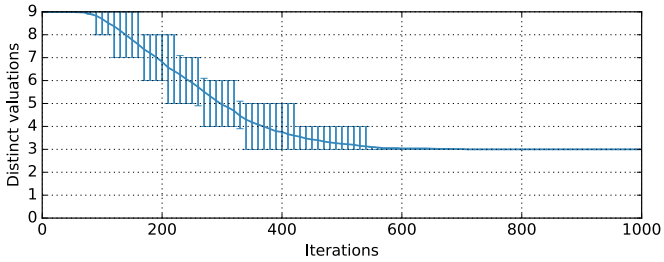


Figure 4: Number of distinct Kleene valuations on $\mathcal{L}$ against iterations for an inconsistency threshold $\gamma = 0.5$.

figure 2 shows a fairly accurate depiction of the convergence at steady state for a typical run.

Given that consensus occurs at random, the primary factor we believe to be driving convergence to these three valuations appears to be that the set $\mathbb{V}_{(1,1)}$ is the majority set in terms of the number of valuations $v \in \mathbb{V}_{\vec{t}}$. This effect is highlighted further for different sentences in Θ. As beliefs are initialised by selecting a valuation v uniformly at random from the set of all Kleene valuations $\mathbb{V}$ on $\mathcal{L}$, there exists a bias in favour of truth assignments with a greater number of corresponding valuations. Indeed this certainly appears to be the case from initial convergence results of the model. However, preliminary studies where agents' beliefs are initially selected uniformly at random across the truth assignments, with valuations then being assigned randomly from $\mathbb{V}_{\vec{t}}$, show that convergence to $(1, 1)$ still occurs.

4.2 $p \wedge q$ and $\neg p \wedge \neg q$

Similarly, we now present results for the sentences $p \wedge q$ and $p \wedge \neg q$. Figure 5 shows again the average number of agents with truth assignments on the sentences Θ against inconsistency threshold γ at steady state. In these experiments, the population converges to the truth assignment $(0, 0)$ on Θ, and for $\mathbb{V}_{(0,0)}$ there are two corresponding valuations which are both completely crisp; these are denoted by the orthopairs $(\{p\}, \{q\})$ and $(\{q\}, \{p\})$. Indeed in figure 7 we see that, just as in the previous experiment, convergence on a single truth assignment on the sentences in Θ occurs in just under 600 iterations, while the population again fails to converge on a single underlying valuation.

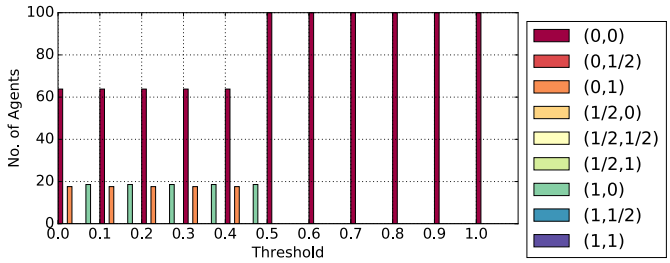


Figure 5: Number of agents with truth assignments on the sentences $p \wedge q$ and $\neg p \wedge \neg q$.

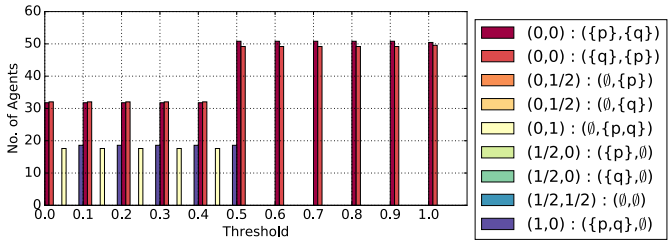


Figure 6: Number of agents with valuations on the propositions p, q for sentences $p \wedge q$ and $\neg p \wedge \neg q$.

Interestingly, the truth assignments $(0, \frac{1}{2})$ and $(\frac{1}{2}, 0)$ also each contain two corresponding valuations on p and q. For $\mathbb{V}_{(0,\frac{1}{2})}$, the two corresponding valuations are $(\emptyset, \{p\})$ and $(\emptyset, \{q\})$, and for $\mathbb{V}_{(\frac{1}{2},0)}$ the two valuations are $(\{p\}, \emptyset)$ and $(\{q\}, \emptyset)$. Note, however, that these valuations are more vague than either of those in $\mathbb{V}_{(0,0)}$. It would appear, therefore, that again the population converges on the most crisp truth assignment on Θ when there does not exist a truth assignment $\vec{t}$ with a corresponding majority set on the underlying valuations. This aligns with our expectations in relation to the properties of the consensus operator which favours stronger truth states over weaker borderline valuation.

4.3 $p \to q, p \to \neg q, \neg p \to q$ and $\neg p \to \neg q$

We now present results for a different set of sentences Θ in which we have increased the number of sentences from two to four. These are: $p \to q$, $p \to \neg q$, $\neg p \to q$ and $\neg p \to \neg q$. In contrast to previous experiments, the valid truth assignments on Θ each correspond to a single valuation. For ease of demonstration, we omit the truth assignments figure shown in previous subsections and focus instead on figure 8 in which the number of agents with valuations on p and q is given. This time, the resulting convergence is rather different from previous models we have studied. While we see that for all $\gamma \in [0, 1]$ the population does converge on a single truth assignment, this truth assignment varies from run to run rather than the population consistently converging on the same $\vec{t}$. This is confirmed in figure 9 which shows the number of distinct valuations as a trajectory for $\gamma = 0.5$. It is clear that, unlike in previous experiments, the population converges to a single valuation and therefore, by necessity, a single truth assignment on Θ. Naturally this convergence is a result of

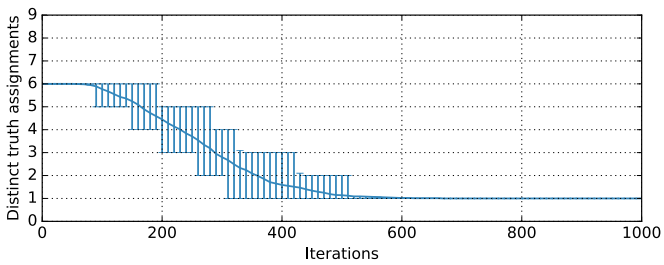


Figure 7: Number of distinct truth assignments on the sentences $p \wedge q$ and $\neg p \wedge \neg q$ against iterations for an inconsistency threshold $\gamma = 0.5$.

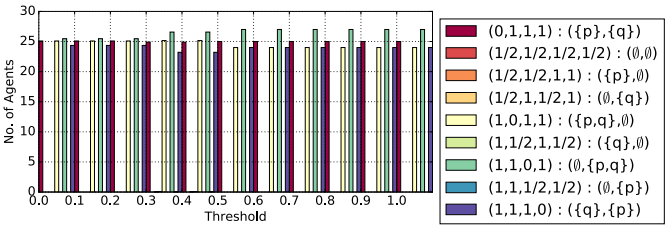


Figure 8: Number of agents with valuations on the propositions p, q for sentences $p \to q$, $p \to \neg q$, $\neg p \to q$ and $\neg p \to \neg q$.

the choice of sentences in Θ where each truth assignment is associated with a single underlying valuation only. However, we see in figure 8 that the population is no longer converging to the same truth assignment every time. Instead, it converges randomly to one of the four crisp truth assignments on Θ. We expect that, due to all four truth assignments being equally crisp, and that all four associated valuations are also crisp, then without any truth assignment possessing a larger number of associated valuations than any other, the population simply converges at random as initially expected prior to running any simulations.

5 Conclusions

Through simulation studies, we have highlighted several important properties of the proposed model and how it differs from previous models of consensus restricted to propositional variables. In particular, we see that convergence at the sentence level does not guarantee convergence at the propositional level unless the chosen set of sentences requires it. We also note that convergence appears to favour the majority set of valuations corresponding to a truth assignment that is crisp at the sentence level, and that when no majority exists amongst crisp truth assignments the population converges at random.

Further analysis of the proposed model is required, including studying the model with increased numbers of propositional variables and sentences to determine what kind of consensus, if any, is achieved. However, we believe that our model presents a promising basis for consensus of compound sentences and, given previous extensions of the consensus operator [Crosscombe and Lawry, 2016], we believe that it

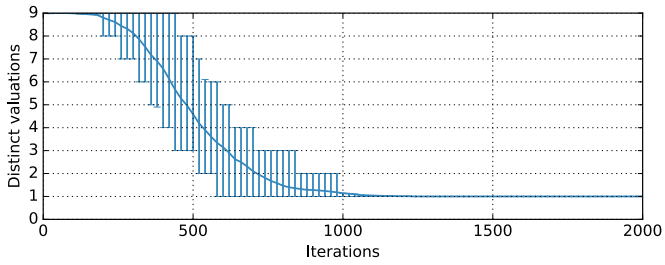


Figure 9: Number of distinct Kleene valuations on $\mathcal{L}$ against iterations for an inconsistency threshold $\gamma = 0.5$.

can be combined with a probabilistic model of uncertainty to allow agents to express both vagueness and uncertainty in their beliefs. In comparison to the related work of [Cholvy, 2016], we believe that allowing agents to combine at random and limiting interactions only by their relative inconsistency, we avoid a seemingly arbitrary preference ordering being assigned to the population for each agent. We also favour pairwise interactions in allowing beliefs to change more naturally over time, as we feel this is more intuitive as opposed to attempting to aggregate a large set of beliefs in one iteration.

References

[Balenzuela *et al.*, 2015] Pablo Balenzuela, Juan Pablo Pinasco, and Viktoriya Semeshenko. The undecided have the key: Interaction-driven opinion dynamics in a three state model. *PLOS ONE*, 10(10):1–21, 10 2015.

[Brambilla *et al.*, 2013] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, Mar 2013.

[Cholvy, 2016] Laurence Cholvy. *Diffusion of Opinion and Influence*, pages 112–125. Springer International Publishing, Cham, 2016.

[Ciucci *et al.*, 2014] Davide Ciucci, Didier Dubois, and Jonathan Lawry. Borderline vs. unknown: comparing three-valued representations of imperfect information. *International Journal of Approximate Reasoning*, 55(9):1866 – 1889, 2014. Weighted Logics for Artificial Intelligence.

[Crosscombe and Lawry, 2016] Michael Crosscombe and Jonathan Lawry. A model of multi-agent consensus for vague and uncertain beliefs. *Adaptive Behavior*, 24(4):249–260, 2016.

[Crosscombe and Lawry, 2017] Michael Crosscombe and Jonathan Lawry. *Exploiting Vagueness for Multi-agent Consensus*, pages 67–78. Springer Singapore, Singapore, 2017.

[de la Lama, M. S. *et al.*, 2006] de la Lama, M. S., Szendro, I. G., Iglesias, J. R., and Wio, H. S. Van kampen's expansion approach in an opinion formation model. *Eur. Phys. J. B*, 51(3):435–442, 2006.

[Deffuant *et al.*, 2002] Guillaume Deffuant, Frederic Amblard, and Grard Weisbuch. How can extremism prevail? a study based on the relative agreement interaction model. *Journal of Artificial Societies and Social Simulation*, 5(4), 2002.

[Degroot, 1974] Morris H. Degroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.

[Dietrich and List, 2017] Franz Dietrich and Christian List. Probabilistic opinion pooling generalized. part one: general agendas. *Social Choice and Welfare*, 48(4):747–786, Apr 2017.

[Grandi *et al.*, 2015] Umberto Grandi, Emiliano Lorini, and Laurent Perrussel. Propositional opinion diffusion. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, pages 989–997, Richland, SC, 2015. International Foundation for Autonomous Agents and Multiagent Systems.

[Hegselmann and Krause, 2002] Rainer Hegselmann and Ulrich Krause. Opinion dynamics and bounded confidence: models, analysis and simulation. *J. Artificial Societies and Social Simulation*, 5, 2002.

[Hegselmann and Krause, 2005] Rainer Hegselmann and Ulrich Krause. Opinion dynamics driven by various ways of averaging. *Comput. Econ.*, 25(4):381–405, June 2005.

[Keefe and Smith, 1997] Rosanna Keefe and Peter Smith. *Vagueness: A Reader*. MIT Press, 1997.

[Krause, 2000] Ulrich Krause. A Discrete Nonlinear and Non-Autonomous Model of Consensus Formation. In S. Elyadi, G. Ladas, J. Popenda, and J. Rakowski, editors, *Communications in Difference Equations*, pages 227–236. Gordon and Breach Pub., Amsterdam, 2000.

[Lawry and Dubois, 2012] Jonathan Lawry and Didier Dubois. A bipolar framework for combining beliefs about vague propositions. In *KR*, 2012.

[Parker and Zhang, 2009] C. A. C. Parker and H. Zhang. Cooperative decision-making in decentralized multiple-robot systems: The best-of-n problem. *IEEE/ASME Transactions on Mechatronics*, 14(2):240–251, April 2009.

[Perron *et al.*, 2009] E. Perron, D. Vasudevan, and M. Vojnovic. Using three states for binary consensus on complete graphs. In *IEEE INFOCOM 2009*, pages 2527–2535, April 2009.

[Schwind *et al.*, 2015] Nicolas Schwind, Katsumi Inoue, Gauvain Bourgne, Sébastien Konieczny, and Pierre Marquis. Belief revision games. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 1590–1596. AAAI Press, 2015.

[Stone, 1961] M. Stone. The opinion pool. *Ann. Math. Statist.*, 32(4):1339–1342, 12 1961.

[Vazquez and Redner, 2004] F Vazquez and S Redner. Ultimate fate of constrained voters. *Journal of Physics A: Mathematical and General*, 37(35):8479, 2004.

Schema Induction From Incomplete Semantic Data

Huan Gao[1,2], **Guilin Qi**[1,2], **Qiu Ji**[3]
[1]School of Computer Science and Engineering, Southeast University, Nanjing, China
[2]Key Laboratory of Computer Network and Information Integration (Southeast University)
Ministry of Education, Nanjing, China
[3]School of Modern Posts & Institute of Modern Posts
Nanjing University of Posts and Telecommunications, Nanjing, China

Abstract

Existing ontology learning or schema induction approaches often adopt the closed world assumption which is opposite to the assumption adopted by the semantic data. This may lead to a lot of noisy negative examples so that existing learning approaches fail to perform well on such incomplete data. In this paper, a novel framework is proposed to automatically obtain disjointness axioms and subclass axioms from incomplete semantic data. This framework first obtains probabilistic type assertions by exploiting a type inference algorithm. Then a mining approach based on association rule mining is proposed to learn high-quality schema information. To address the incompleteness problem of semantic data, the mining model introduces novel definitions to compute the support and confidence for pruning false axioms. Our experimental evaluation shows promising results over several real-life incomplete knowledge bases like DBpedia and LUBM by comparing with existing relevant approaches.

1 Introduction

Producing expressive schema information to enrich existing knowledge bases, which is the task of ontology learning or schema induction, could facilitate many semantic web tasks like ontology reasoning, logical contradiction detection, ontology mapping and object reconciliation Fleischhacker and Völker [2011]; Völker *et al.* [2015]; Nolle *et al.* [2016]. In this paper, we focus on mining two main kinds of axioms, namely disjointness axioms and subclass axioms, and our proposed ontology learning approach could be easily extended to learn other kinds of axioms.

Existing approaches to generate disjointness axioms and subclass axioms can be divided into two categories. One category is to obtain such axioms by labeling relations between two concepts manually like the work in Qi *et al.* [2015]. Obviously, it would be very tedious and even impossible when dealing with large-scale KBs. The other category is to learn terminology axioms automatically. The work in Bühmann *et al.* [2016] learns possible candidate concept expressions which are scored by positive examples and negative examples Fanizzi *et al.* [2008]. For the approaches proposed in Töpper *et al.* [2012] and Völker *et al.* [2015], the confidence is measured by the number of the positive examples and negative examples. This may lead to incorrect terminology axioms learned with a high confidence. The work in Zhu *et al.* [2013] and Völker and Niepert [2011] take missing information as negative examples and they have implemented the systems BelNet$^+$ and GoldMiner respectively.

These learning approaches often adopt Closed World Assumption (CWA) which is opposite to the assumption adopted by the semantic data (i.e., Open World Assumption, OWA)[1]. Due to the incompleteness of the semantic data, a lot of noisy negative examples may be generated so that existing learning approaches usually fail to perform very well. Therefore, learning high-quality disjointness axioms and subclass axioms from the semantic data still remains challenging.

In this paper, we propose a novel framework based on association rules to automatically generate disjointness and subclass axioms for addressing the problem of incomplete semantic data under OWA. This framework consists of two phases. The first phase generates negative examples by a type inference algorithm Paulheim and Bizer [2013]. The algorithm computes a set of type assertions by assigning a probability to indicate how much degree the instance belongs to the concept in such an assertion. In the second phase, novel definitions of support and confidence are given by considering the probabilities and specific association rule mining algorithms are proposed to generate disjointness axioms and subclass axioms. The experimental results are finally provided.

2 Preliminaries

2.1 Knowledge Base

In this paper, we focus on those knowledge bases expressed with RDF statements. Each RDF statement is a triple in the the form $< s, p, o >$, where s, p and o indicate a subject, a property (or predicate) and an object respectively. For convenience, $o(s)$ is used if p indicates "rdf:type" which describes that the individual s belongs to concept o, and $p(s, o)$ is used otherwise.

[1]CWA assumes the truth value of the specified and derivable statements is true, and false otherwise. The main difference between CWA and OWA is that the latter makes the assumption that the truth value of an underivable statement is unknown (i.e., neither true nor false).

Usually, a KB consists of an $ABox$ and a $TBox$. The $ABox$ includes a set of facts describing instances and their relations. The $TBox$ defines the schema information like the relations between classes, domains and ranges of properties. Our task of this paper is to mine $TBox$ information from an $ABox$. It is noted that, we use the terms knowledge base and ontology interchangeable.

2.2 Association Rule Mining

Association rule miningHan *et al.* [2012] is a rule-based method for discovering relations between variables in large databases by constructing transaction tables. Let $\mathcal{I} = \{I_1, I_2, ..., I_m\}$ be a set of items. A set of items that contain k items is called a k-itemset. Each transaction T is a nonempty set of items such that $T \subseteq \mathcal{I}$. For a set A satisfying $A \subseteq T$, the transaction T is said to contain A. An association rule is an implication of the form $A \rightarrow B$, where $A \subset \mathcal{I}$, $B \subset \mathcal{I}$, $A \cap B = \emptyset$, A and B are nonempty.

To decide whether an association rule $A \rightarrow B$ holds or not, the measures of support and confidence are widely adopted.

$$support(A \rightarrow B) = P(A \cup B),$$
$$confidence(A \rightarrow B) = P(B|A) = \frac{support(A \cup B)}{support(A)}.$$

Usually, the support of $A \rightarrow B$ is an absolute support which takes the occurrence frequency of the itemset $\{A, B\}$ as its value. Sometimes, it is also defined as a relative support which is the ratio of the absolute support and the number of all transactions in the transaction table. Confidence of $A \rightarrow B$ is the percentage of transactions containing A that also contain B according to the transaction table. This is taken to be the conditional probability.

2.3 Type Inference

The work in Paulheim and Bizer [2013] provides a statistic method to learn missing type assertions from a noisy KB. For each property in the KB, there is a characteristic distribution of types for the subjects and objects. For example, the property $dbpedia - owl : location$ is used by 247,601 triples in DBpedia and 87% objects of these triples belong to $dbpedia - owl : Place$. This work uses a statistic method to assign certain weight to each property, which reflects its capability of predicting a type. Then the missing types can be inferred by the weights. In this paper, the inferred type assertions are named as probabilistic type assertions (PTA).

3 The Overall Process of SIFS

In this section, we show the overall process to mine disjointness axioms and subclass axioms from incomplete semantic data. The process consisting of four steps: terminology acquisition, type inference, transaction table construction and axiom generation.

3.1 Terminology Acquisition

The process of SIFS starts with obtaining terminology information required by our type inference algorithm. The terminology information includes the type assertions and those relations between a pair of individuals. When a dataset is available for downloading, the required terminology information could be obtained directly by a parser. Otherwise, the information can be obtained through an endpoint if available. In such cases, one type of SPARQL query is used to obtain type assertions with the form $C(i)$, specifying that an instance i belongs to a concept C. The other type is to retrieve triples whose subject and object both are instances.

3.2 PTA Computation

Based on the obtained semantic data, the type inference algorithm given in Paulheim and Bizer [2013] can be applied to enrich an existing knowledge base K. The algorithm computes a set of probabilistic type assertions (PTA for short) in the form of $(C(i), p)$. Here, $p \in [0, 1]$ is the probability of type assertion $C(i)$ and indicates that the instance i has the probability p to belong to concept C.

Given a threshold t, instance i is taken as a positive example if $p \geq t$ and a negative example of concept C otherwise. That is, for concept C,

$$i\ is \begin{cases} postive\ example & p \geq t \\ negative\ example & otherwise. \end{cases}$$

If i is a positive example of concept C, the probability of $\neg C(i)$ is set to be 0. Otherwise, if i is a negative example of concept C, then the probability of $\neg C(i)$ is $1 - p$ and the probability of $C(i)$ is 0. That is,

$$P(C(i)) = \begin{cases} p & i\ is\ a\ positive\ example\ of\ C \\ 0 & otherwise, \end{cases}$$

$$P(\neg C(i)) = \begin{cases} 0 & i\ is\ a\ positive\ example\ of\ C \\ 1 - p & otherwise. \end{cases}$$

Here, $P(s)$ indicates the probability of assertion s.

3.3 Transaction Table Construction

With the generated PTAs, a transaction table can be constructed. In the table, each column (namely an item) presents a concept defined in the original KB or its negation and each row (namely a transaction) corresponds to one individual. Each cell in the table is a value between 0 and 1, which indicates the probability of a type assertion or a probabilistic type assertion. The association rules to be mined are of the form $C_1 \rightarrow \neg C_2$ or $C_1 \rightarrow C_2$. The first rule, used for generating disjointness axioms, means that the instances of C_1 cannot belong to C_2. The second rule, used for generating subclass axioms, indicates that the instances of C_1 must belong to C_2. Note that, the value of a cell is marked as "unknown" if a type assertion is neither stated in the original knowledge base nor inferred by the type inference algorithm.

This is quite different from a traditional transaction table used in association rule mining techniques whose value of a cell is either 0 or 1. Also, in the approach given in the work to learn disjointness axioms Fleischhacker and Völker [2011], the value of a cell is 1 when the corresponding type assertion explicitly stated in the original knowledge base and 0 otherwise.

3.4 Axiom Generation

For mining rules from a transaction table in SIFS, the traditional association rule mining algorithms cannot be applied directly due to the difference between the tables in SIFS and those in CWA. To deal with this problem, novel association rule mining algorithms need to be proposed. It is well known that, the support and confidence are two commonly used measures of rule strength in the field of data mining. Therefore, novel definitions of support and confidence (see Section 4 for details) need to be given before proposing new association rule mining algorithms.

Inspired by the work given in Völker and Niepert [2011], the rules in the form of $C_i \rightarrow C_j$ and $C_i \rightarrow \neg C_j$ are considered to mine subclass axioms and disjointness axioms respectively, where C_i and C_j are concepts. A rule like $C_i \rightarrow C_j$ can be translated to a subclass axiom $C_i \sqsubseteq C_j$ directly. A rule like $C_i \rightarrow \neg C_j$ corresponds to a disjointness axiom $C_i \sqsubseteq \neg C_j$.

4 Association Rule Mining in SIFS

To generate axioms from a transaction table in SIFS, support and confidence have to be redefined. In the following, we present the novel definitions of support and confidence respectively. Then specific algorithms are proposed to mine those rules that are used for generating disjointness axioms and subclass axioms.

4.1 Definition of Support

In SIFS, the support of 1-itemset needs to be calculated first, which indicates how many instances belong to a concept. The support of 1-itemset can be defined as below according to the absolute support of an itemset.

$$support(\{C_i\}) = \sum_{k=1}^{n}(a_{ki}) \tag{1}$$

In the equation, a_{ki} is the value in the kth row and ith column of a transaction table and n is the number of rows. This equation shows that the support of a rule is the sum of all values in the column of C_i in a transaction table.

In order to keep downward closure property of association rule mining, the support must decrease monotonically if more concepts are added into a rule. Thus, the support of a rule like $C_i \rightarrow C_j$ or $C_i \rightarrow \neg C_j$ can be defined as below.

$$support(C_i \rightarrow C_j) = P(\{C_i, C_j\}) = \sum_{k=1}^{n}(\min(a_{ki}, a_{kj})) \tag{2}$$

In the equation, n indicates the number of rows in a transaction table. Besides, a_{ki} indicates the value in the kth row and ith column, and a_{kj} is the value in the kth row and jth column. That is, for a rule like $C_i \rightarrow C_j$ or $C_i \rightarrow \neg C_j$, its support sums the minimal values of a_{ki} and a_{kj} in each row k.

Right now, the support of a rule in SIFS is still an absolute value. This means that a user who assigns a threshold for support has to know the absolute size of the original knowledge base. To deal with this problem, we give a definition of the proportional support. In a naive way, the absolute value of the support of a 1-itemset could be used (see Definition 2) as a denominator.

$$ps(C_i \rightarrow C_j) = \frac{support(C_i \rightarrow C_j)}{support(C_i)} \tag{3}$$

It is noted that, a concept may not have many (probabilistic) type assertions due to the incompleteness of the original knowledge base. For such concepts, we prefer to ignore them which is a commonly used strategy in association rule mining. This will speed up the mining process in SIFS since the searching space could be largely reduced.

4.2 Definition of Confidence

In the context of closed world assumption, the influence of missing data cannot be ignored when defining the confidence of a rule. It is because not considering mising data may make the confidence become larger than the actual one (see the definition of confidence given in Section 2). Thus, wrong rules could be obtained according to those confidences.

To deal with this problem, we define the following negative rules which can be used to compute how many (probabilistic) type assertions violate those rules to be mined:

For the rule $C_i \rightarrow C_j$, its negative rule is $C_i \rightarrow \neg C_j$.
For the rule $C_i \rightarrow \neg C_j$, its negative rule is $C_i \rightarrow C_j$.

It is reasonable to use these negative rules since it is unknown whether a missing instance violates the rules to be mined or not. The proportional support can imply the possibility of the negative rules.

When defining confidence of a rule, it is required to combine the support of the rule and that of its negative rule. The support of a negative rule serves as the constraint to reduce the confidence. For the rule to generate subclass axioms, Equation 4 and 5 are given. Since a disjointness axiom is symmetric, the support of the rule to generate disjointness axioms should be different with that of the rule to generate subclass axioms. We use the operator $\max$ to reflect symmetric. The support of the rule to be mined is given in Equation 6 and the support of the corresponding negative rule can be seen in Equation 7.

$$True(C_i \rightarrow C_j) = ps(C_i \rightarrow C_j) \tag{4}$$

$$False(C_i \rightarrow C_j) = ps(C_i \rightarrow \neg C_j) \tag{5}$$

$$True(C_i \rightarrow \neg C_j) = \max(ps(C_i \rightarrow \neg C_j), ps(C_j \rightarrow \neg C_i)) \tag{6}$$

$$False(C_i \rightarrow \neg C_j) = \max(ps(C_i \rightarrow C_j), ps(C_j \rightarrow C_i)) \tag{7}$$

It is noted that, $True(C_i \rightarrow \neg C_j)$ and $True(C_j \rightarrow \neg C_i)$ are same for the rule to generate disjointness axioms.

Based on the equations from 4 to 7, the confidence of a rule r can be defined as below:

$$Confidence(r) = \frac{True(r)}{True(r) + False(r)} \tag{8}$$

In the equation, r indicates a rule in the form of $C_i \rightarrow C_j$ or $C_i \rightarrow \neg C_j$. This equation normalizes the confidence not by the entire set of facts, but by the facts that support the rule r together with those that violate r according to what we have known.

Algorithm 1: Axioms Mining Algorithm

Input : A transaction table T and the thresholds t_{sup}, t_{ps} and t_{conf}
Output: Axioms to be generated

1 **begin**
2 axioms = candidate_rules = {};
3 **foreach** *column* i *in* T **do**
4 $support(\{C_i\}) = \Sigma_{k=1}^{n} a_{ki}$;
5 **foreach** *row* k *in* T **do**
6 **foreach** *concept pair* $(C_i, C_j)(i \neq j)$ *in* T **do**
7 **if** $support(\{C_i\}) > t_{sup}$ *and* $support(\{C_j\}) > t_1$ **then**
8 $support(\{C_i, C_j\}) += \min(a_{ki}, a_{kj})$;
9 **foreach** *concept pair* $(C_i, C_j)(i \neq j)$ *in* T **do**
10 $ps(C_i \rightarrow C_j) = support(\{C_i, C_j\})/support(\{C_i\})$;
11 **if** $ps(C_i \rightarrow C_j) > t_{ps}$ **then**
12 candidate_rules.add($C_i \rightarrow C_j, ps(C_i \rightarrow C_j)$;
13 **foreach** $r \in$ *candidate_rules* **do**
14 **if** $r == C_i \rightarrow C_j$ *&* C_i *and* C_j *are atomic concepts* **then**
15 $True = support(C_i, C_j)/support(C_i)$;
16 $False = support(C_i, \neg C_j)/support(C_i)$;
17 **if** $r == C_i \rightarrow \neg C_j$ *&* C_i *and* C_j *are atomic concepts* **then**
18 $True = \max(ps(C_i \rightarrow \neg C_j), ps(C_j \rightarrow \neg C_i))$;
19 $False = \max(ps(C_i \rightarrow C_j), ps(C_j \rightarrow C_i))$;
20 $Confidence = True/(True + False)$;
21 **if** $Confidence > t_{conf}$ **then**
22 axioms.add(transRule2Axiom(r));
23 **return** *axioms*;
24 **end**

4.3 Rule Mining Algorithms

Our goal is to mine disjointness axioms and subclass axioms from an incomplete KB under OWA. The mining algorithm (see Algorithm 1) takes a transaction table T and three thresholds, where t_{sup} and t_{ps} are used to filter out those rules with low standard and proportional supports respectively and t_{conf} is to prune candidate rules. The function transRule2Axiom translates an association rule to the corresponding axiom. This algorithm can be easily extended to generate other type of axioms.

Algorithm 1 first computes the absolute support for all 1-itemsets in the input transaction table (see lines 3-4). From line 5 to line 8, the algorithm calculates the absolute support for those 2-itemsets whose 1-itemsets own the support greater than the threshold t_{sup}. For each obtained 2-itemset, the proportional support of the corresponding rule is computed (see line 10). If the proportional support is greater than t_{ps}, the rule will be considered as a candidate one (see lines 11-12). The selection of candidate rules could reduce the search space and speed up the algorithm. After obtaining the candidate rules, the confidence of each rule needs to be calculated for further filtering (see lines 13-22). For each rule r, it is required to check which kind of axiom the rule is suitable for generating. Line 14 and line 17 check if r is in the form to generate subclass axioms or or disjointness axioms respectively. If true, the corresponding probabilities of r to be true and false are computed (see lines 15-16 and lines 18-19). With the obtained probabilities, the confidence value can be obtained (see line 20). If the confidence value is greater than the threshold t_{conf}, rule r needs to be translated to the corresponding axiom which is taken as a new generated axiom (see lines 21-22).

Gold Standards	Number of Concepts		Number of Axioms		
	All	No instances	Subc.	Disj.	Equi.
DBpedia	256	16	257	59,914	0
NTN	19	16	52	6	0
University	43	16	36	17	0
Family	49	4	6	17	14

Table 1: The statistics of the gold standards

5 Experimental Evaluation

Our experiments are performed on a server with Intel Xeon E5-2670 CPU and 128 GB of RAM using 64 bit operating system Ubuntu 14.04. The maximal heap space is set to be 60 GB.

The KBs used in the experiments include DBpedia[2], NTN[3], LUBM[4] and Family[5]. DBpedia is one of the central knowledge bases in linked open data and contains a huge number of facts and relatively small schema information. NTN is a part of Semantic Bible knowledge base. University, a university benchmark, is a customizable and repeatable synthetic data. Family describes the concepts like Brother and Sister and their relationships in a family domain.

For these KBs, gold standards are manually constructed. Namely, a set of disjointness axioms are added to each ontology by assuming the siblings are disjoint. The statistics of the gold standards can be seen in Table 1. This table presents the number of all concepts and those that have no type assertions explicitly declared in an original KB. It also shows the number of subclass, disjointness and equivalent axioms. From the table we can observe that NTN is the most sparse KB since 84% concepts have no instances. The performance of SIFS could be better reflected over such a highly incomplete KB.

To measure the performance of the systems to generate schema information, the traditional precision and recall are used. That is, precision is the fraction of generated axioms that are correct according to the corresponding gold standard and recall is the fraction of correct axioms that are generated.

In our experiments, we compare our system SIFS-P using proportional support, which is the implementation of Algorithm 1, with two relevant systems GoldMiner and BelNet+. The precision and recall of the systems to generate subclass and disjointness axioms can be seen in Figure 1 and Figure 2

[2] http://wiki.dbpedia.org/Downloads351
[3] http://www.semanticbible.com/
[4] http://swat.cse.lehigh.edu/projects/lubm/
[5] https://github.com/fresheye/belnet/blob/master/ontology/family_background.owl

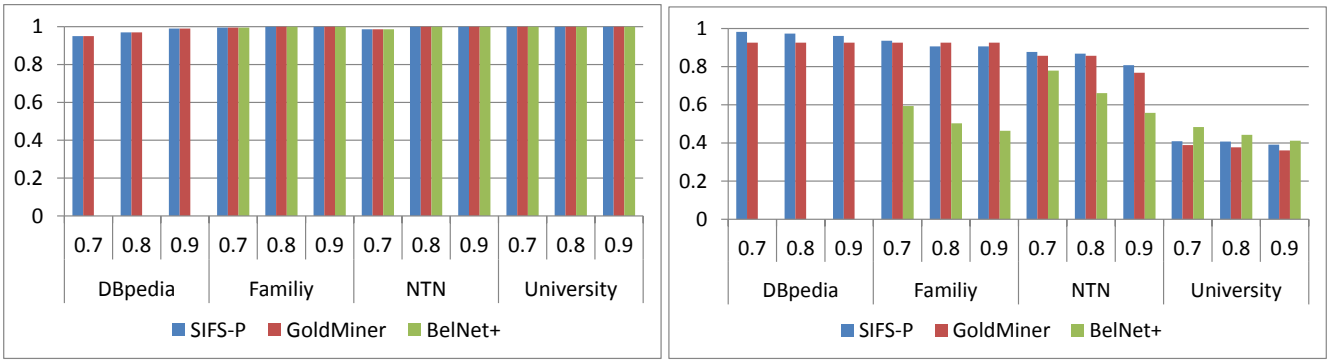


Figure 1: Precision (left) and recall (right) of systems to generate subclass axioms

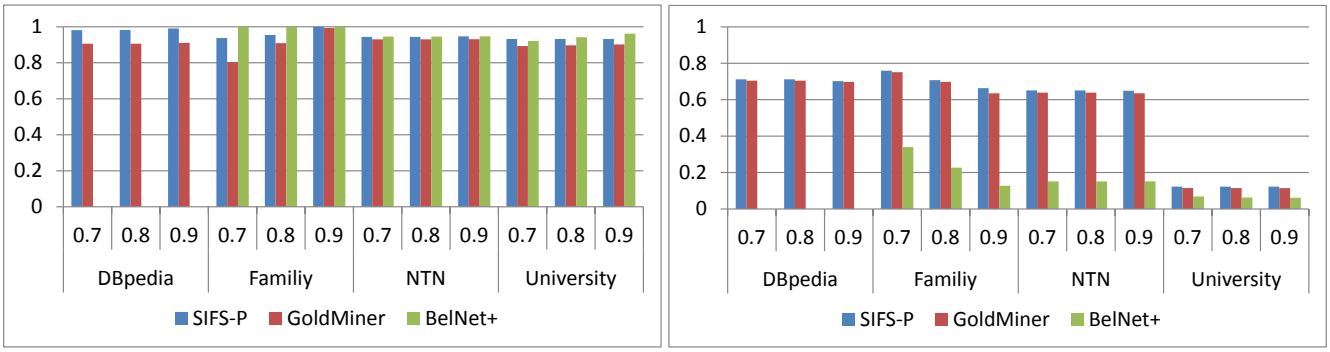


Figure 2: Precision (left) and recall (right) of systems to generate disjointness axioms

respectively. In these figures, 0.7, 0.8 and 0.9 in the horizontal axis indicate the thresholds to filter those rules with lower confidences. Since BelNet$^+$ fails to deal with large data like DBpedia, the comparison is conducted over other three KBs.

We first compare SIFS-P with GoldMiner. When mining subclass axioms, both systems have achieved very high precisions because almost all subclass axioms generated by them are correct. With respect to recall, SIFS-P performs better in most cases since more subclass axioms have been found by SIFS-P. This reflects the advantage of using type inference. That is, more type assertions recommended by type inference can increase the support of a rule and thus more correct subclass axioms could be generated. When mining disjointness axioms, the advantage of using type inference becomes more obvious, especially for DBpedia. When the recalls of SIFS-P and GoldMiner are quite similar, the precisions of SIFS-P are much better.

Comparing SIFS-P with BelNet$^+$, the precisions of BelNet$^+$ are more than 0.9, but 72% of its recalls are no more than 0.6. For SIFS-P, the precisions are similar to those of BelNet$^+$ while 67% of its recalls are more than 0.6. This shows that BelNet$^+$ generates much less disjointness axioms than SIFS-P. BelNet$^+$ depends on the joint probability to learn disjointness axioms. In the training stage, the number of individuals belonging to the pair of concepts is not large enough, which causes that many disjointness axioms are discarded when constructing the network. The experimental results show again that SIFS-P could learn more disjointness axioms whose quality is also high due to the adoption of type inference.

6 Related Work

The most relevant work to ours is the approach given in Völker and Niepert [2011] which uses association rule mining to mine schema information. Their algorithms have been implemented in GoldMiner. This approach is extended in Fleischhacker and Völker [2011] by considering naïve negative association rule mining to learn disjointness axioms. Our systems can be seen as an extension of GoldMiner by considering type inference and proportional support. However, as our experiments have shown, GoldMiner has difficulties to deal with the incompleteness problem.

Another relevant work is given in Zhu *et al.* [2015] which integrates probabilistic inference capability of Bayesian Networks with logical formalism of Description Logics. It considers schema learning as instance classification. To be consistent with OWA, the traditional confusion matrix is extended for considering unknown results. The corresponding algorithms have been implemented in BelNet$^+$. According to our comparison, BelNet$^+$ fails to deal with those cases that the number of instances belonging to the pair of concepts is not large enough.

Our approach is inspired by the rule mining model given in Galárraga *et al.* [2013], which conforms to the OWA. This work introduces novel definitions of support and confidence.

In order to acquire the unknown facts, the notion of functionality is applied to acquire negative examples of a binary relation. Since the goal of this approach is to mine horn logical rules and a type assertion is not functional, it is not suitable to generate axioms from an incomplete KB.

Inductive logic programming (ILP), which marries machine learning and data mining, is often used to learn schema information. In order to induce concept definition axioms from existing instances, Lehmann and his colleagues propose an approach in Lehmann and Hitzler [2010] to generate candidate concept descriptions by a downward refinement operator. Later on, this approach is extended to deal with very large dataset in S *et al.* [2011]. All the relevant algorithms have been implemented in the tool DL-Learner Bühmann *et al.* [2016]. However, noisy negative examples will influence the performance of such approaches.

There are also other works to generate schema information. In Töpper *et al.* [2012], all concepts are mapped to vectors with the same dimension. If the similarity of two concepts is lower than a threshold, they are regarded as disjoint. The authors of Meilicke *et al.* [2008] proposes an appropriate heuristic rule for learning disjointness axioms. This heuristic rule assumes that all sibling concepts are disjoint. A light-weight approach to enrich knowledge is presented in Bühmann and Lehmann [2012], which uses SPARQL queries to learn axioms. The approach given in Völker *et al.* [2007] is a supervised learning approach, whose training set needs to be constructed manually.

7 Conclusion

In this paper, we proposed a novel approach to learning disjointness and subclass axioms from incomplete semantic data under OWA. We first applied the type inference algorithm to generate new probabilistic type assertions. We then introduced novel definitions of support and confidence using negative examples as constraints. The experimental results were provided to compare our system with existing one and showed that SIFS-P performs better with respect to precision and recall in most cases.

In the future, we plan to extend the SIFS-P to learn more kinds of axioms such as the axioms with existential restriction, universal restriction and the limited extensional quantification.

References

Lorenz Bühmann and Jens Lehmann. Universal OWL axiom enrichment for large knowledge bases. In *Proceedings of EKAW 2012*, pages 57–71, 2012.

Lorenz Bühmann, Jens Lehmann, and Patrick Westphal. Dl-learner - A framework for inductive learning on the semantic web. *J. Web Sem.*, 39:15–24, 2016.

Nicola Fanizzi, Claudia d'Amato, and Floriana Esposito. DL-FOIL concept learning in description logics. In *Proceedings of ILP 2008*, pages 107–121, 2008.

Daniel Fleischhacker and Johanna Völker. Inductive learning of disjointness axioms. In *Proceedings of OTM 2011*, pages 680–697, 2011.

Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. AMIE: Association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of WWW 2013*, pages 413–422, 2013.

Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2012.

Jens Lehmann and Pascal Hitzler. Concept learning in description logics using refinement operators. *Machine Learning*, 1-2(78):203–250, 2010.

Christian Meilicke, Johanna Völker, and Heiner Stuckenschmidt. Learning disjointness for debugging mappings between lightweight ontologies. In *Proceedings of EKAW 2008*, pages 93–108, 2008.

Andreas Nolle, Christian Meilicke, Melisachew Wudage Chekol, German Nemirovski, and Heiner Stuckenschmidt. Schema-based debugging of federated data sources. In *ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*, pages 381–389, 2016.

Heiko Paulheim and Christian Bizer. Type inference on noisy RDF data. In *Proceedings of ISWC 2013*, pages 510–525, 2013.

Guilin Qi, Zhe Wang, Kewen Wang, Xuefeng Fu, and Zhiqiang Zhuang. Approximating model-based abox revision in dl-lite: Theory and practice. In *Proceedings of AAAI 2015*, pages 254–260, 2015.

Hellmann S, Lehmann J, and Auer S. Learning of owl class expressions on very large knowledge bases and its applications. *Interoperability Semantic Services and Web Applications*, pages 104–130, 2011.

Gerald Töpper, Magnus Knuth, and Harald Sack. Dbpedia ontology enrichment for inconsistency detection. In *Proceedings of I-SEMANTICS 2012*, pages 33–40, 2012.

Johanna Völker and Mathias Niepert. Statistical schema induction. In *Proceedings of ESWC 2011*, pages 124–138, 2011.

Johanna Völker, Denny Vrandečić, York Sure, and Andreas Hotho. Learning disjointness. In *Proceedings of ESWC 2007*, pages 175–189, 2007.

Johanna Völker, Daniel Fleischhacker, and Heiner Stuckenschmidt. Automatic acquisition of class disjointness. *J. Web Sem.*, 35:124–139, 2015.

Man Zhu, Zhiqiang Gao, J.Z. Pan, Yuting Zhao, Ying Xu, and Zhibin Quan. Ontology learning from incomplete semantic web data by belnet. In *Proceedings of ICTAI 2013*, pages 761–768, 2013.

Man Zhu, Zhiqiang Gao, Jeff Z. Pan, Yuting Zhao, Ying Xu, and Zhibin Quan. Tbox learning from incomplete data by inference in belnet$^+$. *Knowledge-Based Systems*, 75(5):30–40, 2015.

Strength Factors: An Uncertainty System for a Quantified Modal Logic

Naveen Sundar Govindarajulu and **Selmer Bringsjord**
Rensselaer Polytechnic Institute, Troy, NY
{naveensundarg,selmer.bringsjord}@gmail.com

Abstract

We present a new system $\mathcal{S}$ for handling uncertainty in a quantified modal logic (first-order modal logic). The system is based on both probability theory and proof theory and is derived from Chisholm's epistemology. We concretize Chisholm's system by grounding his undefined and primitive (i.e. foundational) concept of **reasonableness** in probability and proof theory. We discuss applications of the system. The system described below is a work in progress; hence we end by presenting a list of future challenges.

1 Introduction

We introduce a new system $\mathcal{S}$ for talking about uncertainty of iterated beliefs in a quantified modal logic with belief operators. The quantified modal logic we use is based on the **deontic cognitive event calculus** ($\mathcal{DCEC}$), which belongs to the family of **cognitive calculi** that have been used in modeling complex cognition. Here, we use a subset of $\mathcal{DCEC}$ that we term **micro cognitive calculus** ($\mu\mathcal{C}$). Specifically, we add a system of uncertainty derived from Chisholm's epistemology [Chisholm, 1987].[1] The system $\mathcal{S}$ is a work in progress and hence the presentation here will be abstract in nature.

One of our primary motivations is to design a system of uncertainty that is easy to use in end-user facing systems. There have been many studies that show that laypeople have difficulty understanding raw probability values (e.g. see [Kaye and Koehler, 1991]); and we believe that our approach borrowed from philosophy can pave the way for systems that can present uncertain statements in a more understandable format to lay users.

$\mathcal{S}$ can be useful in systems that have to interact with humans and provide justifications for their uncertainty. As a demonstration of the system, we apply the system to provide a solution to the lottery paradox. Another advantage of the system is that it can be used to provide uncertainty values for counterfactual statements. Counterfactuals are statements that an agent knows for sure are false. Among other cases, counterfactuals are useful when systems have to explain their actions to users (*If I had not done* α*, then* ϕ *would have happened*). Uncertainties for counterfactuals fall out naturally from our system. Before we discuss the calculus and present $\mathcal{S}$, we go through relevant prior work.

2 Prior Work

Members in the family of cognitive calculi have been used to formalize and automate highly intensional reasoning processes.[2] More recently, using $\mathcal{DCEC}$ we have presented an automation of **the doctrine of double effect** in [Govindarajulu and Bringsjord, 2017].[3] We quickly give an overview of the doctrine to illustrate the scope and expressivity of cognitive calculi such as $\mathcal{DCEC}$. The doctrine of double effect is an ethical principle that has been shown to be used by both untrained laypeople and experts when faced with moral dilemmas; and it plays a central role in many legal systems. Moral dilemmas are situations in which all available options have both good and bad consequences. The doctrine states that an action α in such a situation is permissible *iff* — (1) it is morally neutral; (2) the net good consequences outweigh the bad consequences by some large amount γ; and (3) at least one or more of the good consequences are *intended*, and none of the bad consequences are intended. The conditions require both intensional operators and a calculus (e.g. the event calculus) for modeling commonsense reasoning and the physical world. Other tasks automated by cognitive calculi include the false-belief task [Arkoudas and Bringsjord, 2008] and *akrasia* (succumbing to temptation to violate moral principles) [Bringsjord *et al.*, 2014].[4] Each cognitive calculus is a sorted (i.e. typed) quantified modal logic (also known as sorted first-order modal logic). Each calculus has a well-defined syntax and proof calculus. The proof calculus is based on natural deduction [Gentzen, 1935], and includes all the introduction

[1] See the SEP entry on Chisholm for a quick overview of Chisholm's epistemology: https://plato.stanford.edu/entries/chisholm/#EpiIEpiTerPriFou.

[2] By "intensional processes", we roughly mean processes that take into account knowledge, beliefs, desires, intentions, etc. of agents. Compare with extensional systems such as first-order logic that do not take into account states of minds of other agents. This is not be confused with "intentional" systems which would be modeled with intensional systems. See [Zalta, 1988] for a detailed treatment of intensionality.

[3] This work will be presented at IJCAI 2017.

[4] Arkoudas and Bringsjord [2008] introduced the general family of **cognitive event calculi** to which $\mathcal{DCEC}$ belongs.

and elimination rules for first-order logic, as well as inference schemata for the modal operators and related structures.

On the uncertainty and probability front, there have been many logics of probability, see [Demey *et al.*, 2016] for an overview. Since our system builds upon probabilities, our approach could use a variety of such systems. There has been very little work in uncertainty systems for first-order modal logics. Among first-order systems, the seminal work in [Halpern, 1990] presents a first-order logic with modified semantics to handle probabilistic statements. We can use such a system as the foundation for our work, and use it to define the base probability function **Pr** used below. (Note that we leave **Pr** unspecified for now.)

3 The Formal System

The formal system $\mu\mathcal{C}$ is a modal extension of the the event calculus. The event calculus is a multi-sorted first-order logic with a family of axiom sets. The exact axiom set is not important. The primary sorts in the system are shown below.

Sort	Description
Agent	Human and non-human actors.
Moment or Time	Time points and intervals. E.g. simple, such as t_i, or complex, such as $birthday(son(jack))$.
Event	Used for events in the domain.
ActionType	Action types are abstract actions. They are instantiated at particular times by actors. E.g.: "eating" vs. "jack eats."
Action	A subtype of Event for events that occur as actions by agents.
Fluent	Used for representing states of the world in the event calculus.

Full $\mathcal{DCEC}$ has a suite of modal operators and inference schemata. Here we focus on just two: an operator for belief **B** and an operator for perception **P**. The syntax of and inference schemata of the system are shown below. S is the set of all sorts, f are the core function symbols, t shows the set of terms, and ϕ is the syntax for the formulae.

Syntax

$$S ::= \begin{cases} \mathsf{Object} \mid \mathsf{Agent} \mid \mathsf{Self} \sqsubset \mathsf{Agent} \mid \mathsf{ActionType} \mid \mathsf{Action} \sqsubseteq \mathsf{Event} \mid \\ \mathsf{Moment} \mid \mathsf{Formula} \mid \mathsf{Fluent} \mid \mathsf{Numeric} \end{cases}$$

$$f ::= \begin{cases} action : \mathsf{Agent} \times \mathsf{ActionType} \rightarrow \mathsf{Action} \\ initially : \mathsf{Fluent} \rightarrow \mathsf{Formula} \\ holds : \mathsf{Fluent} \times \mathsf{Moment} \rightarrow \mathsf{Formula} \\ happens : \mathsf{Event} \times \mathsf{Moment} \rightarrow \mathsf{Formula} \\ clipped : \mathsf{Moment} \times \mathsf{Fluent} \times \mathsf{Moment} \rightarrow \mathsf{Formula} \\ initiates : \mathsf{Event} \times \mathsf{Fluent} \times \mathsf{Moment} \rightarrow \mathsf{Formula} \\ terminates : \mathsf{Event} \times \mathsf{Fluent} \times \mathsf{Moment} \rightarrow \mathsf{Formula} \\ prior : \mathsf{Moment} \times \mathsf{Moment} \rightarrow \mathsf{Formula} \end{cases}$$

$$t ::= x : S \mid c : S \mid f(t_1, \ldots, t_n)$$

$$\phi ::= \begin{cases} t : \mathsf{Formula} \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \rightarrow \psi \mid \phi \leftrightarrow \psi \\ \mathbf{P}(a,t,\phi) \mid \mathbf{B}(a,t,\phi) \end{cases}$$

The above calculus lets us formalize statements of the form *"John believes now that Mary perceived that it was raining."* One formalization could be:

$$\exists t < now : \mathbf{B}\Big(john, now, \mathbf{P}\big(mary, t, holds(raining, t)\big)\Big)$$

The figure below shows the inference schemata for $\mu\mathcal{C}$. $R_{\mathbf{P}}$ captures that perceptions get turned into beliefs. $R_{\mathbf{B}}$ is an inference schema that lets us model idealized agents that have their beliefs closed under the $\mu\mathcal{C}$ proof theory. While normal humans are not deductively closed, this lets us model more closely how deliberate agents such as organizations and more strategic actors reason. Assume that there is a background set of axioms Γ we are working with.

Inference Schemata

$$\frac{\mathbf{P}(a,t_1,\phi_1),\ \Gamma \vdash t_1 < t_2}{\mathbf{B}(a,t_2,\phi)}\ [R_{\mathbf{P}}]$$

$$\frac{\mathbf{B}(a,t_1,\phi_1),\ldots,\mathbf{B}(a,t_m,\phi_m),\ \{\phi_1,\ldots,\phi_m\} \vdash \phi,\ \Gamma \vdash t_i < t}{\mathbf{B}(a,t,\phi)}\ [R_{\mathbf{B}}]$$

4 The Uncertainty System $\mathcal{S}$

In the uncertainty system, we augment the belief modal operators with a discrete set of uncertainty factors termed as *strength factors*. The factors are not arbitrary and are based on how derivable a proposition is for a given agent.

Chisholm's epistemology has a primitive undefined binary relation that he terms **reasonableness** with which he defines a scale of strengths for beliefs one might have in a proposition. Note that Chisholm's system is agent free while ours is agent-based. Let $\phi \succ_t^a \psi$ denote that ϕ is more reasonable than ψ to an agent a at time t. We require that $\succ_t^a$ be *asymmetric*: i.e., *irreflexive* and *anti-symmetric*. That is, for all ϕ, $\phi \not\succ_t^a \phi$; and for all ϕ and ψ,

$$(\phi \succ_t^a \psi) \Rightarrow (\psi \not\succ_t^a \phi)$$

We also require that $\succ_t^a$ be transitive. In addition to these conditions, we have the following five requirements governing how $\succ_t^a$ interacts with the logical connectives $\wedge, \neg$ and **B** (the first three conditions can be derived from the defintion of $\succ$ sketched out later):

$[\mathbf{C}_{\wedge_1}]$ $(\psi_1 \succ_t^a \phi_1)$ and $(\psi_2 \succ_t^a \phi_2) \Rightarrow \Big(\psi_1 \succ_t^a \phi_1 \wedge \phi_2\Big)$

$[\mathbf{C}_{\wedge_2}]$ $(\psi_1 \wedge \psi_2 \succ_t^a \phi) \Rightarrow \Big[(\psi_1 \succ_t^a \phi) \text{and } (\psi_2 \succ_t^a \phi)\Big]$

$[\mathbf{C}_{\neg}]$ There is no ϕ such that $(\bot \succ_t^a \phi)$; and for all $\phi (\phi \succ_t^a \bot)$

$[\mathbf{C}_{\mathbf{B}_1}]$ $\Big(\mathbf{B}(a,t,\phi) \succ_t^a \mathbf{B}(a,t,\neg\phi)\Big) \Rightarrow \Big(\mathbf{B}(a,t,\phi) \succ_t^a \neg\mathbf{B}(a,t,\phi)\Big)$

$[\mathbf{C}_{\mathbf{B}_2}]$ For all ϕ, $\begin{bmatrix} \big(\mathbf{B}(a,t,\phi) \succ_t^a \mathbf{B}(a,t,\neg\phi)\big) \text{ or} \\ \big(\mathbf{B}(a,t,\neg\phi) \succ_t^a \mathbf{B}(a,t,\phi)\big) \end{bmatrix}$

We also add a **belief consistency condition** which requires that:

$$\Big(\Gamma \vdash \mathbf{B}^p(a,t,\phi)\Big) \Leftrightarrow \Big(\Gamma \nvdash \mathbf{B}^p(a,t,\neg\phi)\Big)$$

For convenience, we define a new operator, the withholding operator **W** (this is simply syntactic sugar):

$$\mathbf{W}(a,t,\phi) \equiv \neg\mathbf{B}(a,t,\phi) \wedge \neg\mathbf{B}(a,t,\neg\phi)$$

We now reproduce Chisholm's system below. Note the formula used in the definitions below are meta-formula and not strictly in $\mu\mathcal{C}$.

Strength Factor Definitions

Acceptable An agent a at time t finds ϕ acceptable *iff* withholding ϕ is not more reasonable than believing in ϕ.

$$\mathbf{B}^1(a,t,\phi) \Leftrightarrow \begin{cases} \mathbf{W}(a,t,\phi) \not\succ_t^a \mathbf{B}(a,t,\phi); \text{ or} \\ \Big(\neg\mathbf{B}(a,t,\phi) \wedge \neg\mathbf{B}(a,t,\neg\phi)\Big) \not\succ_t^a \mathbf{B}(a,t,\phi) \end{cases}$$

Some Presumption in Favor An agent a at time t has some presumption in favor of ϕ *iff* believing ϕ at t is more reasonable than believing $\neg\phi$ at time t:

$$\mathbf{B}^2(a,t,\phi) \Leftrightarrow \Big(\mathbf{B}(a,t,\phi) \succ_t^a \mathbf{B}(a,t,\neg\phi)\Big)$$

Beyond Reasonable Doubt An agent a at time t has beyond reasonable doubt in ϕ *iff* believing ϕ at t is more reasonable than withholding ϕ at time t:

$$\mathbf{B}^3(a,t,\phi) \Leftrightarrow \begin{cases} \mathbf{B}(a,t,\phi) \succ_t^a \mathbf{W}(a,t,\phi); \text{ or} \\ \Big(\mathbf{B}(a,t,\phi) \succ_t^a \Big(\neg\mathbf{B}(a,t,\phi) \wedge \neg\mathbf{B}(a,t,\neg\phi)\Big)\Big) \end{cases}$$

Evident A formula ϕ is evident to an agent a at time t *iff* ϕ is beyond reasonable doubt and if there is a ψ such that believing ψ is more reasonable for a at time t than believing ϕ, then a is certain about ψ at time t.

$$\mathbf{B}^4(a,t,\phi) \Leftrightarrow \begin{cases} \mathbf{B}^3(a,t,\phi) \wedge \\ \exists\psi : \begin{bmatrix} \mathbf{B}(a,t,\psi) \succ_t^a \mathbf{B}(a,t,\phi) \\ \Rightarrow \mathbf{B}^5(a,t,\psi) \end{bmatrix} \end{cases}$$

Certain An agent a at time t is certain about ϕ *iff* ϕ is beyond reasonable doubt and there is no ψ such that believing ψ is more reasonable for a at time t than believing ϕ.

$$\mathbf{B}^5(a,t,\phi) \Leftrightarrow \begin{cases} \mathbf{B}^3(a,t,\phi) \wedge \\ \neg\exists\psi : \mathbf{B}(a,t,\psi) \succ_t^a \mathbf{B}(a,t,\phi) \end{cases}$$

The above definitions are from Chisholm but more rigorously formalized in $\mu\mathcal{C}$. The definitions and the conditions $\{[\mathbf{C}_{\wedge_1}], [\mathbf{C}_{\wedge_2}, [\mathbf{C}_{\neg}], [\mathbf{C}_{\mathbf{B}_1}], [\mathbf{C}_{\mathbf{B}_1}]\}$ give us the following theorem.

Theorem: Higher Strength subsumes Lower Strength

For any p and q, if $p > q$, we have: $\mathbf{B}^p(a,t,\phi) \Rightarrow \mathbf{B}^q(a,t,\phi)$

Proof: $\mathbf{B}^5 \Rightarrow \mathbf{B}^3$ and $\mathbf{B}^4 \Rightarrow \mathbf{B}^3$ by definition. $\mathbf{B}^5 \Rightarrow \mathbf{B}^4$ by the second clause in the definitions of $\mathbf{B}^4$ and $\mathbf{B}^5$. $\mathbf{B}^3 \Rightarrow \mathbf{B}^1$ by the asymmetry property of $\succ_t^a$.

For $\mathbf{B}^2 \Rightarrow \mathbf{B}^1$, we have a proof by contradiction. Assume that (in shorthand):

$$(\mathbf{B}\phi \succ \mathbf{B}\neg\phi) \text{ but } (\neg\mathbf{B}\phi \wedge \neg\mathbf{B}\neg\phi) \succ \mathbf{B}\phi$$

Using $[\mathbf{C}_{\mathbf{B}_1}]$ on the former and $[\mathbf{C}_{\wedge_2}]$ on the latter, we get

$$\mathbf{B}\phi \succ \neg\mathbf{B}\phi \text{ and } \neg\mathbf{B}\phi \succ \mathbf{B}\phi$$

Using transitivity, we get $\mathbf{B}\phi \succ \mathbf{B}\phi$. This violates irreflexivity, therefore $\mathbf{B}^2 \Rightarrow \mathbf{B}^1$.

For $\mathbf{B}^3 \Rightarrow \mathbf{B}^2$, if the condition for $\mathbf{B}^2$ does not hold, by $\mathbf{C}_{\mathbf{B}_2}$ we have:

$$\mathbf{B}\neg\phi \succ \mathbf{B}\phi$$

Using the condition for $\mathbf{B}^3$ and transitivity, we get

$$\mathbf{B}\neg\phi \succ \neg\mathbf{B}\phi \wedge \neg\mathbf{B}\neg\phi$$

giving us $\mathbf{B}^3\neg\phi$, and we started with $\mathbf{B}^3\phi$. This violates the belief consistency condition. ■

The definitions almost give us $\mathcal{S}$ except for the fact that $\succ_t^a$ is undefined. While Chisholm gives a careful and informal analysis of the relation, he does not provide a more precise definition. Such a definition is needed for automation. We provide a three clause defintion that is based on both probabilities and proof theory.

There are many probability logics that allow us to define probabilities over formulae. They are well studied and understood for propositional and first-order logics. Let $\mathcal{L}$ be the set of all formulae in $\mu\mathcal{C}$. Let $\mathcal{L}_p$ be a pure first-order subset of $\mathcal{L}$. Assume that we have the following partial probability function defined over $\mathcal{L}_p$[5]:

$$\mathbf{Pr} : \mathsf{Agent} \times \mathsf{Moment} \times \mathsf{Formula} \mapsto \mathbb{R}$$

Then we have the first clause of our definition for $\succ_t^a$.

Clause I. Defining $\succ$

If $\mathbf{Pr}(a,t,\phi)$ and $\mathbf{Pr}(a,t,\psi)$ are defined then:

$$\Big(\mathbf{B}(a,t,\phi) \succ_t^a \mathbf{B}(a,t,\psi)\Big) \Leftrightarrow \Big(\mathbf{Pr}(a,t,\phi) > \mathbf{Pr}(a,t,\psi)\Big)$$

We might not always have meaningful probabilities for all propositions. For example, consider propositions of the form *"I believe that Jack believes that* ϕ*."* It is hard to get precise numbers for such statements. In such situations, we might look at the ease of derivation of such statements given a knowledge base Γ. Given two competing statements ϕ and ψ, we can say one is more reasonable than the other if we can easily derive one more than the other from Γ. This assumes that we can derive ϕ and ψ from Γ. We assume we have a cost function $\rho : \mathsf{Proof} \mapsto \mathbb{R}^+$ that lets us compute costs of proofs. There are many ways of specifying such functions. Possible candidates are length of the proof, time for computing the proof, depth vs breadth of the proof, unique symbols used in the proof etc. We leave this choice unspecified but any such function could work here. Let $\vdash_{a,t}$ denote provability w.r.t. to agent a at time t.

Clause II. Defining $\succ$

If one of $\mathbf{Pr}(a,t,\phi)$ and $\mathbf{Pr}(a,t,\psi)$ is not defined, but if $\Gamma \vdash_{a,t} \phi$ and $\Gamma \vdash_{a,t} \psi$:

[5] Something similar to the system in [Halpern, 1990] that accounts for probabilities as statistical information or degrees of belief can work.

$$\Big(\phi \succ_t^a \psi\Big) \Leftrightarrow \Big(\rho\big(\Gamma \vdash_{a,t} \phi\big) < \rho\big(\Gamma \vdash_{a,t} \psi\big)\Big)$$

Clauses I and II might not always be applicable as the premises in the definitions might not always hold. A more common case could be when we cannot derive the propositions of interest from our background set of axioms Γ. For example, if we are interested in the uncertainty values for statements that we know are false, then it should be the case that they be not derivable from our background set of axioms. In this situation, we look at Γ and see what minimal changes we can make to it to let us derive the proposition of interest. Trivially, if we cannot derive ϕ from Γ, we can add it to Γ to derive it, as $\Gamma + \phi \vdash \phi$. This is not desirable for two reasons.

First, simply adding to Γ might result in a contradiction. In such cases we would be looking at removing a minimal set of statements Λ from Γ. Second, we might prefer to add a more simpler set of propositions Θ to Γ rather than ϕ itself to derive ϕ. Recapping, we go from (1) to (2) below:

$$\Gamma \nvdash \phi \quad (1)$$
$$\Gamma \cup \Theta - \Lambda \vdash \phi \quad (2)$$

When we go from (1) to (2) we would like to modify the background axioms as minimally as possible. Assume that we have a similarity function π for sets of formulae. We then choose Θ and Λ as given below ($Con[S]$ denotes that S is consistent):

$$\langle \Theta, \Lambda \rangle = \underset{\langle \Theta, \Lambda \rangle}{\arg\min}\, \pi(\Gamma, \Gamma \cup \Theta - \Lambda); \text{ such that } \begin{cases} \Gamma \cup \Theta - \Lambda \vdash \phi; \text{ and} \\ Con\big[\Gamma \cup \Theta - \Lambda\big] \end{cases}$$

Consider a statement such as *"It rained last week"* when it did not actually rain last week, and another statement such as *"The moon is made of cheese."* Both statements denote things that did not happen, but intuitively it seems that former should be more easier to accept from what we know than the latter. There are many similarity measures which can help convey this. Analogical reasoning is one such possible measure of similarity. If the new formulae are structurally similar to existing formulae, then we might be more justified in accepting such formulae. For example, one such measure could be the analogical measure used by us in [Licato *et al.*, 2013].

Now we have the formal mechanism in place for defining the final clause in our definition for our reasonableness. Let $\delta_t^a(\Gamma, \phi)$ be the distance between Γ and closest consistent set under π that lets us prove ϕ:

$$\delta_t^a(\Gamma, \phi) \equiv \min_{\langle \Theta, \Lambda \rangle} \left\{ \pi\big(\Gamma, \Gamma \cup \Theta - \Lambda\big) \;\middle|\; \begin{array}{l} \big(\Gamma \cup \Theta - \Lambda\big) \vdash_t^a \psi; \text{ and} \\ Con\big[\Gamma \cup \Theta - \Lambda\big] \end{array} \right\}$$

Clause III. Defining $\succ$

If one of $\mathbf{Pr}(a,t,\phi)$ and $\mathbf{Pr}(a,t,\psi)$ is not defined, and one of $\Gamma \vdash_{a,t} \phi$ and $\Gamma \vdash_{a,t} \psi$ does not hold, then

$$\Big(\phi \succ_t^a \psi\Big) \Leftrightarrow \Big[\delta_t^a(\Gamma, \phi) < \delta_t^a(\Gamma, \psi)\Big]$$

The final piece of $\mathcal{S}$ is inference rules for belief propagation with uncertainty values. This is quite straightforward. Inferences propagate uncertainty values from the premises with the lowest strength factor; and inferences happen only with beliefs that are close in their uncertainty values, with maximum difference being parametrized by u, with default $u = 2$.

Inference Schemata for $\mathcal{S}$

$$\frac{\mathbf{P}(a,t_1,\phi_1),\ \ \Gamma \vdash t_1 < t_2}{\mathbf{B}^5(a,t_2,\phi)}\ [R^s_{\mathbf{P}}]$$

$$\frac{\mathbf{B}^{s_1}(a,t_1,\phi_1),\ldots,\mathbf{B}^{s_m}(a,t_m,\phi_m),\{\phi_1,\ldots,\phi_m\} \vdash \phi, \Gamma \vdash t_i < t}{\mathbf{B}^{min(s_1,\ldots,s_m)}(a,t,\phi)}\ [R^s_{\mathbf{B}}]$$
$$\text{with} \quad max(\{s_1,\ldots,s_m\}) - min(\{s_1,\ldots,s_m\}) \leq u$$

5 Usage

In this section, we illustrate $\mathcal{S}$ by applying it solve problems of foundational interest such as the lottery paradox [Kyburg Jr, 1961, p. 197] and a toy version of a more real life example, a murder mystery example (following in the traditions of logic pedagogy). Finally, we very briefly sketch abstract scenarios in which $\mathcal{S}$ can be used to generate uncertainty values for counterfactual statements and to generate explanations for actions.

5.1 Paradoxes: Lottery Paradox

In the lottery paradox, we have a situation in which an agent a comes to believe ϕ and $\neg\phi$ from a seemingly consistent set of premises Γ_L describing a lottery. Our solution to the paradox is that the agent simply has different strengths of beliefs in the proposition and its negation. We first go over the paradox formalized in $\mu\mathcal{C}$ and then present the solution.

Let Γ_L be a meticulous and perfectly accurate description of a 1,000,000,000,000-ticket lottery, of which rational agent a is fully apprised. Assume that from Γ_L it can be proved that either ticket 1 will win or ticket 2 will win or ... or ticket 1,000,000,000,000 will win. Lets write this (exclusive) disjunction as follows (here $\oplus$ is an exclusive disjunction):

$$\Gamma_L \vdash win(t_1) \oplus win(t_2) \oplus \ldots \oplus win(t_{1,000,000,000,000})$$

The paradox has two strands of reasoning. The first strand yields $\mathbf{B}(a, now, \phi)$ and the second strand yields $\mathbf{B}(a, now, \neg\phi)$ with $\phi \equiv \exists t : win(t)$.

Strand 1: Since a believes all propositions in Γ_L, a can then deduce from this the belief that there is at least one ticket that will win, a proposition represented as:

$$\boxed{\mathrm{S}_1} \quad \mathbf{B}\Big(a, now, \exists t : win\big(t\big)\Big)$$

Strand 2: From Γ_L it can be proved that the probability of a particular ticket t_i winning is 10^{-12}.

$$\Big[\mathbf{Pr}\Big(a, now, win\big(t_1\big)\Big) = 10^{-12}\Big] \wedge \Big[\mathbf{Pr}\Big(a, now, win\big(t_2\big)\Big) = 10^{-12}\Big] \wedge \ldots \wedge \Big[\mathbf{Pr}\Big(a, now, win\big(t_{1T}\big)\Big) = 10^{-12}\Big]$$

For the next step, note that the probability of ticket t_1 winning is lower than, say, the probability that if you walk outside a minute from now, you will be flattened on the spot by a door from a 747 that falls from a jet of that type cruising at 35,000 feet. Since you, the reader, have the rational belief that death won't ensue if you go outside (and have this belief precisely because you believe that the odds of your sudden demise in this manner are vanishingly small), the inference to the rational belief on the part of a that t_1 won't win sails through — and this of course works for each ticket. Hence we have as a valid belief (though not derivable in $\mu\mathcal{C}$ from Γ_L):

$$\mathbf{B}\Big(a, now, \neg win\big(t_1\big)\Big) \wedge \mathbf{B}\Big(a, now, \neg win\big(t_2\big)\Big) \wedge \ldots$$
$$\wedge \mathbf{B}\Big(a, now, \neg win\big(t_{1T}\big)\Big)$$

From $R_{\mathbf{B}}$ and above, we get:

$$\mathbf{B}\Big(a, now, \neg win\big(t_1\big) \wedge \neg win\big(t_2\big) \wedge \ldots \wedge \neg win\big(t_{1T}\big)\Big)$$

Applying $R_{\mathbf{B}}$ to the above and Γ_L, we get:

$$\boxed{\mathrm{S}_2} \quad \mathbf{B}\Big(a, now, \neg\exists t : win(t)\Big)$$

The two strands are complete, and we have derived contradictory beliefs labeled S_1 and S_2. Our solution consists of two new uncertainty infused strands that result in beliefs of sufficiently varying strengths that block inferences that could combine them.

Strand 1 and **Strand 2** demonstrate the standard informal reasoning which leads to the paradox. We replicate the reasoning in $\mu\mathcal{C}$ and show that the paradox is not derivable.

Strand 3: Assume that a is certain of all propositions in Γ_L, then using $R^s_{\mathbf{B}}$, we have:

$$\boxed{\mathrm{S}_3} \quad \mathbf{B}^5\big(a, now, \exists t : win(t)\big)$$

Strand 4: Since $\mathbf{Pr}(a, now, win(t_i)) < \mathbf{Pr}(a, now, \neg win(t_i))$, using Clause I and the strength factor definitions, we have now that for all t_i

$$\mathbf{B}^2\big(a, now, \neg win(t_i)\big)$$

Using the reasoning similar to that in Strand 2, we get:

$$\boxed{\mathrm{S}_4} \quad \mathbf{B}^2\Big(a, now, \neg\exists t : win(t)\Big)$$

Strands 3 and 4 resolve the paradox. Note that $R^s_{\mathbf{B}}$ cannot be applied to S_3 and S_4 and churn out arbitrary propositions, as the default value of the u parameter in $R^s_{\mathbf{B}}$ requires beliefs to be no more than 2 levels apart. ■

5.2 Application: Solving a Murder

We look at a toy example in which an agent s has to solve a murder that happened at time t_3. s believes that either Alice or Bob is the murderer. The agent knows that there is a gun gun involved in the murder and that the owner of the gun at t_3 committed the murder. s also knows that Alice is the owner of the gun initially at time t_0.

Presumption in Favor of Alice **Being the Murderer**

From just these facts, the agent has some presumption for believing that Alice is the murderer.

Proof Sketch: All the above statements can be taken as certain beliefs $\mathbf{B}^5$ of s. For convenience, we consider the formulae directly without the belief operators.

In order to prove the above, we need to prove that it is easier for the agent to derive that Alice is the murderer than to derive that Alice is not the murderer. First, to prove the former, the agent just has to assume that Alice's ownership of the gun did not change from t_0 to t_3. Second, in order for the agent to believe that Alice did not commit the murder but Bob committed it, the agent must be willing to admit that something happened to change Alice's ownership of the gun from time t_0 to t_3 that results in Bob owning the gun. One possibility is that Alice simply sold the gun to Bob. Both the scenarios are shown as proofs in the Slate theorem proving workspace [Bringsjord *et al.*, 2008] in the Appendix. Figure 1 shows a proof modulo belief operators of $\mathbf{B}(s, now, Murderer(\mathsf{Alice}))$ from $\Gamma \cup \Theta_1$ and Figure 2 shows a proof of $\mathbf{B}(s, now, \neg Murderer(\mathsf{Alice}))$ from $\Gamma \cup \Theta_2$.

If we assume that Θ_1 and Θ_2 exhaust the space of allowed additions, then it easy to see how syntactic measures of complexity will yield that $\delta^a_t\big(\Gamma, \Gamma \cup \Theta_1\big) < \delta^a_t\big(\Gamma, \Gamma \cup \Theta_2\big)$ as Θ_2 is more complex than Θ_1. This lets us derive that s has some presumption in favor of $Murderer(\mathsf{Alice})$. ■

What happens if the agent knows or has a belief with certainty that Alice's ownership of the gun did not change from t_0 to t_3?

Beyond Reasonable Doubt that Alice **is the Murderer**

If the agent is certain that Alice's ownership of the gun did not change from t_0 till t_3, the agent has beyond reasonable doubt that she is the murderer.

Proof Sketch: In this case we directly have that:

$$\Gamma \vdash \mathbf{B}(s, now, Murderer(\mathsf{Alice}))$$
$$\Gamma \nvdash \neg\mathbf{B}(s, now, Murderer(\mathsf{Alice}))$$
$$\Gamma \nvdash \neg\mathbf{B}(s, now, \neg Murderer(\mathsf{Alice}))$$

In order to flip the last two statements above, we need to modify Γ, but we can derive that Alice is the murderer without any modifications, and since $\delta^a_t(\Gamma, \Gamma) = 0$, it easier to believe Alice is the murderer than to withhold that Alice is the murderer. ■

5.3 Counterfactuals

At time t, assume that an agent a believes in a set of propositions Γ and is interested in propositions $holds(f, t')$ and $holds(g, t')$ with $t' < t$ and:

$$\Gamma \vdash \neg holds(f, t') \wedge \neg holds(g, t')$$

We may need non-trivial uncertainty values, but in this case, **Pr** will assign a trivial value of 0 to both the propositions. We can then look at closest consistent sets to Γ under δ:

$$\Gamma_1 \vdash holds(f, t')$$
$$\Gamma_2 \vdash holds(g, t')$$

Clause III from the definition for reasonableness gives us:

$$\mathbf{B}\Big(a,t,holds(f,t')\Big) \succ_t^a \mathbf{B}\Big(a,t,holds(g,t')\Big)$$
$$\Leftrightarrow$$
$$\delta_t^a(\Gamma,\Gamma_1) < \delta_t^a(\Gamma,\Gamma_2)$$

5.4 Explanations

The definitions of the strength factors and reasonableness above can be used to generate high-level schemas for explanations. These schemas can be used instead of simply presenting raw probability values to end-users. While we have not fleshed out such explanation schemas, we illustrate one possible schema. Say an agent performs an action α on the basis of ϕ. In this case, the agent could generate an explanation that at the highest level simply says that it is more reasonable for the agent to believe ϕ than for the agent to believe in $\neg\phi$. The agent could then further explain why it was reasonable for it by using one of the three clauses in the reasonableness definition.

6 Inference Algorithm Sketch

Describing the inference algorithm in detail is beyond the scope of this paper, but we provide a high-level sketch here.[6] Our proof calculus is simply an extension of standard first-order proof calculus under different modal contexts. For example, if a believes that b believes in a set of propositions Γ and $\Gamma \vdash_{FOL} \psi$, then a believes that b believes ψ. We convert $\mathbf{B}(a,t_a,\mathbf{B}(b,t_b,Q))$ into the pure first-order formula $Q(context(a,t_a,b,t_b))$ and use a first-order prover. The conversion process is a bit more nuanced as we have to handle negations, properly handle substitutions of equalities, uncertainties and transform compound formulae within iterated beliefs.

7 Conclusion and Future Work

We have presented initial steps in building a system of uncertainty that is both probability and proof theory based that could lend itself to (1) solving foundational problems; (2) being useful in applications; (3) generating uncertainty values for counterfactuals; and (4) building understandable explanations.

Shortcomings of $\mathcal{S}$ can be cast as challenges, and many challenges exist, some relatively easy and some quite hard. Among the easy challenges are defining and experimenting with different candidates for **Pr**, ρ, π and δ. On the more difficult side, we have to come up with tractable computational mechanisms for computing the $\min_{\langle\Theta,\Lambda\rangle}$ in the definition for δ. Also on the difficult side, is the challenge of coming up efficient reasoning schemes. While we have an exact inference algorithm, we believe that an approximate algorithm that selectively discards beliefs in a large knowledge base during reasoning will be more useful. Future work also includes comparison with other uncertainty systems and exploration of conditions under which uncertainty values of $\mathcal{S}$ are similar/dissimilar with other systems (thresholded appropriately).

[6] More details can be found here: https://goo.gl/2Vz2nJ

Acknowledgements

We are grateful to the Office of Naval Research for their funding that enabled the research presented in this paper. We are also thankful for the insightful reviews provided by the three anonymous referees.

A Appendix: Slate Proofs

The figures below are vector graphics and can be zoomed to more easily read the contents.

Figure 1: Alice is the murder: $\mathbf{B}\Big(s,t,Murderer(\text{Alice})\Big)$

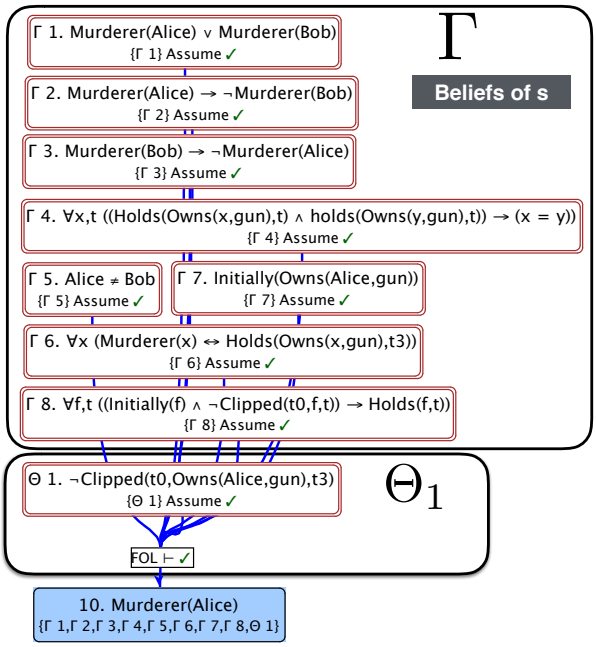
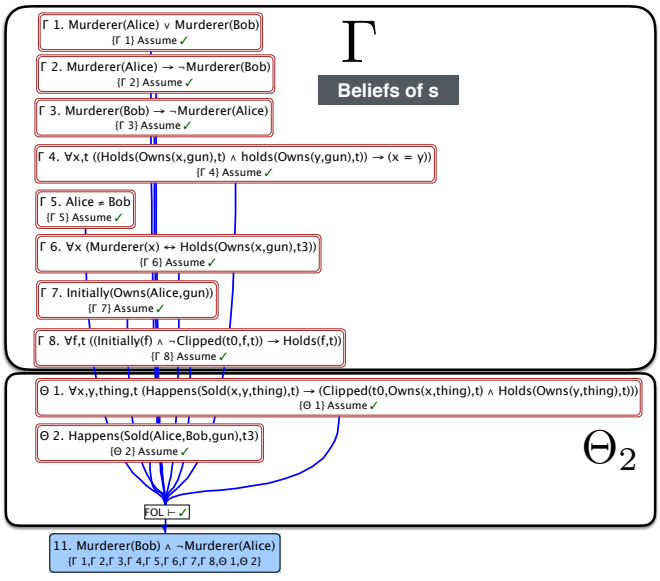


Figure 2: Alice is not the murder: $\mathbf{B}\Big(s,t,\neg Murderer(\text{Alice})\Big)$



References

[Arkoudas and Bringsjord, 2008] Konstantine Arkoudas and Selmer Bringsjord. Toward Formalizing Common-Sense Psychology: An Analysis of the False-Belief Task. In T.-B. Ho and Z.-H. Zhou, editors, *Proceedings of the Tenth Pacific Rim International Conference on Artificial Intelligence (PRICAI 2008)*, number 5351 in Lecture Notes in Artificial Intelligence (LNAI), pages 17–29. Springer-Verlag, 2008.

[Bringsjord *et al.*, 2008] Selmer Bringsjord, Joshua Taylor, Andrew Shilliday, Micah Clark, and Konstantine Arkoudas. Slate: An Argument-Centered Intelligent Assistant to Human Reasoners. In Floriana Grasso, Nancy Green, Rodger Kibble, and Chris Reed, editors, *Proceedings of the 8th International Workshop on Computational Models of Natural Argument (CMNA 8)*, pages 1–10, Patras, Greece, July 21 2008. University of Patras.

[Bringsjord *et al.*, 2014] Selmer Bringsjord, Naveen Sundar Govindarajulu, Daniel Thero, and Mei Si. Akratic Robots and the Computational Logic Thereof. In *Proceedings of ETHICS • 2014 (2014 IEEE Symposium on Ethics in Engineering, Science, and Technology)*, pages 22–29, Chicago, IL, 2014. IEEE Catalog Number: CFP14ETI-POD.

[Chisholm, 1987] Roderick Chisholm. *Theory of Knowledge 3rd ed*. Prentice-Hall, Englewood Cliffs, NJ, 1987.

[Demey *et al.*, 2016] Lorenz Demey, Barteld Kooi, and Joshua Sack. Logic and Probability. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2016 edition, 2016.

[Gentzen, 1935] Gerhard Gentzen. Investigations into Logical Deduction. In M. E. Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–131. North-Holland, Amsterdam, The Netherlands, 1935. This is an English version of the well-known 1935 German version.

[Govindarajulu and Bringsjord, 2017] Naveen Sundar Govindarajulu and Selmer Bringsjord. On Automating the Doctrine of Double Effect. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI 2017)*, 2017. Preprint available at this url: https://arxiv.org/abs/1703.08922.

[Halpern, 1990] Joseph Y Halpern. An Analysis of First-order Logics of Probability. *Artificial intelligence*, 46(3):311–350, 1990.

[Kaye and Koehler, 1991] D. H. Kaye and Jonathan J. Koehler. Can Jurors Understand Probabilistic Evidence? *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, 154(1):75–81, 1991.

[Kyburg Jr, 1961] Henry E Kyburg Jr. *Probability and the Logic of Rational Belief*. Wesleyan University Press, Middletown, CT, 1961.

[Licato *et al.*, 2013] John Licato, Naveen Sundar Govindarajulu, Selmer Bringsjord, Michael Pomeranz, and Logan Gittelson. Analogico-Deductive Generation of Gödel's First Incompleteness Theorem from the Liar Paradox. In Francesca Rossi, editor, *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI–13)*, pages 1004–1009, Beijing, China, 2013. Morgan Kaufmann.

[Zalta, 1988] Edward N Zalta. *Intensional Logic and the Metaphysics of Intentionality*. MIT Press, Cambridge, MA, 1988.

Learning Possibilistic Logic Theories from Default Rules (Abridged Version)

Ondřej Kuželka
Cardiff University, UK
KuzelkaO@cardiff.ac.uk

Jesse Davis
KU Leuven, Belgium
jesse.davis@cs.kuleuven.be

Steven Schockaert
Cardiff University, UK
SchockaertS1@cardiff.ac.uk

Abstract

We introduce a setting for learning possibilistic logic theories from defaults of the form "if alpha then typically beta". An important property of our approach is that it is inherently able to handle noisy and conflicting sets of defaults. Among others, this allows us to learn possibilistic logic theories from crowdsourced data and to approximate propositional Markov logic networks using heuristic MAP solvers. *This short paper is an abridged version of [Kuželka* et al.*, 2016].*

1 Introduction

Structured information plays an increasingly important role in applications such as information extraction, question answering and robotics. With the notable exceptions of CYC and WordNet, most of the knowledge bases that are used in such applications have at least partially been obtained using some form of crowdsourcing (e.g. Freebase, Wikidata, ConceptNet). To date, such knowledge bases are mostly limited to facts (e.g. Trump is the current president of the US) and simple taxonomic relationships (e.g. every president is a human). One of the main barriers to crowdsourcing more complex domain theories is that most users are not trained in logic. This is exacerbated by the fact that often (commonsense) domain knowledge is easiest to formalize as defaults (e.g. birds typically fly), and, even for non-monotonic reasoning (NMR) experts, it can be challenging to formulate sets of default rules without introducing inconsistencies (w.r.t. a given NMR semantics) or unintended consequences.

In this paper, we propose a method for learning consistent domain theories from crowdsourced examples of defaults and non-defaults. Since these examples are provided by different users, who may only have an intuitive understanding of the semantics of defaults, together they will typically be inconsistent. The problem we consider is to construct a set of defaults which is consistent w.r.t. the System P semantics [Kraus *et al.*, 1990], and which entails as many of the given defaults and as few of the non-defaults as possible. Taking advantage of the relation between System P and possibilistic logic [Benferhat *et al.*, 1997], we treat this as a learning problem, in which we need to select and stratify a set of propositional formulas.

2 Background: Possibilistic logic

A stratification of a propositional theory $\mathcal{T}$ is an ordered partition of the set of formulas in $\mathcal{T}$. A theory in possibilistic logic [Dubois *et al.*, 1994] is a set of formulas of the form (α, λ), with α a propositional formula and $\lambda \in]0, 1]$ a certainty weight. These certainty weights are interpreted in a purely ordinal fashion, hence a possibilistic logic theory is essentially a stratification of a propositional theory. The strict λ-cut $\Theta_{\overline{\lambda}}$ of a possibilistic logic theory Θ is defined as $\Theta_{\overline{\lambda}} = \{\alpha \,|\, (\alpha, \mu) \in \Theta, \mu > \lambda\}$. The inconsistency level $inc(\Theta)$ of Θ is the lowest certainty level λ in $[0, 1]$ for which the classical theory $\Theta_{\overline{\lambda}}$ is consistent. An inconsistency-tolerant inference relation $\vdash_{poss}$ for possibilistic logic can then be defined as follows: $\Theta \vdash_{poss} \alpha$ iff $\Theta_{\overline{inc(\Theta)}} \models \alpha$. We will write $(\Theta, \alpha) \vdash_{poss} \beta$ as an abbreviation for $\Theta \cup \{(\alpha, 1)\} \vdash_{poss} \beta$. It can be shown that $\Theta \vdash_{poss} (\alpha, \lambda)$ can be decided by making $O(\log_2 k)$ calls to a SAT solver, with k the number of certainly levels in Θ [Lang, 2001].

3 Learning from Default Rules

In this section, we formally describe a new learning setting for possibilistic logic called *learning from default rules*. We assume a finite alphabet Σ is given. An example is a default rule over Σ and a hypothesis is a possibilistic logic theory over Σ. A hypothesis h predicts the class of an example $e = \alpha \,|\!\sim \beta$ by checking if h covers e, in the following sense.

Definition 1 (Covering)**.** *A hypothesis $h \in \mathcal{H}$* covers *an example $e = \alpha \,|\!\sim \beta$ if $(h, \alpha) \vdash_{poss} \beta$.*

The hypothesis h predicts positive, i.e. $h(\alpha \,|\!\sim \beta) = 1$, iff h covers e, and else predicts negative, i.e. $h(\alpha \,|\!\sim \beta) = -1$.

Example 1. *Let us consider the following set of examples*

$$\mathcal{S} = \{(bird \wedge antarctic \,|\!\sim \neg flies, 1), (bird \,|\!\sim \neg flies, -1)\}$$

The following hypotheses over the alphabet {bird,flies, antarctic} cover all positive and no negative examples:

$$h_1 = \{(bird, 1), (antarctic \rightarrow \neg flies, 1)\}$$
$$h_2 = \{(flies, 0.5), (antarctic \rightarrow \neg flies, 1)\}$$
$$h_3 = \{(antarctic \rightarrow \neg flies, 1)\}$$

The learning task can be formally described as follows: **Given:** *A multi-set $\mathcal{S}$ which is an i.i.d. sample from a set*

of default rules over a given finite alphabet Σ. **Do:** *Learn a possiblistic logic theory that covers all positive examples and none of the negative examples in* $\mathcal{S}$. This definition assumes that $\mathcal{S}$ is perfectly separable, i.e. it is possible to perfectly distinguish positive examples from negative examples. In practice, we often relax this requirement, and instead aim to find a theory that minimizes the training set error. Similar to learning in graphical models, this learning task can be decomposed into *parameter learning* and *structure learning*. In our context, the goal of parameter learning is to convert a set of propositional formulas into a possibilistic logic theory, while the goal of structure learning is to decide what that set of propositional formulas should be.

Example 2. *Let* $\mathcal{S} = \{(\textit{penguin} \mid\sim \textit{bird}, 1), (\textit{bird} \mid\sim \textit{flies}, 1), (\textit{penguin} \mid\sim \neg\textit{flies}, 1), (\mid\sim \textit{bird}, -1), (\textit{bird} \mid\sim \textit{penguin}, -1)\}$ *and* $\mathcal{T} = \{\textit{bird}, \textit{flies}, \textit{penguin}, \neg\textit{penguin} \vee \neg\textit{flies}\}$. *A stratification of* $\mathcal{T}$ *which minimizes the training error on the examples from* $\mathcal{S}$ *is* $\mathcal{T}^* = \{(\textit{bird}, 0.25), (\textit{penguin}, 0.25), (\textit{flies}, 0.5), (\neg\textit{penguin} \vee \neg\textit{flies}, 1)\}$ *which is equivalent to* $\mathcal{T}^{**} = \{(\textit{flies}, 0.5), (\neg\textit{penguin} \vee \neg\textit{flies}, 1)\}$ *because* $\textit{inc}(\mathcal{T}^*) = 0.25$. *Note that* $\mathcal{T}^{**}$ *correctly classifies all examples except* $(\textit{penguin} \mid\sim \textit{bird}, 1)$.

Given a set of examples $\mathcal{S}$, we write $\mathcal{S}^+ = \{\alpha | (\alpha, 1) \in \mathcal{S}\}$ and $\mathcal{S}^- = \{\alpha | (\alpha, -1) \in \mathcal{S}\}$). A stratification $\mathcal{T}^*$ of a theory $\mathcal{T}$ is a *separating stratification* of $\mathcal{S}^+$ and $\mathcal{S}^-$ if it covers all examples from $\mathcal{S}^+$ and no examples from $\mathcal{S}^-$. Because arbitrary stratifications can be chosen, there is substantial freedom to ensure that negative examples are not covered[1]. Unfortunately, the problem of finding a separating stratification is computationally hard.

Theorem 1. *Deciding whether a separating stratification exists for given* $\mathcal{T}$, $\mathcal{S}^+$ *and* $\mathcal{S}^-$ *is a* Σ_2^P*-complete problem.*

Another important parameter besides computational complexity is *sample complexity* which can be determined using the Vapnik-Chervonenkis (VC) dimension. Hence, we also determine the VC of the set of possible stratifications of a propositional theory. Let us write $\textit{Strat}(\mathcal{T})$ for the set of all stratifications of a theory $\mathcal{T}$, and let $\textit{Strat}^{(k)}(\mathcal{T})$ be the set of all stratifications with at most k levels. The following proposition provides an upper bound for the VC dimension and can be proved by bounding the cardinality of $\textit{Strat}^{(k)}(\mathcal{T})$.

Theorem 2. *Let* $\mathcal{T}$ *be a set of* n *propositional formulas. Then* $VC(\textit{Strat}^{(k)}(\mathcal{T})) \leq n \log_2 k$.

The next theorem establishes a lower bound on the VC dimension of stratifications with at most k levels.

Theorem 3. *For every* k, n, $k \leq n$, *there is a propositional theory* $\mathcal{T}$ *consisting of* n *formulas such that*

$$VC(\textit{Strat}^{(k)}(\mathcal{T})) \geq \frac{1}{4} n(\log_2 k - 1).$$

4 Experiments with a Heuristic Algorithm

We implemented a heuristic algorithm, which combined structure learning and parameter learning, and we evaluated it in two different applications: learning of domain theories from crowdsourced default rules and approximating MAP inference in propositional Markov logic networks. As we are not aware of any existing methods that can learn a consistent logical theory from a set of noisy defaults, there are no baseline methods to which our method can directly be compared. However, if we fix a target literal l, we can train standard classifiers to predict for each propositional context α whether the default $\alpha \mid\sim l$ holds. This can only be done consistently with "parallel" rules, where the literals in the consequent do not appear in antecedents. We thus compared our method to three traditional classifiers on two crowdsourced datasets of parallel rules. In the second experiment, approximating MAP inference, we did not restrict ourselves to parallel rules. In this case, only our method can guarantee that the predicted defaults will be consistent. This would also be the case if we did not ask the crowdsourcers only about "parallel" rules. The experimental results and details of the methodology are described in the full version of this paper [Kuželka *et al.*, 2016].

[1]Note that, as we show in [Kuželka *et al.*, 2016], to decide whether a separating stratification exists it is not sufficient to compute the Z-ranking because of the presence of negative examples.

5 Future Work

There are several important directions for future work. Although our implementation is capable of working with tens of thousands of defaults, it still does not scale to datasets of the sizes of knowledge bases such as FreeBase. Scalability is therefore an important issue. Also while possibilistic logic is a natural choice for representing the learned default rule theory, the framework of learning from default rules, which we introduced here, could as well work with other representations of default rules which might be more suitable for certain domains. On the applications side, it would be interesting to apply the method to learning from symtoms and diagnoses/treatments.

Acknowledgments This work has been supported by a grant from the Leverhulme Trust (RPG-2014-164). Jesse Davis is partially supported by the KU Leuven Research Fund (C22/15/015), and FWO-Vlaanderen (G.0356.12, SBO-150033).

References

[Benferhat *et al.*, 1997] S. Benferhat, D. Dubois, and H. Prade. Nonmonotonic reasoning, conditional objects and possibility theory. *Artificial Intelligence*, 92(1-2):259–276, 1997.

[Dubois *et al.*, 1994] D. Dubois, J. Lang, and H. Prade. Possibilistic logic. In D. Nute D. Gabbay, C. Hogger J. Robinson, editor, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3, pages 439–513. Oxford University Press, 1994.

[Kraus *et al.*, 1990] S. Kraus, D. Lehmann, and M. Magidor. Nonmonotonic reasoning, preferential models and cumulative logics. *Artif. Intelligence*, 44(1-2):167–207, 1990.

[Kuželka *et al.*, 2016] Ondřej Kuželka, Jesse Davis, and Steven Schockaert. Learning Possibilistic Logic Theories from Default Rules. In *Proceedings of IJCAI*, 2016.

[Lang, 2001] J. Lang. Possibilistic logic: complexity and algorithms. In J. Kohlas and S. Moral, editors, *Algorithms for Uncertainty and Defeasible Reasoning*, volume 5 of *Handbook of Defeasible Reasoning and Uncertainty Management Systems (Gabbay D., Smets P. Eds.)*, pages 179–220. Kluwer, 2001.

An overview: Axiomatizations of probabilities with non-standard ranges*

Zoran Ognjanović, Miodrag Rašković, Zoran Marković, Angelina Ilic-Stepić,
Mathematical Institute of the Serbian Academy of Sciences and Arts
Nebojša Ikodinović, Aleksandar Perović, Dragan Doder,
University of Belgrade
ikodinovic@matf.bg.ac.rs

1 Introduction

The aim of probabilistic logics is to capture rules of reasoning about uncertain knowledge. In [Ognjanović and Rašković, 1999; 2000; Ognjanović *et al.*, 2009; 2016] we presented axiomatizations and proofs of completeness and decidability wrt. the standard real-valued probability functions. However, real-valued probabilities are not always adequate to model different types of uncertainty, as it is the case in default reasoning. In this paper we describe some logics related to probabilities with non-standard ranges:

- the finite set $\{0, \frac{1}{n}, \frac{2}{n}, \ldots, \frac{n-1}{n}, 1\}$,
- the unit interval of rational numbers $[0,1]_{\mathbb{Q}}$ or some other recursive subsets of $[0,1]$,
- the unit interval of Hardy field $[0,1]_{\mathbb{Q}(\varepsilon)}$,
- some partially ordered countable commutative monoid with the least element, e.g., $[0,1]_{\mathbb{Q}} \times [0,1]_{\mathbb{Q}}$,
- a closed ball in the field $\mathbb{Q}_p$ of p-adic numbers, and
- the set of complex numbers $\mathbb{C}$.

These different types of ranges impose challenges in axiomatizations and we provide appropriate techniques to resolve those issues.

In this overview we will consider logics without iterations of probability operators, but this restriction is not essential and in a similar way we can work with higher-order probabilities. So, as the basic logic we will consider the probability logic denoted LPP_2 which enriches propositional calculus with probabilistic operators of the form $P_{\geq s}$ with the intended meaning "probability is at least s", and the corresponding possible-world semantics with a finitely additive probability measure on sets of worlds. An LPP_2-model is a tuple $\mathbf{M} = \langle W, H, \mu, v\rangle$ where:

- W is a nonempty set of objects called worlds,
- H is an algebra of subsets of W,
- μ is a finitely additive probability measure, $\mu : H \rightarrow [0,1]$, and
- v provides for each world $w \in W$ a classical valuation,

and satisfiability is defined such that:

- $\mathbf{M} \models P_{\geq s}\alpha$ iff $\mu(\{w : v(w)(\alpha) = \top\}) \geq s$.

Note that compactness does not hold for LPP_2, e.g., the set of formulas $\{\neg P_{=0}p\} \cup \{P_{<1/n}p : n \in \mathbb{N}\}$ is finitely satisfiable, but not satisfiable. Usually, similar examples can be given for other probabilistic logics, and in these cases we use infinitary rules, as for example:

- From $A \rightarrow P_{\geq s-\frac{1}{k}}\alpha$, for every integer $k \geq \frac{1}{s}$, and $s > 0$, infer $A \rightarrow P_{\geq s}\alpha$,

for providing strongly complete axiomatizations for the studied systems. All considered logics are decidable.

2 Finite ranges

Let n be a fixed positive integer, and $\mathrm{Range} = \{0, 1/n, \ldots, (n-1)/n, 1\}$ be the range of probability functions. If $s \in [0,1)$, then s^+ denotes $\min\{r \in \mathrm{Range} : s < r\}$. Since the range of probabilities is fixed and finite, the following is a characteristic valid formula:

- $P_{>s}\alpha \rightarrow P_{\geq s^+}\alpha$.

Now compactness holds, and using this formula as an axiom, it is possible to give a finite strongly complete axiomatization that formalize reasoning about probabilities with the range Range [Ognjanović *et al.*, 2009].

The paper [Djordjević *et al.*, 2004] proves the completeness theorem wrt. to the class of all probabilistic models whose measures have arbitrary finite ranges (without the requirement that the range is fixed in advance).

3 Recursive ranges

Let Range be the unit interval of a recursive field. An infinitary rule suitable for obtaining strong completeness for reasoning about probabilities with the range Range is

- From $A \rightarrow P_{\neq s}\alpha$, for every $s \in \mathrm{Range}$, infer $A \rightarrow \bot$.

Note that this rule allows us to syntactically determine the range of probabilities.

4 Approximate conditional probabilities

A useful example of recursive fields mentioned in the previous section is Hardy field which is a recursive non-Archimedean field. It contains all rational functions of a fixed positive infinitesimal ε (i.e., $|\varepsilon| < 1/n$, $n \in \mathbb{N}$).

*Supported by Minister of Education, Science and Technological Development of the Republic of Serbia, through Matematički institut SANU.

In this case, we use also the conditional probability operators: $CP_{\leq s}(\alpha, \beta)$, $CP_{\geq s}(\alpha, \beta)$, and $CP_{\approx s}(\alpha, \beta)$ with the intended meaning the conditional probability of α given β is at most s, at least s, and approximately s, respectively.

We can use $CP_{\approx 1}(\alpha, \beta)$ to model the default "if β, then generally α". It is shown [Rašković *et al.*, 2008]:

- If we restrict our logic, and consider the language of defaults and finite default bases, the entailment coincides with the one in the system P.
- If we consider the language of defaults and arbitrary default bases, more conclusions can be obtained in our system than in the system P.
- When we consider our full language, we can express probabilities of formulas, negations of defaults, combinations of defaults and other formulas etc.

5 Logics with unordered or partially ordered ranges

It is possible (as Keynes proposed) to consider probabilities than cannot always be compared, in which case the range of probability functions should be partially ordered, as is the case with lattices with the additional underlying structure (e.g., $[0,1]_{\mathbb{Q}} \times [0,1]_{\mathbb{Q}}$ with the product (coordinatewise) order: $(a_1, b_1) < (a_2, b_2)$ iff $a_1 < a_2$, and $b_1 < b_2$). Probability functions with such ranges naturally arose in various phenomena involving quantum physics, incomparability, and indeterminacy. Theoretical background can be found in so called vector valued measure theory. Logics developed for such probability functions are described in [Ikodinović *et al.*, 2013].

6 p-adic numbers and complex numbers

The interest in p-adic numbers has extended far beyond the first applications in number theory, to theory of distributions, differential and pseudodifferential equations, spectral theory and p-adic probability theory (mainly developed by Andrei Khrennikov). Khrennikov's goal was to provide a firm mathematical background for certain peculiarities in quantum physics such as negative probabilities that arose in Wigners distribution on the phase space and Diracs distributions (relativistic quantization). To formalize Khrennikov's measure theoretic approach, we note that, since $\mathbb{Q}_p$ is not an ordered field, we cannot use the standard probability operators ($P_{\geq s}$), and as a suitable choice we introduce the operators of the form $K_{r,\rho}\alpha$, with the intended meaning "the probability of α is in the ball $K[r, \rho] = \{a \in \mathbf{Q}_p \ : \ |r - a|_p \leqslant \rho\}$". For the resulting logics we can prove strong completeness and decidability [Ilic-Stepić *et al.*, 2012; Ilić-Stepić *et al.*, 2014; Ilić-Stepić and Ognjanović, 2015].

Complex valued probabilities have also proven to be useful in applications, for example it is possible to consider relativistic quantum mechanics based on complex probability theory. In this approach, a wave function is not treated as the state of the system", but represents the best estimate of the complex probability of finding particle at some point in a measure space. It says what is known about the system, and the collapse of the wave function represents learning a new fact about the system and therefore leads to calculation of new complex probabilities. Similarly as in the p-adic case, it is possible to use complex balls and/or squares to estimate probabilities of events [Ilic-Stepić and Ognjanović, 2014].

References

[Djordjević *et al.*, 2004] Radosav Djordjević, Miodrag Rašković, and Zoran Ognjanović. Completeness theorem for propositional probabilistic models whose measures have only finite ranges. *Archive for Mathematical Logic*, 43:557 – 563, 2004.

[Ikodinović *et al.*, 2013] Nebojša Ikodinović, Miodrag Rašković, Zoran Marković, and Zoran Ognjanović. Logics with generalized measure operators. *Journal of Multiple-Valued Logic and Soft Computing*, 20(5-6):527–555, 2013.

[Ilic-Stepić and Ognjanović, 2014] Angelina Ilic-Stepić and Zoran Ognjanović. Complex valued probability logics. *Publications de l'Institut Mathematique*, n.s. tome 95 (109):73–86, 2014.

[Ilić-Stepić and Ognjanović, 2015] Angelina Ilić-Stepić and Zoran Ognjanović. Logics for reasoning about processes of thinking with information coded by p-adic numbers. *Studia Logica*, 103:145–174, 2015.

[Ilic-Stepić *et al.*, 2012] Angelina Ilic-Stepić, Zoran Ognjanović, Nebojša Ikodinović, and Aleksandar Perović. A p-adic probability logic. *Mathematical Logic Quarterly*, 58(4-5):263–280, 2012.

[Ilić-Stepić *et al.*, 2014] Angelina Ilić-Stepić, Zoran Ognjanović, and Nebojša Ikodinović. Conditional p-adic probability logic. *International Journal of Approximate Reasoning*, 55(9):1843–1865, 2014.

[Ognjanović and Rašković, 1999] Zoran Ognjanović and Miodrag Rašković. Some probability logics with new types of probability operators. *Journal of Logic and Computation*, 9(2):181 – 195, 1999.

[Ognjanović and Rašković, 2000] Zoran Ognjanović and Miodrag Rašković. Some first-order probability logics. *Theoretical Computer Science*, 247(1-2):191 – 212, 2000.

[Ognjanović *et al.*, 2009] Zoran Ognjanović, Miodrag Rašković, and Zoran Marković. Probability logics. In *Zbornik radova, subseries Logic in computer science, 12 (20)*, pages 35–111. Matematički institut SANU, 2009.

[Ognjanović *et al.*, 2016] Zoran Ognjanović, Miodrag Rašković, and Zoran Marković. *Probability Logics: Probability-Based Formalization of Uncertain Reasoning*. Springer, 2016.

[Rašković *et al.*, 2008] Miodrag Rašković, Zoran Marković, and Zoran Ognjanović. A logic with approximate conditional probabilities that can model default reasoning. *International Journal of Approximate Reasoning*, 49(1):52–66, 2008.

Towards Argumentation-based Classification

Matthias Thimm
Universität Koblenz-Landau
Germany

Kristian Kersting
Technische Universität Darmstadt
Germany

1 Introduction

Classification is the problem of categorizing new observations by using a classifier learnt from already categorized examples. In general, the area of *machine learning* [Mitchell, 1997] has brought forth a series of different approaches to deal with this problem, from decision trees to support vector machines and others. Recently, approaches to *statistical relational learning* [De Raedt *et al.*, 2016] even take the perspective of knowledge representation and reasoning into account by developing models on more formal logical and statistical grounds. In this position paper, we envisage to significantly generalize this reasoning aspect of machine learning towards the use of *computational models of argumentation* [Baroni *et al.*, 2011], a popular approach to commonsense reasoning, for reasoning within machine learning. More concretely, we consider the following two-step classification approach. In the first step, rule learning algorithms are used to extract frequent patterns and rules from a given data set. The output of this step comprises a huge number of rules (given fairly low confidence and support parameters) and these cannot directly be used for the purpose of classification as they are usually inconsistent with one another. Therefore, in the second step, we interpret these rules as the input for approaches to structured argumentation, such as ASPIC$^+$ [Modgil and Prakken, 2014] or DeLP [Garcia and Simari, 2004]. Using the argumentative inference procedures of these approaches and given a new observation, the classification of the new observation is determined by constructing arguments on top of these rules for the different classes and determining their justification status.

The use of argumentation techniques allows to obtain classifiers, which are by design able to *explain* their decisions, and therefore addresses the recent need for *Explainable AI*: classifications are accompanied by a dialectical analysis showing why arguments for the conclusion are preferred to counterarguments. Argumentation techniques in machine learning also allows the easy integration of additional expert knowledge in form of arguments.

While there are some previous works considering the combination of machine learning and computational argumentation techniques—see e. g. [Možina *et al.*, 2008; Riveret and Governatori, 2016]—, the proposed two-step process offers a novel perspective on this combination, which is likely to bring new insights on the general relationship between machine learning and knowledge representation and reasoning. Preliminary experiments already suggest that our framework can yield performance comparable to state-of-the-art, while being explainable.

2 Proposed approach

We illustrate the goals of our envisioned approach using a classical example for a (multi-class) classification problem, the "Animals with Attributes" data set[1] (we only consider the base package with the class/attribute table). This dataset describes 50 animals, e. g. ox, mouse, dolphin, using 85 binary attributes such as "swims", "black", and "arctic". Using a first-order logic representation this data can be represented as a set of ground literals such as

$$swims(dolphin), \neg black(dolphin), \neg arctic(dolphin), \ldots$$

Now given the truth values of some attributes of a new animal, say a kangaroo, the classification tasks consists of predicting the values of the remaining attributes, e. g. given the fact that a kangaroo is orange and that it hops, does it live in the arctic? We address this task by first applying *association rule mining* such as the well-know Apriori algorithm [Agrawal and Srikant, 1994]. The output is a set of association rules such as "animals with flippers usually live in the ocean" which can be modeled as

$$flippers(X) \rightarrow ocean(X)$$

As the rules are mined based on frequent patterns and ignore logical coherency, they may be contradictory to each in other in certain cases. For example, another mined rule could be

$$big(X) \rightarrow \neg ocean(X)$$

saying that big animals usually do not live in the ocean. However, as a dolphin both has flippers and is big, the above two rules would therefore result in a contradiction and no meaningful classification could be given in this case. It is not surprising that rule mining algorithms are rarely used for classification purposes in this manner. We, however, add another second step to our classification approach by taking the output of the rule mining algorithm, i. e., a set of rules, as the input of an approach to structured argumentation such as ASPIC$^+$ [Modgil and Prakken, 2014] or DeLP [Garcia and Simari, 2004]. In these approaches, rules are not just applied in a

[1] http://attributes.kyb.tuebingen.mpg.de

direct fashion but arguments are build for all alternative conclusions and compared through e. g. a dialectical procedure in order to determine a consistent set of conclusions. Assume that another rule mined in the first step is

$$big(X), blue(X) \rightarrow ocean(X)$$

meaning that big and blue animals do indeed live in the ocean. Using *specificity* [Stolzenburg *et al.*, 2003] as a comparison criterion between conflicting arguments the conflict can be resolved because this final argument defeats the less specific second argument. We call this general approach *Argumentation-based Classification* (AbC). It is customizable by employing different rule mining algorithms in the first step and different approaches to structured argumentation in the second step. Moreover, besides using classical (qualitative) approaches to structured argumentation in the second step we can also make use of argumentative approaches incorporating quantitative uncertainty such as [Rienstra, 2012; Alsinet *et al.*, 2008]. By doing so, we can make use of additional quantitative information of the rules mined in the first step. For example, the confidence value of a rule can be interpreted as a conditional probability, i. e., the ratio of the probability of the conjunction of the head and body of the rule over only the body of the rule. This information can be used during the argumentation process in order to make more accurate predictions.

Making use of argumentation in classification allows the user to also inspect the reasoning process of why a certain prediction has been made, i. e., the resulting argumentative classification approaches are explainable by design. Formalisms such as DeLP conduct a dialectical analysis where all arguments contributing to the matter of deciding whether a certain statement is true. This analysis can be shown to the user in order to explain why a certain decision has been made. For example, above we would get the explanation "A dolphin lives in the ocean because it is blue, despite the fact that it is big". Users can then evaluate this reasoning and, if they are not satisfied with the explanation, pose a new argument for a different conclusion.

3 Preliminary results and conclusion

In order to assess the feasibility of our envisaged approach, we already implemented a first version of Argumentation-based Classification using the standard Apriori algorithm [Agrawal and Srikant, 1994] for rule mining and DeLP [Garcia and Simari, 2004] as the structured argumentation approach. We applied the rule miner to the "Animals with Attributes" data set with minimum confidence 0.9 and minimum support 0.8. We only mined rules with up to 3 elements in the body and 1 element in the conclusion. All rules with confidence value 1 were interpreted as strict rules, the remaining rules were interpreted as defeasible rules. This resulted in 254 strict and 621 defeasible rules. To these rules we added all but one randomly chosen attribute fact of some randomly chosen animal and asked DeLP whether the remaining attribute is warranted (note that DeLP has a three-valued answering behaviour: yes/no/undecided). We repeated this experiment 1000 times. While in about 70% of the times, DeLP could not classify the attribute (answer "undecided") it never misclassified any attribute and therefore classified 30% correctly, e. g., it never answered "no" when the correct answer was "yes". However, slightly changing the parameters of the experiment (such as minimum support and minimum confidence) would increase and decrease these values, while still not misclassifying any attribute. Note that using the mined rules directly as a classifier results in inconsistent classifications most of the time. We found these initial results encouraging and it is likely that a more careful setup, analysis, and evaluation will improve them significantly.

References

[Agrawal and Srikant, 1994] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings VLDB'94*, pages 487–499, 1994.

[Alsinet *et al.*, 2008] T. Alsinet, C. I. Chesñevar, L. Godo, and G. R. Simari. A logic programming framework for possibilistic argumentation: Formalization and logical properties. *Fuzzy Sets and Systems*, 159(10):1208–1228, 2008.

[Baroni *et al.*, 2011] P. Baroni, M. Caminada, and M. Giacomin. An Introduction to Argumentation Semantics. *The Knowledge Engineering Review*, 26(4):365–410, 2011.

[De Raedt *et al.*, 2016] L. De Raedt, K. Kersting, S. Natarajan, and D. Poole. *Statistical Relational Artificial Intelligence: Logic, Probability, and Computation*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2016.

[Garcia and Simari, 2004] A. Garcia and Guillermo R. Simari. Defeasible Logic Programming: An Argumentative Approach. *Theory and Practice of Logic Programming*, 4(1–2):95–138, 2004.

[Mitchell, 1997] Tom Mitchell. *Machine Learning*. McGraw-Hill Education, 1997.

[Modgil and Prakken, 2014] S. Modgil and H. Prakken. The ASPIC+ framework for structured argumentation: a tutorial. *Argument and Computation*, 5:31–62, 2014.

[Možina *et al.*, 2008] M. Možina, M. Guid, J. Krivec, A. Sadikov, and I. Bratko. Fighting knowledge acquisition bottleneck with argument based machine learning. In *Proceedings ECAI'08*, 2008.

[Odom and Natarajan, 2016] P. Odom and S. Natarajan. Actively interacting with experts: A probabilistic logic approach. In *Proceedings ECML PKDD 2016*, part II pages 527–542, 2016.

[Rienstra, 2012] T. Rienstra. Towards a probabilistic Dung-style argumentation system. In *Proceedings AT2012*, 2012.

[Riveret and Governatori, 2016] R. Riveret and G. Governatori. On learning attacks in probabilistic abstract argumentation. In *Proceedings AAMAS'16*, 2016.

[Stolzenburg *et al.*, 2003] F. Stolzenburg, A. Garcia, C. I. Chesnevar, and G. R. Simari. Computing generalized specificity. *Journal of Non-Classical Logics*, 13(1):87–113, 2003.

From First-Order Logic to Assertional Logic

Yi Zhou
School of Computing, Engineering and Mathematics
Western Sydney University, Australia

Abstract

First-Order Logic (FOL) is widely regarded as one of the most important foundations for knowledge representation. Nevertheless, in this paper, we argue that FOL has several critical issues for this purpose. Instead, we propose an alternative called assertional logic, in which all syntactic objects are categorized as set theoretic constructs including individuals, concepts and operators, and all kinds of knowledge are formalized by equality assertions. We first present a primitive form of assertional logic that uses minimal assumed knowledge and constructs. Then, we show how to extend it by definitions, which are special kinds of knowledge, i.e., assertions. We argue that assertional logic, although simpler, is more expressive and extensible than FOL. As a case study, we show how assertional logic can be used to unify logic and probability, and more building blocks in AI.

1 Introduction

Classical First-Order Logic (FOL) is widely regarded as one of the most important foundations of symbolic AI. FOL plays a central role in the field of Knowledge Representation and Reasoning (KR). Many of its fragments (such as propositional logic, modal and epistemic logic, description logics), extensions (such as second-order logic, situation calculus and first-order probabilistic logic) and variants (such as Datalog and first-order answer set programming) have been extensively studied in the literature [Brachman and Levesque, 2004; van Harmelen *et al.*, 2008].

Nevertheless, AI researchers have pointed out several issues regarding using FOL for knowledge representation and reasoning, mostly from the reasoning point of view. First, FOL is computationally very difficult. Reasoning about FOL is a well-known undecidable problem. Also, FOL is monotonic in the sense that adding new knowledge into a first-order knowledge base always results in more consequences. However, human reasoning is sometimes nonmonotonic.

In this paper, we argue that FOL also has some critical issues from the knowledge representation point of view. First of all, although FOL is considered natural for well-trained logicians, it is not simple and flexible enough for knowledge engineers with less training. One reason is the distinction and hierarchy between term level, predicate level and formula level. From my own experience as a teacher in this subject, although strongly emphasized in the classes, many students failed to understand why a predicate or a formula cannot be in the scope of a function. Another reason is the notion of free occurrences of variables. For instance, it is not easily understandable for many students why the GEN inference rule has to enforce the variable occurrence restrictions. Last but not least, arbitrary nesting also raises issues. Although natural from a mathematical point of view, a nested formula, e.g., $(x \vee \neg(y \wedge z)) \wedge (\neg y \vee \neg x)$ is hard to be understood and used.

Secondly, FOL has limitations in terms of expressive power. FOL cannot quantify over predicates/functions. This can be addressed by extending FOL into high-order logic. Nevertheless, high-order logic still cannot quantify over formulas. As a consequence, FOL and high-order logic are not able to represent an axiom or an inference rule in logic, such as *Modus Ponens*. Flexible quantification beyond the term level is needed in applications. As an example, in automated solving mathematical problems, we often use proof by induction. To represent this, we need to state that for some statement P with a number parameter, if that P holds for all numbers less than k implies that P holds for the number k as well, then P holds for all natural numbers. Here, P is a statement at a formula level, possibly with complex sub-statements within itself. Hence, in order to represent and use proof by induction, we need to quantify over P that is at a formula level.

Thirdly, FOL is hard to be extended with new building blocks. FOL itself cannot formalize some important AI notions including probability, actions, time etc., which are needed in a wide range of applications. For this purpose, AI researchers have made significant progresses on extending FOL with these notions separately, such as first-order probabilistic logic [Bacchus, 1990; Halpern, 1990], situation calculus [Levesque *et al.*, 1991; Lin, 2008], CTL [Clarke and Emerson, 1982] etc. Each is a challenging task in the sense that it has to completely re-define the syntax as well as the semantics. However, combing these notions together, even several of them, seems an extremely difficult task. Moreover, there are many more building blocks to be incorporated in applications. For instance, consider task planning for home service robots [Keller *et al.*, 2010]. It is necessary to represent actions, probability, time and more building blocks such

as preferences altogether at the same time.

To address these issues, we propose assertional logic, in which all syntactic objects are categorized as set theoretic constructs including individuals, concepts and operators, and all kinds of knowledge are uniformly formalized by equality assertions of the form $a = b$, where a and b are either atomic individuals or compound individuals. Semantically, individuals, concepts and operators are interpreted as elements, sets and functions respectively in set theory, and knowledge of the form $a = b$ means that the two individuals a and b are referring to the same element.

We first present the primitive form of assertional logic that uses minimal assumed knowledge and primitive constructs. Then, we show how to extend it with more building blocks by definitions, which are special kinds of knowledge, i.e., assertions used to define new individuals, concepts and operators. Once these new syntactic objects are defined, they can be used as a basis to define more. As an example, we show how to define multi-assertions by using Cartesian product, and nested assertions by using multi-assertions.

We show that assertional logic, although simpler, is more expressive and extensible than FOL. As a case study, we show how to extend assertional logic for unifying logic and probability, and more important AI building blocks including time. Note that our intention is not to reinvent the wheel of these building blocks but to borrow existing excellent work on formalizing these building blocks separately and to assemble them within one framework (i.e., assertional logic) so that they can live happily ever after.

2 Meta Language and Prior Knowledge

One cannot build something from nothing. Hence, in order to establish assertional logic, we need some basic knowledge. Of course, for the purpose of explanation, we need an informal meta language whose syntax and semantics are pre-assumed. As usual, we use a natural language such as English. Nevertheless, this meta language is used merely for explanation and it should not affect the syntax as well as the semantics of anything defined formally.

Only a meta level explanation language is not enough. Other than this, we also need some core objects and knowledge, whose syntax and semantics are pre-assumed as well. These are called *prior objects* and *prior knowledge*. For instance, when defining real numbers, we need some prior knowledge about natural numbers; when defining probability, we need some prior knowledge about real numbers.

In assertional logic, we always treat the equality symbol "=" as a prior object. There are some prior knowledge associated with the equality symbol. For instance, "=" is an equivalence relation satisfying reflexivity, symmetricity, and transitivity. Also, "=" satisfies the general substitution property, that is, if $a = b$, then a can be used to replace b anywhere. Other than the equality symbol, we also assume some prior objects and their associated prior knowledge in set theory [Halmos, 1960], including set operators such as set union and Cartesian product, Boolean values, set builder notations and natural numbers.

3 Assertional Logic: the Primitive Form

In this section, we present the primitive form of assertional logic. As the goal of assertional logic is to syntactically represent knowledge in application domains, there are two essential tasks, i.e., how to capture the syntax of the domain and how to represent knowledge in it.

3.1 Capturing the syntax

Given an application domain, a *syntactic structure* (*structure* for short if clear from the context) of the domain is a triple $\langle \mathcal{I}, \mathcal{C}, \mathcal{O} \rangle$, where $\mathcal{I}$ is a collection of *individuals*, representing objects in the domain, $\mathcal{C}$ a collection of *concepts*, representing groups of objects sharing something in common and $\mathcal{O}$ a collection of *operators*, representing relationships and connections among individuals and concepts. Concepts and operators can be nested and considered as individuals as well. If needed, we can have concepts of concepts, concepts of operators, concepts of concepts of operators and so on.

An operator could be multi-ary, that is, it maps a tuple of individuals into a single one. Each multi-ary operator O is associated with a *domain* of the form $(C_1, \ldots, C_n)$, representing all possible values that the operator O can operate on, where $C_i, 1 \leq i \leq n$, is a concept. We call n the *arity* of O. For a tuple $(a_1, \ldots, a_n)$ matching the domain of an operator O, i.e., $a_i \in C_i, 1 \leq i \leq n$, O maps $(a_1, \ldots, a_n)$ into an individual, denoted by $O(a_1, \ldots, a_n)$.

Operators are similar to functions in FOL but differs in two essential ways. First, operators are many-sorted as $C_1, \ldots, C_n$ could be different concepts. More importantly, $C_1, \ldots, C_n$ could be high-order constructs, e.g., concepts of concepts, concepts of operators.

3.2 Representing knowledge

Let $\langle \mathcal{I}, \mathcal{C}, \mathcal{O} \rangle$ be a syntactic structure. A *term* is an individual, either an atomic individual $a \in \mathcal{I}$ or the result $O(a_1, \ldots, a_n)$ of an operator O operating on some individuals $a_1, \ldots, a_n$. We also call the latter *compound individuals*.

An *assertion* is of the form

$$a = b, \tag{1}$$

where a and b are two terms. Intuitively, an assertion of the form (1) is a piece of knowledge in the application domain, claiming that the left and the right side refer to the same object. A *knowledge base* is a set of assertions. Terms and assertions can be considered as individuals as well.

Similar to concepts that group individuals, we use schemas to group terms and assertions. A *schema term* is either an atomic concept $C \in \mathcal{C}$ or of the form $O(C_1, \ldots, C_n)$, where $C_i, 1 \leq i \leq n$ are concepts. Essentially, a schema term represents a set of terms, in which every concept is grounded by a corresponding individual. That is, $O(C_1, \ldots, C_n)$ is the collection $\{O(a_1, \ldots, a_n)\}$, where $a_i \in C_i, 1 \leq i \leq n$ are individuals. Then, a *schema assertion* is of the same form as form (1) except that terms can be replaced by schema terms. Similarly, a schema assertion represents a set of assertions.

We say that a schema term/assertion *mentions* a set $\{C_1, \ldots, C_n\}$ of concepts if $C_1, \ldots, C_n$ occur in it, and *only mentions* if $\{C_1, \ldots, C_n\}$ contains all concepts mentioned in

it. Note that it could be the case that two or more different individuals are referring to the same concept C in schema terms and assertions. In this case, we need to use different *copies* of C, denoted by $C^1, C^2, \ldots$, to distinguish them. For instance, all assertions $x = y$, where x and y are human, are captured by the schema assertion $Human^1 = Human^2$. On the other side, in a schema, the same copy of a concept C can only refer to the same individual. For instance, $Human = Human$ is the set of all assertions of the form $x = x$, where $x \in Human$.

3.3 The semantics

We propose a set theoretic semantics for assertional logic. Since we assume set theory as the prior knowledge, in the semantics, we freely use those individuals (e.g., the empty set), concepts (e.g., the set of all natural numbers) and operators (e.g., the set union operator) without explanation.

An *interpretation* (also called a *possible world*) is a pair $\langle \Delta, .^I \rangle$, where Δ is a domain of elements, and $.^I$ is a mapping function that admits all prior knowledge, and maps each individual into a domain element in Δ, each concept into a set in Δ and each n-ary operator into an n-ary function in Δ. The mapping function $.^I$ is generalized for terms by mapping $O(a_1, \ldots, a_n)$ to $O^I(a_1^I, \ldots, a_n^I)$. Similar to terms and assertions, interpretations can also be considered as individuals to be studied.

It is important to emphasize that an interpretation has to admit all prior knowledge. For instance, since we assume set theory, suppose that an interpretation maps two individuals x and y as the same element a in the domain, then the concepts $\{x\}$ and $\{y\}$ must be interpreted as $\{a\}$, and $x = y$ must be interpreted as $a = a$.

Let I be an interpretation and $a = b$ an assertion. We say that I is a *model* of $a = b$, denoted by $I \models a = b$ iff $.^I(a) = .^I(b)$, also written $a^I = b^I$. Let KB be a knowledge base. We say that I is a model of KB, denoted by $I \models KB$, iff I is a model of all assertions in KB. We say that an assertion A is a *property* of KB, denoted by $KB \models A$, iff all models of KB are also models of A. In particular, we say that an assertion A is a *tautology* iff it is modeled by all interpretations.

Since we assume set theory as our prior knowledge, we directly borrow some set theoretic constructs. For instance, we can use $\cup(C_1, C_2)$ (also written as $C_1 \cup C_2$) to denote a new concept that unions two concepts C_1 and C_2. Applying this to assertions, we can see that assertions of the primitive form (1) can indeed represent many important features in knowledge representation. For instance, the *membership assertion*, stating that an individual a is an instance of a concept C, is the following assertion $\in (a, C) = \top$ (also written as $a \in C$). The *containment assertion*, stating that a concept C_1 is contained by another concept C_2, is the following assertion $\subseteq (C_1, C_2) = \top$ (also written as $C_1 \subseteq C_2$). The *range declaration*, stating that the range of an operator O operating on some concepts $C_1, \ldots, C_n$ equals to another concept C, is the following assertion $O(C_1, \ldots, C_n) = C$.

4 Extensibility via Definitions

As argued in the introduction section, extensibility is a critical issue for knowledge representation. In assertional logic, we use *definitions* for this purpose. Definitions are (schema) assertions used to define new syntactic objects (including individuals, concepts and operators) based on existing ones. Once these new syntactic objects are defined, they can be used to define more. Note that definitions are nothing extra but special kinds of knowledge (i.e. assertions).

We start with defining new individuals. An individual definition is an assertion of the form

$$a = t, \tag{2}$$

where a is an atomic individual and t is a term. Here, a is the individual to be defined. This assertion claims that the left side a is defined as the right side t. For instance, $0 = \emptyset$ means that the individual 0 is defined as the empty set.

Defining new operators is similar to defining new individuals except that we use schema assertions instead. Let O be an operator to be defined and $(C_1, \ldots, C_n)$ its domain. An operator definition is a schema assertion of the form

$$O(C_1, \ldots, C_n) = T, \tag{3}$$

where T is a schema term that mentions concepts only from $C_1, \ldots, C_n$.

Since a schema assertion represents a set of assertions, essentially, an operator definition of the form (3) defines the operator O by defining the value of $O(a_1, \ldots, a_n)$ one-by-one, where $a_i \in C_i, 1 \leq i \leq n$. For instance, for defining the successor operator $Succ$, we can use the schema assertion $Succ(\mathbb{N}) = \{\mathbb{N}, \{\mathbb{N}\}\}$, meaning that, for every natural number n, the successor of n, is defined as $\{n, \{n\}\}$.

Defining new concepts is somewhat different. As concepts are essentially sets, we directly borrow set theory notations for this purpose. There are four ways to define a new concept.
Enumeration Let $a_1, \ldots, a_n$ be n individuals. Then, the collection $\{a_1, \ldots, a_n\}$ is a concept, written as

$$C = \{a_1, \ldots, a_n\}. \tag{4}$$

For instance, we can define the concept $Digits$ by $Digits = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.
Operation Let C_1 and C_2 be two concepts. Then, $C_1 \cup C_2$ (the union of C_1 and C_2), $C_1 \cap C_2$ (the intersection of C_1 and C_2), $C_1 \setminus C_2$ (the difference of C_1 and C_2), $C_1 \times C_2$ (the Cartesian product of C_1 and C_2), 2^{C_1} (the power set of C_1) are concepts. Operation can be written by assertions as well. For instance, the following assertion

$$C = C_1 \cup C_2 \tag{5}$$

states that the concept C is defined as the union of C_1 and C_2. As an example, one can define the concept Man by $Man = Human \cap Male$.
Restricted Comprehension Let C be a concept and $A(C)$ a schema assertion that only mentions concept C. Then, individuals in C satisfying A, denoted by $\{x \in C | A(x)\}$ (or simply $C|A(C)$), form a concept, written as

$$C' = C|A(C). \tag{6}$$

For instance, we can define the concept $Male$ by $Male = \{Animal \mid Sex(Animal) = male\}$, meaning that $Male$ consists of all animals whose sexes are male.

Replacement Let O be an operator and C a concept on which O is well defined. Then, the individuals mapped from C by O, denoted by $\{O(x) \mid x \in C\}$ (or simply $O(C)$), form a concept, written as

$$C' = O(C). \tag{7}$$

For instance, we can define the concept $Parents$ by $Parents = ParentOf(Human)$, meaning that it consists of all individuals who is a $ParentOf$ some human.

Definitions can be incremental. We may define some syntactic objects first. Once defined, they can be used to define more. One can always continue with this incremental process. For instance, in arithmetic, we define the successor operator first. Once defined, it can be used to define the add operator, which is further served as a basis to define more.

Since terms and assertions can be considered as individuals, we can define new type of terms and assertions by definitions. As an example, we extend assertions of the form (1) into multi-assertions by using Cartesian product. We first define multi-assertions of a fixed number of assertions. Given a number n, we define a new operator M_n with arity n by the following schema assertion:

$$M_n(C_1 = D_1, \ldots, C_n = D_n) = (C_1, \ldots, C_n) = (D_1, \ldots, D_n),$$

where $C_i, D_i, 1 \leq i \leq n$, are concepts of terms. Notice that, $(C_1, \ldots, C_n) = (D_1, \ldots, D_n)$ is a single assertion of the form (1). In this sense, an n-ary multi-assertion is just a syntax sugar. Then, we define the concept of multi-assertions:

$$Multi - Assertion = \bigcup_{1 \leq i \leq \infty} M_i(\mathcal{A}^1, \ldots, \mathcal{A}^i),$$

where $\mathcal{A}^1, \ldots, \mathcal{A}^i$ are i copies of standard assertions. For convenience, we use $Assertion_1, \ldots, Assertion_n$ to denote an n-ary multi-assertion. Once multi-assertion is defined, it can be used to define more syntactic objects.

As an example, we use multi-assertion to define nested assertions. We first define nested terms as follows:

$$\begin{aligned} Nested - Term &= Term \cup N - Term \\ N - Term &= Op(Nested - Term). \end{aligned}$$

Then, nested assertions can be defined as

$$Nested{-}Assertion = Nested{-}Term = Nested{-}Term.$$

Again, once nested assertion is defined, it can be used as basis to define more, so on and so forth. Using nested assertions can simplify the representation task. However, one cannot overuse it since, essentially, every use of a nested term introduces a new individual.

5 Embedding FOL into Assertional Logic

In the previous section, we show how to extend assertions of the primitive form (1) into multi-assertions and nested assertions. In this section, we continue with this task to show how to define more complex forms of assertions with logic connectives, including propositional connectives and quantifiers.

We start with the propositional case. Let $\mathcal{A}$ be the concept of nested assertions. We introduce a number of operators over $\mathcal{A}$ in assertional logic, including $\neg(\mathcal{A})$ (for *negation*), $\wedge(\mathcal{A}^1, \mathcal{A}^2)$ (for *conjunction*), $\vee(\mathcal{A}^1, \mathcal{A}^2)$ (for *disjunction*) and $\rightarrow (\mathcal{A}^1, \mathcal{A}^2)$ (for *implication*).

There could be different ways to define these operators in assertional logic. Let $a = a'$ and $b = b'$ be two (nested) assertions. The propositional connectives are defined as follows:

$$\begin{aligned} \neg(a = a') &= \{a\} \cap \{a'\} = \emptyset \\ \wedge(a = a', b = b') &= (\{a\} \cap \{a'\}) \cup (\{b\} \cap \{b'\}) = \{a, a', b, b'\} \\ \vee(a = a', b = b') &= (\{a\} \cap \{a'\}) \cup (\{b\} \cap \{b'\}) \neq \emptyset \\ \rightarrow (a = a', b = b') &= (\{a, a'\} \setminus \{a\} \cap \{a'\}) \cup (\{b\} \cap \{b'\}) \neq \emptyset, \end{aligned}$$

where $a \neq a'$ is used to also denote $\neg(a = a')$. One can observe that the ranges of all logic operators are nested assertions. Hence, similar to multi-assertion and nested assertion, propositional logic operators are syntactic sugar as well.

Now we consider to define operators for quantifiers, including $\forall$ (for the *universal* quantifier) and $\exists$ (for the *existential* quantifier). The domain of quantifiers is a pair $(C, A(C))$, where C is a concept and $A(C)$ is a schema assertion that only mentions C.

The quantifiers are defines as follows:

$$\forall(C, A(C)) = C|A(C) = C \tag{8}$$

$$\exists(C, A(C)) = C|A(C) \neq \emptyset \tag{9}$$

Intuitively, $\forall(C, A(C))$ is true iff those individuals x in C such that $A(x)$ holds equals to the concept C itself, that is, for all individuals x in C, $A(x)$ holds; $\exists(C, A(C))$ is true iff those individuals x in C such that $A(x)$ holds does not equal to the empty set, that is, there exists at least one individual x in C such that $A(x)$ holds. We can see that the ranges of quantifiers are nested assertions as well. In this sense, quantifiers are also syntactic sugar of the primitive form.

Note that quantifiers defined here are ranging from an arbitrary concept C. If C is a concept of all atomic individuals and all quantifiers range from the same concept C, then these quantifiers are first-order. Nevertheless, the concepts could be different. In this case, we have many-sorted FOL. Moreover, C could be complex concepts, e.g., a concept of all possible concepts. In this case, we have monadic second-order logic. Yet C could be many more, e.g., a concept of assertions, a concept of concepts of terms etc. In this sense, the quantifiers become high-order. Finally, the biggest difference is that C can even be a concept of assertions so that quantifiers in assertional logic can quantify over assertions (corresponding to formulas in classical logics), while this cannot be done in classical logics including high-order logic.

It can be verified that all tautologies in propositional logic and FOL (e.g., De-Morgan's laws) are also tautologies in assertional logic. For space reasons, we leave the theorems and their proofs to a full version of this paper.

6 Incorporating Probability and More

Probability is another important building block for knowledge representation. In the last several decades, with the development of uncertainty in artificial intelligence, a number of influential approaches [Bacchus, 1990; Gaifman, 1964; Hailperin, 1984; Milch, 2006; Pearl, 1988; Richardson and Domingos, 2006] have been developed, and important applications have been found in machine learning, natural language processing etc.

Normally, to incorporate probability in logic, one has to complete redefine the whole semantics since the integrations between probability and logic connectives and quantifiers are complicated. In this section, we show how this can be easily done in assertional logic. The key point is, although the interactions between logic and probability are complicated, their interactions with assertions of the basic form (1) is relatively simple. As shown in the previous section, the interactions between logic and assertions can be defined by a few lines. In this section, following Gaifman's idea [1964], we show that this is indeed the case for integrating assertions with probability as well. Then, the interactions between logic and probability will be automatically established via assertions.

6.1 Integrating assertions with probability

Since operations over real numbers are involved in defining probability, we need to assume a theory of real number as our prior knowledge.

Gaifman [1964] proposed to define the probability of a logic sentence by the sum of the probabilities of the possible worlds satisfying it. Following this idea, in assertional logic, we introduce an operator Pr (for probability) over the concept $\mathcal{A}$ of assertions. The range of Pr is the concept of real numbers. For each possible world w, we assign an associated weight W_w, which is a positive real number. Then, for an assertion A, the probability of A, denoted by $Pr(A)$, is define by the following schema assertion:

$$Pr(A) = \frac{\Sigma_{w,w\models A}\ W_w}{\Sigma_w\ W_w}. \tag{10}$$

This definition defines the interactions between probability and assertions. In case that there are a number of infinite worlds, we need to use measure theory. Nevertheless, this is beyond the scope of our paper.

Once we have defined the probability $Pr(A)$ of an assertion A as a real number, we can directly use it inside other assertions. In this sense, $Pr(A) = 0.5$, $Pr(A) \geq 0.3$, $Pr(A) \geq Pr(\forall(C, B(C))) - 0.3$, $Pr(A) \times 0.6 \geq 0.4$ and $Pr(Pr(A) \geq 0.3) \geq 0.3$ are all valid assertions. We are able to vefiry some properties about probability, for instance, Kolmogorov's first and second probability axioms.

We also extend this definition for conditional probability. We again introduce a new operator Pr over pairs of two assertions. Following a similar idea, the conditional probability $Pr(A_1, A_2)$ of an assertion A_1 providing another assertion A_2, also denoted by $Pr(A_1|A_2)$, is defined by the following schema assertion:

$$Pr(A_1|A_2) = \frac{\Sigma_{w,w\models A_1, w\models A_2}\ W_w}{\Sigma_{w,w\models A_2}\ W_w}. \tag{11}$$

Again, once conditional probability is defined as a real number, we can use it arbitrarily inside other assertions. Similarly, we can verify some properties about conditional probabilities, including the famous Bayes' theorem, i.e.,

$$Pr(A_1) \times Pr(A_2|A_1) = Pr(A_2) \times Pr(A_1|A_2).$$

for all assertions A_1 and A_2.

6.2 Interactions between logic and probability via assertions

Although we only define probabilities for assertions of the basic form, the interactions between probability and other building blocks, e.g., logic, are automatically established since assertions connected by logic operators can be reduced into the primitive form. In this sense, we can investigate some properties about the interactions between logic and probability. For instance, it can be observed that Kolmogorov's third probability axiom is a tautology in assertion logic. That is, let $A_1, \ldots, A_n$ be n assertions that are pairwise disjoint. Then, $Pr(A_1 \vee \cdots \vee A_n) = Pr(A_1) + \cdots + Pr(A_n)$.

It can be verified that many axioms and properties regarding the interactions between logic and probability are tautologies in assertional logic, for instance, the additivity axiom: $Pr(\phi) = Pr(\phi \wedge \psi) + Pr(\phi \wedge \neg\psi)$ and the distributivity axiom: $\phi \equiv \psi$ implies that $Pr(\phi) = Pr(\psi)$, for any two assertions ϕ and ψ. In this sense, assertional logic can also be used to validate existing properties about the interactions of logic and probability. In addition, it may foster new discoveries, e.g., the interactions between higher-order logic and probability and some properties about nested probabilities.

Note that we do not intend to reinvent the wheel of defining probability nor its interactions with logic. All definitions about (conditional) probability are borrowed from the literature. Instead, we take probability as a case study to show how one building block (e.g., logic) and another (e.g., probability) can be interacted through assertions without going deeper into the interactions between themselves.

6.3 More building blocks

More critically, there are many more important building blocks to be incorporated. It is barely possible to clarify the interactions among them all. Nevertheless, it becomes possible to unify them altogether in assertional logic as one only needs to consider the interactions between these building blocks and the basic form of assertions separately. Consequently, the interactions among these building blocks themselves will be automatically established via assertions, as what we did for unifying logic and probability.

As another case study, we consider how to formalize time in assertional logic. Time itself can be understood in different ways such as time points, time interval, LTL and CTL [Allen, 1983; Clarke and Emerson, 1982; Pnueli, 1977]. Following the same idea of incorporating probability, we only need to consider the interactions between time and assertions. In this paper, we only report the simple case of integrating assertions with time points. Let Tp be a concept of time points. We introduce a new operator $\mathtt{t}$ whose domain is a pair $(\mathcal{I}, Tp)$. Intuitively, $\mathtt{t}(i, tp), i \in \mathcal{T}, tp \in Tp$, is the value of individual i at time point tp. Then, we introduce temporal formulas, a new Boolean operator $\mathcal{T}$ whose domain is a pair $(\mathcal{A}, Tp)$, by the following schema assertion:

$$\mathcal{T}(a = b, tp) = \mathtt{t}(a, tp) = \mathtt{t}(b, tp). \tag{12}$$

Then, the interactions between time points and logic connectives and probability can be automatically established. We are able to investigate some properties. For instance, for all

assertions A and B, $\mathcal{T}(A, tp) = \top$ iff $\mathcal{T}(\neg A, tp) = \bot$; $\mathcal{T}(A \wedge B, tp) = \top \models \mathcal{T}(A, tp) = \top$ etc. Hence, we have an integrated formalism combing logic, probability and time points in assertional logic.

7 Discussion, Related Work and Conclusion

In this paper, we argue that, for the purpose of knowledge representation, classical first-order logic has some critical issues, including simplicity, flexibility, expressivity and extensibility. To address these issues, we propose assertional logic instead, in which the syntax of an application domain is captured by individuals (i.e., objects in the domain), concepts (i.e., groups of objects sharing something in common) and operators (i.e., connections and relationships among objects), and knowledge in the domain is simply captured by equality assertions of the form $a = b$, where a and b are terms.

In assertional logic, without redefining the semantics, one can extend a current system with new syntactic objects by definitions, which are special kinds of knowledge (i.e., assertions). Once defined, these syntactic objects can be used to define more. This can be done for assertional logic itself. We extend the primitive form of assertional logic with multi-assertions and nested assertions as well as logic connectives and quantifiers. We further consider how to extend assertional logic with other important AI building blocks. The key point is that, when one wants to integrate a new building block in assertional logic, she only needs to formalize it as syntactic objects (including individuals, concepts and operators) and defines its interactions with the basic form of assertions (i.e., $a = b$). Then, the interactions between this building block and others will be automatically established since all complicated assertions can essentially be reduced into the basic form. As a case study, we briefly discuss how to incorporate probability and time points in this paper.

Of course, assertional logic is deeply rooted in first-order logic. Individuals, concepts and operators are analogous to constants, unary predicates and functions respectively, and assertions are originated from equality atoms. Nevertheless, they differ from many essential aspects. Firstly, individuals can be high-order objects, e.g., concepts and assertions, so are concepts and operators. Secondly, assertional logic is naturally many-sorted, that is, the domain of an operator can be a tuple of many different concepts including high-order ones. Thirdly, concepts play a central role in assertional logic, which is natural for human knowledge representation. While concepts can be formalized as unary predicates in FOL, they are not specifically emphasized. Fourthly, in assertional logic, all kinds of knowledge are uniformly formalized in the same form of equality assertions. As shown in Section 5, complicated logic sentences are defined as equality assertions as well by embedding connectives and quantifiers as operators over assertions. Fifthly, following the above, although connectives, quantifiers and nesting can be defined in assertional logic, they are not considered as primitive constructs. In this sense, they will only be used on demand when necessary. We argue that this is an important reason that makes assertional logic simpler than FOL. Sixthly, in assertional logic, the simple form of $a = b$ is expressive as a and b can be high-order constructs and can be inherently related within the rich syntactic structure. In contrast, equality atoms in FOL do not have this power. Last but not least, assertional logic directly embraces extensibility within its own framework by syntactic definitions. For instance, to define quantifiers, assertional logic only needs two lines (see Equations 8 and 9) without redefining a whole new semantics, which is much simpler than FOL.

Assertional logic is also inspired by many approaches in symbolic AI. Traditionally in FOL, there is a strict hierarchy from the term level to the formula level. To some extent, this is broken in situation calculus [Levesque *et al.*, 1991; Lin, 2008] and game description language [Thielscher, 2016] that have to quantify over situations, actions and fluents and to directly talk about whether a fluent holds in a particular situation by a meta-predicate $Hold$. Assertional logic goes much further by completely demolishing the distinction and hierarchy between term level, predicate level and formula level. In assertional logic, one can flexibly use, e.g., atoms and predicates in the scope of a function as long as they match its domain definition. Also, one can quantify over any well-defined concepts, including a concept of assertions. This makes assertional logic even more expressive than high-order logic that cannot quantifier over formulas.

Another inspiration comes from the family of description logics [Baader *et al.*, 2003], where the terminologies individuals and concepts are borrowed from. The family of description logics allows a certain level of extensibility. By interpreting individuals, concepts and roles as domain elements, unary predicates and binary predicates respectively, one can extend the basic description logics with more building blocks (e.g., nominal, number restrictions, inverse and transitive roles etc.) based on the same foundational semantics. Also, one can freely assemble those building blocks into different fragments of description logics such as ALC, SHIQ, SHION and so on. However, many important AI building blocks, e.g., actions, probability, time, rules, etc. are still difficult to be incorporated by this method. Some interesting pioneering work have been done to consider more extensibility in description logics [Baader and Hanschke, 1991; Borgida, 1999; Giacomo *et al.*, 2011; Kutz *et al.*, 2004]. Nevertheless, they differ from assertional logic that embraces extensibility at a syntactic level instead of a semantic one.

This paper is only concerned with the representation task and the definition task. We leave the reasoning task and the learning task to our future work. We shall propose a different reasoning approach that focuses on efficient but not necessarily complete reasoning. Based on which, we further consider how to learn knowledge from experiences. As we shall see, the flexibility and extensibility of assertional logic play a critical role for knowledge reasoning and knowledge learning. We shall present this in another paper. Nevertheless, we argue that representation and definition are worth study on their own merits. Such successful stories include entity-relationship diagram, semantic network and many more. Besides, extending assertional logic with some important AI building blocks, e.g., actions and their effects, is indeed challenging and worth pursuing.

References

[Allen, 1983] James F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843, November 1983.

[Baader and Hanschke, 1991] Franz Baader and Philipp Hanschke. A scheme for integrating concrete domains into concept languages. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'91, pages 452–457, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc.

[Baader *et al.*, 2003] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, New York, NY, USA, 2003.

[Bacchus, 1990] Fahiem Bacchus. *Representing and Reasoning with Probabilistic Knowledge: A Logical Approach to Probabilities*. MIT Press, Cambridge, MA, USA, 1990.

[Borgida, 1999] Alexander Borgida. Extensible knowledge representation: the case of description reasoners. *J. Artif. Intell. Res. (JAIR)*, 10:399–434, 1999.

[Brachman and Levesque, 2004] Ronald J. Brachman and Hector J. Levesque. *Knowledge Representation and Reasoning*. Elsevier, 2004.

[Clarke and Emerson, 1982] Edmund M. Clarke and E. Allen Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In *Logic of Programs, Workshop*, pages 52–71, London, UK, UK, 1982. Springer-Verlag.

[Gaifman, 1964] Haim Gaifman. Concerning measures in first order calculi. *Israel J. Math.*, 2:1–18, 1964.

[Giacomo *et al.*, 2011] Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Higher-order description logics for domain metamodeling. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI'11, pages 183–188. AAAI Press, 2011.

[Hailperin, 1984] Theodore Hailperin. Probability logic. *Notre Dame J. Formal Logic*, 25:198–212, 1984.

[Halmos, 1960] Paul Halmos. *Naive Set Theory*. Van Nostrand, 1960. Reprinted by Springer-Verlag, Undergraduate Texts in Mathematics, 1974.

[Halpern, 1990] Joseph Y. Halpern. An analysis of first-order logics of probability. *Artif. Intell.*, 46(3):311–350, 1990.

[Keller *et al.*, 2010] Thomas Keller, Patrick Eyerich, and Bernhard Nebel. Task planning for an autonomous service robot. In *Proceedings of the 33rd Annual German Conference on Advances in Artificial Intelligence*, KI'10, pages 358–365, Berlin, Heidelberg, 2010. Springer-Verlag.

[Kutz *et al.*, 2004] O. Kutz, C. Lutz, F. Wolter, and M. Zakharyaschev. E-connections of abstract description systems. *Artificial Intelligence*, 156(1):1–73, 2004.

[Levesque *et al.*, 1991] Hector Levesque, Fiora Pirri, and Ray Reiter. Foundations for the situation calculus. *Electronic Transactions on Artificial Intelligence*, Vol. 2(1998), Issue 3-4:pp. 159–178, 1991.

[Lin, 2008] Fangzhen Lin. Situation calculus. In *Handbook of Knowledge Representation*, pages 649–669. 2008.

[Milch, 2006] Brian Christopher Milch. *Probabilistic Models with Unknown Objects*. PhD thesis, Berkeley, CA, USA, 2006. AAI3253991.

[Pearl, 1988] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[Pnueli, 1977] Amir Pnueli. The temporal logic of programs. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science*, SFCS '77, pages 46–57, Washington, DC, USA, 1977. IEEE Computer Society.

[Richardson and Domingos, 2006] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006.

[Thielscher, 2016] Michael Thielscher. GDL-III: A proposal to extend the game description language to general epistemic games. In *ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*, pages 1630–1631, 2016.

[van Harmelen *et al.*, 2008] Frank van Harmelen, Vladimir Lifschitz, and Bruce W. Porter, editors. *Handbook of Knowledge Representation*, volume 3 of *Foundations of Artificial Intelligence*. Elsevier, 2008.