7th International Workshop on Uncertainty Reasoning for the Semantic Web

Proceedings

edited by

Fernando Bobillo
Rommel Carvalho
Paulo C. G. da Costa
Claudia d'Amato
Nicola Fanizzi
Kathryn B. Laskey
Kenneth J. Laskey
Thomas Lukasiewicz
Trevor Martin
Matthias Nickles
Michael Pool

Bonn, Germany, October 23, 2011

collocated with
the 10th International Semantic Web Conference
– ISWC 2011 –

Foreword

This volume contains the papers presented at the 7th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2011), held as a part of the 10th International Semantic Web Conference (ISWC 2011) at Bonn, Germany, October 23, 2011. It contains 8 technical papers and 3 position papers, which were selected in a rigorous reviewing process, where each paper was reviewed by at least four program committee members.

The International Semantic Web Conference is a major international forum for presenting visionary research on all aspects of the Semantic Web. The International Workshop on Uncertainty Reasoning for the Semantic Web is an exciting opportunity for collaboration and cross-fertilization between the uncertainty reasoning community and the Semantic Web community. Effective methods for reasoning under uncertainty are vital for realizing many aspects of the Semantic Web vision, but the ability of current-generation web technology to handle uncertainty is extremely limited. Recently, there has been a groundswell of demand for uncertainty reasoning technology among Semantic Web researchers and developers. This surge of interest creates a unique opening to bring together two communities with a clear commonality of interest but little history of interaction. By capitalizing on this opportunity, URSW could spark dramatic progress toward realizing the Semantic Web vision.

We wish to thank all authors who submitted papers and all workshop participants for fruitful discussions. We would like to thank the program committee members and external referees for their timely expertise in carefully reviewing the submissions.

October 2011

Fernando Bobillo
Rommel Carvalho
Paulo C. G. da Costa
Claudia d'Amato
Nicola Fanizzi
Kathryn B. Laskey
Kenneth J. Laskey
Thomas Lukasiewicz
Trevor Martin
Matthias Nickles
Michael Pool

Workshop Organization

Program Chairs

Fernando Bobillo (University of Zaragoza, Spain)
Rommel Carvalho (George Mason University, USA)
Paulo C. G. da Costa (George Mason University, USA)
Claudia d'Amato (University of Bari, Italy)
Nicola Fanizzi (University of Bari, Italy)
Kathryn B. Laskey (George Mason University, USA)
Kenneth J. Laskey (MITRE Corporation, USA)
Thomas Lukasiewicz (University of Oxford, UK)
Trevor Martin (University of Bristol, UK)
Matthias Nickles (University of Bath, UK)
Michael Pool (Vertical Search Works, Inc., USA)

Program Committee

Fernando Bobillo (University of Zaragoza, Spain)
Silvia Calegari (University of Milano-Bicocca, Italy)
Rommel Carvalho (George Mason University, USA)
Paulo C. G. da Costa (George Mason University, USA)
Fabio Gagliardi Cozman (Universidade de São Paulo, Brazil)
Claudia d'Amato (University of Bari, Italy)
Nicola Fanizzi (University of Bari, Italy)
Marcelo Ladeira (Universidade de Brasília, Brazil)
Kathryn B. Laskey (George Mason University, USA)
Kenneth J. Laskey (MITRE Corporation, USA)
Thomas Lukasiewicz (University of Oxford, UK)
Trevor Martin (University of Bristol, UK)
Matthias Nickles (University of Bath, UK)
Jeff Z. Pan (University of Aberdeen, UK)
Rafael Peñaloza (TU Dresden, Germany)
Michael Pool (Vertical Search Works, Inc., USA)
Livia Predoiu (University of Mannheim, Germany)
Guilin Qi (Southeast University, China)
Celia Ghedini Ralha (Universidade de Brasília, Brazil)

David Robertson (University of Edinburgh, UK)
Daniel Sánchez (University of Granada, Spain)
Thomas Scharrenbach (University of Zurich, Switzerland)
Sergej Sizov (University of Koblenz-Landau, Germany)
Giorgos Stoilos (University of Oxford, UK)
Umberto Straccia (ISTI-CNR, Italy)
Andreas Tolk (Old Dominion University, USA)
Peter Vojtáš (Charles University Prague, Czech Republic)

Table of Contents

Technical Papers

Position Papers

Technical Papers

Building A Fuzzy Knowledge Body for Integrating Domain Ontologies

Konstantin Todorov[1], Peter Geibel[2], and Céline Hudelot[1]

[1]Laboratory MAS, École Centrale Paris
[2]TU Berlin, Sekr. FR 5-8, Fakultät IV

Abstract. This paper deals with the problem of building a common knowledge body for a set of domain ontologies in order to enable their sharing and integration in a collaborative framework. We propose a novel hierarchical algorithm for concept fuzzy set representation mediated by a reference ontology. In contrast to the original concept representations based on instances, this enables the application of methods of fuzzy logical reasoning in order to characterize and measure the degree of the relationships holding between concepts from different ontologies. We presenta an application of the approach in the multimedia domain.

1 Introduction

In collaborative contexts, multiple independently created ontologies often need to be brought together in order to enable their interoperability. These ontologies have an impaired collaborative functionality, due to heterogeneities coming from the decentralized nature of their acquisition, differences in scopes and application purposes and mismatches in syntax and terminology.

We present an approach to building a combined knowledge body for a set of domain ontologies, which captures and exposes various relations holding between the concepts of the domain ontologies, such as their relative generality or specificity, their shared commonality or their complementarity. This can be very useful in a number of real-life scenarioss, especially in collaborative platforms. Let us imagine a project which includes several partners, each of which has its own vocabulary of semantically structured terms that describes its activity. The proposed framework would allow every party to keep its ontology and work with it, but query the combined knowledge body whenever collaboration is necessary. Examples of such queries can be: "which concept of a partner P_1 is closest to my concept A", or "give me those concepts of all of my partners which are equally distant to my concept B", or "find me a concept from partner P_2 which is a strong subsumer of my concept C", or "what are the commonality and specificity between my concept A and my partner's concept D".

We situate our approach in a fuzzy framework, where every domain concept is represented as a fuzzy set of the concepts of a particular *reference* ontology. This can be seen as a projection of all domain source concepts onto a common semantical space, where distances and relations between any two concepts can be

expressed under fixed criteria. In contrast to the original instance-representation, we can apply methods of fuzzy logical reasoning in order to characterize the relationship between concepts from different ontologies. In addition, the fuzzy representations allow for quantifying the degree to which a certain relation holds.

The paper is structured as follows. Related work is presented in the next section. Background in the field of fuzzy sets, as well as main definitions and problems from the ontology matching domain are overviewed in Section 3. We present the concept fuzzification algorithm in Section 4, before we discuss how the combined knowledge body can be constructed in Section 5. Experimental results and conclusions are presented in Sections 6 and 7, respectively.

2 Related Work

Fuzzy set theory generalizes classical set theory [19] allowing to deal with imprecise and vague data. A way of handling imprecise information in ontologies is to incorporate fuzzy reasoning into them. Several papers by Sanchez, Calegari and colleagues [4], [5], [13] form an important body of work on fuzzy ontologies where each ontology concept is defined as a fuzzy set on the domain of instances and relations on the domain of instances and concepts are defined as fuzzy relations.

Work on fuzzy ontology matching can be classified in two families: (1) approaches extending crisp ontology matching to deal with fuzzy ontologies and (2) approaches addressing imprecision of the matching of (crisp or fuzzy) concepts. Based on the work on approximate concept mapping by Stuckenschmidt [16] and Akahani *et al.* [1], Xu *et al.* [18] suggested a framework for the mapping of fuzzy concepts between fuzzy ontologies. With a similar idea, Bahri *et al.* [2] propose a framework to define similarity relations among fuzzy ontology components. As an example of the second family of approaches, we refer to [8] where a fuzzy approach to handling matchinging uncertainty is proposed. A matching approach based on fuzzy conceptual graphs and rules is proposed in [3]. To define new intra-ontology concept similarity measures, Cross *et al.* [6] model a concept as a fuzzy set of its ancestor concepts and itself, using as a membership degree function the Information Content (IC) of concept with respect to its ontology.

Crisp instance-based ontology matching, relying on the idea that concept similarity is accounted for by the similarity of their instances, has been overviewed broadly in [7]. We refer particularly to the Caiman approach which relies on estimating concepts similarity by measuring class-means distances [10].

3 Background and Preliminaries

In this section, we introduce basics from fuzzy set theory and discuss aspects of the ontology matching problem.

3.1 Fuzzy Sets

A fuzzy set $\mathcal{A}$ is defined on a given domain of objects X by the function

$$\mu_{\mathcal{A}} : X \longmapsto [0, 1], \tag{1}$$

which expresses the degree of membership of every element of X to $\mathcal{A}$ by assigning to each $x \in X$ a value from the interval $[0, 1]$ [19]. The fuzzy power set of X, denoted by $\mathcal{F}(X, [0, 1])$, is the set of all membership functions $\mu : X \longmapsto [0, 1]$.

We recall several fuzzy set operations by giving definitions in terms of Gödel semantics [15]. The *intersection* of two fuzzy sets $\mathcal{A}$ and $\mathcal{B}$ is given by a t-norm function $T(\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x)) = \min(\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x))$. The *union* of $\mathcal{A}$ and $\mathcal{B}$ is given by $S(\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x)) = \max(\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x))$ where S is a t-conorm. The *complement* of a fuzzy set $\mathcal{A}$, denoted by $\neg\mathcal{A}$, is defined by the membership function $\mu_{\neg\mathcal{A}}(x) = 1 - \mu_{\mathcal{A}}(x)$. We consider the Gödel definition of a fuzzy *implication*

$$\mu_{\mathcal{A}\to\mathcal{B}}(x) = \begin{cases} 1, & \text{if } \mu_{\mathcal{A}}(x) \leq \mu_{\mathcal{B}}(x), \\ \mu_{\mathcal{B}}(x), & \text{otherwise.} \end{cases} \tag{2}$$

3.2 Ontologies, Heterogeneity and Ontology Matching

An *ontology* consists of a set of semantically related *concepts* and provides in an explicit and formal manner knowledge about a given domain of interest [7]. We are particularly interested in ontologies, whose concepts come equipped with a set of associated instances, defined as it follows.

Definition 1 (Crisp Ontology). *Let C be a set of concepts, is_a $\subseteq C \times C$, R a set of relations on C, I a set of instances, and $g : C \to 2^I$ a function that assigns a subset of instances from I to each concept in C. We require that is_a and g are compatible, i.e., that is_a$(A', A) \leftrightarrow g(A') \subseteq g(A)$ holds for all $A', A \in C$. In particular, this entails that is_a has to be a partial order. With these definitions, the quintuple*

$$O = (C, \textit{is\_a}, R, I, g)$$

forms a crisp ontology.

Above, a concept is modeled *intensionally* by its relations to other concepts, and *extensionally* by a set of instances assigned to it via the function g. By assumption, every instance can be represented as a real-valued vector, defined by a fixed number of variables of some kind (the same for all the instances in I).

Ontology heterogeneity occurs when two or more ontologies are created independently from one another over similar domains. Heterogeneity may be observed on *linguistic or terminological*, on *conceptual* or on *extensional* level [7]. *Ontology matching* is understood as the process of establishing relations between the elements of two or more heterogeneous ontologies. Different matching techniques have been introduced over the past years in order to resolve different types of heterogeneity [9].

Instance-based, or *extensional ontology matching* gathers a set of approaches around the central idea that ontology concepts can be represented as sets of related instances and the similarity measured on these sets reflects the semantic similarity between the concepts that these instances populate.

3.3 Crisp Concept Similarities

Consider the ontologies $O_1 = (C_1, is\_a_1, R_1, I_1, g_1)$ and $O_{ref} = (X, is\_a_{ref}, R_{ref}, I_{ref}, g_{ref})$. We rely on the straightforward idea that determining the similarity $sim(A, x)$ of two concepts $A \in C_1$ and $x \in X$ consists in comparing their instance sets $g_1(A)$ and $g_{ref}(x)$. For doing so, we need a similarity measure for instances $\mathbf{i}^A$ and $\mathbf{i}^x$, where $\mathbf{i}^A \in g_1(A)$ and $\mathbf{i}^x \in g_{ref}(x)$. We have used the scalar product and the cosine $s(\mathbf{i}^A, \mathbf{i}^x) = \frac{\langle \mathbf{i}^A, \mathbf{i}^x \rangle}{\|\mathbf{i}^A\| \|\mathbf{i}^x\|}$. Based on this similarity measure for elements, the similarity measure for the sets can be defined by computing the similarity of the mean vectors corresponding to class prototypes [10]:

$$sim_{proto}(A, x) = s\Big(\frac{1}{|g_1(A)|} \sum_{j=1}^{|g_1(A)|} \mathbf{i}_j^A, \frac{1}{|g_{ref}(x)|} \sum_{k=1}^{|g_{ref}(x)|} \mathbf{i}_k^x\Big). \tag{3}$$

Note that other approaches of concept similarity can be employed as well, like the variable selection approach in [17]. In the context of our study, we have used the method that both works best and is less complex. A hierarchical application of the similarity measure for the concepts of two ontologies is presented in [17].

4 A Hierarchical Algorithm for Concept Fuzzification

Let $\Omega = \{O_1, ..., O_n\}$ be a set of (crisp) ontologies that will be referred to as *source ontologies* defined as in Def. 1. The set of concepts $C_\Omega = \bigcup_{i=1}^n C_i$ will be referred to as the set of *source concepts*. The ontologies from the set Ω are assumed to share similar functionalities and application focuses and to be heterogeneous in the sense of some of the heterogeneity types described in Section 3.2. A certain complementarity of these resources can be assumed: they could be defined with the same application scope, but on different levels, treating different and complementary aspects of the same application problem.

Let $O_{ref} = (X, is\_a_{ref}, R_{ref}, I_{ref}, g_{ref})$ be an ontology, called a *reference* ontology whose concepts will be called *reference concepts*. In contrast to the source ontologies, the ontology O_{ref} is assumed to be a less application dependent, generic knowledge source. As a consequence of Def. 1, the ontologies in Ω and O_{ref} are populated.

The fuzzification procedure that we propose relies on the idea of scoring every source concept by computing its similarities with the reference concepts, using the similarity measure (3). A source concept A will be represented by a function of the kind

$$\mu_{\mathcal{A}}(x) = score_A(x), \forall x \in X, \tag{4}$$

where $score_A(x)$ is the similarity between the concept A and a given reference concept x. Since *score* takes values between 0 and 1, (4) defines a fuzzy set. We will refer to such a fuzzy set as the *fuzzified* concept A denoted by $\mathcal{A}$.

In order to fuzzify the concepts of a source ontology O_1, we propose the following hierarchical algorithm. First, we assign score-vectors, i.e. fuzzy membership functions to all leaf-node concepts of O_1. Every non-leaf node, if it does not contain instances (documents) of its own, is scored as the maximum of the scores of its children for every $x \in X$. If a non-leaf node has directly assigned instances (not inherited from its children), the node is first scored on the basis of these instances with respect to the reference ontology, and then as the maximum of its children and itself. To illustrate, let a concept A have children A' and A'' and let the non-empty function $g^*(A)$ represent the instances assigned directly to the concept A. We compute the following similarity scores for this concept w.r.t. the set X :

$$score_A(x) = max\{score_{A'}(x), score_{A''}(x), score_{g^*(A)}(x)\}, \forall x \in X. \quad (5)$$

Above, $score_{g^*(A)}(x)$ conventionally denotes the similarity obtained for the concept A and a reference concept x by only taking into account the documents assigned directly to A. The algorithm is given in Alg. 1.

It is worth noting that assigning the max of all children to the parent for every x leads to a convergence to uniformity of the membership functions for nodes higher up in the hierarchy. Naturally, the functions of the higher level concepts are expected to be less "specific" than those of the lower level concepts. A concept in a hierarchical structure can be seen as the union of its descendants, and a union corresponds to taking the max (an approach underlying the single link strategy used in clustering).

The hierarchical scoring procedure has the advantage that every x-score will be larger for a parent node than those for any of its children, and it holds that $\mu_{\mathcal{A}'}(x) \rightarrow \mu_{\mathcal{A}}(x) = 1$ for all x and all children $\mathcal{A}'$ of $\mathcal{A}$. From computational viewpoint, the procedure which only scores the populated nodes is less expensive, compared to scoring all nodes one by one.

5 Building a Combined Knowledge Body

The construction of a combined knowledge body for a set of source ontologies aims at making explicit the relations that hold among their concepts, across these ontologies. To these ends, we apply the fuzzy set representations acquired in the previous section. In what follows, we consider two source ontologies O_1 and O_2 but note that all definitions can be extended for multiple ontologies. Let $\mathcal{C}_\Omega = \{\mathcal{A}_1, ..., \mathcal{A}_{|C_1|}, \mathcal{B}_1, ..., \mathcal{B}_{|C_2|}\}$ be the union of the concept sets of O_1 (the $\mathcal{A}$-concepts) and O_2 (the $\mathcal{B}$-concepts). We introduce several relations and operations that can be computed over $\mathcal{C}_\Omega$ and will be used for constructing a combined reduced knowledge body that contains the concepts of interest.

5.1 Fuzzy Concept Relations

The implication $A' \rightarrow A$ holds for any A' and A such that $is\_a(A', A)$. We provide a definition for a fuzzy subsumption of two fuzzified concepts $\mathcal{A}'$ and $\mathcal{A}$ based on the fuzzy implication (2).

Function score(concept A, ontology O_{ref}, sim. measure sim)
begin
  for $i = 1, ..., |X|$ **do**
    sim$[i] = sim(A, x_i)$ // $x_i \in X$
  return sim
end
Procedure hierachicalScoring(ontology O, ontology O_{ref}, sim. measure sim)
begin

1. Let C be the list of concepts in O.
2. Let L be a list of nodes, initially empty
3. Do until C is empty:
   (a) Let L' be the list of nodes in C that have only children in L
   (b) $L = \text{append}(L, L')$
   (c) $C = C - L'$
4. Iterate over L (first to last), with A being the current element:
   if $children(A) = \emptyset$ **then**
     $score(A) = score(A, O_{ref}, sim)$
   else
     if $g^*(A) \neq \emptyset$ **then**
       $score(A) = max\{max_{B \in children(A)} score(B), score(A, O_{ref}, sim)\}$
     else
       $score(A) = max_{B \in children(A)} score(B)$

  return $score(A), \forall A \in C$
end

Algorithm 1: An algorithm for hierarchical scoring of the source concepts.

Definition 2 (Fuzzy Subsumption). *The subsumption $\mathcal{A}'$ is a $\mathcal{A}$ is defined and denoted in the following manner:*

$$\textbf{is\_a}(\mathcal{A}', \mathcal{A}) = \inf_{x \in X} \mu_{\mathcal{A}' \to \mathcal{A}}(x) \tag{6}$$

Equation (6) defines the fuzzy subsumption as a degree between 0 and 1 to which one concept is the subsumer of another. It can be shown that **is_a**, similarly to its crisp version, is reflexive and transitive (i.e. a quasi-order). In addition, the hierarchical procedure for concept fuzzification introduced in the previous section assures that **is_a**$(A', A) = 1$ holds for every child-parent concept pair, i.e. the crisp subsumption relation is preserved by the fuzzification process.

Taking the example of a collaborative platform from the introduction, computing the fuzzy **is_a** between two concepts allows for answering a user query regarding generality and specificity of their partners concepts with respect to a given target concept.

We provide a definition of a fuzzy ontology which follows directly from the fuzzification of the source concepts and their **is_a** relations introduced above.

Definition 3 (Fuzzy Ontology). *Let $\mathcal{C}$ be a set of (fuzzy) concepts,* **is_a** $: \mathcal{C} \times \mathcal{C} \to [0, 1]$ *a fuzzy is_a-relationship, $\mathcal{R}$ a set of fuzzy relations on $\mathcal{C}$, i.e., $\mathcal{R}$*

contains relations $r : \mathcal{C}^n \to [0,1]$*, where* n *is the arity of the relation (for the sake of presentation, we only consider binary relations),* $\mathcal{X}$ *a set of objects, and* $\phi : \mathcal{C} \to \mathcal{F}(\mathcal{X}, [0,1])$ *a function that assigns a membership function to every fuzzy concept in* $\mathcal{C}$*. We require that* **is_a** *and* ϕ *are compatible, i.e., that* $\mathbf{is\_a}(\mathcal{A}', \mathcal{A}) = \inf_x \mu_{\mathcal{A}' \to \mathcal{A}}(x)$ *holds for all* $\mathcal{A}', \mathcal{A} \in \mathcal{C}$*. In particular, it can be shown that this entails that* **is_a** *is a fuzzy quasi-order. With these definitions, the quintuple*

$$\mathcal{O} = (\mathcal{C}, \mathbf{is\_a}, \mathcal{R}, \mathcal{X}, \phi)$$

forms a fuzzy ontology.

Above, the set $\mathcal{X}$ is defined as a set of abstract objects. In our setting, these are the concepts of the reference ontology, i.e. $\mathcal{X} = X$. The set $\mathcal{C}$ is any subset of $\mathcal{C}_\Omega$. In case $\mathcal{C} = \mathcal{C}_1$, where $\mathcal{C}_1$ is the set of fuzzified concepts of the ontology O_1, $\mathcal{O}$ defines a fuzzy version of the crisp source ontology O_1. In case $\mathcal{C} = \mathcal{C}_\Omega$, $\mathcal{O}$ defines *a common knowledge body* for the two source ontologies. Note that the membership values of the reference concepts entail fuzzy membership values for the documents populating the reference concepts. However, we will work directly with the concepts scores in what follows.

Based on the subsumption relation defined above, we will define equivalence of two concepts in the following manner.

Definition 4 (Fuzzy θ-Equivalence). *Fuzzy* θ*-equivalence between a concept* $\mathcal{A}$ *and a concept* $\mathcal{B}$*, denoted by* $\mathcal{A} \backsim_\theta \mathcal{B}$ *holds if and only if* $\mathbf{is\_a}(\mathcal{A}, \mathcal{B}) > \theta$ *and* $\mathbf{is\_a}(\mathcal{B}, \mathcal{A}) > \theta$*, where* θ *is a parameter between* 0 *and* 1*.*

The equivalence relation allows to define classes of equivalence on the set $\mathcal{C}_\Omega$. In the collaborative framework described in the introduction, this can be used for querying concepts equivalent (up to a degree defined by the user) to a given user concept from the set of their partners concepts.

5.2 Similarity Measures for Fuzzy Concepts

We propose several measures of closeness of two fuzzy concepts $\mathcal{A}$ and $\mathcal{B}$. We begin by introducing a straightforward measure given by

$$\rho_{\text{base}}(\mu_\mathcal{A}, \mu_\mathcal{B}) = 1 - \max_{x \in X} |\mu_\mathcal{A}(x) - \mu_\mathcal{B}(x)|. \tag{7}$$

We consider a similarity measure based on the Euclidean distance:

$$\rho_{\text{eucl}}(\mu_\mathcal{A}, \mu_\mathcal{B}) = 1 - \|\mu_\mathcal{A} - \mu_\mathcal{B}\|_2 \,, \tag{8}$$

where $\|x\|_2 = \left(\sum_{x \in X} |x|^2\right)^{1/2}$ is the ℓ^2-norm. Several measures of fuzzy set compatibility can be applied, as well. Zadeh's partial matching index between two fuzzy sets is given by

$$\rho_{\text{sup-min}}(\mu_\mathcal{A}, \mu_\mathcal{B}) = \sup_{x \in X} T(\mu_\mathcal{A}(x), \mu_\mathcal{B}(x)). \tag{9}$$

Finally, the Jaccard coefficient is defined by

$$\rho_{\text{jacc}}(\mu_{\mathcal{A}}, \mu_{\mathcal{B}}) = \frac{\sum_x T(\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x))}{\sum_x S(\mu_{\mathcal{A}}(x), \mu_{\mathcal{B}}(x))}\,. \tag{10}$$

It is required that at least one of the functions $\mu_{\mathcal{A}}$ or $\mu_{\mathcal{B}}$ takes a non-zero value for some x. T and S are as defined in Section 3.

The similarity measures listed above provide different information as compared to the relations introduced in the previous subsection. Subsumption and equivalence characterize the structural relation between concepts, whereas similarity measures closeness between set elements. The two types of information are to be used in a complementary manner within the collaboration framework.

5.3 Quantifying Commonality and Relative Specificity

The union of two fuzzy concepts can be decomposed into three components, each quantifying, respectively, the commonality of both concepts, the specificity of the first compared to the second and the specificity of the second compared to the first expressed in the following manner

$$S(\mathcal{A}, \mathcal{B}) = (\mathcal{A}\mathcal{B}) + (\mathcal{A} - \mathcal{B}) + (\mathcal{B} - \mathcal{A}). \tag{11}$$

Each of these components is defined as follows and, respectively, accounts for:

$$\mathcal{A}\mathcal{B} \;= T(\mathcal{A}, \mathcal{B}) \;\; // \text{ what is common to both concepts;} \tag{12}$$

$$\mathcal{A} - \mathcal{B} = T(\mathcal{A}, \neg\mathcal{B}) \; // \text{ what is characteristic for A;} \tag{13}$$

$$\mathcal{B} - \mathcal{A} = T(\mathcal{B}, \neg\mathcal{A}) \; // \text{ what is characteristic for B.} \tag{14}$$

Several merge options can be provided to the user with respect to the values of these three components. In case $\mathcal{A}\mathcal{B}$ is significantly larger than each of $\mathcal{A}-\mathcal{B}$ and $\mathcal{B}-\mathcal{A}$, the two concepts can be merged into their union. In case one of $\mathcal{A}-\mathcal{B}$ or $\mathcal{B}-\mathcal{A}$ is larger than the other two components, the concepts can be merged to either $\mathcal{A}$ or $\mathcal{B}$.

6 Experiments

We situate our experiments in the multimedia domain, opposing two complementary heterogeneous ontologies containing annotated pictures. We chose, on one hand, LSCOM [14] initially built in the framework of TRECVID[1] and populated with the development set of TRECVID 2005. Since this set contains images from broadcast news videos, LSCOM is particularly adapted to annotate this kind of content, thus contains abstract and specific concepts (e.g. Science_Technology, Interview_On_Location). On the other hand, we used

[1] http://www-nlpir.nist.gov/projects/tv2005/

WordNet [11] populated with the LabelMe dataset [12], referred to as the LabelMe ontology. Contrarily to LSCOM, this ontology is very general, populated with photographs from daily life and contains concepts such as CAR, COMPUTER, PERSON, etc. The parts of the two multimedia ontologies used in the experiments are shown in Figure 1.

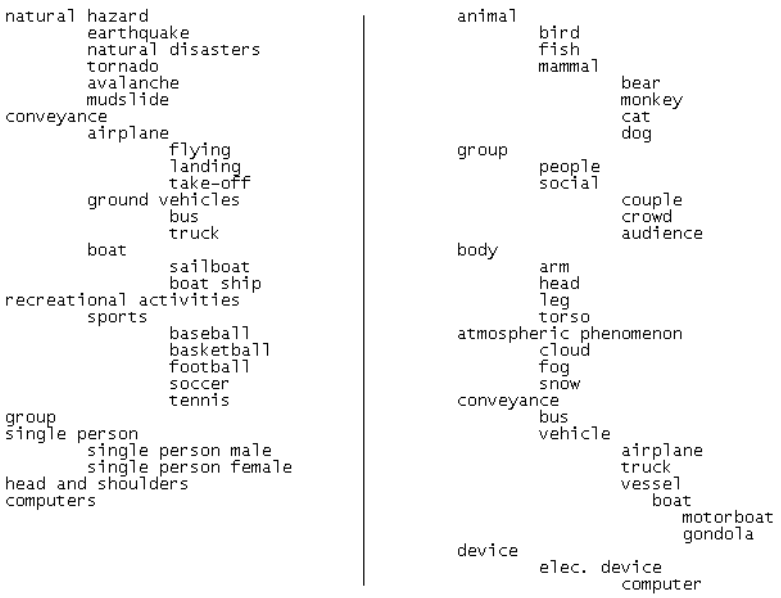


Fig. 1. The LSCOM (left) and the LabelMe (right) ontologies.

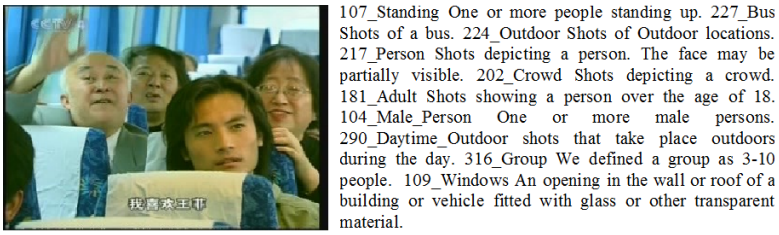


Fig. 2. The LSCOM concept Bus: a visual and a textual instance.

A text document has been generated for every image of the two ontologies, by taking the names of all concepts that an image contains in its annotation, as well as the (textual) definitions of these concepts (the LSCOM definitions for

TRECVID images or the WordNet glosses for LabelMe images). An example of a visual instance of a multimedia concept and the constructed textual description is given in Figure 2. Several problems related to this representation are worth noting. The LSCOM keyword descriptions sometimes depend on negation and exclusion which are difficult to handle in a simple bag-of-words approach. Taking the WordNet glosses of the terms in LabelMe introduces problems related to polysemy and synonymy. Additionally, a scene often consists of several objects, which are frequently not related to the object that determines the class of the image. In such cases, the other objects in the image act as noise.

Concept $\mathcal{A}$:	LSCM:truck vs.	LSCM:sports vs.	LM:computer vs.	LM:animal vs.
Concept $\mathcal{B}$:	LSCM:gr.vehicle	LSCM:basketball	LM:elec. device	LM:bird
$\mathbf{is\_a}(\mathcal{A},\mathcal{B})$	1	0.007	1	0.004
$\mathbf{is\_a}(\mathcal{B},\mathcal{A})$	0.012	1	0.011	1
$\mathbf{is\_a}^{\text{mean}}(\mathcal{A},\mathcal{B})$	1	0.052	1	0.062
$\mathbf{is\_a}^{\text{mean}}(\mathcal{B},\mathcal{A})$	0.326	1	0.07	1
Base Sim.	0.848	0.959	0.915	0.390
Eucl. Sim.	0.835	0.908	0.854	0.350
SupMin Sim.	0.435	0.545	0.359	0.309
Jacc. Sim.	0.870	0.814	0.733	0.399
Cosine Sim.	0.974	0.994	0.975	0.551

Concept $\mathcal{A}$:	LM:gondola vs.	LSCM:group vs.	LSCM:truck vs.	LSCM:truck vs.
Concept $\mathcal{B}$:	LSCM:boat_ship	LM:audience	LM:vehicle	LM:conveyance
$\mathbf{is\_a}(\mathcal{A},\mathcal{B})$	0.016	0.006	0.022	0.022
$\mathbf{is\_a}(\mathcal{B},\mathcal{A})$	0.009	1	0.012	0.012
$\mathbf{is\_a}^{\text{mean}}(\mathcal{A},\mathcal{B})$	0.86	0.022	0.748	0.769
$\mathbf{is\_a}^{\text{mean}}(\mathcal{B},\mathcal{A})$	0.167	1	0.301	0.281
Base Sim.	0.72	0.78	0.58	0.58
Eucl. Sim.	0.66	0.71	0.40	0.38
SupMin Sim.	0.069	0.082	0.22	0.22
Jacc. Sim.	0.49	0.42	0.54	0.52
Cosine Sim.	0.69	0.82	0.66	0.67

Table 1. Examples of pairs of matched intra-ontology concepts (above) and cross-ontology concepts (below), column-wise.

In order to fuzzify our source concepts, we have applied the hierarchical scoring algorithm from Section 4 independently for each of the source ontologies. As a reference ontology, we have used an extended version of the Wikipedia's so-called main topic classifications (adding approx. 3 additional concepts to every first level class), containing more than 30 categories. For each topic category, we included a set of corresponding documents from the Inex 2007 corpus.

The new combined knowledge body has been constructed by first taking the union of all fuzzified source concepts. For every pair of concepts, we have computed their Gödel subsumptional relations, as well as the degree of their similar-

ities (applying the measures from Section 5.2 and the standard cosine measure). Apart from the classical Gödel subsumption defined in (6), we consider a version of it which takes the average over all x instead of the smallest value, given as $\mathbf{is\_a}^{\text{mean}}(\mathcal{A}', \mathcal{A}) = \text{avg}_{x \in X} \mu_{\mathcal{A}' \to \mathcal{A}}(x)$. The results for several intra-ontology concepts and several cross-ontology concepts are given in Table 1. Fig. 3 shows a fragment of the common fuzzy ontology built for LSCOM and LabelMe. The labels of the edges of the graph correspond to the values of the fuzzy subsumptions between concepts.

We will underline several shortcomings that need to be adressed in future work. Due to data heterogeneity, it appears that the fuzzy **is_a**-structure is reflected better within one single ontology, as compared to cross-ontology relations which are more interesting. Additionally, some part_of relations are expressed as subsumptional (e.g. *torso* **is_a** *person*) which is a natural effect in view of the instance-representations. Indeed, the textual representation of images needs to be improved by accounting for the limitations discussed earlier in this section.

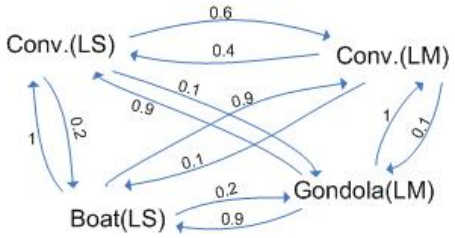


Fig. 3. A fragment of the common fuzzy ontology of LSCOM (LS) and LabelMe (LM).

Note that computing the common fuzzy ontology is inexpensive, once we have in hand the fuzzy representations of the source concepts made available by the hierarchical scoring algiorithm.

7 Conclusion and Open Ends

Whenever collaboration between knowledge resources is required, it is important to provide procedures which make explicit to users the relations that hold between different terms of these resources. In an attempt to solve this problem, we have proposed a fuzzy theoretical approach to build a common ontology for a set of source ontologies which contains these relations, as well as the degrees to which they hold, and can be queried upon need by different parties within a collaborative framework.

In future work, we will investigate the impact of the choice of a reference ontology onto the concept fuzzification and the quality of the constructed fuzzy common ontology. Additionally, the approach will be extended with elements of OWL 2, including relations and axioms between instances which is not covered by the ontology definition used in this work.

References

1. J.-I. Akahani, K. Hiramatsu, and T. Satoh. Approximate query reformulation based on hierarchical ontology mapping. In *In Proc. of Intl Workshop on SWFAT*, pages 43–46, 2003.
2. A. Bahri, R. Bouaziz, and F. Gargouri. Dealing with similarity relations in fuzzy ontologies. In *Fuzzy Systems Conference, 2007. FUZZ-IEEE 2007. IEEE International*, pages 1–6. IEEE, 2007.
3. P. Buche, J. Dibie-Barthélemy, and L. Ibanescu. Ontology mapping using fuzzy conceptual graphs and rules. In *ICCS Supplement*, pages 17–24, 2008.
4. S. Calegari and D. Ciucci. Fuzzy ontology, fuzzy description logics and fuzzy-owl. In *WILF*, pages 118–126, 2007.
5. S. Calegari and E. Sanchez. A fuzzy ontology-approach to improve semantic information retrieval. In *URSW*, 2007.
6. V. Cross and X. Yu. A fuzzy set framework for ontological similarity measures. In *WCCI 2010, FUZZ-IEEE 2010*, pages 1 – 8. IEEE Compter Society Press, 2010.
7. J. Euzenat and P. Shvaiko. *Ontology Matching*. Springer-Verlag, 1 edition, 2007.
8. A. Ferrara, D. Lorusso, G. Stamou, G. Stoilos, V. Tzouvaras, and T. Venetis. Resolution of conflicts among ontology mappings: a fuzzy approach. OM'08 at ISWC, 2008.
9. A. Gal and P. Shvaiko. Advances in web semantics i. chapter Advances in Ontology Matching, pages 176–198. Springer-Verlag, Berlin, Heidelberg, 2009.
10. M. S. Lacher and G. Groh. Facilitating the exchange of explicit knowledge through ontology mappings. In *Proceedings of the 14th FLAIRS Conf.*, pages 305–309. AAAI Press, 2001.
11. G.A. Miller. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.
12. B.C. Russell, A. Torralba, K.P. Murphy, and W.T. Freeman. Labelme: A database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1), 2008.
13. E. Sanchez and T. Yamanoi. Fuzzy ontologies for the semantic web. *Flexible Query Answering Systems*, pages 691–699, 2006.
14. J.R. Smith and S.F. Chang. Large-scale concept ontology for multimedia. *IEEE Multimedia*, 13(3):86 – 91, 2006.
15. U. Straccia. Towards a fuzzy description logic for the semantic web (preliminary report). In Asuncin Gomez-Perez and Jrme Euzenat, editors, *The Semantic Web: Research and Applications*, volume 3532 of *Lecture Notes in Computer Science*, pages 73–123. Springer Berlin / Heidelberg, 2005.
16. H. Stuckenschmidt. Approximate information filtering on the semantic web. In M. Jarke, G. Lakemeyer, and J. Koehler, editors, *KI 2002*, volume 2479 of *LNCS*, pages 195–228. Springer Berlin / Heidelberg, 2002.
17. K. Todorov, P. Geibel, and K.-U. Khnberger. Mining concept similarities for heterogeneous ontologies. In P. Perner, editor, *Advances in Data Mining. Applications and Theoretical Aspects*, volume 6171 of *LNCS*, pages 86–100. Springer Berlin / Heidelberg, 2010.
18. B. Xu, D. Kang, J. Lu, Y. Li, and J. Jiang. Mapping fuzzy concepts between fuzzy ontologies. In R. Khosla, R. J. Howlett, and L. C. Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems*, volume 3683 of *Lecture Notes in Computer Science*, pages 199–205. Springer Berlin / Heidelberg, 2005.
19. L.A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338 – 353, 1965.

Estimating Uncertainty of Categorical Web Data

Davide Ceolin, Willem Robert van Hage, Wan Fokkink, and Guus Schreiber

VU University Amsterdam
De Boelelaan 1081a
1081HV Amsterdam, The Netherlands
{d.ceolin,w.r.van.hage,w.j.fokkink,guus.schreiber}@vu.nl

Abstract. Web data often manifest high levels of uncertainty. We focus on categorical Web data and we represent these uncertainty levels as first or second order uncertainty. By means of concrete examples, we show how to quantify and handle these uncertainties using the Beta-Binomial and the Dirichlet-Multinomial models, as well as how take into account possibly unseen categories in our samples by using the Dirichlet Process.

Keywords: Uncertainty, Bayesian statistics, Non-parametric statistics, Beta-Binomial, Dirichlet-Multinomial, Dirichlet Process

1 Introduction

The World Wide Web and the Semantic Web offer access to an enormous amount of data and this is one of their major strengths. However, the uncertainty about these data is quite high, due to the multi-authoring nature of the Web itself and to its time variability: some data are accurate, some others are incomplete or inaccurate, and generally, such a reliability level is not explicitly provided.

We focus on the real distribution of these Web data, in particular of categorical Web data, regardless of whether they are provided by documents, RDF (see [27]) statements or other means. Categorical data are the among the most important types of Web data, because they include also URIs. We do not look for correlations among data, but we stick to estimating how category proportions distribute over populations of Web data.

We assume that any kind of reasoning that might produce new statements (e.g. subsumption) has already taken place. Hence, unlike for instance Fukuoe et al. (see [10]), that apply probabilistic reasoning in parallel to OWL (see [26]) reasoning, we will propose some models to address uncertainty issues on top of that kind of reasoning layers. These models, namely the parametric Beta-Binomial and Dirichlet-Multinomial, and the non-parametric Dirichlet Process, will use first and second order probabilities and the generation of new classes of observations, to derive safe conclusions on the overall populations of our data, given that we are deriving those from possibly biased samples.

First we will describe the scope of these models (section 2), second we will introduce the concept of conjugate prior (section 3), and then two classes of

models: parametric (section 4) and non-parametric (section 5). Finally we will discuss the results and provide conclusions (section 6).

2 Scope of this work

2.1 Empirical evidence from the Web

Uncertainty is often an issue in case of empirical data. This is especially the case with empirical Web data, because the nature of the Web increases the relevance of this problem but also offers means to address it, as we will see in this section. The relevance of the problem is related to the utilization of the mass of data that any user can find over the network: can one safely make use of these data? Lots of data are provided on the Web by entities the reputation of which is not surely known. In addition to that, the fact that we access the Web by crawling, means that we should reduce our uncertainty progressively, as long as we increment our knowledge. Moreover, when handling our samples it is often hard to determine how representative such a sample is of the entire population, since often we do not own enough sure information about it.

On the other hand, the huge amount of Web data gives also a solution for managing this reliability issue, since it can hopefully provide the evidence necessary to limit the risk when using a certain data set.

Of course, even within the Web it can be hard to find multiple sources asserting about a given fact of interest. However, the growing dimension of the Web makes it reasonable to believe in the possibility to find more than one data set about the given focus, at least by means of implicit and indirect evidence.

This work aims showing how it is possible to address the described issues by handling such empirical data, categorical empirical data in particular, by means of the Beta-Binomial, Dirichlet-Multinomial and Dirichlet Process models.

2.2 Requirements

Our approach will need to be quite elastic in order to cover several issues, as described below. The non-triviality of the problem comes in a large part from the impossibility to directly handle the sampling process from which we derive our conclusions. The requirements that we will need to meet are:

Ability to handle incremental data acquisition The model should be incremental, in order to reflect the process of data acquisition: as long as we collect more data (even by crawling), our knowledge will reflect that increase.

Prudence It should derive prudent conclusions given all the available information. In case not enough information is available, the wide range of possible conclusions derivable will clearly make it harder to set up a decision strategy.

Cope with biased sampling The model should deal with the fact that we are not managing a supervised experiment, that is, we are not randomly sampling from the population. We are using an available data set to derive safe consequences, but these data could, in principle, be incomplete, inaccurate or biased, and we must take this into account.

Ability to handle samples from mixtures of probability distributions The data we have at our disposal may have been drawn from diverse distributions, so we can't use the central limit theorem, because it relies on the fact that the sequence of variables is identically distributed. This implies the impossibility to make use of estimators that approximate by means of the Normal distribution.

Ability to handle temporal variability of parameters Data distributions can change over time, and this variability has to be properly accounted.

Complementarity with higher order layers The aim of the approach is to quantify the intrinsic uncertainty in the data provided by the reasoning layer, and, in turn, to provide to higher order layers (time series analysis, decision strategy, trust, etc.) reliable data and/or metadata.

2.3 Related work

The models adopted here are applied in a variety of fields. For the parametric models, examples of applications are: topic identification and document clustering (see [18, 6]), quantum physics (see [15]), and combat modeling in the naval domain (see [17]). What these heterogeneous fields have in common is the presence of multiple levels of uncertainty (for more details about this, see sect. 4).

Also non-parametric models are applied in a wide variety of fields. Examples of these applications include document classification [3] and haplotype inference [30]. These heterogeneous fields have in common with the previous application the presence of several layers of uncertainty, but they also show lack of prior information about the number of parameters. These concepts will be treated in section 5 where even the Wilcoxon sign-ranked test (see [29]), used for validation purposes, falls into the non-parametric models class.

As to our knowledge, the chosen models have not been applied to categorical Web data yet. We propose to adopt them, because, as the following sections will show, they fit the requirements previously listed.

3 Prelude: Conjugate priors

To tackle the requirements described in the previous section, we adopt some bayesian parametric and non-parametric models in order to be able to answer questions about Web data.

Conjugate priors (see [12]) are the "leit motiv", common to all the models adopted here. The basic idea starts from the Bayes theorem (1): given a prior knowledge and our data, we update the knowledge into a posterior probability.

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)} \tag{1}$$

This theorem describes how it is possible to compute the posterior probability, $P(A|B)$, given the prior probability of our data, $P(A)$, the likelihood of the model, given the data, $P(B|A)$, and the probability of the model itself, $P(B)$.

When dealing with continuous probability distributions, the computation of the posterior distribution by means of Bayes theorem can be problematic, due to the need to possibly compute complicated integrals. Conjugate priors allow us to overcome this issue: when prior and posterior probability distributions belong to the same exponential family, the posterior probability can be obtained by updating the prior parameters with values depending on the observed sample (see also [9]). Exponential families are classes of probability distributions having their density functions sharing the form $f(x) = e^{a(q)b(x)+c(q)+d(x)}$, with q a known parameter and a, b, c, d known functions. Exponential families include many important probability distributions, like the Normal, Binomial, Beta, etc., see [5]. So, if X is a random variable that distributes as defined by the function $P(p)$ (for some parameter or vector of parameters p) and, in turn, p distributes as $Q(\alpha)$ for some parameter (or vector of parameters) α called "hyperparameter"), and P belongs to the same exponential family as Q,

$$p \sim Q(\alpha),\ X \sim P(p)$$

then, after having observed *obs*,

$$p \sim Q(\alpha')$$

where $\alpha' = f(\alpha, obs)$, for some function f. For example, the Beta distribution is the conjugate of the Binomial distribution. This means that the Beta, shaped by the prior information and by the observations, defines the range within which the parameter p of the Binomial will probably be situated, instead of directly assigning to it the most likely value. Other examples of conjugate priors are: Dirichlet, which is conjugate to the Multinomial, and Gaussian, which is conjugate to itself.

Conjugacy guarantees ease of computation, which is a desirable characteristic when dealing with very big data sets as Web data sets often are. Moreover, the model is incremental, and this makes it fit the crawling process with which Web data are obtained, because crawling, in turn, is an incremental process. Both the heterogeneity of the Web and the crawling process itself increase the uncertainty of Web data. The probabilistic determination of the parameters of the distributions adds a smoothing factor that helps to handle this uncertainty.

4 Parametric bayesian models for categorical Web data

In this section we will handle situations where the number of categories is known a priori, by using the Dirichlet-Multinomial model and its special case with two categories, i.e. the Beta-Binomial model [9]. As generalized versions of the Binomial and Multinomial distribution, they describe the realization of sequences of mutually exclusive events. Categorical data can be seen as examples of such sequences. These models are parametric, since the number and type of parameters is given a priori, and they can also be classified as "empirical bayesian models". This further classification means that they can be seen as an approximation of a full hierarchical bayesian model, where the prior hyperparameters are set to their maximum likelihood values according to the analyzed sample.

4.1 Case study 1: Deciding between alternatives - ratio estimation

Suppose that a museum has to annotate a particular item I of its collection. Suppose further, that the museum does not have expertise in the house about that particular subject and, for this reason, in order to correctly classify the item, it seeks judgments from outside people, in particular from Web users that provide evidence of owning the desired expertise.

After having collected judgements, the museum faces two possible classifications for the item, C1 and C2. C1 is supported by four experts, while C2 by only one expert. We can use these numbers to estimate a probability distribution that resembles the correct distribution of C1 and C2 among all possible annotations.

A basic decision strategy that could make use of this probability distribution, could accept a certain classification only if its probability is greater or equal to a given threshold (e.g. 0.75). If so, the Binomial distribution representing the sample would be treated as representative of the population, and the sample proportions would be used as parameters of a Bernoulli distribution about the possible classifications for the analyzed item: $P(class(I) = C1) = 4/5 = 0.8,\ P(class(I) = C2) = 1/5 = 0.2$. (A Bernoulli distribution describes the possibility that one of two alternative events happens. One of these events happens with probability p, the other one with probability $1 - p$. A Binomial distribution with parameters n, p represents the outcome of a sequence of n Bernoulli trials having all the same parameter p.)

However, this solution shows a manifest leak. It provides to the decision strategy layer the probabilities for each of the possible outcomes, but these probabilities are based on the current available sample, with the assumption that it correctly represents the complete population of all existing annotations. This assumption is too ambitious. (Flipping a coin twice, obtaining a heads and a tails, does not guarantee that the coin is fair, yet.)

In order to overcome such a limitation, we should try to quantify how much we can rely on the computed probability. In other words, if the previously computed probability can be referred as a "first order" probability, what we need to compute now is a "second order" probability (see [15]). Given that the conjugate prior for the Binomial distribution representing our data is the Beta distribution, the model becomes:

$$p \sim Beta(\alpha, \beta),\ X \sim Bin(p, n) \tag{2}$$

where $\alpha = \#evidence_{C1} + 1$ and $\beta = \#evidence_{C2} + 1$.

By analyzing the shape of the conjugate prior Beta(5,2), we can be certain enough about the probability of C1 being safely above our acceptance threshold. In principle, our sample could be drawn by a population distributed with a $40\% - 60\%$ proportion. If so, given the threshold of acceptance of 0.75, we would not be able to take a decision based on the evidence. However, the quantification of that proportion would only be possible if we know the population. Given that we do not have such information, we need to estimate it, by computing (3), where we can see how the probability of the parameter p being above the threshold is less than 0.5. This manifests the need for more evidence: our sample suggests to

accept the most popular value, but the sample itself does not guarantee to be representative enough of the population.

$$P(p \geq 0.75) = 0.4660645,\ p \sim Beta(5, 2) \tag{3}$$

Table 1 shows how the confidence in the value p being above the threshold grows as long as we increase the size of the sample, when the proportion is kept. By applying the previous strategy (0.75 threshold) also to the second order probability, we will still choose C1, but only if supported by a sample of size at least equal to 15.

Table 1: The proportion within the sample is kept, so the most likely value for p is always exactly that ratio. However, given our 0.75 threshold, we are sure enough only if the sample size is 15 or higher.

#C1	$\#C2$	$P(p \geq 0.75)$ $p \sim Beta(\#C1 + 1, \#C2 + 1)$
4	1	0.4660645
8	2	0.5447991
12	3	0.8822048

Finally, these considerations could also be done on the basis of the Beta-Binomial distribution, which is a probability distribution representing a Binomial which parameter p is randomly drawn from a Beta distribution. The Beta-Binomial summarizes model (2) in one single function (4). We can see from Table 2 that the expected proportion of the probability distribution approaches the ratio of the sample (0.8), as the sample size grows. If so, the sample is regarded as a better representative of the entire population and the Beta-Binomial, as sample size grows, will converge to the Binomial representing the sample (see Fig. 1).

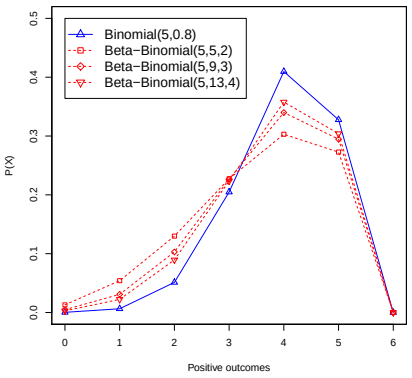


Fig. 1: Comparison between Binomial and Beta-Binomial with increasing sample size. As the sample size grows, Beta-Binomial approaches Binomial.

$$X \sim BetaBin(n, \alpha, \beta) = p \sim Beta(\alpha, \beta), X \sim Bin(n, p) \tag{4}$$

4.2 Case study 2: deciding proportions - confidence intervals estimation

The Linked Open Piracy[1] is a repository of piracy attacks that happened around the world in the period 2005 - 2011, derived from reports retrieved from the ICC-

[1] http://semanticweb.cs.vu.nl/lop

X	$E(X)$	$p = E(X)/n$
BetaBin(5,5,2)	3.57	0.71
BetaBin(5,9,3)	3.75	0.75
BetaBin(5,13,4)	3.86	0.77

Table 2: The sample proportion is kept, but the "expected proportion" p of Beta-Binomial passes the threshold only with a large enough sample. $E(X)$ is the expected value.

CCS website.[2] Attack descriptions are provided, in particular covering their type (boarding, hijacking, etc.), place, time, as well as ship type.

Data about attacks is provided in RDF format, and a SPARQL (see [28]) endpoint permits to query the repository. Such a database is very useful, for instance, for insurance companies to properly insure ships. The premium should be related to both ship conditions and their usual route. The Linked Open Piracy repository allows an insurance company to estimate the probability to be victim of a particular type of attacks, given the programmed route. Different attack types will imply different risk levels.

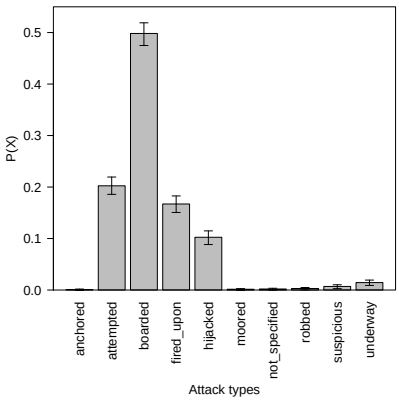


Fig. 2: Attack type proportion and confidence intervals

However, directly estimating the probability of a new attack given the dataset, would not be correct, because, although derived from data published from an official entity like the Chamber of Commerce, the reports are known to be incomplete. This fact clearly affects the computed proportions, especially because it is likely that this incompleteness is not fully random. There are particular reasons why particular attack types or attacks happening in particular zones are not reported. Therefore, beyond the uncertainty about the type of next attack happening (first order uncertainty), there will be an additional uncertainty order due to the uncertainty in the proportions themselves. This can be handled by a parametric model that will allow to estimate the parameters of a Multinomial distribution. The model that we are going to adopt is the multivariate version of the model described in section 4, that is, the Dirichlet-Multinomial model (see [6, 17, 18]):

$$Attacks \sim \text{Multinom}(params),\ params \sim \text{Dirichlet}(\alpha) \tag{5}$$

where α is the vector of observations per attack type (incremented by one unit each, as the α and β parameters of Beta probability distribution). By adopting this model, we are able to properly handle the uncertainty carried by our sample, due to either time variability (over the years, attack type proportions could have changed) or biased samples. Drawing the parameters of our Multinomial

[2] http://www.icc-ccs.org/

distribution from a Dirichlet distribution instead of directly estimating them, allows us to compensate for this fact, by smoothing our attacks distribution. As a result of the application of this model, we can obtain an estimate of confidence intervals for the proportions of the attack types (with 95% of significance level, see (6)). These confidence intervals depend both on the sample distribution and on its dimension (Fig. 2).

$$\forall p \in param, CI_p = (p - \theta_1, p + \theta_2), P(p - \theta_1 \leq p \leq p + \theta_2) = 0.95 \qquad (6)$$

5 Non-parametric bayesian models

In some situations, the previously described parametric models do not fit our needs, because they set a priori the number of categories, but this is not always possible. In the previous example, we considered and handled uncertainty due to the possible bias of our sample. The proportions showed by our sample could be barely representative of the entire population because of a non-random bias, and therefore we were prudent in estimating densities, even not discarding entirely those proportions. However, such an approach lacks in considering another type of uncertainty: we could not have seen all the possible categories and we are not allowed to know all of them a priori. Our approach was to look for the prior probability to our data in the n-dimensional simplex, where n is the number of categories, that is, possible attack types. Now such an approach is no more sufficient to address our problem. What we should do is to add yet another hierarchical level and look for the right prior Dirichlet distribution in the space of the probability distributions over probability distributions (or space of simplexes). Non-parametric models differ from parametric models in that the model structure is not specified a priori but is instead determined from data. The term non-parametric is not meant to imply that such models completely lack parameters, but that the number and nature of the parameters are flexible and not set in advance. Hence, these models are also called "distribution free".

5.1 Dirichlet Process

Dirichlet Processes [8] are a generalization of Dirichlet distributions, since they correspond to probability distributions of Dirichlet probability distributions. They are stochastic processes, that is, sequences of random variables (distributed as Dirichlet distributions) which value depends on the previously seen ones. Using the so-called "Chinese Restaurant Process" representation (see [22]), it can be described as follows:

$$X_n = \begin{cases} X_k^* & \text{with probability } \frac{num_{n-1}(X_k^*)}{n-1+\alpha} \\ \text{new draw from } H & \text{with probability } \frac{\alpha}{n-1+\alpha} \end{cases} \qquad (7)$$

where H is the continuous probability measure ("base distribution") from which new values are drawn, representing our prior best guess. Each draw from H will

return a different value with probability 1. α is an aggregation parameter, inverse of the variance: the higher α, the smaller the variance, which can be interpreted as the confidence value in the base distribution H: the higher the α value is, the more the Dirichlet Process resembles H. The lower the α is, the more the value of the Dirichlet Process will tend to the value of the empirical distribution observed. Each realization of the process is discrete and is equivalent to a draw from a Dirichlet distribution, because, if

$$G \sim DP(H, \alpha) \tag{8}$$

is a Dirichlet Process, and $\{B\}_{i=1}^{n}$ are partitions of S, we have that

$$(G(B_1)...G(B_n)) \sim Dirichlet(\alpha H(B_1)...\alpha H(B_n)) \tag{9}$$

If our prior Dirichlet Process is (8), given (9) and the conjugacy between Dirichlet and Multinomial distribution, our posterior Dirichlet Process (after having observed n values θ_i) can be represented as one of the following two representations:

$$(G(B_1)...G(B_n))|\theta_1...\theta_n \sim Dirichlet(\alpha H(B_1) + n_{\theta_1}...\alpha H(B_n) + n_{\theta_n}) \tag{10}$$

$$G \mid \theta_1...\theta_n \sim DP\left(\alpha + n, \frac{\alpha}{\alpha + n}H + \frac{n}{\alpha + n}\frac{\Sigma_{i=1}^{n}\delta_{\theta_i}}{n}\right) \tag{11}$$

where δ_{θ_i} is the Dirac delta function (see [4]), that is, the function having density only in θ_i. The new base function will therefore be a merge of the prior H and the empirical distribution, represented by means of a sum of Dirac delta's. The initial status of a Dirichlet Process posterior to n observations, is equivalent to the *nth* status of the initial Dirichlet Process that produced those observations (see De Finetti theorem, [13]).

The Dirichlet process, starting from a (possibly non-informative) "best guess", as long as we collect more data, will approximate the real probability distribution. Hence, it will correctly represent the population in a prudent (smoothed) way, exploiting conjugacy like the Dirichlet-Multinomial model, that approximates well the real Multinomial distribution only with a large enough data set (see section 4). The improvement of the posterior base distribution is testified by the increase of the α parameter, proportional to the number of observations.

5.2 Case study 3: Classification of piracy attacks - unseen types generation

We aim at predicting the type distributions of incoming attack events. In order to build an "infinite category" model, we need to allow for event types to be randomly drawn from an infinite domain. Therefore, we map already observed attack types with random numbers in [0..1] and, since all events are a priori equally likely, then new events will be drawn from the Uniform distribution, $U(0, 1)$, that is our base distribution (and is a measure over [0..1]). The model then is:

– $type_1 \sim DP(U(0,1), \alpha)$: the prior over the first attack type in region R;
– $attack_1 \sim Categorical(type_1)$: type of the first attack in R during $year_y$.

After having observed $attack_{1...n}$ during $year_y$, our posterior process becomes:

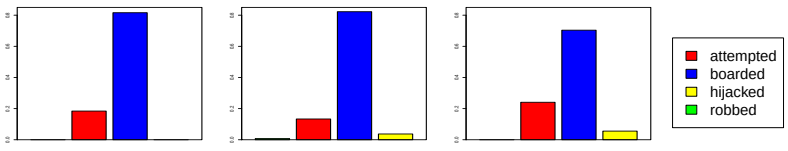
$$type_{n+1} \mid attack_{1...n} \sim DP\left(\alpha + n, \frac{\alpha}{\alpha + n} U(0,1) + \frac{n}{\alpha + n} \frac{\Sigma_{i=1}^{n} \delta_{attack_i}}{n}\right)$$

where α is a low value, given the low confidence in $U(0,1)$, and $type_{n+1}$ is the prior of $attack_{n+1}$, that happens during $year_{y+1}$. A Categorical distribution is a Bernoulli distribution with more than two possible outcomes (see Section 4).

Results Focusing on each region at time, we simulate all the attacks that happened there in $year_{y+1}$. Names of new types generated by simulation are matched to the actual $year_{y+1}$ names, that do not occur in $year_y$, in order of decreasing probability. The simulation is compared with a projection of the proportions of $year_n$ over the actual categories of $year_{n+1}$. The comparison is made by measuring the distance of our simulation and of the projection from the real attack types proportions of $year_{y+1}$ using the the Manhattan distance (see [16]). This metric simply sums, for each attack type, the difference between the real $year_{y+1}$ probability and the one we forecast. Hence, it can be regarded as an error measure. Table 3 summarizes the results over the entire dataset.[3] Our simulation reduces the distance (i.e. the error) with respect to the projection, as confirmed by a Wilcoxon signed-rank test [29] at 95% significance level. (This non-parametric statistical hypothesis test is used to determine whether one of the means of the population of two samples is smaller/greater than the other.) The simulation improves when large amount of data is available and the category cardinality varies, as in case of Region India, which results are reported in Fig. 3 and 4a.

Table 3: Averages and variances of the error of the two forecasts. The simulation gets a better performance.

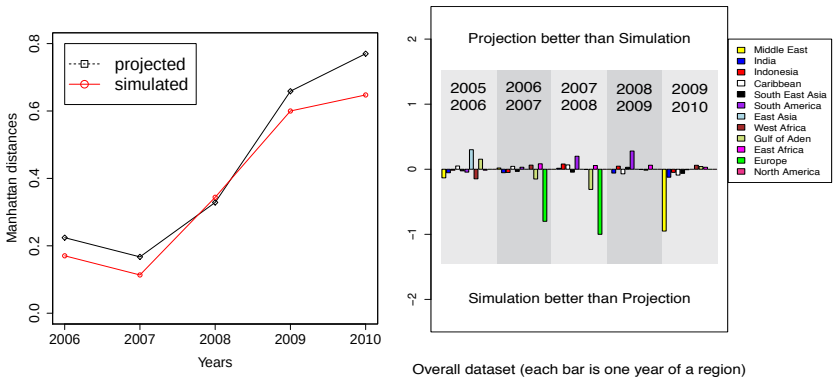
	Simulation	Projection
Average distance	0.29 △	0.35
Variance	0.09 △	0.21



(a) Year 2005 (b) Simulation (c) Year 2006

Fig. 3: Comparison between the projection forecast and the simulation forecast with the real-life year 2006 data of region India.

[3] The code can be retrieved at `http://www.few.vu.nl/~dceolin/DP/Dir.R`



(a) Errors (b) Distances differences

Fig. 4: Error distance from real distribution of the region India (fig. 4a) and differences of the error of forecast based on simulation and on projection (fig. 4b). Positive difference means that the projection predicts better than our simulation.

6 Conclusions and future work

The fact that our proposed models fit well with the expressed requirements is apparently a good hypothesis to continue to explore, because we have seen how it is possible to handle such uncertainty and to transform it in a smoothing factor of the probability distribution that we estimate given our evidence, by allowing the parameters of our distributions to be probabilistically determined. Moreover, we have built models able to produce reliable forecasts also when not every class is know a priori. We also provided case study validation of the suggested models.

The set of models will be extended to deal with concrete domain data (e.g. time intervals, measurements), for instance, by adopting the Normal or the Poisson Process (see [9]). Moreover, automatic model selection will be investigated, in order to choose the best model also when the limited information about our problems could make more models suitable. From a pure Web perspective, our models will be extended to properly handle contributions coming from different sources together with their reputation. This means, on one side, considering also provenance (like in [1]) and, on the other side, using Mixture Models ([23]), Nested ([24]) and Hierarchical Dirichlet Processes ([25]), eventually employing Markov Chain Monte Carlo algorithms (see [7, 21]) to handle lack of conjugacy.

References

1. D. Ceolin, P. Groth, and W. R. van Hage. Calculating the trust of event descriptions using provenance. In *SWPM 2010 Proceedings*, 2010.
2. D. Ceolin, W.R. van Hage, and W. Fokkink. A trust model to estimate the quality of annotations using the web. In *WebSci10*, 2010.

3. M. Davy and J. Tourneret. Generative supervised classification using dirichlet process priors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32:1781–1794, 2010.
4. P. Dirac. *Principles of quantum mechanics*. Oxford at the Clarendon Press, 1958.
5. Andersen E. Sufficiency and exponential families for discrete sample spaces. *Journal of the American Statistical Association*, 65:1248–1255, 9 1970.
6. C. Elkan. Clustering documents with an exponential-family approximation of the Dirichlet compound multinomial distribution. In *ICML*, volume 148, pages 289–296. ACM, 2006.
7. M. D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90:577–588, 1994.
8. T. S. Ferguson. A Bayesian Analysis of Some Nonparametric Problems. *The Annals of Statistics*, 1(2):209–230, 1973.
9. D. Fink. A Compendium of Conjugate Priors. Technical report, Cornell University, 1995.
10. A. Fokoue, M. Srivatsa, and R. Young. Assessing trust in uncertain information. In *ISWC*, pages 209–224, 2010.
11. A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. CRC Press, 2003.
12. R. Schlaifer H. Raiffa. *Applied statistical decision theory*. M.I.T. Press, 1968.
13. M. Hazewinkel. *Encyclopaedia of Mathematics*, chapter De Finetti theorem. Springer, 2001.
14. B. He, M. Patel, Z. Zhang, and K. C. Chang. Accessing the deep web. *Commun. ACM*, 50:94–101, May 2007.
15. J. Hilgevoord and J. Uffink. Uncertainty in prediction and in inference. *Foundations of Physics*, 21:323–341, 1991.
16. E. F. Krause. *Taxicab Geometry*. Dover, 1987.
17. P. Kvam and D. Day. The multivariate polya distribution in combat modeling. *Naval Research Logistics (NRL)*, 48(1):1–17, 2001.
18. R. E. Madsen, D. Kauchak, and C. Elkan. Modeling word burstiness using the Dirichlet distribution. In *ICML*, ICML '05, pages 545–552. ACM, 2005.
19. T. Minka. Estimating a Dirichlet distribution. Technical report, Microsoft Research, 2003.
20. P. Müller N. L. Hjort, C. Holmes and S. G. Walker. *Bayesian Nonparametrics*. Cambridge University Press, 2010.
21. R. M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and graphical statistics*, 9(2):249–265, 2000.
22. J. Pitman. Exchangeable and partially exchangeable random partitions. *Probab. Theory Related Fields*, 102(2):145–158, 1995.
23. Carl Edward Rasmussen. The infinite gaussian mixture model. In *In Advances in Neural Information Processing Systems 12*, pages 554–560. MIT Press, 2000.
24. A. Rodriguez, D. B. Dunson, and A. E. Gelfand. The nested dirichlet process. *Journal of the American Statistical Assoc.*, 103(483):1131–1144, September 2008.
25. Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Assoc.*, 101(476):1566–1581, 2006.
26. W3C. OWL Reference, August 2011. http://www.w3.org/TR/owl-ref/.
27. W3C. Resource Definition Framework, August 2011. http://www.w3.org/RDF/.
28. W3C. SPARQL, August 2011. http://www.w3.org/TR/rdf-sparql-query/.
29. F. Wilcoxon. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80–83, 1945.
30. E. Xing. Bayesian Haplotype Inference via the Dirichlet Process. In *ICML*, pages 879–886. ACM Press, 2004.

An Evidential Approach for Modeling and Reasoning on Uncertainty in Semantic Applications

Amandine Bellenger[1,2], Sylvain Gatepaille[1], Habib Abdulrab[2], Jean-Philippe Kotowicz[2]

[1] Cassidian – An EADS Company
Parc d'Affaires des Portes, 27106 Val-de-Reuil, France
{Amandine.Bellenger, Sylvain.Gatepaille}@eads.com
[2] LITIS Laboratory
INSA de Rouen, Avenue de l'Université, 76801 Saint-Etienne-du-Rouvray, France
{Habib.Abdulrab, Jean-Philippe.Kotowicz}@insa-rouen.fr

Abstract. Standard semantic technologies propose powerful means for knowledge representation as well as enhanced reasoning capabilities to modern applications. However, the question of dealing with uncertainty, which is ubiquitous and inherent to real world domain, is still considered as a major deficiency. We need to adapt those technologies to the context of uncertain representation of the world. Here, this issue is examined through the evidential theory, in order to model and reason about uncertainty in the assertional knowledge of the ontology. The evidential theory, also known as the Dempster-Shafer theory, is an extension of probabilities and proposes to assign masses on specific sets of hypotheses. Further on, thanks to the semantics (hierarchical structure, constraint axioms and properties defined in the ontology) associated to hypotheses, a consistent frame of this theory is automatically created to apply the classical combinations of information and decision process offered by this mathematical theory.

Keywords: Ontologies, OWL, Uncertainty, Dempster-Shafer Theory, Belief Functions, Semantic Similarity.

1 Introduction

Uncertainty is an important characteristic of data and information handled by real-world applications. The term "uncertainty" refers to a variety of forms of imperfect knowledge, such as incompleteness, vagueness, randomness, inconsistency and ambiguity. In this approach, we consider only the epistemic uncertainty, due to lack of knowledge (incompleteness) and the inconsistency, due to conflicting testimonies or reports. This paper presents a proposal on a possible way to tackle the issue of representing and reasoning on this type of uncertainty in semantic applications, by using the Dempster–Shafer theory [1], also known as "evidential theory" or "belief function theory". The general objective of our applications is to form the most informative and consistent view of the situation, observed by multiple sources. These observations populate our domain ontology. Thus, we consider that the uncertainty

has to be embodied in the instantiation rather than in the structural knowledge of ontology. One of our requirements is that a source can assign a belief on any instance without worrying of any level of granularity or disjointness of these instances. For example, one source could assign a belief on an instance of class *Vehicle* and, at the same time, another belief on an instance of type *Car*, which inherits from the class *Vehicle*.

The following section of this paper introduces the basic definitions and notations of the Dempster–Shafer theory. Section 3 presents our ontology modeling of the representation of uncertainty, using evidential theory. In the fourth section, we address how to reason with the evidential theory while benefiting from the semantics included in the domain ontology. Section 5 proposes to position our approach by comparing it with already existing works in the domain of uncertainty and the Semantic Web.

2 Basis of Dempster-Shafer Theory

The Dempster–Shafer theory [1] allows the combination of distinct evidence from different sources in order to calculate a global amount of belief for a given hypothesis. It is often presented as a generalization of the probability theory. It permits to manage uncertainties as well as inaccuracies and ignorance.

2.1 Frame of Discernment

Let Ω be the universal set, also called the discernment frame. It is the set of all the N states (hypothesis) under consideration: $\Omega = \{H_1, H_2, ..H_N\}$.

The universal set is supposed to be exhaustive and all hypotheses are exclusives. Exhaustivity refers to the closed-world principle. From this universal set, we can define a set, noted 2^{Ω}. It is called the power set and is the set of all possible sub-sets of Ω, including the empty set. It is defined as follows:

$$2^{\Omega} = \{A | A \subseteq \Omega\} = \{\emptyset, \{H_1\}, ..., \{H_N\}, \{H_1, H_2\}, ..., \Omega\}.$$

2.2 Basic Mass Assignment and Belief Measures

A source, who believes that one or more states in the power set of Ω might be true, can assign belief mass to these states. Formally, a mass function is defined by:

$$m : 2^{\Omega} \rightarrow [0,1] . \quad (1)$$

It is also called a basic belief assignment and it has two properties:

$$m(\emptyset) = 0 \quad \text{and} \quad \sum_{A \in 2^{\Omega}} m(A) = 1 . \quad (2)$$

This quantity differs from a probability since the total mass can be given either to singleton hypothesis H_n or to composite ones.

The main other belief measures are belief and plausibility. Belief *bel(A)* for a set *A* is defined as the sum of all the masses of the subsets of the set of interest:

$$bel(A) = \sum_{B|B\subseteq A} m(B) \qquad \forall A \subseteq \Omega . \tag{3}$$

It is the degree of evidence that directly supports the given hypothesis *A* at least in part, forming a lower bound. The plausibility *pl(A)* is the sum of all the masses of the sets *B* that intersect the set of interest *A*:

$$pl(A) = \sum_{B|B\cap A\neq \emptyset} m(B) \qquad \forall A \subseteq \Omega . \tag{4}$$

pl(A) can be interpreted as the part of belief which could be potentially allocated to *A*, taking into account the elements that do not contradict this hypothesis. It is seen as an upper bound.

2.4 Information Fusion

Modeling by masses through the evidential theory would be useless without an adequate combination enabling the fusion of a set of information sources. This is especially the role of the Dempster's rule of combination. Namely, it combines two independent sets of mass assignments (i.e. from difference sources). The combination (called the joint mass) is calculated from the two sets of masses m_1 and m_2 in the following manner:

$$(m_1 \oplus m_2)(A) = \begin{cases} \dfrac{\sum_{B\cap C=A} m_1(B)m_2(C)}{1-K_{12}} & A \neq \emptyset \\ 0 & A = \emptyset \end{cases} . \tag{5}$$

$$\text{where} \qquad K_{12} = \sum_{B\cap C=\emptyset} m_1(B)m_2(C) . \tag{6}$$

K is a measure of the amount of conflict between the two mass sets. K is ranging from 0 to 1. Dempster's rule corresponds to the normalized conjunctive operator. Other combination rules exist, such as the disjunctive combination and other operators that reassign the amount of conflict differently [2].

3 DS-Ontology Modeling

The first step of our approach is to model and represent the uncertainty through ontologies. Modeling is proposed through a specific ontology that needs to be imported in the initial domain ontology. This initial domain ontology is the ontology we want to instantiate in an uncertain way. The imported ontology is called DS-Ontology. It is described in the following.

3.1 Structural Knowledge of the DS-Ontology

This ontology is a formal representation of the theory of Dempster-Shafer, as it proposes a shared understanding of the main concepts: mass, belief, plausibility, source, etc. It is non-domain specific, since one can use it in every area of knowledge. It has been coded in OWL2 language [3]. Hereafter is an informal schema of the terminology of DS-Ontology.

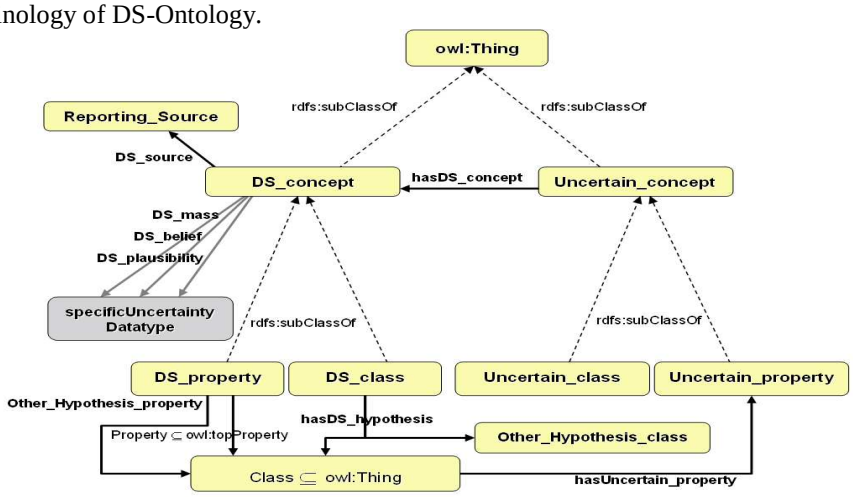


Fig. 1. Informal ontology structure schema. Yellow boxes represent OWL classes. Grey ones refer to datatypes (XML ones and user defined datatype). Arrows symbolize properties. Resources appearing without namespace prefix come from the DS-Ontology whose namespace is http://DS-Ontology.owl.

The main classes are *Uncertain_concept* and *DS_concept*. The *DS_concept* class links the hypothesis, with the source and the numerical amount of belief related to the hypothesis. The hypothesis consists either of a singleton or a union of hypotheses. Hypotheses are in fact instances of the domain ontology. Instances are either individuals of classes or instances of properties. The *Uncertain_concept* class links together all the *DS_concept* that are related to the same context. Indeed, the uncertainty is embodied by several candidate instances (with an assigned belief) and the uncertainty is concretely instantiated through one instance of *Uncertain_concept*. *Uncertain_concept* enables to retrieve the set of hypotheses under consideration, i.e. the power set 2^{Ω}.

In order to represent uncertainty both on individuals and on asserted properties, *DS_concept* and *Uncertain_concept* have been specialized. They are specified in subclasses *XX_class* and *XX_property* (XX prefix representing both DS and Uncertain). *Uncertain_concept* is now an equivalent class to the union of *Uncertain_property* and *Uncertain_class*, while the latter two are disjoint. Respectively, this holds for *DS-concept* and its subclasses.

The *hasDS_hypothesis* object property relates an instance of *DS_class* to a set of candidate individuals. Concerning candidate properties, things have been done differently. Indeed, OWL properties are not first-class citizens, contrary to OWL

classes; as such OWL properties cannot be related to each others: a property cannot be the subject or object of another property. To get around this, an object property *hasUncertain_property* has been introduced. The original subject of the candidate property is the subject of *hasUncertain_property*. The domain of *hasUncertain_property* is intuitively the class *Uncertain_Property*. Then, *DS_Property* instances are directly the subject of the candidate properties while their object remains unchanged.

An illustration of the use of the DS-Ontology is given in the next section.

As with the Dempster-Shafer theory, the modeling of ignorance is made possible. It is realized through an instance of *DS_concept* linked to all hypothetical instances. Ontologies evolve within the open world assumption. However, the original evidential theory assumes a closed world and that is why the measure of the amount of conflict exists. Therefore, we should for instance opt for an Open Extended World extension of the Dempster-Shafer theory [4]. Applied to ontologies, it consists in modeling another concept, with prefix: "*Other_Hypothesis*". This element is included in the DS-Ontology (both as a class and a property) and is asserted if needed to embody hypothesis, which does not correspond to any already defined concept in the domain ontology.

We represent numerical evidential belief through a s*pecificUncertaintyDatatype* which is a user-defined datatype defined in our DS-Ontology to restrict its value to an *xsd:double* ranging from 0 to 1.

In our model, *Uncertain_concept* and *DS_concept* are classes that let grouping together collected pieces of information about an uncertain instance we want to model and reason about. It can be viewed as a reification process, where an addressable object is created as a proxy for non-addressable objects. Informally, reification is often referred to as "making something a first-class citizen" within the scope of a particular system. Reification is one of the most frequently used techniques of conceptual analysis and knowledge representation. Even if RDF language enables reification process [5], we choose to model explicitly in an ontology our full representation, instead of using annotations not defined in the ontology. As a consequence, the uncertainty extension of OWL through the DS-Ontology is completely compliant with the basic principle of OWL ontologies to structure knowledge in two levels: structural and assertional.

3.2 Instantiation Example

Our applications aim mainly at observing real world situations through different perspectives (sources) and give an understandable and fused analysis of what is going on in this situation to the final decision maker. This simplified scenario involves here two distinct sources. One is a human while the other is an automatic sensor, such as radar. They both want to express that something is going into a specific direction; the "something" entity is the same object for both sources; however, they are not sure about how to identify this object. Indeed, the radar source can only distinguish a land vehicle from an aircraft; it assigns here a more important belief on the fact that it is an instance of a land vehicle. The second source is a human, who has a slight and far away view of the situation is assigning different beliefs to an instance of car which

looks like red, or a fire truck or a more imprecisely one to a land vehicle. In most cases, we do not have to assess the belief assigned to hypotheses by ourselves, it is directly given by the sources according to their condition of use (e.g. meteorology, proximity, etc.) and we apply possibly a weakening coefficient according to the source reliability. The structural knowledge of this domain is modeled through an ontology (http://ontology-uri.owl), whose hierarchical structure is captured in figure 2.

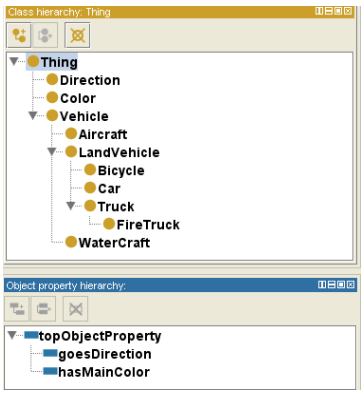


Fig. 2. Protégé snapshot of the structural knowledge of the ontology.

In addition to the hierarchical structure of the knowledge, domain and range of properties are also defined, as well as additional information concerning a priori information about the world. For instance, in this domain ontology, it is mentioned that a fire truck individual is always associated to the property *hasMainColor* with the value red. According to the sources and to the domain ontology, the assertional knowledge of this ontology involves:

- http://ontology-uri.owl#direction: an individual of class http://ontology-uri.owl#Direction
- respectively #landVehicle for class #LandVehicle
- respectively #aircraft for class #Aircraft
- respectively #fireTruck for class #FireTruck
- respectively #red for class #Color
- respectively #car for class #Car which is linked to the individual #red through the #hasMainColor property.

The set of candidate instances are: {#landVehicle, #aircraft, #fireTruck, #car}. We refer here to IRI instances only with their local name, omitting the namespace.
Regarding the Dempster-Shafer theory, the masses are assigned by the sources as:

- $m_{radar}(\{\#landVehicle\}) = 0.6$; $m_{radar}(\{\#aircraft\}) = 0.1$; $m_{radar}(\{\#landVehicle, \#aircraft\}) = 0.3$
- $m_{human}(\{\#car\}) = 0.2$; $m_{human}(\{\#fireTruck\}) = 0.4$; $m_{human}(\{\#landVehicle\}) = 0.4$

This domain ontology imports the DS-Ontology, in order to represent all these pieces of knowledge within the domain ontology. Two more individuals are created to represent the sources:

- #human for class http://DS-Ontology.owl#Reporting_Source
- #radar for class http://DS-Ontology.owl#Reporting_Source

The following figure illustrates through a non-formal ontological schema, how the instances are linked together.

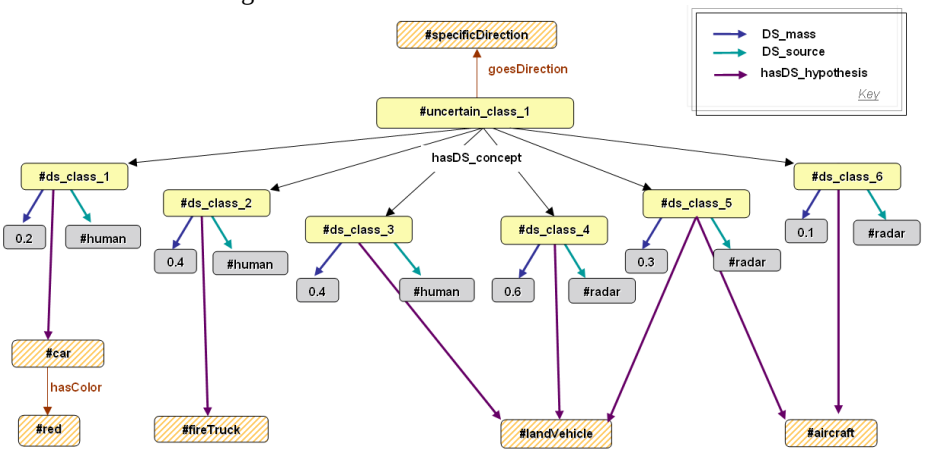


Fig. 3. Uncertain individuals scenario

4 Evidential Reasoning on DS-Ontology

Once the uncertainty contained in the information has been represented, reasoning processes have to be conducted to fuse the different observation and eventually decide of the instance with the most likelihood. This section has to be viewed as the chronological steps that are realized by the system in order to reason on the uncertain pieces of information represented through the DS-Ontology.

4.1 Generate automatically the Discernment Frame

One *Uncertain_concept* instance of the DS-Ontology groups a set of candidate instances together (either individuals or properties). From this set of instances, we want to determine automatically a consistent frame of discernment, according to the Dempster-Shafer theory. The underlying assumptions of the theory are: an exhaustive frame of discernment and the exclusivity of elements of Ω (see section 2.1). In this paper, we have already managed the first constraint within the Open World assumption of ontologies in the modelling of the DS-Ontology. The second constraint of the frame of discernment is the exclusivity of its elements. This implies that each singleton hypothesis (i.e. the elements of Ω) are disjoint. In other words, if H_1 and H_2 are two singletons, we cannot have $H_1 \subset H_2$ or even $H_1 \cap H_2 \neq \emptyset$. In the instantiation example, #fireTruck and #car individuals are semantically "included" in #landVehicle. As there is an inclusion, #fireTruck and #car individuals have also a non-null intersection with the #landVehicle individual. Moreover, #fireTruck has a non-null intersection with #car. Indeed, these two individuals are sharing many

characteristics in common: they are both land vehicles and their main colors are in both cases red.

To deal with this second constraint, we take into account the explicit and inferred semantics of the domain ontology to generate the discernment frame. The granularity of the set of candidate instances affects the generation of the discernment frame. The semantic will help us determining the inclusion of hypotheses as well as the semantic similarity between instances. The whole set of candidate instances will help us fixing a threshold for semantic distances.

4.1.1 Semantic Inclusion/Intersection

The semantic inclusion is quite straightforward to determine. Indeed, in case the instances are property assertions, for example if a property P1 has for ancestor P2, then we say that P1 is included in P2. Otherwise, in case the instances are individuals and they have zero or the same properties (or some included property), then there is an inclusion. In all other cases, the inclusion does not hold.

Concerning semantic intersection, things go a little further. First of all, logically, if two instances have already a semantic inclusion, then they also have a non-null semantic intersection. In all other cases, we will consider that two instances have a non-null intersection when their semantic similarity is exceeding a certain threshold. More specifically, our similarity measure is a global function, which combines existing similarity measure defined in literature. As for individuals, it is a mixture of similarity measure of their respective types and similarity measures concerning their relations. Wu & Palmer similarity measure [6] is used to qualify the similarity between two instances based on their respective type. It takes into account the distance that separates two types in the hierarchy and their position with the root. Equation (7) depicts their formula. $C1$ and $C2$ are two classes. Class C is the immediate mother-class of $C1$ and $C2$ that subsumes both classes. $depth(C)$ function is the number of edges separating C from the root. $depth_C(Ci)$ is the number of edges which separate Ci from the root while passing by C.

$$conSim(C1, C2) = \frac{2 * depth(C)}{depth_C(C1) + depth_C(C2)}. \quad (7)$$

The other combined similarity measures count the number of identical properties versus the number of different properties related to the two individuals. This is calculated both for object properties and datatype properties. On equation (8), $I1$ and $I2$ are the two individuals for which the global semantic similarity measure is calculated. For object properties (respectively for datatype properties), $nbProp(I)$ is the number of object properties (resp. of datatype properties) of individual I. $nbPropComm(I1,I2)$ is the number of common properties - identical predicate and related individual or value - for the two individuals $I1$ and $I2$. These three similarity measures focusing on the similarity of the types of individuals and their characteristics (through the datatype and object properties) are combined through a weighted mean.

$$propSim(I1, I2) = \frac{2 * nbPropComm(I1, I2)}{nbProp(I1) + nbProp(I2)}. \quad (8)$$

Once the cross-similarity measure of the set of all candidate instances is calculated, the threshold is fixed through a clustering method. The threshold is thus varying according to all the computed semantic similarities. This process permits to adapt the granularity of the set of candidate instances. It translates our general impression that the concept of a compact car is closer to the concept of minivan than of a plane's; however the concept of a compact car is closer to the concept of plane than of a book's. In the first case, the intersection should be brought by the pair (compact car, minivan), whereas in the latter, it should be brought by the pair of (compact car, plane). It should be noted that, in both cases, the concepts of compact car and plane have the same semantic similarity. As a consequence, the semantic intersection is seen as a Boolean condition on the similarity measure exceeding the threshold.
Finally, we consider the evidential set inclusion (respectively intersection) as equivalent to the semantic inclusion (respectively intersection). In case of our scenario, the intersection and inclusion are graphically represented on the figure below.

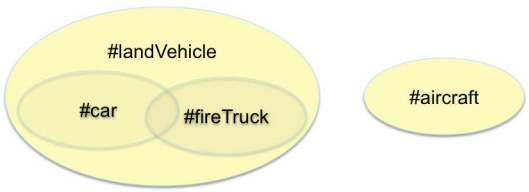


Fig. 4. Inclusion and intersection of candidate instances

4.1.2 From the Set of Candidates Instances to the Discernment Frame

Once the intersection and inclusion of candidate instances identified, we are able to set up a consistent frame of discernment. For this, we reframe the set of candidate instances into single or composite disjoint hypotheses.
In case of a discovered intersection between two candidate instances #inst1 and #inst2, #inst1 is reformulated as the union of two singletons $\{H_1, H_{inters}\}$ and #inst2 as $\{H_2, H_{inters}\}$. In case of discovered inclusions between two candidate instances #inst1 and #inst2, where #inst1 is included in #inst2, #inst1 is represented by a single hypothesis $\{H_1\}$ and #inst2 by the union of hypotheses $\{H_2, H_1\}$. Single hypotheses, grouped together, constitute the frame of discernment. In fact, each initial candidate instance belongs to the power-set of the frame of discernment. Taking our scenario, each candidate instance can now be decomposed as such:

- #aircraft = $\{H_1\}$
- #car = $\{H_2, H_3\}$
- #fireTruck = $\{ H_3, H_4\}$
- #landVehicle = $\{H_2, H_3, H_4, H_5\}$

Indeed, relying on Figure 4, #aircraft instance has no intersection nor inclusion; thus, it constitutes a single hypothesis within the frame of discernment. The non-null intersection, between #fireTruck and #car instances, has been modeled through a common and shared single hypothesis: H_3. Finally, the inclusion brought by #landVehicle results in the union of the set of single hypotheses of #fireTruck and #car, in addition to its own singleton H_5.

4.2 Use Dempster-Shafer Calculations on DS-Ontology

Once the discernment frame has been obtained, we can reformulate in the Dempster-Shafer formalism, the basic mass assignment of the scenario:

- $m_{radar}(\{H_2, H_3, H_4, H_5\}) = 0.6$; $m_{radar}(\{H_1\}) = 0.1$; $m_{radar}(\{H_1, H_2, H_3, H_4, H_5\}) = 0.3$
- $m_{human}(\{H_2, H_3\}) = 0.2$; $m_{human}(\{H_3, H_4\}) = 0.4$; $m_{human}(\{H_2, H_3, H_4, H_5\}) = 0.4$

We are now able to apply directly the classical combination rules found in the Dempster-Shafer theory, and then go through the decision process.

5 Related Work

During the last decade, approaches considering both uncertainty and the Semantic Web have been proposed. In this section, we mention some of them in order to position and compare our work. We consider their goal, underlying mathematical theory and processes.

Fuzzy and rough set theories aim to model vagueness and uncertainty. Regarding fuzzy sets, classes are considered to have unsharp definitions. fuzzyDL approach [7] aims to represent and reason about a membership function specifying the degree to which an instance belongs to a class. Even if it could be interesting to take into account fuzzy aspect of hypotheses especially those formulated by human sources, it is not the purpose of our approach to model more precisely our knowledge, but to decide among multi hypotheses and have a more coherent and reliable view of the situation. Approaches in [8, 9] are relying on rough set theory – which considers the indiscernability between objects. In that case, classes are not restricted to a crisp representation; they may be coarsely described with approximations. In [9], the author is using rough classes to generate new subclasses or relations by mining an important set of instances already existing. This can be part of the ontology engineering process. The goal is here also different to ours; however, some notions and process are similar. First, the design of a rough OWL ontology can be seen as the matching piece to our DS-Ontology for the Dempster-Shafer theory. Moreover, the use of p-indistinguishable properties notion for two individuals can be linked to our so-called common properties in Equation (8) when processing the similarity measure between two instances. Finally, descriptions for lower and upper approximation – through intersection and inclusion considerations - remind us the definition of the exclusivity of our frame of discernment; however, they consider here intersection and inclusion between two classes whereas we calculate it between two individuals.

Probabilistic adaptations or extensions (Pr-OWL [10], BayesOWL [11], Fire [12]) are more relevant to our objective of assessing the most likelihood instances that holds. However, probabilities suffer from the lack of ignorance and imprecision management in comparison to evidential theory.

Approaches in [13, 14 and 15] are more related to our chosen mathematical theory as they directly deal with evidential theory. [13] and [14] transform uncertain statements in belief networks. However, these network representations are themselves extensions of evidential theory. Moreover, they do not take into account the semantic attached to the hypotheses, in order to consider the most conflicting hypotheses or on

the inverse the implied hypotheses. Looking this way, they can be considered complementary to ours. A recent published approach [15] is concentrating on uncertain reasoning on instances of an ontology using the evidential theory and some similarity measures. While we handle the same mentioned tools, our process and aspiration are quite different. Indeed, their main objective is to propose an alternative ABox inductive reasoning - by classifying individuals (determining their class- or role- memberships or value for datatype properties) through a prediction based on an evidential nearest neighbor procedure. Their reasoning addresses here another way to tackle automatic inference from a classical ontology. This automatic inference aims to derive new or implicit knowledge about the current representation of the world, on the basis of the asserted knowledge. Whereas, our current reasoning goal is to rely on the semantic description of candidate instances (hypothesis) describing a same and unique entity or phenomenon in order to decide which candidate instances should be chosen.

Other reports enlarging the state-of-the-art to all ontology languages can be found in [17, 18].

6 Conclusion and Future Work

This paper proposes a solution in order to handle uncertainty within ontologies. Our approach is relying on current W3C standards. Modeling of uncertainty is realized through an imported pre-defined ontology: the DS-Ontology. Uncertain instantiation of the domain ontology is performed through the use of this imported DS-Ontology. The DS-Ontology relies on the theory of Dempster-Shafer, which manages uncertainty, as well as imprecision and ignorance. This paper has underlined some key issues that have to be dealt when implementing such parallelism between a formal mathematical theory to manage uncertainty and semantic world. The assumption of Open World in ontologies is one of these issues. Reasoning on uncertainty is made possible through an automatic generation of the frame of discernment. For that purpose, Boolean semantic operators, such as the intersection and inclusion, have been developed based on the semantic expressivity of the domain ontology. As a consequence, this paper provides a double and mutual contribution in the domains of the Semantic Web and of uncertain theories, which benefits clearly from the semantics of the hypotheses.

Further researches are also in discussion to extend the reasoning over the Boolean inclusion and intersection of candidate instances. Indeed, it could be interesting to keep the semantic similarity degree (which is a real ranging from 0 to 1) and use it instead of Boolean notions within the theory of Dempster-Shafer. This could be made by rearranging the basic measures of belief and plausibility and of the rules of combination.

References

1. Shafer, G.: A mathematical theory of evidence. Princeton University press Princeton, NJ, Volume 1 (1976)
2. Martin, A., Osswald, C., Dezert, J., Smarandache, F.: General combination rules for qualitative and quantitative beliefs, Journal of Advances in Information Fusion 3, 67-89, (2008)
3. Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S.: OWL 2 Web Ontology Language Primer. W3C Recommendation 27 October 2009, World Wide Web Consortium (2009)
4. Royère, C., Gruyer, D., Cherfaoui, V.: Data association with believe theory, In Proceedings of the Third International Conference on Information Fusion, Paris, France (2000)
5. Manola, F., Miller, E., McBride, B.: RDF Primer W3C Recommendation (2004)
6. Wu, Z., Palmer, M.: Verb semantics and lexical selection. In Proceedings of the 32nd Annual Meeting of the Associations for Computational Linguistics, pages 133–138, Las Cruces, New Mexico (1994)
7. Bobillo, F., Straccia, U.: FuzzyDL: An expressive fuzzy description logic reasoner, in IEEE International Conference on Fuzzy Systems 2008, pages 923–930. IEEE (2008)
8. Schlobach, S., Klein, M. C, Peelen, L.: Description Logics with Approximate Definitions: Precise Modeling of Vague Concepts. In Proceedings of the 20th International Joint Conference on Artificial Intelligence, IJCAI 07, Hyderabad, India (2007)
9. Keet, C.M.: Ontology engineering with rough concepts and instances. 17th International Conference on Knowledge Engineering and Knowledge Management (EKAW'10), P. Cimiano and H.S. Pinto (Eds.), Lisbon, Portugal, Springer LNAI 6317, 507-517 (2010)
10. Costa, P.C.G., Laskey, K.B.: PR-OWL: A framework for probabilistic ontologies, In Proceeding of the 2006 conference on Formal Ontology in Information Systems: Proceedings of the Fourth International Conference, pages 237–249, IOS Press (2006)
11. Ding, Z., Peng, Y., Pan, R.: BayesOWL: Uncertainty modeling in semantic web ontologies, Soft Computing in Ontologies and Semantic Web, pages 3–29 (2006)
12. Simou, N., Kollias, S.: Fire: A fuzzy reasoning engine for imprecise knowledge, In K-Space PhD Students Workshop, Berlin, Germany, volume 14, Citeseer (2007)
13. Essaid, A., Yaghlane, B.B.: BeliefOWL: An Evidential Representation in OWL Ontology, In International Semantic Web Conference, International Workshop on Uncertainty Reasoning for the Semantic Web, Washington DC, USA, page 77. Citeseer (2009)
14. Nikolov, A., Uren, V., Motta, E., de Roeck, A.: Using the Dempster-Shafer Theory of Evidence to Resolve ABox Inconsistencies, Uncertainty Reasoning for the Semantic Web I Lecture Notes in Computer Science, Volume 5327/2008, 143-160 (2008)
15. Fanizzi, N., d'Amato, C., Esposito, F.: Evidential Nearest-Neighbors Classification for Inductive ABox Reasoning. in Workshop on Uncertainty Reasoning for the Semantic Web URSW 2009, Washington D.C. - USA (2009)
16. Bellenger, A., Gatepaille, S.: Uncertainty in Ontologies: Dempster-Shafer Theory for Data Fusion Applications, in Workshop on the Theory of Belief Functions, Brest – France (2010)
17. Predoiu, L., Stuckenschmidt, H.: Probabilistic Models for the Semantic Web The Semantic Web for Knowledge and Data Management: Technologies and Practices (2009)
18. Lukasiewicz, T., Straccia, U.: Managing uncertainty and vagueness in description logics for the semantic web, Web Semantics: Science, Services and Agents on the World Wide Web, Elsevier, 6, 291-308 (2008)

Representing Sampling Distributions In P-$\mathcal{SROIQ}$

Pavel Klinov[1] and Bijan Parsia[2]

[1] University of Arizona, AZ, USA pklinov@email.arizona.edu
[2] The University of Manchester, UK bparsia@cs.man.ac.uk

Abstract. We present a design for a (fragment of) Breast Cancer ontology encoded in the probabilistic description logic P-$\mathcal{SROIQ}$ which supports determining the consistency of distinct statistical experimental results which may be described in diverse ways. The key contribution is a method for approximating sampling distributions such that the inconsistency of the approximation implies the statistical inconsistency of the continuous distributions.

1 Introduction

The current amount of knowledge about breast cancer is overwhelming. For example, a meta-study conducted in 2006 by Key et al. [4] covered 98 *unique* studies focused only on the impact of a single risk factor, alcohol consumption. At the same time there are no common knowledge bases which would combine and formally represent findings produced by the multitude of studies.[3] This makes it difficult to have a global view of breast cancer risk factors and, consequently, develop tools like risk assessment calculators.

The probabilistic description logic P-$\mathcal{SROIQ}$ can be used to represent general knowledge about breast cancer in the form of a probabilistic ontology (the BRC ontology) [5]. However, a general knowledge ontology need not support risk entailments for various combinations of risk factors — that is, compete (poorly) with narrowly specific risk calculators[4] which have a direct implementation of simple equations derived from statistical risk models (such as the Gail model [2]). Instead, its main goal is to formally and unambiguously describe the background theory of breast cancer embracing as many reliable findings as possible and serving as a common knowledge base for more specific tools, such as risk assessment calculators or decision support systems. This sort of task seems to be a better fit for a probabilistic logic.

The set of use cases for the general knowledge ontology is wider than for the BRC ontology. In addition to maintaining a birds-eye view of breast cancer, it may be used for finding and analyzing inconsistencies in outcomes of different

[3] There are some lower level databases, such as ROCK (http://rock.icr.ac.uk/)—a cancer specific functional genomic database. However, they do not explicitly represent case study findings and do not support such services as risk assessment.
[4] Such as http://www.cancer.gov/bcrisktool

studies. It can support studying mechanisms of interactions between risk factors, for example, how alcohol consumption affects estrogen level. Finally, it may play a useful role in planning and coordination of future medical studies by helping to identify the most controversial or insufficiently studied risk factors or exposures.

In this paper, we present a design of general P-$\mathcal{SROIQ}$ ontology about breast cancer (i.e., the BRC ontology) which incorporates a substantial amount of statistical knowledge. While we do not present a fully fleshed out instance of this design, we do tackle a major representational challenge, namely, the representation of the statistical results of experiments. We present a method for approximate representations of different sampling distributions and their use in determining consistency between experimental data.

2 Preliminaries of P-$\mathcal{SROIQ}$

P-$\mathcal{SROIQ}$ [8] is a probabilistic extension of the DL $\mathcal{SROIQ}$ [3]. It provides means for expressing probabilistic relationships between arbitrary $\mathcal{SROIQ}$ concepts and a certain class of probabilistic relationships between classes and individuals. Any $\mathcal{SROIQ}$, and thus OWL 2 DL (as it can be seen as a notational variant of $\mathcal{SROIQ}$), ontology can be used as a basis for a P-$\mathcal{SROIQ}$ ontology, which facilitates transition from classical to probabilistic ontologies. We presume the reader is reasonbly familiar with class/object oriented description logics such as $\mathcal{SROIQ}$, though very little in this paper turns on specific details.

The only syntactic construct in P-$\mathcal{SROIQ}$ (in addition to all of the $\mathcal{SROIQ}$ syntax) is the conditional constraint.

Definition 1 (Conditional Constraint). *A conditional constraint is an expression of the form $(D|C)[l,u]$, where C and D are concept expressions in $\mathcal{SRIQ}$ (i.e., $\mathcal{SROIQ}$ without nominals) called* ***evidence*** *and* ***conclusion****, respectively, and $[l,u] \subseteq [0,1]$ is a closed real-valued interval. In the case where C is $\top$ the constraint is called* ***unconditional****.*

Ontologies in P-$\mathcal{SROIQ}$ are separated into a classical and a probabilistic part. It is assumed that the set of individual names N_I is partitioned onto two sets: classical individuals N_{CI} and probabilistic individuals N_{PI}.

Definition 2 (PTBox, PABox, and Probabilistic Knowledge Base). *A* ***probabilistic TBox*** *(PTBox) is a pair $PT = (\mathcal{T}, \mathcal{P})$ where $\mathcal{T}$ is a classical (finite) $\mathcal{SROIQ}$ TBox and $\mathcal{P}$ is a finite set of conditional constraints. A* ***probabilistic ABox*** *(PABox) is a finite set of conditional constraints associated with a probabilistic individual $o_p \in N_{PI}$. A* ***probabilistic knowledge base*** *(or a probabilistic ontology) is a triple $PO = (\mathcal{T}, \mathcal{P}, \{\mathcal{P}_{o_p}\}_{o_p \in N_{PI}})$, where the first two components define a PTBox and the last is a a set of PABoxes.*

Informally, a PTBox constraint $(D|C)[l,u]$ expresses a conditional statement of the form "if a *randomly* chosen individual is an instance of C, the probability of it being an instance of D is in $[l,u]$". A PABox constraint, which we write

as $(D|C)_o[l,u]$ where o is a probabilistic individual, states that "if a *specific* individual (that is, o) is an instance of C, the probability of it being an instance of D is in $[l,u]$". For more details we refer the reader to [8].

3 The Classical Part

The classical part of a P-$\mathcal{SROIQ}$ ontology (or OWL part) provides a medical vocabulary which can be used on its own in a variety of applications or used in the representation of probabilistic knowledge. In this paper we focus on providing an OWL terminology for probabilistic statements. The ontology contains the following main class hierarchies (taxonomies):

Taxonomy of breast cancers Breast cancer is a heterogeneous disease. Some risk factors can be associated with increase in risk of developing one particular type of breast cancer and not the other. Thus it is important to classify types of breast cancer. In particular, our ontology distinguishes breast cancers by hormone receptor status. Estrogen and progesterone positive breast cancers are modeled using concepts `ERPositiveBRC` and `PRPositiveBRC` while their complements are modeled using `ERNegativeBRC` and `PRNegativeBRC` (we use shorthands ER+/- and PR+/- with obvious meaning.). Another important classification is based on histology. The ontology distinguishes between invasive and non-invasive (e.g. in situ) cancers.

Taxonomy of risk factors Dozens of risk factors are known so far. Some are established and strongly associate with increased risks, such as BRCA1(2) gene mutations, while others are controversial. The ontology should provide vocabulary for both to support current and future findings. It includes a taxonomy of concepts rooted at `RiskFactor`. We distinguish between known risk factors (those which can be reported via a questionnaire, such as alcohol intake) and inferred risk factors which require medical examination.

Taxonomy of risks The ontology differentiates absolute and relative risks of developing breast cancer. Absolute risks are further divided into the lifetime risk and the short-term risk. Relative risks are divided into increased and reduced risks. Level of increases is a continuous variable which requires discretization (see below).

The last two taxonomies induce the corresponding classifications of women, i.e., classes of women w.r.t. risk factors and w.r.t. risk. For example, any risk factors `RF` gives rise to a class of women $\texttt{Woman} \sqcap \exists \texttt{hasRiskFactor.RF}$. Women having various combinations of risk factors are modeled as conjunctive concept expressions. Analogously, given a certain kind of risk `R` the expression $\texttt{Woman} \sqcap \exists \texttt{hasRisk.R}$ models those women who are in the risk group `R`, for example, have moderately increased risk of developing ER+ breast cancer. These taxonomies of women may or may not be explicitly present in the ontology. In other words, it is possible, but not essential, to generate a concept name for each interesting class of women since P-$\mathcal{SROIQ}$ (and our reasoner Pronto) allows for complex concept expressions in conditional constraints.

A future, more complete version of the ontology would certainly make use of existing bio-medical ontologies which cover substantial portions of the domain either by direct reuse or by ontology alignment techniques.

4 The Method for Approximating Distributions in the Probabilistic Part

The probabilistic part of the ontology captures statistical background knowledge about breast cancer. We distinguish between knowledge which explicitly associates quantifies specific risk factors and more general statistical relationships which are not necessarily risk related. The distinction could be useful for importing knowledge from other medical ontologies. We begin with the latter.

General statistical knowledge mostly includes relationships between various risk factors. For example, Ashkenazi Jew women are more likely to develop BRCA gene mutations, while early menarche, late first child (or no live births), lack of breastfeeding and alcohol consumption all increase levels of estrogen in blood.[5] Such relationships are important because they can help to infer the presence of some risk factors given the set of known factors. They are typically easy to represent by using conditional constraints of the form `(Woman`$\sqcap\exists$`hasRiskFactor.RFY|Woman`$\sqcap\exists$`hasRiskFactor.RFX)[l,u]` which says that the chances of having risk factor RFY given RFX are between l and u. One possible source of complications is continuous variables, e.g. the level of estrogen, which are discussed below.

Most of statistical findings available in medical literature quantitatively describe risk increase for categories of women with specific risk factors. Such findings are presented by giving estimated parameters of a probability distribution where the random variable represents the relative risk of a random woman in the population. Such parameters include the estimated mean value and the estimated variance. Table 1 presents an example of the reported association between alcohol intake and the risk increase among postmenopausal women taken from [10]. There are two main difficulties with representing this kind of data in P-$\mathcal{SROIQ}$. First, the risk increase is a continuous random variable so it needs to be discretized. Second, the available language supports only conditional constraints so a straightforward encoding of probability distributions is not possible.

Table 1. Example of a reported association between alcohol intake and the risk of hormone receptor-specific breast cancer (excerpt from [10])

Alcohol (g)	**ER+**	**ER-**	**PR+**	**PR-**
	RR (95% CI)	**RR (95% CI)**	**RR (95% CI)**	**RR (95% CI)**
0	1.00 (ref)	1.00 (ref)	1.00 (ref)	1.00 (ref)
≤4	1.06 (0.91 - 1.22)	1.40 (1.00 - 1.96)	1.04 (0.89 - 1.23)	1.24 (0.95 - 1.62)
≥4	1.07 (0.90 - 1.26)	1.64 (1.14 - 2.35)	1.12 (0.93 - 1.34)	1.28 (0.96 - 1.71)

[5] See `http://tinyurl.com/4jpsdvk`

Discretization of a continuous variable is technically straightforward. We introduce a set of disjoint concept names each of which models women in the corresponding group of risk. Specifically, we define concepts `WomenAtWeakRisk`, `WomenAtModerateRisk` and `WomenAtHighRisk` with the obvious meanings described using OWL 2 datatype support to describe the exact boundaries. We have chosen ranges (1, 1.5], (1.5, 3.0] and (3.0, $+\inf$) respectively.[6]

The inability to represent distributions is a more severe limitation. It leaves the modeler with the only option of *approximating* the continuous distribution using a finite set of points. In other words, each distribution, for example, risk increase for women consuming a certain amount of alcohol, can be approximated by specifying the probability that a *randomly* taken woman with the given exposure belongs to a specific group of risk, i.e. `WomenAtWeakRisk`, `WomenAtModerateRisk` or `WomenAtHighRisk`. This *is* the semantics of P-$\mathcal{SROIQ}$ conditional constraints.

Assuming that the random variable is real-valued, a standard way of approximating a continuous distribution is to take each interval and compute the probability that the variable takes on a value in that interval. Then the approximation of a distribution $Pr(x)$ w.r.t. a finite set of intervals U is simply a function $\hat{Pr}$ such that $\hat{Pr}(U_i) = \int\limits_{U_i} Pr(x)\mathrm{d}x$.

Unfortunately, this approximation of results of statistical experiments is unsatisfactory because it maps every interval to a single point. The problem is that *any* arbitrarily small difference between two or more sampling distributions will results in conflicting probabilistic statements for every interval (because the point-valued probabilities will be different) even though the results can confirm each other from a purely statistical point of view. Consequently this approach does not support working with results reported by multiple studies.

Our goal is to approximate sampling distributions in P-$\mathcal{SROIQ}$ in a *statistically coherent* way. Informally it means that satisfiability of probabilistic formulas representing two or more sampling distributions must agree with their mutual statistical consistency, i.e., whether they support a common statistical hypothesis. The hypothesis, in this case, is that there exists a distribution (not necessarily a unique one) over G with parameters μ, σ such that it is supported by all sampling distributions with the required level of confidence.

We assume a (finite) population G of size N_G and a random variable X which is normally distributed across G. We also make the realistic assumption that G is large enough so that evaluating X for all members of G is not feasible. A common approach is to take one or more random samples from G, evaluate X for them and estimate the actual distribution over G based on the sampling distributions. We use μ, σ to denote the mean and the variance of the actual distribution and $\overline{X^{(i)}}, S^{(i)}$ for the mean and the variance of the sample $X^{(i)}$. For simplicity we finally assume that the population distribution is normal.

The mainstream approach for comparing two or more sampling distributions is based on statistical hypothesis tests. For example, given two normal distribu-

[6] The choice of intervals is obviously ambiguous but this issue is orthogonal to the approximation method presented in this paper.

tions $\overline{X^{(1)}}, S^{(1)}, \overline{X^{(2)}}, S^{(1)}$ it is common to take $\overline{X^{(1)}} - \overline{X^{(2)}}$, which is a normally distributed random variable, and perform a *z-test* (or a Student's t-test depending on the sample sizes) to see if the difference can be taken as 0 with the required level of confidence. It amounts to calculating standard errors of the mean (SE) for both distributions and then computing the difference *in units of SE*. If the probability of observing such difference given the null hypothesis,[7] which can be found in standard tables, is low enough, e.g., ≤ 0.05, a statistician would accept the hypothesis that both distributions are consistent.

Our approach is slightly different from the outlined above. It is not based on tests but on *confidence regions* for sampling distributions. The approach, which generalizes confidence intervals and dates back to Mood [9], is to estimate a region $\mathcal{R}_\gamma$ in the parameter space for (μ, σ^2) such that it will contain the μ, σ^2 pair of the actual distribution $100(1-\gamma)\%$ times as the number of estimations goes to infinity. More formally, a $100(1-\gamma)\%$ confidence region $\mathcal{R}_\gamma$ is a *random set* for parameters (μ, σ^2) based on a group of independent normally distributed variables X (i.e., a sample) such that [1]:[8]

$$P((\mu, \sigma^2) \in \mathcal{R}_\gamma) = 1 - \gamma, \text{ for all } (\mu, \sigma^2) \tag{1}$$

Informally, the confidence region specifies how far sampling distributions can deviate from the population distribution while supporting it with $100(1-\gamma)\%$ confidence. Following Mood [9] we will show that for the normal distribution the region is a convex set and, therefore can be represented by boundary values of (μ, σ^2) such that *any* sampling distribution inside the boundary will be consistent with the current distribution.

Consider the sample $X_1, \ldots, X_n$ where all X_i are independent random variables with the normal distribution $(N(\mu, \sigma^2))$. Then $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$ and $S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2$, i.e., the sample mean and the sample variance, are random variables. It is well known that $\overline{X}$ has the normal distribution $N(\mu, \frac{\sigma^2}{n})$ (or, equivalently, $\frac{\overline{X}-\mu}{\sigma/\sqrt{n}} \sim N(0,1)$) while $(n-1)S^2/\sigma^2$ has the chi-square distribution with $n-1$ degrees of freedom [9].

The standard tables for $N(0,1)$ and χ^2_{n-1} provide numbers a, b, c such that for fixed p_1, p_2 the following equalities hold [1]:

$$P(-a < \frac{\overline{X} - \mu}{\sigma/\sqrt{n}} < a) = p_1,$$
$$P(b < (n-1)S^2/\sigma^2 < c) = p_2$$

[7] The null hypothesis is a default position which, in this case, could be that the population mean is different from at least one of $\overline{X^{(1)}}, \overline{X^{(2)}}$.

[8] We deliberately leave out a precise definition of random set. For the purposes of this paper it is sufficient to think of a random set as of a random variable which takes on subsets of some space.

The crucial fact is that the two random variables are independent (see [9] for a proof) which implies that:

$$p_1 p_2 = P(-a < \frac{\overline{X} - \mu}{\sigma/\sqrt{n}} < a, b < \frac{(n-1)S^2}{\sigma^2} < c) =$$
$$P(\overline{X} - a\frac{\sigma}{\sqrt{n}} < \mu < \overline{X} + a\frac{\sigma}{\sqrt{n}}, \frac{(n-1)S^2}{c} < \sigma^2 < \frac{(n-1)S^2}{b})$$

Thus, the $100(p_1)(p_2)\%$ confidence region for (μ, σ^2) takes the following form:

$$\mathcal{R}_{p_1,p_2}(\overline{X}, S) = \left\{(\mu, \sigma^2) : \ \overline{X} - \alpha\frac{\sigma}{\sqrt{n}} < \mu < \overline{X} + \alpha\frac{\sigma}{\sqrt{n}}, \right. \tag{2}$$
$$\left. \frac{(n-1)S^2}{\gamma} < \sigma^2 < \frac{(n-1)S^2}{\beta} \right\}$$

Figure 1 shows the joint confidence region $\mathcal{R}$ in the parameter space (μ, σ^2). Note that it is possible, although technically messy, to generalize the definition (2) to the case of several independent sampling distributions. The simultaneous confidence region for k samples $X^{(1)}, \ldots, X^{(k)}$ will be a region in the $2k$-dimensional parameter space which projections on each plane $(\mu^{(i)}, (\sigma^{(i)})^2)$ will look as (2). Then the notion of consistency of sampling distributions can be defined as follows (we limit the attention to two samples for clarity):

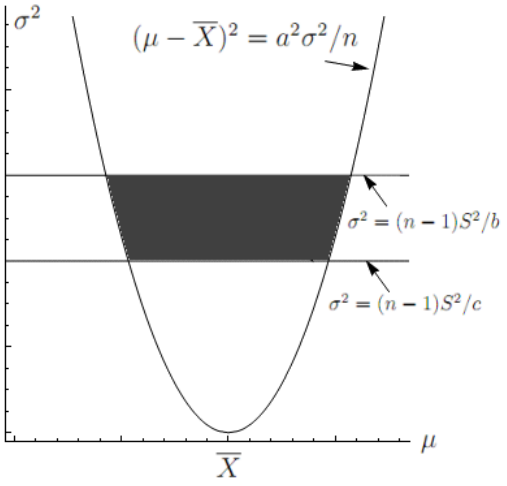


Fig. 1. Joint confidence region for (μ, σ^2)

Definition 3. *Let $Pr(X^{(1)}), Pr(X^{(2)})$ be distributions on two samples $X^{(1)}, X^{(2)}$ drawn independently from a population G. They are said to be consistent with confidence $100p\%$ if there exists a point (μ, σ^2) which belongs to both $\mathcal{R}_p(\overline{X^{(1)}}, S^{(1)})$ and $\mathcal{R}_p(\overline{X^{(2)}}, S^{(2)})$.*

Now we can return to the issue of approximating a continuous sampling distribution by a discrete set of points. Assume that the domain E of a continuous real-valued random variable X is a disjoint union of a finite number of intervals $U = \{(-\infty, r_1], (r_1, r_2], \ldots, (r_{l-1}, r_l], (r_l, +\infty)\}$. Then the *approximation* of the sampling distribution $Pr(X)$ with mean and variance $(\overline{X}, S^2)$ is the function $\hat{Pr}$ which maps each interval U_i to the following real-valued set:

$$\hat{Pr}(U_i; \overline{X}, S) = \{g(\mu, \sigma^2) | (\mu, \sigma^2) \in \mathcal{R}_{p_1,p_2}(\overline{X}, S)\} \tag{3}$$
$$g(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \int\limits_{U_i} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \mathrm{d}x$$

Now we are ready to define the notion of approximate consistency of sampling distribution with respect to a set of intervals U:

Definition 4. *Two sampling distributions $Pr(X^{(1)}), Pr(X^{(2)})$ are approximately consistent given a finite set of intervals U if $\hat{Pr}(U_i; \overline{X^{(1)}}, S^{(1)}) \cap \hat{Pr}(U_i; \overline{X^{(2)}}, S^{(2)})$ is non-empty for all $U_i \in U$.*

As with any approximation, the utility of approximations of sampling distributions depends on what conclusions they help to draw about the distributions themselves. Given that we are interested in the matter of consistency, it is important to understand the relationships between the notions of consistency and approximate consistency of sampling distributions. Fortunately, consistency implies approximate consistency regardless of partitioning of the real line:

Theorem 1. *If two sampling distributions $Pr(X^{(1)}), Pr(X^{(2)})$ are consistent, then they are approximately consistent for any choice of real-valued intervals.*

Proof. For the distribution $Pr(X^{(1)})$ a confidence region $\mathcal{R}_{p_1,p_2}(\overline{X^{(1)}}, S^{(1)})$ is connected (see Definition 2). The function $g(\mu, \sigma^2)$ (Definition 3) is continuous on it which implies that for any U_i, the set $\hat{Pr}(U_i; \overline{X^{(1)}}, S^{(1)})$ is a real-valued interval (l_1, u_1). Now consider a point $\mu_0, \sigma_0^2 \in \mathcal{R}_{p_1,p_2}(\overline{X^{(1)}}, S^{(1)}) \cap \mathcal{R}_{p_1,p_2}(\overline{X^{(2)}}, S^{(2)})$ which exists since the distributions are consistent. It follows that $l_1 < g(\mu_0, \sigma_0^2) < u_1$ (and analogously $l_2 < g(\mu_0, \sigma_0^2) < u_2$ for $\hat{Pr}(U_i; \overline{X^{(2)}}, S^{(2)})$), so $g(\mu_0, \sigma_0^2)$ is a common point for both approximations on U_i. As such the distributions are approximately consistent.

The following corollary from the above theorem is at heart of our method. As we demonstrate below, the inconsistency of approximations can be proved by logical reasoning in P-$\mathcal{SROIQ}$ (i.e., by solving the probabilistic satisfiability problem), which means that the result enables approximate reasoning about

sampling distributions in a purely logical way. Even though the power of such reasoning is currently limited to consistency checking, its integration with OWL/DL reasoning and the ability to use common, formally defined terminology for representation of statistical experiments is promising.

Corollary 1. *If sampling distributions $Pr(X^{(1)}), Pr(X^{(2)})$ are approximately inconsistent for some choice of real-valued intervals, then they are inconsistent.*

5 Example of Approximate Modelling

Now we present an example of approximate representation of sampling distributions in P-$\mathcal{SROIQ}$. The task is to take two results of statistical experiments aimed at investigating associations between alcohol consumption and the increased risk of breast cancer among postmenopausal women. Unfortunately it is common for medical papers to not explicitly present all parameters that characterize results of their statistical analyses. Typically, only the estimated mean and the confidence interval are presented while, for example, the kind of distribution is left to the reader to infer from other information. Due to that fact and because the approach above has only been developed for normal distributions, we illustrate it on an artificial example. The information given in the example is analogous to that given in medical literature, e.g. [10, 11], but is complete in the sense that all parameters and the type of sampling distributions are known.

Example 1. Consider two hypothetical papers which report results of independent studies of associations between alcohol consumption among postmenopausal women and their relative risk of developing breast cancer. According to study A the mean relative risk (RR) of ER+ breast cancer for women drinking $\geq$ 4g of ethanol a day is 1.8 and has variance of 0.5. Study B has reported that the mean RR of ER+ breast cancer for the same level of drinking is 2.2 (variance 0.7). The number of cases in the studies was 230 and 150 respectively.

We propose the following four step procedure for an approximate representation of statistical results, similar to those in the example above, in P-$\mathcal{SROIQ}$:

1. Preparing concepts The first step is to define the concepts/roles used to describe the distribution. In our case evidence concepts should describe categories of women with respect to specific risk factors, e.g. alcohol intake, while conclusion concepts describe groups of women stratified by risk increase. For instance, the concept expression $\mathtt{C} \equiv \mathtt{Woman} \sqcap \exists \mathtt{hasRiskFactor}.(\mathtt{Postmenopause} \sqcap \mathtt{ModerateConsumption})$ is used to model postmenopausal women with moderate level of alcohol intake.[9] On the other hand the expression:

$$D \equiv \mathtt{Woman} \sqcap \exists \mathtt{hasRisk}.(\mathtt{ModeratelyIncreasedRisk} \sqcap \exists \mathtt{riskOf}.\mathtt{ERPositiveBRC})$$

[9] The level of intake is a continuous variable which we also split onto categories `LimitedConsumption`, `ModerateConsumption` and `HeavyConsumption` which correspond to ≤ 4, $4 - 9.9$ and ≥ 10g of ethanol per day.

models women who are at moderately increased risk of developing ER-positive breast cancer. Using these expressions the modeler can specify the probability than a random women the class `C` also belong the risk group `D` as `(D|C)[l,u]`.

2. Determining parameters of sampling distributions (if required) Sometimes parameters of sampling distributions can be determined from other information. For example, knowing the kind of distribution, sample mean, sample size, confidence interval and the methodology of its estimation, one can calculate the sample variance.[10] In our case it is not needed as the distributions are normal and the parameters are known.

3. Choosing intervals Choice of intervals for an approximation of a continuous random variable is driven by balancing the quality of the approximation (i.e., how closely it models the continuous distribution) and the number of statements required. The latter has a direct impact on performance. For Example 1 we use three concepts `WomenAtWeakRisk`, `WomenAtModerateRisk` and `WomenAtHighRisk` which correspond to relative risk intervals of `(1,1.5]`, `(1.5, 3.0]` and `(3.0,`$+\infty$`)` respectively.

4. Computing the approximation The final (and the central) step is to compute probability intervals for the statements that approximate the continuous distribution. Each statement specifies the lower and upper probabilities that the continuous random variable X will fall into an interval U_i given that parameters of the distribution can vary within the confidence region (2). More formally, given the interval U_i, e.g. `(1,1.5]` for `WomenAtWeakRisk`, and the sampling distribution $(\overline{X}, S^2)$ the interval $[l_i, u_i]$ can be computed by solving the following non-linear optimization problem ():

$$\begin{aligned} & l_i \text{ (resp. } u_i) = \ min \text{ (resp. } max) \ g(\mu, \sigma^2) \text{ s.t.} \\ & (\mu, \sigma^2) \in \mathcal{R}_{p_1,p_2}(\overline{X}, S) \\ & g(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \int\limits_{U_i} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \mathrm{d}x \end{aligned} \tag{4}$$

In other words, $[l_i, u_i]$=$[inf\ \hat{Pr}(U_i; \overline{X}, S), sup\ \hat{Pr}(U_i; \overline{X}, S)]$.

The last preparatory step is to calculate confidence regions according to (2). The 95% confidence regions for distributions $(\overline{X^{(1)}}, S^{(1)}), (\overline{X^{(2)}}, S^{(2)})$ in Example 1 (abbreviated as $R^{(1)}_{0.95}$ and $R^{(2)}_{0.95}$) are defined by the following inequalities:

$$R^{(1)}_{0.95} = \left\{ (\mu, \sigma^2) : \ 1.8 - \frac{2.241\sigma}{\sqrt{230}} < \mu < 1.8 + \frac{2.241\sigma}{\sqrt{230}}, 0.409 < \sigma^2 < 0.623 \right\}$$

$$R^{(2)}_{0.95} = \left\{ (\mu, \sigma^2) : \ 2.2 - \frac{2.241\sigma}{\sqrt{150}} < \mu < 2.2 + \frac{2.241\sigma}{\sqrt{150}}, 0.548 < \sigma^2 < 0.923 \right\}$$

[10] The variable $T = (\overline{X} - \mu)/(S/\sqrt{n})$ has the t-dustribution with $n-1$ degrees of freedom. Confidence interval is standardly computed as $[\overline{X} - a, \overline{X} + a]$ where $a = t_{\frac{1-\alpha}{2}, n-1} \frac{S}{\sqrt{n}}$ ($t_{\frac{1-\alpha}{2}, n-1}$ is the $\alpha-$percentile of the Student distribution). If the confidence interval and α are known, then S can be calculated.

Now the optimization problem (4) can be solved numerically[11] to obtain the following approximations for both sampling distributions:

$$\begin{aligned}
inf\ \hat{Pr}((1,1.5];\overline{X^{(1)}},S^{(1)}) &= 0.219 & sup\ \hat{Pr}((1,1.5];\overline{X^{(1)}},S^{(1)}) &= 0.298\\
inf\ \hat{Pr}((1.5,3.0];\overline{X^{(1)}},S^{(1)}) &= 0.655 & sup\ \hat{Pr}((1.5,3.0];\overline{X^{(1)}},S^{(1)}) &= 0.878\\
inf\ \hat{Pr}((3.0,+\infty);\overline{X^{(1)}},S^{(1)}) &= 0.239 & sup\ \hat{Pr}((3.0,+\infty);\overline{X^{(1)}},S^{(1)}) &= 0.586\\
inf\ \hat{Pr}((1,1.5];\overline{X^{(2)}},S^{(2)}) &= 0.116 & sup\ \hat{Pr}((1,1.5];\overline{X^{(2)}},S^{(2)}) &= 0.224\\
inf\ \hat{Pr}((1.5,3.0];\overline{X^{(2)}},S^{(2)}) &= 0.562 & sup\ \hat{Pr}((1.5,3.0];\overline{X^{(2)}},S^{(2)}) &= 0.769\\
inf\ \hat{Pr}((3.0,+\infty);\overline{X^{(2)}},S^{(2)}) &= 0.189 & sup\ \hat{Pr}((3.0,+\infty);\overline{X^{(2)}},S^{(2)}) &= 0.568
\end{aligned}$$

So, for this example, the sampling distributions are approximately represented in P-$\mathcal{SROIQ}$ using the following conditional constraints.

$\{(\mathtt{W} \sqcap \exists \mathtt{hR}.(\mathtt{WeaklyIncreasedRisk} \sqcap \exists \mathtt{riskOf}.\mathtt{ERPositiveBRC})|C)[0.219, 0.298],$
$(\mathtt{W} \sqcap \exists \mathtt{hR}.(\mathtt{ModeratelyIncreasedRisk} \sqcap \exists \mathtt{riskOf}.\mathtt{ERPositiveBRC})|C)[0.655, 0.878],$
$(\mathtt{W} \sqcap \exists \mathtt{hR}.(\mathtt{StronglyIncreasedRisk} \sqcap \exists \mathtt{riskOf}.\mathtt{ERPositiveBRC})|C)[0.239, 0.586]\}$
and
$\{(\mathtt{W} \sqcap \exists \mathtt{hR}.(\mathtt{WeaklyIncreasedRisk} \sqcap \exists \mathtt{riskOf}.\mathtt{ERPositiveBRC})|C)[0.116, 0.224],$
$(\mathtt{W} \sqcap \exists \mathtt{hR}.(\mathtt{ModeratelyIncreasedRisk} \sqcap \exists \mathtt{riskOf}.\mathtt{ERPositiveBRC})|C)[0.562, 0.769],$
$(\mathtt{W} \sqcap \exists \mathtt{hR}.(\mathtt{StronglyIncreasedRisk} \sqcap \exists \mathtt{riskOf}.\mathtt{ERPositiveBRC})|C)[0.189, 0.568]\}$
where $\mathtt{W}$ and $\mathtt{hR}$ abbreviate $\mathtt{Woman}$ and $\mathtt{hasRisk}$, respectively and
$\mathtt{C} \equiv \mathtt{W} \sqcap \exists \mathtt{hasRiskFactor}.(\mathtt{Postmenopause} \sqcap \mathtt{ModerateConsumption})$

Probabilistic consistency of the above set of statements can be proved by solving the probabilistic satisfiability problem (PSAT). Modern algorithms can decide PSAT for over a thousand of P-$\mathcal{SROIQ}$ statements (in addition to thousands of OWL axioms), so the method could be computationally practical [6].

6 Conclusion

Checking consistency of sampling distributions in P-$\mathcal{SROIQ}$ may well appear cumbersome and pointless given that the same task can be done in a much simpler way and without any logical reasoning, e.g. via testing or by analyzing confidence regions. However, our aim is *not* to reduce statistical testing to logical reasoning (that aim is indeed pointless). Our aim is to represent results of statistical experiments using *common, unambiguously defined logical vocabulary* and be able to reason about them. Even though probabilistic reasoning about statistical results is currently limited to approximate consistency checking, the

[11] We use Wolfram Mathematica for this purpose.

potential benefits are in combining it with reasoning about the classical knowledge. For example, the BCRA ontology contains a little taxonomy of breast cancers by hormone receptor status. This enables us to combine results of the studies which are of different levels of granularity. For instance, Sellers et al. [10] report associations between alcohol intake and ER(+/-) breast cancer risk, while Suzuki et al. [11] divide it further to ER(+/-)PR(+/-) risks. In that simple case non-logical reasoning about the reported results becomes much less straightforward, while studies can also distinguish histologic types of breast cancer (see [7]). In such complex situations reasoning about findings does involve reasoning about background knowledge, e.g. the taxonomy of breast cancers, so a combination of OWL and probabilistic reasoning is potentially beneficial.

References

1. Arnold, B.C., Shavelle, R.M.: Joint confidence sets for the mean and variance of a normal distribution. The American Statistician 52(2), 133–140 (1998)
2. Gail, M.H., Brinton, L.A., Byar, D.P., Corle, D.K., Green, S.B., Schairer, C., Mulvihill, J.J.: Projecting individualized probabilities of developing breast cancer for white females who are being examined annually. Journal of the National Cancer Institute 81(25), 1879–1886 (1989)
3. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible $\mathcal{SROIQ}$. In: Knowledge Representation and Reasoning. pp. 57–67 (2006)
4. Key, J., Hodgson, S., Omar, R.Z., Jensen, T.K., Thompson, S.G., Boobis, A.R., Davies, D.S., Elliott, P.: Meta-analysis of studies of alcohol and breast cancer with consideration of the methodological issues. Cancer Causes Control 17, 759–770 (2006)
5. Klinov, P., Parsia, B.: Probabilistic modeling and OWL: A user oriented introduction into P-$\mathcal{SHIQ}$(D). In: OWL: Experiences and Directions (2008), `http://www.webont.org/owled/2008/papers/owled2008eu_submission_32.pdf`
6. Klinov, P., Parsia, B.: A hybrid method for probabilistic satisfiability. In: CADE. pp. 354–368 (2011)
7. Lew, J.Q., Freedman, N.D., Leitzmann, M.F., Brinton, L.A., Hoover, R.N., Hollenbeck, A.R., Schatzkin, A., Park, Y.: Alcohol and risk of breast cancer by histologic type and hormone receptor status in postmenopausal women the nih-aarp diet and health study. American Journal of Epidemiology 170(3), 308–317 (2009)
8. Lukasiewicz, T.: Expressive probabilistic description logics. Artificial Intelligence 172(6-7), 852–883 (2008)
9. Mood, A.M.: Introduction to the Theory of Statistics. McGraw-Hill (1950)
10. Sellers, T.A., Vierkant, R.A., Cerhan, J.R., Gapstur, S.M., Vachon, C.M., Olson, J.E., Pankratz, V.S., Kushi, L.H.: Interaction of dietary folate intake, alcohol, and risk of hormone receptor-defined breast cancer in a prospective study of postmenopausal women. Cancer Epidemiology, Biomarkers and Prevention 11, 1104–1107 (2002)
11. Suzuki, R., Ye, W., Rylander-Rudqvist, T., Saji, S., Colditz, G.A., Wolk, A.: Alcohol and postmenopausal breast cancer risk defined by estrogen and progesterone receptor status: A prospective cohort study. Journal of the National Cancer Institute 97(21), 1601–1608 (2005)

Finite Lattices Do Not Make Reasoning In $\mathcal{ALCI}$ Harder

Stefan Borgwardt and Rafael Peñaloza
{stefborg,penaloza}@tcs.inf.tu-dresden.de

Theoretical Computer Science, TU Dresden, Germany

Abstract. We consider the fuzzy logic $\mathcal{ALCI}$ with semantics based on a finite residuated lattice. We show that the problems of satisfiability and subsumption of concepts in this logic are ExpTime-complete w.r.t. general TBoxes and PSpace-complete w.r.t. acyclic TBoxes. This matches the known complexity bounds for reasoning in crisp $\mathcal{ALCI}$.

1 Introduction

OWL 2, the current standard ontology language for the semantic web, is based on the crisp description logic (DL) $\mathcal{SROIQ}(D)$. As a crisp logic, it is not well suited to express vague or imprecise concepts, such as HighTemperature, that can be found in numerous domains; prominently, in the biomedical area.

Fuzzy extensions of DLs have been studied for over a decade, and the literature on the topic is very extensive (see [15] for a survey). However, most of those approaches are based on the very simple *Zadeh semantics* where conjunction is interpreted as the minimum, with truth values ranging over the interval $[0, 1]$ of rational numbers. The last lustrum has seen a shift towards more general semantics for treating vagueness. On the one hand, the use of continuous t-norms as the underlying interpretation function for conjunction was proposed in [14]. On the other hand, [18] allows lattice-based truth values, but still restricts to Zadeh-like semantics.

Most of the work since then has focused on t-norm-based semantics over the unit interval; yet, ontologies are usually restricted to be unfoldable or acyclic [4–6]. Indeed, very recently it has been shown that general concept inclusion axioms (GCIs) can cause undecidability even in fuzzy DLs based on $\mathcal{ALC}$ [2, 3, 9, 11]. These results motivate restricting the logics, e.g. to finitely-valued semantics.

If one considers the Łukasiewicz t-norm over finitely many values, then reasoning is decidable even for very expressive DLs, as shown in [7] through a reduction to crisp reasoning. When restricted to $\mathcal{ALC}$ without terminological axioms, concept satisfiability is PSpace-complete as in the crisp case [10].[1] In the presence of general TBoxes, this problem becomes ExpTime-complete [8, 9], again matching the complexity of the crisp case, even if arbitrary (finite) lattices and t-norms are allowed. However, the complexity of subsumption of concepts

[1] The paper [10] considers a syntactic variant of fuzzy $\mathcal{ALC}$ with only one role.

was left as an open problem, as the standard reduction used in crisp DLs does not work with general t-norm semantics.

In this paper, we improve these complexity results to the fuzzy logic $\mathcal{ALCI}_L$ over finite lattices with general and acyclic TBoxes. More precisely, we show that in this logic, concept satisfiability is ExpTime-complete w.r.t. general TBoxes, and PSpace-complete w.r.t. acyclic TBoxes. Moreover, the same complexity bounds also hold for deciding subsumption between concepts.

2 Preliminaries

We will first give a short introduction to residuated lattices, which will be used for defining the semantics of our logic.[2] Afterwards, we recall some results from automata theory that will allow us to obtain tight upper bounds for the complexity of deciding satisfiability and subsumption of concepts.

2.1 Residuated Lattices

A *lattice* is an algebraic structure $(L, \vee, \wedge)$ over a *carrier set* L with two binary operations *join* $\vee$ and *meet* $\wedge$ that are idempotent, associative, and commutative and satisfy the absorption laws $\ell_1 \vee (\ell_1 \wedge \ell_2) = \ell_1 = \ell_1 \wedge (\ell_1 \vee \ell_2)$ for all $\ell_1, \ell_2 \in L$. L induces the ordering $\ell_1 \leq \ell_2$ iff $\ell_1 \wedge \ell_2 = \ell_1$ for all $\ell_1, \ell_2 \in L$. L is called *distributive* if $\vee$ and $\wedge$ distribute over each other, *finite* if L is finite, and *bounded* if it has a *minimum* and a *maximum* element, denoted as $\mathbf{0}$ and $\mathbf{1}$, respectively. It is *complete* if joins and meets of arbitrary subsets $T \subseteq L$, denoted by $\bigvee_{t \in T} t$ and $\bigwedge_{t \in T} t$ respectively, exist. Every finite lattice is also bounded and complete. Whenever it is clear from the context, we will simply use the carrier set L to represent the lattice $(L, \vee, \wedge)$.

A *De Morgan lattice* is a bounded distributive lattice extended with an involutive and anti-monotonic unary operation $\sim$, called *(De Morgan) negation*, satisfying the De Morgan laws $\sim(\ell_1 \vee \ell_2) = \sim\ell_1 \wedge \sim\ell_2$ and $\sim(\ell_1 \wedge \ell_2) = \sim\ell_1 \vee \sim\ell_2$ for all $\ell_1, \ell_2 \in L$.

A *residuated lattice* is a lattice L extended with two binary operators $\otimes$ (called *t-norm*) and $\Rightarrow$ (called *residuum*) such that $\otimes$ is associative, commutative, and has $\mathbf{1}$ as its unit and for every $\ell_1, \ell_2, \ell_3 \in L$, $\ell_1 \otimes \ell_2 \leq \ell_3$ iff $\ell_2 \leq \ell_1 \Rightarrow \ell_3$ holds. In a complete residuated lattice L, $\ell_1 \Rightarrow \ell_2 = \bigvee\{x \mid \ell_1 \otimes x \leq \ell_2\}$.[3] A simple consequence of this is that for every $\ell_1, \ell_2 \in L$, (i) $\mathbf{1} \Rightarrow \ell_1 = \ell_1$, and (ii) $\ell_2 \leq \ell_2$ iff $\ell_1 \Rightarrow \ell_2 = \mathbf{1}$. Additionally, the t-norm $\otimes$ is always monotonic.

In a residuated De Morgan lattice L, one can define the *t-conorm* $\oplus$ as $\ell_1 \oplus \ell_2 := \sim(\sim\ell_1 \otimes \sim\ell_2)$. For example, the meet operator $\ell_1 \wedge \ell_2$ defines a t-norm; its t-conorm is $\ell_1 \vee \ell_2$.

[2] For a more comprehensive view on residuated lattices, we refer the reader to [13, 12].

[3] We could also define the operator $\Rightarrow$ using this supremum, even if the complete lattice L is not residuated without affecting the results from Section 4.

In the following section, we will describe the fuzzy description logic $\mathcal{ALCI}_L$, whose semantics uses the residuum $\Rightarrow$ and the negation $\sim$. We emphasize, however, that the reasoning algorithm presented in Section 4 can be used with any choice of operators, as long as these are computable. In particular this means that our algorithm could also deal with other variants of fuzzy semantics, e.g. so-called Zadeh semantics [8, 18].

2.2 PSPACE Automata

To obtain upper bounds for the complexity of reasoning in $\mathcal{ALCI}_L$, we will make a reduction to the emptiness problem of looping automata on infinite trees. These automata receive as input the (unlabeled) infinite k-ary tree K^* for $K := \{1, \ldots, k\}$ with $k \in \mathbb{N}$. The *nodes* of this tree are represented as words in K^*: the empty word ε represents the root node, and ui represents the i-th successor of the node u. A *path* is a sequence $v_1, \ldots, v_m$ of nodes such that $v_1 = \varepsilon$ and each v_{i+1} is a direct successor of v_i.

Definition 1 (looping automaton). *A* looping automaton (LA) *is a tuple* $\mathcal{A} = (Q, I, \Delta)$ *where* Q *is a finite set of* states, $I \subseteq Q$ *a set of* initial states, *and* $\Delta \subseteq Q \times Q^k$ *the* transition relation. *A* run *of* $\mathcal{A}$ *is a mapping* $r : K^* \to Q$ *assigning states to each node of* K^* *such that* $r(\varepsilon) \in I$ *and for every* $u \in K^*$ *we have* $(r(u), r(u1), \ldots, r(uk)) \in \Delta$. *The* emptiness problem *for LA is to decide whether a given LA has a run.*

The emptiness of LA can be decided in polynomial time using a bottom-up approach [19]. Alternatively, one can use a top-down approach, which relies on the fact that if there is a run, then there is also a periodic run. To speed up the top-down search, one wants to find the period of a run as early as possible. This motivates the notion of *blocking automata*.

Definition 2 (m-blocking). *Let* $\mathcal{A} = (Q, \Delta, I)$ *be a looping automaton. We say that* $\mathcal{A}$ *is* m-blocking *for* $m \in \mathbb{N}$ *if every path* $v_1, \ldots, v_m$ *of length* m *in a run* r *of* $\mathcal{A}$ *contains two nodes* v_i *and* v_j *(*$i < j$*) such that* $r(v_j) = r(v_i)$.

Clearly, every looping automaton is m-blocking for every $m > |Q|$. However, the main interest in blocking automata arises when one can find a smaller bound on m. One way to reduce this limit is through a so-called *faithful* family of functions.

Definition 3 (faithful). *Let* $\mathcal{A} = (Q, \Delta, I)$ *be a looping automaton on* k*-ary trees. The family of functions* $f_q : Q \to Q$ *for* $q \in Q$ *is* faithful *w.r.t.* $\mathcal{A}$ *if for all* $q, q_0, q_1, \ldots, q_k \in Q$,

- *if* $(q, q_1, \ldots, q_k) \in \Delta$, *then* $(q, f_q(q_1), \ldots, f_q(q_k)) \in \Delta$, *and*
- *if* $(q_0, q_1, \ldots, q_k) \in \Delta$, *then* $(f_q(q_0), f_q(q_1), \ldots, f_q(q_k)) \in \Delta$.

The subautomaton $\mathcal{A}^S = (Q, \Delta^S, I)$ *of* $\mathcal{A}$ *induced by this family has the transition relation* $\Delta^S = \{(q, f_q(q_1), \ldots, f_q(q_k)) \mid (q, q_1, \ldots, q_k) \in \Delta\}$.

Lemma 4 ([1]). *Let $\mathcal{A}$ be a looping automaton and $\mathcal{A}^S$ its subautomaton induced by a faithful family of functions. $\mathcal{A}$ has a run iff $\mathcal{A}^S$ has a run.*

The construction that we will present in Section 4 produces automata that are exponential on the size of the input. For such cases, it has been shown that if the automata are m-blocking for some m bounded polynomially on the size of the input (that is, logarithmically on the size of the automaton), then the emptiness test requires only polynomial space.

Definition 5 (PSPACE on-the-fly construction). *Assume that we have a set $\mathfrak{I}$ of inputs and a construction that yields, for every $\mathfrak{i} \in \mathfrak{I}$, an $m_\mathfrak{i}$-blocking automaton $\mathcal{A}_\mathfrak{i} = (Q_\mathfrak{i}, \Delta_\mathfrak{i}, I_\mathfrak{i})$ working on $k_\mathfrak{i}$-ary trees. This construction is called a* PSPACE on-the-fly construction *if there is a polynomial P such that, for every input $\mathfrak{i}$ of size n*

- *$m_\mathfrak{i} \leq P(n)$ and $k_\mathfrak{i} \leq P(n)$,*
- *every element of $Q_\mathfrak{i}$ is of a size bounded by $P(n)$, and*
- *one can non-deterministically guess in time bounded by $P(n)$ an element of $I_\mathfrak{i}$, and, for a state $q \in Q_\mathfrak{i}$, a transition from $\Delta_\mathfrak{i}$ with first component q.*

Theorem 6 ([1]). *If the looping automata $\mathcal{A}_\mathfrak{i}$ are obtained from the inputs $\mathfrak{i} \in \mathfrak{I}$ by a* PSPACE *on-the-fly construction, then the emptiness problem for $\mathcal{A}_\mathfrak{i}$ can be decided in* PSPACE.

In Section 5 we will use this theorem to give PSPACE upper bounds on the complexity of reasoning in the logic $\mathcal{ALCI}_L$, which we introduce next.

3 The Fuzzy Logic $\mathcal{ALCI}_L$

For the rest of this paper, L denotes a fixed residuated, complete De Morgan lattice with the t-norm $\otimes$. The fuzzy description logic $\mathcal{ALCI}_L$ is a generalization of the crisp DL $\mathcal{ALCI}$ that uses the elements of L as truth values, instead of just the Boolean *true* and *false*. The syntax of $\mathcal{ALCI}_L$ is the same as in $\mathcal{ALCI}$: given the sets $\mathsf{N_C}$ and $\mathsf{N_R}$ of concept and role names, the set of *complex roles* is $\mathsf{N_R} \cup \{r^- \mid r \in \mathsf{N_R}\}$, and $\mathcal{ALCI}_L$ concepts are built using the syntactic rule

$$C ::= A \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \neg C \mid \exists s.C \mid \forall s.C \mid \top \mid \bot,$$

where $A \in \mathsf{N_C}$ and s is a complex role. For a complex role s, the *inverse of s* (denoted by $\overline{s}$) is s^- if $s \in \mathsf{N_R}$ and r if $s = r^-$.

The semantics of this logic is based on interpretation functions that map every concept C to a function specifying the membership degree of every domain element to C.

Definition 7 (semantics of $\mathcal{ALCI}_L$). *An* interpretation *is a pair $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ where $\Delta^\mathcal{I}$ is a non-empty (crisp)* domain *and $\cdot^\mathcal{I}$ is a function that assigns to every concept name A and every role name r functions $A^\mathcal{I} : \Delta^\mathcal{I} \to L$ and $r^\mathcal{I} : \Delta^\mathcal{I} \times \Delta^\mathcal{I} \to L$, respectively. The function $\cdot^\mathcal{I}$ is extended to $\mathcal{ALCI}_L$ concepts as follows for every $x \in \Delta^\mathcal{I}$:*

– $\top^{\mathcal{I}}(x) = \mathbf{1},\ \bot^{\mathcal{I}}(x) = \mathbf{0}$,
– $(C \sqcap D)^{\mathcal{I}}(x) = C^{\mathcal{I}}(x) \otimes D^{\mathcal{I}}(x),\ \ (C \sqcup D)^{\mathcal{I}}(x) = C^{\mathcal{I}}(x) \oplus D^{\mathcal{I}}(x)$,
– $(\neg C)^{\mathcal{I}}(x) = \sim C^{\mathcal{I}}(x)$,
– $(\exists s.C)^{\mathcal{I}}(x) = \bigvee_{y \in \Delta^{\mathcal{I}}} s^{\mathcal{I}}(x,y) \otimes C^{\mathcal{I}}(y)$,
– $(\forall s.C)^{\mathcal{I}}(x) = \bigwedge_{y \in \Delta^{\mathcal{I}}} s^{\mathcal{I}}(x,y) \Rightarrow C^{\mathcal{I}}(y)$,

where $(r^-)^{\mathcal{I}}(x,y) = r^{\mathcal{I}}(y,x)$ *for all* $x, y \in \Delta^{\mathcal{I}}$ *and* $r \in \mathsf{N_R}$.

Notice that, unlike in crisp $\mathcal{ALCI}$, existential and universal quantifiers are not dual to each other, i.e. in general, $(\neg\exists s.C)^{\mathcal{I}}(x) = (\forall s.\neg C)^{\mathcal{I}}(x)$ does not hold.

The axioms of this logic also have an associated lattice value, which expresses the degree to which the restriction must be satisfied.

Definition 8 (axioms). Terminological axioms *are* (labeled) concept definitions *of the form* $\langle A \doteq C, \ell\rangle$ *or* (labeled) general concept inclusions (GCIs) $\langle C \sqsubseteq D, \ell\rangle$, *where* $A \in \mathsf{N_C}$, C, D *are* $\mathcal{ALCI}_L$ *concepts, and* $\ell \in L$.

A general TBox *is a finite set of GCIs. An* acyclic TBox *is a finite set of concept definitions such that every concept name occurs at most once in the left-hand side of an axiom, and there is no cyclic dependency between definitions. A* TBox *is either a general TBox or an acyclic TBox.*[4]

An interpretation $\mathcal{I}$ satisfies *the concept definition* $\langle A \doteq C, \ell\rangle$ *if for every* $x \in \Delta^{\mathcal{I}}$, $(A^{\mathcal{I}}(x) \Rightarrow C^{\mathcal{I}}(x)) \otimes (C^{\mathcal{I}}(x) \Rightarrow A^{\mathcal{I}}(x)) \geq \ell$ *holds. It satisfies the GCI* $\langle C \sqsubseteq D, \ell\rangle$ *if for every* $x \in \Delta^{\mathcal{I}}$, $C^{\mathcal{I}}(x) \Rightarrow D^{\mathcal{I}}(x) \geq \ell$. $\mathcal{I}$ *is a* model *of the TBox* $\mathcal{T}$ *if it satisfies all axioms in* $\mathcal{T}$.

If $\mathcal{T}$ is an acyclic TBox, then all concept names occuring on the left-hand side of some axiom of $\mathcal{T}$ are called *defined*, all others are called *primitive*. If $\mathcal{T}$ is a general TBox, then all concept names appearing in it are *primitive*. A concept is an *atom* if it is either a primitive concept name, or it is a quantified concept, i.e. a concept of the form $\exists s.C$ or $\forall s.C$ for some complex role s and concept C.

We emphasize here that $\mathcal{ALCI}$ is a special case of $\mathcal{ALCI}_L$, where the underlying lattice contains only the elements $\mathbf{0}$ and $\mathbf{1}$, which may be interpreted as *false* and *true*, respectively, and the t-norm and t-conorm are just conjunction and disjunction, respectively. Accordingly, one can generalize the reasoning problems for $\mathcal{ALCI}$ to the use of other lattices. We will focus on deciding strong ℓ-satisfiability and ℓ-subsumption [8].

Definition 9 (satisfiability, subsumption). *Let* C, D *be* $\mathcal{ALCI}_L$ *concepts,* $\mathcal{T}$ *a TBox, and* $\ell \in L$. C *is* strongly ℓ-satisfiable *w.r.t.* $\mathcal{T}$ *if there is a model* $\mathcal{I}$ *of* $\mathcal{T}$ *and an* $x \in \Delta^{\mathcal{I}}$ *such that* $C^{\mathcal{I}}(x) \geq \ell$. C *is* ℓ-subsumed *by* D *w.r.t.* $\mathcal{T}$ *if every model* $\mathcal{I}$ *of* $\mathcal{T}$ *is also a model of* $\langle C \sqsubseteq D, \ell\rangle$.

In previous work we have shown that satisfiability is undecidable in $\mathcal{ALC}_L$ [9], and hence also in $\mathcal{ALCI}_L$, in general. For this reason, we assume that L is

[4] Notice that we do not consider mixed TBoxes. We could allow axioms of the form $\langle A \sqsubseteq C, \ell\rangle$ in acyclic TBoxes, as long as they do not introduce cyclic dependencies. To avoid overloading the notation, we exclude this case.

finite for the rest of this paper. As we will show in the next sections, under this restriction we obtain the same complexity upper bounds for deciding satisfiability and subsumption as in the crisp case; that is, the lattice based semantics do not increase the complexity of the logic.

4 Deciding Strong Satisfiability and Subsumption

Recall that the semantics of the quantifiers require the computation of a supremum or infimum of the membership degrees of a possibly infinite set of elements of the domain. To obtain an effective decision procedure, one usually restricts reasoning to witnessed models [14].

Definition 10 (witnessed model). *Let $n \in \mathbb{N}$. A model $\mathcal{I}$ of a TBox $\mathcal{T}$ is called* n-witnessed *if for every $x \in \Delta^{\mathcal{I}}$ and every concept of the form $\exists r.C$ there are n elements $x_1, \ldots, x_n \in \Delta^{\mathcal{I}}$ such that*

$$(\exists r.C)^{\mathcal{I}}(x) = \bigvee_{i=1}^{n} r^{\mathcal{I}}(x, x_i) \otimes C^{\mathcal{I}}(x_i),$$

and analogously for the universal restrictions $\forall r.C$. In particular, if $n = 1$, then the suprema and infima from the semantics of $\exists r.C$ and $\forall r.C$ become maxima and minima, respectively. In this case, we simply say that $\mathcal{I}$ is witnessed.

We can restrict reasoning to n-witnessed models w.l.o.g.: since L is finite, we always have the n-witnessed model property for some $n \in \mathbb{N}$.

Lemma 11. *If the cardinality of the largest antichain of L is n, then $\mathcal{ALCI}_L$ has the n-witnessed model property.*

To simplify the description of the algorithm, in the following we consider $n = 1$. The algorithm and the proofs of correctness can easily be adapted for any other $n \in \mathbb{N}$.

Our algorithm for deciding satisfiability and subsumption of concepts exploits the fact that a TBox $\mathcal{T}$ has a model iff it has a well-structured tree model, called a *Hintikka tree*. Intuitively, Hintikka trees are abstract representations of models that explicitly express the membership value of all "relevant" concepts. We will construct automata that have exactly these Hintikka trees as their runs, and use the initial states to verify that an element in the model verifies the satisfiability or violates the subsumption condition, respectively. Reasoning is hence reduced to the emptiness test of these automata.

We denote by $\mathsf{sub}(C, \mathcal{T})$ the set of all subconcepts of C and of the concepts A, E, and F for all axioms $\langle E \sqsubseteq F, \ell \rangle$ or $\langle A \doteq F, \ell \rangle$ in $\mathcal{T}$. The nodes of the Hintikka trees are labeled with so-called Hintikka functions over the domain $\mathsf{sub}(C, \mathcal{T}) \cup \{\rho\}$, where ρ is an arbitrary new element, which will be used to express the degree with which the role relation to the parent node holds.

Definition 12 (Hintikka function). *A* Hintikka function *for $C, \mathcal{T}$ is a partial function $H : \mathsf{sub}(C, \mathcal{T}) \cup \{\rho\} \to L$ such that:*

(i) H *is defined for* ρ *and for all atoms,*
(ii) *if* $H(D \sqcap E)$ *is defined, then* $H(D)$ *and* $H(E)$ *are also defined and it holds that* $H(D \sqcap E) = H(D) \otimes H(E)$,
(iii) *if* $H(D \sqcup E)$ *is defined, then* $H(D)$ *and* $H(E)$ *are also defined and it holds that* $H(D \sqcup E) = H(D) \oplus H(E)$,
(iv) *if* $H(\neg D)$ *is defined, then* $H(D)$ *is defined and* $H(\neg D) = \sim H(D)$.

It is compatible *with the concept definition* $\langle A \doteq E, \ell \rangle$ *if, whenever* $H(A)$ *is defined, then* $H(E)$ *is defined and* $(H(A) \Rightarrow H(E)) \otimes (H(E) \Rightarrow H(A)) \geq \ell$.[5] *It is* compatible *with the GCI* $\langle E \sqsubseteq F, \ell \rangle$ *if* $H(E)$ *and* $H(F)$ *are always defined and* $H(E) \Rightarrow H(F) \geq \ell$ *holds.*

The Hintikka trees have a fixed arity k determined by the number of existential and universal restrictions, i.e. concepts of the form $\exists s.F$ or $\forall s.F$, contained in $\mathsf{sub}(C, \mathcal{T})$. Intuitively, each successor will act as the witness for one of these restrictions. Since we need to know which successor in the tree corresponds to which restriction, we fix an arbitrary bijection

$$\varphi : \{E \mid E \in \mathsf{sub}(C, \mathcal{T}) \text{ is of the form } \exists s.F \text{ or } \forall s.F\} \to K.$$

Definition 13 (Hintikka condition). *The tuple* $(H_0, H_1, \ldots, H_k)$ *of Hintikka functions for* $C, \mathcal{T}$ *satisfies the* Hintikka condition *if:*

(i) *For every existential restriction* $\exists s.G \in \mathsf{sub}(C, \mathcal{T})$
- $H_{\varphi(\exists s.G)}(G)$ *is defined and* $H_0(\exists s.G) = H_{\varphi(\exists s.G)}(\rho) \otimes H_{\varphi(\exists s.G)}(G)$, *and*
- $H_{\varphi(E)}(G)$ *is defined and* $H_0(\exists s.G) \geq H_{\varphi(E)}(\rho) \otimes H_{\varphi(E)}(G)$ *for every restriction* $E \in \mathsf{sub}(C, \mathcal{T})$ *of the form* $\exists s.F$ *or* $\forall s.F$.

(ii) *For every universal restriction* $\forall s.G \in \mathsf{sub}(C, \mathcal{T})$
- $H_{\varphi(\forall s.G)}(G)$ *is defined and* $H_0(\forall s.G) = H_{\varphi(\forall s.G)}(\rho) \Rightarrow H_{\varphi(\forall s.G)}(G)$,
- $H_{\varphi(E)}(G)$ *is defined and* $H_0(\forall s.G) \leq H_{\varphi(E)}(\rho) \Rightarrow H_{\varphi(E)}(G)$ *for every restriction* $E \in \mathsf{sub}(C, \mathcal{T})$ *of the form* $\exists s.F$ *or* $\forall s.F$.

(iii) *For every existential restriction* $\exists s.G \in \mathsf{sub}(C, \mathcal{T})$ *and every restriction* $E \in \mathsf{sub}(C, \mathcal{T})$ *of the form* $\exists \overline{s}.F$ *or* $\forall \overline{s}.F$, $H_0(G)$ *is defined and* $H_{\varphi(E)}(\exists s.G) \geq H_{\varphi(E)}(\rho) \otimes H_0(G)$.

(iv) *For every universal restriction* $\forall s.G \in \mathsf{sub}(C, \mathcal{T})$ *and every restriction* $E \in \mathsf{sub}(C, \mathcal{T})$ *of the form* $\exists \overline{s}.F$ *or* $\forall \overline{s}.F$, $H_0(G)$ *is defined and* $H_{\varphi(E)}(\forall s.G) \leq H_{\varphi(E)}(\rho) \Rightarrow H_0(G)$.

The tuple is compatible *with the axiom* t *if the Hintikka functions* $H_0, \ldots, H_k$ *are compatible with* t.

Condition (i) makes sure that an existential restriction $\exists s.G$ is witnessed by its designated successor $\varphi(\exists s.G)$ and all other s-successors do not contradict the witness. Condition (iii) deals with inverse roles, ensuring that the $\overline{s}$-restrictions are propagated backwards through the s-relation. Conditions (ii) and (iv) treat the universal restrictions analogously.

[5] This method, called *lazy unfolding*, is only correct for acyclic TBoxes.

A *Hintikka tree* for $C, \mathcal{T}$ is an infinite k-ary tree $\mathbf{T}$ labeled with compatible Hintikka functions for $C, \mathcal{T}$ such that $\mathbf{T}(\varepsilon)(C)$ is defined and the tuple $(\mathbf{T}(u), \mathbf{T}(u1), \ldots, \mathbf{T}(uk))$ satisfies the Hintikka condition for every node $u \in K^*$. The definition of compatibility ensures that all axioms are satisfied at any node of the Hintikka tree, while the Hintikka condition makes sure that the tree is in fact a witnessed model.

The proof of the following theorem uses arguments similar to those in [1]. The main difference is the presence of successors witnessing the universal restrictions.

Theorem 14. *Let C be an $\mathcal{ALCI}_L$ concept, $\mathcal{T}$ a TBox, and $\ell \in L$. Then C is strongly ℓ-satisfiable w.r.t. $\mathcal{T}$ (in a witnessed model) iff there is a Hintikka tree $\mathbf{T}$ for $C, \mathcal{T}$ such that $\mathbf{T}(\varepsilon)(C) \geq \ell$.*

Proof (Sketch). Every witnessed model $\mathcal{I}$ of $\mathcal{T}$ with a domain element $x \in \Delta^{\mathcal{I}}$ for which $C^{\mathcal{I}}(x) \geq \ell$ holds can be *unraveled* into a Hintikka tree $\mathbf{T}$ for $C, \mathcal{T}$ as follows. We start by labeling the root node by the (total) Hintikka function that records the membership values of x for each concept from $\mathsf{sub}(C, \mathcal{T})$. We then create successors of the root by considering every $E \in \mathsf{sub}(C, \mathcal{T})$ of the form $\exists s.F$ or $\forall s.F$ and finding the witness $y \in \Delta^{\mathcal{I}}$ for this restriction. We create a new node for y which is the $\varphi(E)$-th successor of the root node and is labeled by a Hintikka set H with $H(\rho) = s^{\mathcal{I}}(x, y)$. The fact that $\mathcal{I}$ is a model of $\mathcal{T}$ ensures that these successors satisfy the Hintikka condition. By continuing this process, we construct a Hintikka tree $\mathbf{T}$ for $C, \mathcal{T}$ for which $\mathbf{T}(\varepsilon)(C) \geq \ell$ holds.

Conversely, we show that a Hintikka tree can be seen as a witnessed model with domain K^* and interpretation function given by the Hintikka functions. Notice that from the partial function labeling each node we can obtain a valuation for each concept name that satisfies all the axioms in $\mathcal{T}$. Indeed, if $\mathcal{T}$ is a general TBox, then every concept name is primitive, and hence the valuation is already defined. The fact that the Hintikka function is compatible with all the axioms in $\mathcal{T}$ implies that every node satisfies the TBox. On the other hand, if $\mathcal{T}$ is an acyclic TBox, and H is undefined for some concept names, then consider an axiom $\langle A \doteq C, \ell \rangle$ for which $H(A)$ is undefined, but $H(B)$ is defined for every atom appearing in C. The acyclicity of $\mathcal{T}$ ensures that such an axiom always exists. Thus, we can compute a value for $H(C)$ that still satisfies the conditions of Definition 12. If we set $H(A) := H(C)$, then H is still compatible with $\mathcal{T}$. By an induction argument, we can define a compatible total Hintikka function, and thus a valuation for every concept name that satisfies $\mathcal{T}$.

For this valuation to be an interpretation, it only remains to be shown that the semantics of the existential and universal restrictions are satisfied. This is ensured by the Hintikka condition. The choice of the successors also ensures that the interpretation is witnessed. As explained above, it is compatible, and hence also a model of $\mathcal{T}$. Thus, if there is a Hintikka tree $\mathbf{T}$ for $C, \mathcal{T}$ with $\mathbf{T}(\varepsilon)(C) \geq \ell$, then C is strongly ℓ-satisfiable w.r.t. $\mathcal{T}$. □

Hintikka trees can also be used for deciding (non-)subsumption between $\mathcal{ALCI}_L$ concepts. The proof of the following theorem is analogous to the one of Theorem 14.

Theorem 15. *Let C, D be $\mathcal{ALCI}_L$ concepts, $\mathcal{T}$ a TBox, and $\ell \in L$. Then C is not ℓ-subsumed by D (in a witnessed model) iff there is a Hintikka tree $\mathbf{T}$ for $C \sqcap D, \mathcal{T}$ such that $\mathbf{T}(\varepsilon)(C) \Rightarrow \mathbf{T}(\varepsilon)(D) \not\geq \ell$.*[6]

Notice that this does not yield a reduction from subsumption to satisfiability, since the residuum $\Rightarrow$ cannot in general be expressed using only the t-norm, t-conorm and negation, and in Theorem 14 the value of C at the root is restricted to a value greater or equal to ℓ, while Theorem 15 negates this restriction.

From the last two theorems it follows that satisfiability and subsumption of $\mathcal{ALCI}_L$ concepts can be reduced to deciding the existence of a Hintikka tree with additional restrictions in the root. By building looping automata whose runs correspond exactly to those Hintikka trees, we reduce $\mathcal{ALCI}_L$ reasoning to the emptiness problem of these automata. For the following, we focus only on deciding satisfiability and explain the minor modifications required for deciding subsumption.

Definition 16 (Hintikka automaton). *Let C be an $\mathcal{ALCI}_L$ concept, $\mathcal{T}$ a TBox, and $\ell \in L$. The* Hintikka automaton *for $C, \mathcal{T}, \ell$ is $\mathcal{A}_{C,\mathcal{T},\ell} = (Q, I, \Delta)$, where Q is the set of all compatible Hintikka functions for $C, \mathcal{T}$, I contains all Hintikka functions H with $H(C) \geq \ell$, and Δ is the set of all $(k+1)$-tuples of Hintikka functions that satisfy the Hintikka condition.*

The runs of $\mathcal{A}_{C,\mathcal{T},\ell}$ are exactly the Hintikka trees $\mathbf{T}$ having $\mathbf{T}(\varepsilon)(C) \geq \ell$. Thus, C is strongly ℓ-satisfiable w.r.t. $\mathcal{T}$ iff $\mathcal{A}_{C,\mathcal{T},\ell}$ is not empty. To obtain an automaton deciding ℓ-subsumption between C and D, one needs only modify the set of initial states I to contain all Hintikka functions H with $H(C) \Rightarrow H(D) \not\geq \ell$. In that case, we have that C is ℓ-subsumed by D iff the automaton is empty.

The size of the automaton $\mathcal{A}_{C,\mathcal{T},\ell}$ is exponential in the input $C, \mathcal{T}$. Hence, we have an ExpTime algorithm for this logic. For general TBoxes, this gives a tight upper bound for the complexity of satisfiability and subsumption, since these problems are already ExpTime-hard for crisp $\mathcal{ALC}$ [16].

Theorem 17. *Deciding strong satisfiability and subsumption in $\mathcal{ALCI}_L$ w.r.t. general TBoxes is* ExpTime*-complete.*

5 PSpace Results for Acyclic TBoxes

If one restricts to acyclic TBoxes, then the upper bound obtained by the emptiness test of the automaton from Definition 16 does not match the PSpace lower bound given by crisp $\mathcal{ALCI}$ with acyclic TBoxes. We will now improve this upper bound and show that satisfiability and subsumption of $\mathcal{ALCI}_L$ concepts w.r.t. acyclic TBoxes are also PSpace-complete problems.

The idea is to modify the construction of the Hintikka automata into a PSpace on-the-fly construction. Notice that $\mathcal{A}_{C,\mathcal{T},\ell}$ satisfies all but one of the

[6] Using $C \sqcap D$ only ensures that $\mathbf{T}(\varepsilon)(C)$ and $\mathbf{T}(\varepsilon)(D)$ are defined, but imposes no further restriction on their values.

conditions from Definition 5: (i) the arity of the automata is given by the number of existential and universal concepts in $\mathsf{sub}(C, \mathcal{T})$; (ii) every Hintikka function has size bounded by $|\mathsf{sub}(C, \mathcal{T})|$; (iii) building a state or a transition of the automaton requires only guessing values for all concepts in $\mathsf{sub}(C, \mathcal{T})$ and then verifying that this is indeed a valid state or transition, which can be done in time polynomial in $|\mathsf{sub}(C, \mathcal{T})|$. However, it is easy to build runs of the automata constructed by this reduction where blocking occurs only after exponentially many transitions, violating the first condition of PSPACE on-the-fly constructions.

We will use a faithful family of functions to obtain a reduced automaton that guarantees blocking after at most polynomially many transitions, thus obtaining the PSPACE upper bound. The idea is that it suffices to consider only transitions that reduce the maximal role depth (w.r.t. $\mathcal{T}$) in the support of the states.

The *role depth w.r.t.* $\mathcal{T}$ ($\mathsf{rd}_\mathcal{T}$) of $\mathcal{ALCI}_L$ concepts is recursively defined as follows: $\mathsf{rd}_\mathcal{T}(A) = \mathsf{rd}_\mathcal{T}(\top) = \mathsf{rd}_\mathcal{T}(\bot) = 0$ for every primitive concept name A; $\mathsf{rd}_\mathcal{T}(C \sqcap D) = \mathsf{rd}_\mathcal{T}(C \sqcup D) = \max\{\mathsf{rd}_\mathcal{T}(C), \mathsf{rd}_\mathcal{T}(D)\}$; $\mathsf{rd}_\mathcal{T}(\neg C) = \mathsf{rd}_\mathcal{T}(C)$; $\mathsf{rd}_\mathcal{T}(\exists r.C) = \mathsf{rd}_\mathcal{T}(\forall r.C) = \mathsf{rd}_\mathcal{T}(C) + 1$; and $\mathsf{rd}_\mathcal{T}(A) = \mathsf{rd}_\mathcal{T}(C)$ for every definition $\langle A \doteq C, \ell\rangle \in \mathcal{T}$. For a Hintikka function H for $C, \mathcal{T}$, we denote as $\mathsf{support}(H)$ the set of all concepts in $\mathsf{sub}(C, \mathcal{T})$ such that $H(C)$ is defined and $H(C) > \mathbf{0}$. We define $\mathsf{rd}_\mathcal{T}(H)$ as the maximum $\mathsf{rd}_\mathcal{T}(D)$ such that $D \in \mathsf{support}(H)$.

Definition 18 (functions f_H). *Let H and H' be two states of $\mathcal{A}_{C,\mathcal{T},\ell}$ with $\mathsf{rd}_\mathcal{T}(H) = n$. The function $f_H(H')$ is given by:*

$$f_H(H')(D) = \begin{cases} \mathbf{0} & \text{if } D \text{ is an atom and } \mathsf{rd}_\mathcal{T}(D) \geq n \\ H'(D) & \text{if } \mathsf{rd}_\mathcal{T}(D) < n \\ \textit{undefined} & \text{otherwise.} \end{cases}$$

$$f_H(H')(\rho) = \begin{cases} \mathbf{0} & \text{if } \mathsf{support}(H) = \emptyset \\ H'(\rho) & \text{otherwise.} \end{cases}$$

Since $\mathcal{T}$ is acyclic, the function $f_H(H')$ defined above is still a Hintikka function for $C, \mathcal{T}$ compatible with all the axioms in $\mathcal{T}$.

Lemma 19. *The family of mappings f_H for states H of $\mathcal{A}_{C,\mathcal{T},\ell}$ from Definition 18 is faithful w.r.t. $\mathcal{A}_{C,\mathcal{T},\ell}$.*

Proof. Let $(H, H_1, \ldots, H_k)$ be a valid transition of $\mathcal{A}_{C,\mathcal{T},\ell}$. We need to show that $(H, f_H(H_1), \ldots, f_H(H_k))$ is also a transition, i.e. that it satisfies the Hintikka condition. We show in detail only the proof for the restriction (i) from Definition 13, as the others can be treated analogously.

For $\exists s.G \in \mathsf{sub}(C, \mathcal{T})$, the value $H_{\varphi(E)}(G)$ is defined for all restrictions E of the form $\exists s.F$ or $\forall s.F$ in $\mathsf{sub}(C, \mathcal{T})$. If $\mathsf{rd}_\mathcal{T}(\exists s.G) > \mathsf{rd}_\mathcal{T}(H)$, then $H(\exists s.G) = \mathbf{0}$, and all the values $f_H(H_{\varphi(E)})(G)$ are $\mathbf{0}$. Thus, the inequalities are trivially satisfied. Otherwise, $\mathsf{rd}_\mathcal{T}(G) < \mathsf{rd}_\mathcal{T}(\exists s.G) \leq \mathsf{rd}_\mathcal{T}(H)$, and thus the values $H_{\varphi(E)}(G)$ are not changed by applying f_H. If the values $H_{\varphi(E)}(\rho)$ are also left unchanged, all inequalities remain satisfied. Otherwise, $H(\exists s.G) = \mathbf{0}$, and all the values $f_H(H_{\varphi(E)})(\rho)$ are $\mathbf{0}$. Thus, the inequalities are again trivially satisfied. □

By Lemma 4, $\mathcal{A}_{C,\mathcal{T},\ell}$ is empty iff the induced subautomaton $\mathcal{A}^S_{C,\mathcal{T},\ell}$ is empty.

Theorem 20. *The construction of $\mathcal{A}^S_{C,\mathcal{T},\ell}$ from an $\mathcal{ALCI}_L$ concept C, $\ell \in L$, and an acyclic TBox $\mathcal{T}$ is a* PSPACE *on-the-fly construction.*

Proof. As described before, we only need to show that the automata $\mathcal{A}^S_{C,\mathcal{T},\ell}$ are m-blocking for some m bounded polynomially in $|\mathsf{sub}(C,\mathcal{T})|$. We show that this holds for $m = \max\{\mathsf{rd}_\mathcal{T}(D) \mid D \in \mathsf{sub}(C,\mathcal{T})\} + 2$.

By definition of $\mathcal{A}^S_{C,\mathcal{T},\ell}$, every transition decreases the maximal role depth of the support of the state. Hence, after at most $\max\{\mathsf{rd}_\mathcal{T}(D) \mid D \in \mathsf{sub}(C,\mathcal{T})\}$ transitions, we reach a state H, where $H(D) = \mathbf{0}$ if D is an atom and undefined otherwise, and hence, $\mathsf{support}(H) = \emptyset$. From the next transition on, all the states additionally satisfy that $H(\rho) = \mathbf{0}$. Hence, after at most m transitions, we find two states that are equal. Since $m \leq |\mathsf{sub}(C,\mathcal{T})| + 2$, $\mathcal{A}^S_{C,\mathcal{T},\ell}$ satisfies the requirements for a PSPACE on-the-fly construction. □

This shows that emptiness of $\mathcal{A}^S_{C,\mathcal{T},\ell}$ and hence also of $\mathcal{A}_{C,\mathcal{T},\ell}$ is in PSPACE. This yields the desired PSPACE upper bound for satisfiability and similar arguments can be made for subsumption. PSPACE-hardness follows from PSPACE-hardness of satisfiability and subsumption w.r.t. the empty TBox in $\mathcal{ALC}$ [17].

Theorem 21. *Deciding strong satisfiability and subsumption in $\mathcal{ALCI}_L$ w.r.t. acyclic TBoxes is* PSPACE*-complete.*

Notice that the definitions of Hintikka functions and Hintikka trees are independent of the operators used. One could have chosen the residual negation $\ominus \ell := \ell \Rightarrow \mathbf{0}$ to interpret the constructor $\neg$, or the Kleene-Dienes implication $\ell_1 \Rightarrow \ell_2 := \sim \ell_1 \vee \ell_2$ instead of the residuum. The only restrictions are that the semantics must be truth functional, i.e. the value of a formula must depend only on the values of its direct subformulas, and the underlying operators must be computable. We could also use the traditional semantics for concept definitions in which $\otimes$ is replaced by the simple meet t-norm $\wedge$.

We also point out that the algorithm can be modified for reasoning w.r.t. n-witnessed models for $n > 1$. One needs only extend the arity of the Hintikka trees to account for n witnesses for each quantified formula in $\mathsf{sub}(C,\mathcal{T})$; the arity of $\mathcal{A}_{C,\mathcal{T},\ell}$ grows polynomially in n. This does not affect the complexity upper bounds from the automata, and hence Theorems 17 and 21 still hold.

6 Conclusions

We have shown that reasoning in $\mathcal{ALCI}_L$ is not harder than in the underlying crisp DL $\mathcal{ALCI}$. More precisely, strong ℓ-satisfiability and ℓ-subsumption can be decided in EXPTIME for general TBoxes and in PSPACE for acyclic TBoxes. This extends the complexity results from [8–10] and demonstrates that automata can show PSPACE results even for fuzzy description logics, as in the crisp case [1]. This paper provides a small step towards reasoning services for fuzzy generalizations of the current standard ontology languages, like $\mathcal{SROIQ}(D)$.

In the future, we want to study the influence of additional DL constructors and axioms on the complexity of the reasoning tasks. In particular, transitive roles, which are covered by the results in [1], have not been considered in this paper. Although in the crisp case they do not increase the complexity of checking satisfiability, it is not straightforward to generalize the methods used to show this to residuated De Morgan lattices.

Satisfiability w.r.t. general TBoxes and residuated total orders has been shown to be undecidable [9], but it remains open to find subclasses of infinite lattices and t-norms for which the problem is decidable. Over the unit interval, the product and Łukasiewicz t-norms cause undecidability w.r.t. witnessed models [3, 11]; for arbitrary models decidability is unknown in these cases.

References

1. F. Baader, J. Hladik, and R. Peñaloza. Automata can show PSPACE results for description logics. *Inform. Comput.*, 206(9-10):1045–1056, 2008.
2. F. Baader and R. Peñaloza. Are fuzzy description logics with general concept inclusion axioms decidable? In *Proc. FuzzIEEE'11*, pages 1735–1742. IEEE, 2011.
3. F. Baader and R. Peñaloza. On the undecidability of fuzzy description logics with GCIs and product t-norm. In *Proc. FroCoS'11*, pages 55–70. Springer-Verlag, 2011.
4. F. Bobillo, F. Bou, and U. Straccia. On the failure of the finite model property in some fuzzy description logics. *Fuzzy Set. Syst.*, 172(1):1–12, 2011.
5. F. Bobillo and U. Straccia. A fuzzy description logic with product t-norm. In *Proc. Fuzz-IEEE'07*, pages 1–6. IEEE, 2007.
6. F. Bobillo and U. Straccia. Fuzzy description logics with general t-norms and datatypes. *Fuzzy Set. Syst.*, 160(23):3382–3402, 2009.
7. F. Bobillo and U. Straccia. Reasoning with the finitely many-valued Łukasiewicz fuzzy description logic $\mathcal{SROIQ}$. *Inf. Sci.*, 181(4):758–778, 2011.
8. S. Borgwardt and R. Peñaloza. Description logics over lattices with multi-valued ontologies. In *Proc. IJCAI'11*, pages 768–773. AAAI Press, 2011.
9. S. Borgwardt and R. Peñaloza. Fuzzy ontologies over lattices with t-norms. In *Proc. DL'11*, pages 70–80. CEUR Workshop Proceedings, 2011.
10. F. Bou, M. Cerami, and F. Esteva. Finite-valued Łukasiewicz modal logic is PSPACE-complete. In *Proc. IJCAI'11*, pages 774–779. AAAI Press, 2011.
11. M. Cerami and U. Straccia. On the undecidability of fuzzy description logics with GCIs with Łukasiewicz t-norm. 2011. `arXiv:1107.4212v3 [cs.LO]`.
12. G. De Cooman and E. E. Kerre. Order norms on bounded partially ordered sets. *J. Fuzzy Math*, 2:281–310, 1993.
13. G. Grätzer. *General Lattice Theory, Second Edition*. Birkhäuser Verlag, 2003.
14. P. Hájek. Making fuzzy description logic more general. *FSS*, 154(1):1–15, 2005.
15. T. Lukasiewicz and U. Straccia. Managing uncertainty and vagueness in description logics for the semantic web. *J. Web Semant.*, 6(4):291–308, 2008.
16. K. Schild. A correspondence theory for terminological logics: Preliminary report. In *Proc. IJCAI'91*, pages 466–471. Morgan Kaufmann, 1991.
17. M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artif. Intell.*, 48(1):1–26, 1991.
18. U. Straccia. Description logics over lattices. *Int. J. Unc. Fuzz.*, 14(1):1–16, 2006.
19. M. Y. Vardi and P. Wolper. Automata-theoretic techniques for modal logics of programs. *J. Comput. Syst. Sci.*, 32(2):183–221, 1986.

Learning Terminological Naïve Bayesian Classifiers
Under Different Assumptions on Missing Knowledge

Pasquale Minervini, Claudia d'Amato, and Nicola Fanizzi

LACAM – Dipartimento di Informatica – Università degli Studi di Bari "Aldo Moro"
via E. Orabona, 4 - 70125 Bari - Italia
pasquale.minervini@uniba.it, {claudia.damato, fanizzi}@di.uniba.it

Abstract. Knowledge available through Semantic Web standards can easily be missing, generally because of the adoption of the Open World Assumption (i.e. the truth value of an assertion is not necessarily known). However, the rich relational structure that characterizes ontologies can be exploited for handling such missing knowledge in an explicit way. We present a Statistical Relational Learning system designed for learning terminological naïve Bayesian classifiers, which estimate the probability that a generic individual belongs to the target concept given its membership to a set of Description Logic concepts. During the learning process, we consistently handle the lack of knowledge that may be introduced by the adoption of the Open World Assumption, depending on the varying nature of the missing knowledge itself.

1 Introduction

On the Semantic Web (SW) [2] difficulties arise when trying to model real-world domains using purely logical formalisms, since real-world knowledge generally involves some degree of uncertainty or imprecision. In recognition of the need to soundly represent uncertain knowledge, the World Wide Web Consortium (W3C) created, in 2007, the Uncertainty Reasoning for the World Wide Web Incubator Group [1] (URW3-XG), with the aim of identifying the requirements for reasoning with and representing the uncertain knowledge in Web-based information.

Several approaches to representation and inference with knowledge enriched with probabilistic information have been proposed: some extend knowledge representation formalisms actually used in the SW; others rely on probabilistic enrichment of Description Logics or logic programming formalisms.

Motivation

The main problem of applying these approaches in real settings is given by the fact that they almost always assume the availability of probabilistic information. However, except of seldom cases, this information would be hardly known in advance. Having a method that, exploiting available information on the data, i.e. an already designed and

[1] http://www.w3.org/2005/Incubator/urw3/

populated ontology, is able to capture the necessary probabilistic information would be of great help.

Also, when relying on SW knowledge bases for reasoning with the Open World Assumption (OWA) (e.g. when OWL is considered as a syntactic variant of some Description Logic [1]), it is not always possible to know the truth value of an assertion: under OWA, a statement is true or false only if its truth value can be formally derived. As a consequence, there can be some cases (e.g. determining if an individual is a member of a given concept) for which the truth value cannot be determined (it cannot be derived neither that the individual is instance of the considered concept nor that the individual is instance of the negated concept). This is opposed by the *Closed World Assumption* (CWA), employed by a multitude of first order logic fragments and in the Data Base setting where every statement that cannot be proved to be true, is assumed to be false.

Related Work

Within the SW, Machine Learning (ML) is going to cover a relevant role in the analysis of distributed data sources described using SW standards [24], with the aim of discovering new and refining existing knowledge. A collection of ML approaches oriented to SW have already been proposed in literature, ranging from propositional and single-relational (e.g. SPARQL-ML [14], or based on low-rank matrix approximation techniques such as in [24, 25]) to multi-relational (e.g. distance-based [6, 9] or kernel-based [10, 3]).

In the class of multi-relational learning methods, *Statistical Relational Learning* [13] (SRL) one seem particularly appealing, being designed to learn in domains with both a complex relational and a rich probabilistic structure; the URW3-XG provided in [16] a large group of situations in which knowledge on the SW needs to represent uncertainty, ranging from recommendation and extraction/annotation to belief fusion/opinion pooling and healthcare/life sciences. There have already been some proposals regarding the adaptation and application of SRL systems to the SW, e.g. [7] proposes to employ Markov Logic Networks [21] for first-order probabilistic inference and learning within the SW, and [18] proposes to learn first-order probabilistic theories in a probabilistic extension of the $\mathcal{ALC}$ Description Logic named CR$\mathcal{ALC}$.

However, such ML techniques make strong assumptions about the nature of the missing knowledge (e.g. both matrix completion methods and the technique proposed in [18] inherently assume data is *Missing at Random* [23], while Markov Logic Networks resort to Closed World Assumption during learning). Learning from incomplete knowledge bases by adopting methods not coherent with the nature of the missing knowledge itself (e.g. expecting it to be *Missing at Random* while it is *Informatively Missing*) can lead to misleading results with respect to the real model followed by the data [22].

We realised a SRL system for incrementally inducing a terminological naïve Bayesian classifier, i.e. a naïve Bayesian network modelling the conditional dependencies between a learned set of Description Logic (complex) concepts and a target atomic concept the system aims to learn. Our system is focused to the SW, being able to learn classifiers with a structure which is both logically and statistically rich, and to deal with the missing knowledge resulting from the adoption of the OWA with methods that are

consistent with the assumed nature of the missing knowledge (i.e. *Missing Completely at Random*, *Missing at Random* or *Informatively Missing*). In the rest of this paper, we will first describe Bayesian Networks (and some extensions we will employ to deal with some potentially problematic cases); then we will describe our probabilistic-logic model, terminological Bayesian classifiers, and the problem of learning it from a set of training individuals and a Description Logic knowledge base. In the last part, we will describe our learning algorithm, and the adaptations to learn under different assumptions on the ignorance model.

2 Bayesian Networks and Robust Bayesian Estimation

Graphical models [19] (GMs) are a popular framework to compactly describe the joint probability distribution for a set of random variables, by representing the underlying structure through a series of modular factors. Depending on the underlying semantics, GMs can be grouped into two main classes: *directed graphical models*, which found on directed graphs, and *undirected graphical models*, founding on undirected graphs.

A Bayesian network (BN) is a directed GM which represents the conditional dependencies in a set of random variables by using a directed acyclic graph (DAG) $\mathcal{G}$ augmented with a set of conditional probability distributions $\theta_{\mathcal{G}}$ associated with $\mathcal{G}$'s vertices. In such graph, each vertex corresponds to a random variable X_i (e.g. an observable quantity, a set of unknown parameters etc.) and each edge indicates a *direct influence* relation between the two random variables; this allows to define *conditional independence* relationships between the variables, which are independent from any of their non-descendants, given the value of their parent variables.

A BN stipulates a set of *conditional independence assumptions*, also called *local Markov assumptions*, over its set of random variables: each vertex X_i in the DAG is conditionally independent of any subset $S \subseteq Nd(X_i)$ of vertices that are not descendants of X_i given a joint state of its parents:

$$\forall X_i : \Pr(X_i \mid S, parents(X_i)) = \Pr(X_i \mid parents(X_i));$$

where the function $parents(X_i)$ returns the parent vertices of X_i in the DAG representing the BN. The conditional independence assumption allows to represent the *joint probability distribution* $\Pr(X_1, \ldots, X_n)$ defined by a BN over a set of random variables $\{X_1, \ldots, X_n\}$ as a production of the individual probability distributions, conditional on their parent variables:

$$\Pr(X_1, \ldots, X_n) = \prod_{i=1}^{n} \Pr(X_i \mid parents(X_i));$$

As a result, it is possible to define $\Pr(X_1, \ldots, X_n)$ by only specifying, for each vertex X_i in the graph, the conditional probability distribution $\Pr(X_i \mid parents(X_i))$.

Given a BN specifying a joint probability distribution over a set of variables, it is possible to evaluate inference queries by marginalization, like calculating the posterior probability distribution for a set of query variables given some observed event (i.e. assignment of values to the set of evidence variables).

Exact inference for general BNs is an NP-hard problem, but algorithms exist to efficiently infer in restricted classes of networks, such as variable elimination, which has linear complexity in the number of vertices if the BN is a singly connected network [15]. Approximate inference methods also exist in literature, such as *Monte Carlo* algorithms, that provide approximate answers whose accuracy depends on the number of samples generated. Other methods in this family, such as *belief propagation* or *variational methods*, approximate sums of random variables through their means [15].

However, finding an optimal structure for a BN may not be trivial: the number of possible structures for a DAG is super-exponential ($\mathcal{O}(2^{f(n)})$, with $f(n) = n^{1+\epsilon}$, $\epsilon > 0$) in the size of its vertices ($r_4 = 543$, $r_8 \approx 7,8 \times 10^{11}$, $r_{12} \approx 5,2 \times 10^{26}$), making it impractical, in many cases, to perform an exhaustive search through the space of possible structures. Therefore, in our approach, we tried to find an acceptable trade-off between efficiency and expressiveness, so to make our method suitable for a context like SW: we decided to focus on a particular subclass of Bayesian networks, i.e. *naïve Bayesian networks*, modelling the dependencies between a set of random variables $\mathcal{F} = \{F_1, \ldots, F_n\}$, also called *features*, and a random variable C, also called *class*, so that each pair of features are independent of each other given the class, i.e. $\forall F_i, F_j \in \mathcal{F} : i \neq j \Rightarrow (F_i \perp\!\!\!\perp F_j \,|\, C)$.

This kind of models is especially interesting since they proved to be effective also in contexts in which the underlying independence assumptions are violated [8], even outperforming more current approaches [4].

However, defining a BN requires a number of precise probability assessments which, as we will see, will not be always possible to obtain. A generalisation of naïve Bayesian networks to probability intervals is the *robust Bayesian estimator* [20] (RBE): each conditional probability in the network is a *probability interval* characterised by its *lower* and *upper bounds*, defined respectively as $\underline{\Pr}(A) = \min_{\Pr \in \mathcal{P}} \Pr(A)$ and $\overline{\Pr}(A) = \max_{\Pr \in \mathcal{P}} \Pr(A)$.

The main problem with this approach is assigning class labels, after having calculated the posterior probability intervals: if the two resulting intervals do not overlap, it is possible to apply the so called *stochastic dominance criterion*, which assigns a generic individual a to a target concept C iff $\underline{\Pr}(C(a)) > \overline{\Pr}(\neg C(a))$. If the intervals overlap, to avoid undecidability, it is still possible to use a weaker criterion, called *weak dominance criterion* [20] by representing each probability interval into a single probability value represented by its middle point, which indeed underlies some assumptions on the distribution of the missing values.

A similar approach, founded on *imprecise probability theory*, is presented in [5] and proposes using a *Credal network* (structurally similar to a BN, but where the conditional probability densities belong to convex sets of mass functions) to represent uncertainty about network parameters.

3 Terminological Naïve Bayesian Classifiers

The learning problem we intend to focus on consists in learning a terminological naïve Bayesian classifier $\mathcal{N}_\mathcal{K}$; this is defined as a naïve BN modelling the dependency relations between a set of Description Logic (DL) concepts (also referred to as *feature concepts*) and a target atomic concept C, given a set of training individuals. Feature

concepts may eventually be complex, and the training individuals are distinguished in *positive*, *negative* and *neutral*, belonging respectively to the target concept C, $\neg C$ and or whose membership of C is unknown. A DL Knowledge Base (KB) $\mathcal{K}$ is typically constituted by (at least) two main components, a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$:

- **TBox** – which introduces the *terminology* of an application domain, in terms of axioms describing concept hierarchies;
- **ABox** – which contains *assertions* (ground axioms) about named individuals in terms of this terminology.

A terminological Bayesian classifier can be defined as follows:

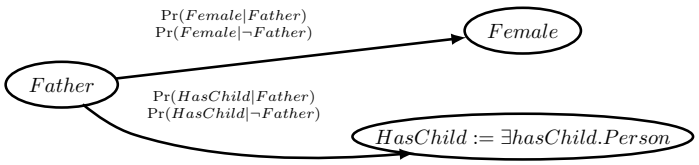
Definition 1 (Terminological Bayesian Classifier). *A terminological Bayesian classifier $\mathcal{N}_{\mathcal{K}}$, with respect to a DL KB $\mathcal{K}$, is defined as a pair $\langle \mathcal{G}, \Theta_{\mathcal{G}} \rangle$, where:*

- *$\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ is a directed acyclic graph, in which:*
  - *$\mathcal{V} = \{F_1, \ldots, F_n, C\}$ is a set of vertices, each F_i representing a DL (eventually complex) concepts defined over $\mathcal{K}$ and C representing a target atomic concept;*
  - *$\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges, modelling the independence relations between the elements of $\mathcal{V}$;*
- *$\Theta_{\mathcal{G}}$ is a set of* conditional probability distributions *(CPD), one for each $V \in \mathcal{V}$, representing the conditional probability of the feature concept given the state of its parents in the graph.*

in which the membership probability of a generic individual a to the target concept C (or $\neg C$) is estimated using BN inference techniques given the membership of a to the concepts in $\mathcal{V}$.

In particular, a terminological naïve Bayesian Classifier is characterised by the following structure: $\mathcal{E} = \{\langle C, F_i \rangle \mid i \in \{1, \ldots, n\}\}$ (i.e. each feature concept is independent from the other feature concept, given the value of the target atomic concept).

Example 1 (Example of Terminological Naïve Bayesian Classifier). Given a set of DL feature concepts $\mathcal{F} = \{Female, HasChild := \exists hasChild.Person\}$ [2] and a target concept $Father$, a terminological naïve Bayesian classifier expressing the target concept in terms of the feature concepts is the following:



Let $\mathcal{K}$ be a DL KB and a a generic individual so that $\mathcal{K} \models HasChild(a)$ and the membership of a to the concept $Female$ is not known, i.e. $\mathcal{K} \not\models Female(a) \wedge \mathcal{K} \not\models \neg Female(a)$. It is possible to infer, through the given network, the probability that the individual a is a member of the target atomic concept $Father$:

[2] In examples, variable names are used instead of complex feature concepts for brevity

$$\Pr(Father(a)) = \frac{\Pr(Father)\Pr(HasChild \mid Father)}{\sum_{Father' \in \{Father, \neg Father\}} \Pr(Father')\Pr(HasChild \mid Father')};$$

In the following we define the problem of learning a terminological Bayesian classifier $\mathcal{N}_\mathcal{K}$ given a DL KB $\mathcal{K}$ and the training individuals $Ind_C(\mathcal{A})$:

Definition 2 (Terminological Bayesian Classifier Learning Problem). *Our terminological naïve Bayesian classifier learning problem consists in finding a network $\mathcal{N}_K^*$ that maximizes the quality of the network with respect to the training instances and a specific scoring function; formally:*

Given

- *a* target concept C *we aim to learn;*
- *a DL KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, where the ABox $\mathcal{A}$ contains membership assertions about individuals and C, while the TBox $\mathcal{T}$ does not contain assertions involving C;*
- *the disjoint sets of of positive, negative and neutral examples for C, denoted with $Ind_C^+(\mathcal{A})$, $Ind_C^-(\mathcal{A})$ and $Ind_C^0(\mathcal{A})$, so that:*
  - $\forall a \in Ind_C^+(\mathcal{A}) : C(a) \in \mathcal{A}$,
  - $\forall a \in Ind_C^-(\mathcal{A}) : \neg C(a) \in \mathcal{A}$,
  - $\forall a \in Ind_C^0(\mathcal{A}) : C(a) \notin \mathcal{A} \wedge \neg C(a) \notin \mathcal{A}$;
- *a* scoring function *specifying the quality of an induced terminological Bayesian classifier $\mathcal{N}_\mathcal{K}$ with respect to the samples in $Ind_C(\mathcal{A}) = \bigcup_{v \in \{+,-,0\}} Ind_C^v(\mathcal{A})$ and a scoring criterion;*

Find *a network $\mathcal{N}_\mathcal{K}^*$ maximizing the score function with respect to the samples:*

$$\mathcal{N}_\mathcal{K}^* \leftarrow \arg\max_{\mathcal{N}_\mathcal{K}} score(\mathcal{N}_\mathcal{K}, Ind_C(\mathcal{A}))).$$

Our search space, to find the optimal network $\mathcal{N}_\mathcal{K}^*$, may be too large to explore exhaustively; therefore our learning algorithm, outlined in Alg. 1, works by greedily searching the space of features (i.e. DL complex concepts) for the ones that maximize the score of the induced network, with respect to a scoring function, and incrementally building the resulting network. While the features are added one by one, the search in the space of DL complex concepts is made through a beam search, employing the $\rho_\downarrow^{cl}$ closure of the downward refinement operator $\rho_\downarrow$ described in [17].

For each new complex concept being evaluated, the algorithm creates a new set of concepts $\mathcal{V}'$ and finds the optimal structure (under a given set of constraints) $\mathcal{E}'$ (which, in the case of terminological naïve Bayesian classifiers, is already defined) and the corresponding maximum likelihood parameters $\Theta_{\mathcal{G}'}$ (which may vary depending on the assumptions on the nature of the ignorance model), then scores the new network with respect to a scoring criterion.

Different Assumptions on the Ignorance Model

Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL KB; under OWA, it is not always possible to know if a generic DL assertion α is or is not entailed by $\mathcal{K}$ (i.e. there may be cases in which $\mathcal{K} \not\models \alpha \wedge \mathcal{K} \not\models \neg\alpha$). This allows us to characterize our lack of knowledge about concept-memberships through the probability distribution of the ignorance model [23]. Given a generic concept C, a generic individual a and a DL KB $\mathcal{K}^*$, let $\mathcal{I}$ be an *ignorance model* from which we extract a fragment of $\mathcal{K}^*$, $\mathcal{I}(\mathcal{K}^*) = \mathcal{K}$ (so that $\forall \alpha : \mathcal{K} \models \alpha \Rightarrow \mathcal{K}^* \models \alpha \wedge \mathcal{K}^* \models \alpha \not\Rightarrow \mathcal{K} \models \alpha$). Let denote $\mathcal{N}_\mathcal{K}$ as a probabilistic model that, from a DL KB $\mathcal{K}$, calculates the probability that the concept-membership relation between C and a is unknown. We can say that the ignorance model underlying the concept-membership relation between a and C in $\mathcal{K}$ (with respect to a, $\mathcal{K}^*$ and the aforementioned probabilistic model) is:

- **MCAR** (Missing Completely at Random) – when the probability for such concept-membership to be missing is independent from the knowledge on a available in $\mathcal{K}^*$: $\Pr(\mathcal{K} \not\models C(a) \wedge \mathcal{K} \not\models \neg C(a) \mid \mathcal{K}^*) = \Pr(\mathcal{K} \not\models C(a) \wedge \mathcal{K} \not\models \neg C(a))$;
- **MAR** (Missing At Random) – when the probability for such concept-membership to be missing depends on the knowledge on a available in $\mathcal{K}$: $\Pr(\mathcal{K} \not\models C(a) \wedge \mathcal{K} \not\models \neg C(a) \mid \mathcal{K}^*) = \Pr(\mathcal{K} \not\models C(a) \wedge \mathcal{K} \not\models \neg C(a) \mid \mathcal{K})$;
- **NMAR** (Not Missing At Random, also referred to as **IM**, Informatively Missing) – when the probability for such concept-membership to be missing depends on the knowledge on a available in $\mathcal{K}^*$: $\Pr(\mathcal{K} \not\models C(a) \wedge \mathcal{K} \not\models \neg C(a) \mid \mathcal{K}^*) \neq \Pr(\mathcal{K} \not\models C(a) \wedge \mathcal{K} \not\models \neg C(a) \mid \mathcal{K})$.

Algorithm 1 Algorithm for Learning Terminological Bayesian Classifiers

Require: DL KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, Concept $Start$, Beam Width w,
Search depth d, Maximum concept description length $maxLen$,
Positive, Negative, Neutral training individuals $Ind_C(\mathcal{A})$;
Ensure: $\mathcal{N}_\mathcal{K} = \langle \mathcal{G}, \Theta_\mathcal{G} \rangle, \mathcal{G} = \langle \mathcal{V} \leftarrow \{C\}, \mathcal{E} \leftarrow \emptyset \rangle$;

repeat
  $Best \leftarrow \emptyset$; $Beam \leftarrow \{Start\}$; $NewBeam \leftarrow \emptyset$;
  repeat
    for $c \in Beam$ **do**
      for $c' \in \{\rho^{cl}_\downarrow(c) \mid |c'| \leq min(|c| + d, maxLen)\}$ **do**
        $\mathcal{N}'_\mathcal{K} \leftarrow optimalNetwork(\mathcal{V} \cup \{c'\}, Ind_C(\mathcal{A}))$;
        $s' \leftarrow score(\mathcal{N}'_\mathcal{K}, Ind_C(\mathcal{A}))$;
        $NewBeam \leftarrow NewBeam \cup \{\langle \mathcal{N}'_\mathcal{K}, s' \rangle\}$;
      end for
    end for
    $Best \leftarrow \arg\max_{\langle \mathcal{N}'_\mathcal{K}, s' \rangle}(s' : \langle \mathcal{N}'_\mathcal{K}, s' \rangle \in NewBeam \cup \{Best\})$;
    $Beam \leftarrow selectFrom(NewBeam, w)$; $NewBeam \leftarrow \emptyset$;
  until stopping criterion on $Beam$;
  $\mathcal{N}_\mathcal{K} \leftarrow \mathcal{N}'_\mathcal{K} : \langle \mathcal{N}'_\mathcal{K}, s' \rangle = Best$;
until stopping criterion on $\mathcal{N}_\mathcal{K}$;

Specifically, in our algorithm, the outer loop (lines 1-15) greedily searches for a new (complex) concept definition whose addition increases the network's quality on the given sample instances (determined by a scoring function $score$). The search through the space of concept definitions is performed in the inner loop (lines 3-13) through a beam search: starting from a beginning concept $Start$, for each refinement level, all refinements up to a given length are memorized in a priority queue $NewBeam$ (sorted according to the score associated to the network generated by adding them to the set of feature concepts) from which only the k with the highest score are selected, by the selection function $selectFrom$, to be refined in the next iteration.

The functions $optimalNetwork$ and $score$ are used, respectively, to find the optimal Bayesian network structure between the nodes in the network (eventually under a set of constraints, like in the naïve Bayes case or some of its extensions) and for scoring a classifier (to compare its effectiveness with others). However, those two functions are sensitive to the assumptions made about the ignorance model.

When the assumed ignorance model is **MCAR**, we are allowed to use an approach called *available case analysis* [15], in which we build an unbiased estimator of the network parameters, based only on available knowledge. A scoring function we realised for such case is the network's log-likelihood on training data, calculated only on positive and negative training individuals, ignoring the available knowledge about the concept-membership relations between such individuals and the target concept C, and defined as:

$$\mathcal{L}(\mathcal{N}_\mathcal{K} \mid Ind_C(\mathcal{A})) = \log \Pr(\mathcal{N}_\mathcal{K}) + \sum_{a \in Ind_C^+(\mathcal{A})} \log \Pr(C(a) \mid \mathcal{N}_\mathcal{K}) + \sum_{a \in Ind_C^-(\mathcal{A})} \log \Pr(\neg C(a) \mid \mathcal{N}_\mathcal{K});$$

Another approach we implemented consisted in ranking both positive and negative training individuals a according to $P(C(a) \mid \mathcal{N}_\mathcal{K})$, and then calculating the area under the Precision-Recall curve using different acceptance thresholds.

Under the naïve Bayes assumption, there is no need to perform a search for finding the optimal network, since the structure is already fixed (each node except the target concept node has only one parent, i.e. the target concept node); otherwise, finding a network structure which is optimal under some criterion (e.g. the BIC score [15]) may require an exhaustive search in the space of possible structures. However, tree-augmented naïve Bayesian networks (which allow for a tree structure among feature nodes), it is possible to efficiently compute the optimal structure employing the method in [12], making it appealing for real-life applications requiring efficiency and scalability.

In the **MAR** case, a possible solution for learning models accounting for missing knowledge is to use the Expectation-Maximization (EM) algorithm, MCMC sampling or the gradient ascent method [15]. We use EM to learn terminological naïve Bayesian classifiers from MAR data. In our approach, outlined in Alg. 2, we first heuristically estimate network's parameters by only using available data; then, in order to find the maximum likelihood parameters with respect to both observed and missing knowledge, we consider individuals whose membership to a particular concept description D is not known as several fractional individuals belonging, with different weights (corresponding to the posterior probability of their class membership), to both the components D and $\neg D$.

Formally, the EM algorithm for parameters learning explores the space of possible parameters through an iterative hill-climbing search, converging to a (local) maximum likelihood estimate of the unknown parameters, where the (log-)likelihood (which we also use as scoring criterion) is defined as follows:

$$\mathcal{L}(\mathcal{N}_{\mathcal{K}} \mid Ind_C(\mathcal{A})) = \log \Pr(\mathcal{N}_{\mathcal{K}}) + \sum_{a \in Ind_C^0(\mathcal{A})} \sum_{C' \in \{C, \neg C\}} \log \Pr(C'(a) \mid \mathcal{N}_{\mathcal{K}}) \Pr(C' \mid \mathcal{N}_{\mathcal{K}}) + \sum_{a \in Ind_C^+(\mathcal{A})} \log \Pr(C(a) \mid \mathcal{N}_{\mathcal{K}}) + \sum_{a \in Ind_C^-(\mathcal{A})} \log \Pr(\neg C(a) \mid \mathcal{N}_{\mathcal{K}});$$

At each iteration, the EM algorithm applies the following two steps:

- **Expectation step** – using available data and the current network parameters, calculate a distribution over possible completions for the missing knowledge;
- **Maximization step** – considering each possible completion as a fully available data case (weighted by its probability), calculate next parameters using (weighted) frequency counting.

In our use of the EM algorithm, the E-step calculates the concept-membership posterior probability (inferencing through the network) of each individual whose concept-membership relation in unknown, thus completing the data through so called *expected counts*. Then, the M-step calculates a new estimate of the network's conditional probability distributions by using expected counts, maximizing the log-likelihood of both available and missing data with respect to a network $\mathcal{N}_{\mathcal{K}}$.

About finding optimal structures for networks with less restrictions on their structure (such as tree-augmented naïve BNs or unrestricted BNs) from MAR data, it is possible to employ the Structural EM (SEM) algorithm [11]. In SEM, the maximization step is performed both in the space of structures $\mathcal{G}$ and in the space of parameters $\Theta_{\mathcal{G}}$, by first searching a better structure and then the best parameters associated to the given structure; it can be proven that, if the search procedure finds a structure that is better than the one used in the previous iteration with respect to e.g. the BIC score, then the structural EM algorithm will monotonically improve the score.

When knowledge is **NMAR**, it is generally possible to extend the probabilistic model to produce one where the MAR assumption holds; e.g. if a feature concept F_i follows a NMAR ignorance model, with respect to a generic individual a and a DL KB $\mathcal{K}$, we can consider its observability as an additional variable (e.g. $Y_i = 0$ iff $\mathcal{K} \not\models F_i(a) \wedge \mathcal{K} \not\models \neg F_i(a)$, $Y_i = 1$ otherwise) in our probabilistic model, so that F_i's ignorance model satisfies the MAR assumption (since its missingness depends on an always observable variable).

An alternate solution is recurring to *robust Bayesian estimation* [20] (RBE), to learn conditional probability distributions without making any sort of assumption about the nature of the missing data. RBE finds probability intervals instead of single probability values, obtained by taking in account all the possible fillings of the missing knowledge; the width of inferred intervals is therefore directly proportional to the quantity of missing knowledge considered during the learning process. To score each new induced network, we employ the framework proposed in [26] to compare credal networks, while

Algorithm 2 Outline for our implementation of the EM algorithm for parameter learning in a terminological Bayesian classifier assuming the underlying ignorance model is MAR.

function $ExpectedCounts(\mathcal{N}_{\mathcal{K}}, Ind_C(\mathcal{A}))$

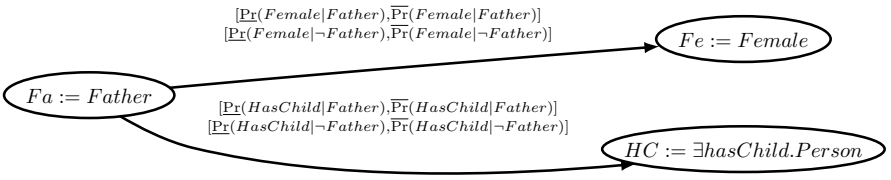
$\mathcal{N}_{\mathcal{K}} = \langle \mathcal{G}, \Theta_{\mathcal{G}} \rangle, \mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$;
for $X_i \in \mathcal{V}$ **do**
  for $\langle x_i, \pi_{x_i} \rangle \in vals(X_i, parents(X_i))$ **do**
    $\{\bar{n}(x_i, \pi_{x_i})$ contains the expected count for $(X_i = x_i, parents(X_i) = \pi_{x_i})\}$
    $\bar{n}(x_i, \pi_{x_i}) \leftarrow 0$;
  end for
end for
for $a \in Ind_C(\mathcal{A})$ **do**
  for $X_i \in \mathcal{V}$ **do**
    for $\langle x_i, \pi_{x_i} \rangle \in vals(X_i, parents(X_i))$ **do**
      {Each expected count $\bar{n}(x_i, \pi_{x_i})$ is obtained summing out the probability assignments to the concept memberships $(X_i = x_i, parents(X_i) = \pi_{x_i})$ for each individual, calculated using the background knowledge $\mathcal{K}$ and, if they are only partially known, inferring through the network $\mathcal{N}_{\mathcal{K}}$}
      $\bar{n}(x_i, \pi_{x_i}) \leftarrow \bar{n}(x_i, \pi_{x_i}) + \Pr(x_i, \pi_{x_i} \mid \mathcal{N}_{\mathcal{K}})$;
    end for
  end for
end for
return $\{\bar{n}(x_i, \pi_{x_i}) \mid X_i \in \mathcal{V}, \langle x_i, \pi_{x_i} \rangle \in vals(X_i, parents(X_i))\}$;

function $ExpectationMaximization(\mathcal{N}^0_{\mathcal{K}}, Ind_C(\mathcal{A}))$

{The network was first initialized with arbitrary heuristic parameters $\Theta^0_{\mathcal{G}}$}
$\mathcal{N}^0_{\mathcal{K}} = \langle \mathcal{G}, \Theta^0_{\mathcal{G}} \rangle, \mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$;
$t \leftarrow 0$;
repeat
  $\{\bar{n}(x_i, \pi_{x_i})\} \leftarrow ExpectedCounts(\mathcal{N}_{\mathcal{K}}, Ind_C(\mathcal{A}))$;
  for $X_i \in \mathcal{V}$ **do**
    for $\langle x_i, \pi_{x_i} \rangle \in vals(X_i, parents(X_i))$ **do**
      $\theta^{t+1}_{\mathcal{G}}(x_i, \pi_{x_i}) \leftarrow \frac{\bar{n}(x_i, \pi_{x_i})}{\sum_{x'_i \in vals(X_i)} \bar{n}(x'_i, \pi_{x_i})}$;
    end for
  end for
  $t \leftarrow t + 1$;
  $\mathcal{N}^t_{\mathcal{K}} = \langle \mathcal{G}, \Theta^t_{\mathcal{G}} \rangle$;
  {The EM loop ends when improvements to the network's log-likelihood go below a certain threshold}
until $\mathcal{L}(\mathcal{N}^t_{\mathcal{K}} = \langle \mathcal{G}, \Theta^t_{\mathcal{G}} \rangle) \mid Ind_C(\mathcal{A})) - \mathcal{L}(\mathcal{N}^{t-1}_{\mathcal{K}} = \langle \mathcal{G}, \Theta^{t-1}_{\mathcal{G}} \rangle \mid Ind_C(\mathcal{A})) \leq \tau$;
return $\mathcal{N}^t_{\mathcal{K}}$;

we do not have implemented yet a method to search for structures other than naïve Bayesian.

Example 2 (Example of Terminological Naïve Bayesian Classifier using Robust Bayesian Estimation). The following is a terminological naïve Bayesian classifier using robust Bayesian estimation for inferring posterior probability intervals in presence of NMAR knowledge. In this networks, conditional probability tables associated to each node contain probability intervals instead of probability values, each defined by its upper and lower bound.



Inference, using such network, can be performed as follows – given a generic individual a and given that $\mathcal{K} \models HC(a)$, the posterior probability interval that a is a member of Fa is represented by the probability interval $[\underline{\Pr}(Fa \mid HC), \overline{\Pr}(Fa \mid HC)]$, where:

$$\underline{\Pr}(Fa(a)) = \underline{\Pr}(Fa \mid HC) = \frac{\underline{\Pr}(HC \mid Fa)\underline{\Pr}(Fa)}{\underline{\Pr}(HC \mid Fa)\underline{\Pr}(Fa) + \overline{\Pr}(HC \mid \neg Fa)\overline{\Pr}(\neg Fa)};$$

$$\overline{\Pr}(Fa(a)) = \overline{\Pr}(Fa \mid HC) = \frac{\overline{\Pr}(HC \mid Fa)\overline{\Pr}(Fa)}{\overline{\Pr}(HC \mid Fa)\overline{\Pr}(Fa) + \underline{\Pr}(HC \mid \neg Fa)\underline{\Pr}(\neg Fa)};$$

4 Conclusions and Future Work

We presented a Statistical Relational Learning method designed for learning terminological naïve Bayesian classifiers, a ML method based on the naïve Bayes assumption for estimating the probability that a generic individual belongs to a certain target concept, given its membership relation to an induced set of complex Description Logic concepts. We gave a characterisation of the lack of knowledge that may be introduced by the OWA depending on the underlying ignorance model, and handled such missing knowledge under different assumptions on the nature of missing knowledge itself (i.e. *Missing Completely at Random*, *Missing at Random* or *Informatively Missing*). In the future, we aim at estimating computationally the ignorance model followed by each feature, at developing new methods to exploit the potential information contained in knowledge's missingness and evaluate our methods' effectiveness on real world ontologies.

References

[1] OWL 2 Web Ontology Language Direct Semantics (October 2009), `http://www.w3.org/TR/owl2-direct-semantics/`

[2] Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American 284(5), 34–43 (May 2001)
[3] Bicer, V., et al.: Relational kernel machines for learning from graph-structured rdf data. In: Antoniou, G., et al. (eds.) ESWC (1). LNCS, vol. 6643, pp. 47–62. Springer (2011)
[4] Caruana, R., Niculescu-Mizil, A.: An empirical comparison of supervised learning algorithms. In: ICML2006. pp. 161–168. ACM, New York, NY, USA (2006)
[5] Corani, G., Zaffalon, M.: Naive credal classifier 2: an extension of naive bayes for delivering robust classifications. In: DMIN. pp. 84–90 (2008)
[6] d'Amato, C., Fanizzi, N., Esposito, F.: Query answering and ontology population: an inductive approach. In: ESWC 2008. pp. 288–302. Springer (2008)
[7] Domingos, P., et al.: Uncertainty reasoning for the semantic web i. chap. Just Add Weights: Markov Logic for the Semantic Web, pp. 1–25. Springer (2008)
[8] Domingos, P., Pazzani, M.J.: On the optimality of the simple bayesian classifier under zero-one loss. Machine Learning 29(2-3), 103–130 (1997)
[9] Fanizzi, N., et al.: Reduce: A reduced coulomb energy network method for approximate classification. In: Aroyo, L., et al. (eds.) ESWC. pp. 323–337. Springer (2009)
[10] Fanizzi, N., D'Amato, C., Esposito, F.: Learning with kernels in description logics. In: ILP 2008. pp. 210–225. Springer (2008)
[11] Friedman, N.: The Bayesian structural EM algorithm. In: UAI 1998. pp. 129–138. Morgan Kaufmann Publishers Inc., San Francisco, CA (1998)
[12] Friedman, N., Geiger, D., Goldszmidt, M., Provan, G., Langley, P., Smyth, P.: Bayesian network classifiers. In: Machine Learning. pp. 131–163 (1997)
[13] Getoor, L., Taskar, B.: Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning). The MIT Press (2007)
[14] Kiefer, C., et al.: Adding Data Mining Support to SPARQL via Statistical Relational Learning Methods. In: ESWC 2008. LNCS, vol. 5021, pp. 478–492. Springer (2008)
[15] Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press (2009)
[16] Laskey, K.J., Laskey, K.B.: Uncertainty reasoning for the world wide web: Report on the urw3-xg incubator group. In: Bobillo, F., et al. (eds.) URSW. CEUR Workshop Proceedings, vol. 423. CEUR-WS.org (2008)
[17] Lehmann, J., Hitzler, P.: Concept learning in description logics using refinement operators. Mach. Learn. 78, 203–250
[18] Luna, J.E.O., Cozman, F.G.: An algorithm for learning with probabilistic description logics. In: Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.) URSW. CEUR Workshop Proceedings, vol. 527, pp. 63–74. CEUR-WS.org (2009)
[19] Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1988)
[20] Ramoni, M., Sebastiani, P.: Robust learning with missing data. Mach. Learn. 45, 147–170 (October 2001)
[21] Richardson, M., Domingos, P.: Markov logic networks. Mach. Learn. 62, 107–136 (February 2006)
[22] Rodrigues De Morais, S., Aussem, A.: Exploiting data missingness in bayesian network modeling. In: IDA 2009. pp. 35–46. Springer (2009)
[23] Rubin, D.B.: Inference and missing data. Biometrika 63(3), 581–592 (1976)
[24] Tresp, V., et al.: Uncertainty reasoning for the semantic web i. chap. Towards Machine Learning on the Semantic Web, pp. 282–314. Springer (2008)
[25] Tresp, V., Huang, Y., Bundschus, M., Rettinger, A.: Materializing and querying learned knowledge. In: IRMLeS 2009 (2009)
[26] Zaffalon, M., Corani, G., Mauá, D.: Utility-based accuracy measures to empirically evaluate credal classifiers. In: ISIPTA 2011. pp. 401–410. Innsbruck (2011)

A Distribution Semantics for Probabilistic Ontologies

Elena Bellodi, Evelina Lamma, Fabrizio Riguzzi, and Simone Albani

ENDIF – University of Ferrara, Via Saragat 1, I-44122, Ferrara, Italy
{elena.bellodi,evelina.lamma,fabrizio.riguzzi}@unife.it
simone.albani@student.unife.it

Abstract. We present DISPONTE, a semantics for probabilistic ontologies that is based on the distribution semantics for probabilistic logic programs. In DISPONTE each axiom of a probabilistic ontology is annotated with a probability. The probabilistic theory defines thus a distribution over normal theories (called worlds) obtained by including an axiom in a world with a probability given by the annotation. The probability of a query is computed from this distribution with marginalization. We also present the system BUNDLE for reasoning over probabilistic OWL DL ontologies according to the DISPONTE semantics. BUNDLE is based on Pellet and uses its capability of returning explanations for a query. The explanations are encoded in a Binary Decision Diagram from which the probability of the query is computed.

1 Introduction

Representing probabilistic knowledge and reasoning with it is fundamental in order to realize the full vision of the Semantic Web, due to the ubiquity of uncertainty in the real world and on the Web [24]. Various authors have advocated the use of probabilistic ontologies, see e.g. [17], and many proposals have been put forward for allowing ontology languages, and OWL in particular, to represent uncertainty.

Similarly, in the field of logic programming, there has been much work on introducing uncertainty in the programs. Among the various proposals, the distribution semantics [22] has emerged as one of the most effective approaches and it underlies many languages such as PRISM [22], ICL [19], Logic Programs with Annotated Disjunctions [26] and ProbLog [3]. In this semantics a probabilistic logic program defines a probability distribution over a set of normal logic programs (called *worlds*). The distribution is extended to a joint distribution over worlds and queries; the probability of a query is obtained from this distribution by marginalization. In general, the problem of integrating logic and probability has been much studied lately, with proposals such as Markov Logic [20], Multi Entity Bayesian Networks [12] and Probabilistic Relational Models [10].

In this paper we propose to apply this approach to ontology languages and, in particular, to the OWL DL fragment, that is based on the description logic $\mathcal{SHOIN}(\mathbf{D})$. However, the approach is applicable in principle to any description

logic. We called the approach DISPONTE for "DIstribution Semantics for Probabilistic ONTologiEs" (Spanish for "get ready"). The idea is to annotate each axiom of a theory with a probability and assume that each axiom is independent of the others. A probabilistic theory defines thus a distribution over normal theories (worlds) obtained by including an axiom in a world with a probability given by the annotation. The probability of a query is again computed from this distribution with marginalization.

We also present the system BUNDLE for "Binary decision diagrams for Uncertain reasoNing on Description Logic thEories" that performs inference over probabilistic OWL DL ontologies. BUNDLE uses the inference techniques developed for probabilistic logic programs under the distribution semantics [8,21] and, in particular, the use of Binary Decision Diagrams (BDDs) for encoding explanations to queries and for computing their probability.

BUNDLE is based on the Pellet reasoner [23] for OWL DL and exploits its capability of returning explanations for queries in the form of a set of sets of axioms from which BUNDLE builds a BDD for computing the probability. In this way we provide an effective reasoning system for DISPONTE.

The paper is organized as follows. Section 2 describes the distribution semantics for logic programs while Section 3 presents DISPONTE. Section 4 illustrates BUNDLE and Section 5 discusses current limitations of DISPONTE and BUNDLE. Section 6 describes related works while Section 7 concludes the paper.

2 The Distribution Semantics in Probabilistic Logic Programming

The probabilistic logic programming languages based on the distribution semantics differ in the way they define the distribution over logic programs. Each language allows probabilistic choices among atoms in clauses. Let us consider ProbLog [3] which is the language with the simplest syntax. A ProbLog program T is composed of a normal logic program T_C and a set of probabilistic facts T_P. Each probabilistic fact is of the form $p_i :: F_i.$ where p_i is a probability (i.e. $p_i \in [0, 1]$) and F_i is a atom. This means that every grounding of F_i is a Boolean random variable that assumes *true* value with probability p_i and *false* with probability $1 - p_i$.

Let us call T_F the set of atoms obtained by removing the probabilistic annotation from the probabilistic facts. Let us consider the case in which $T_C \cup T_F$ does not contain function symbols so that its Herbrand base is finite. Let us call $ground(T)$ the grounding of a normal program T. Since there are no function symbols, $ground(T_C \cup T_F)$ is finite and so is the grounding $ground(T_F)$ obtained by grounding the probabilistic atoms with constants from the Herbrand universe of $T_C \cup T_F$. So each probabilistic fact F_i has a finite set of groundings.

A substitution is a set of couples V/c where V is a variable and c is a constant. A substitution θ_j is applied to a logic atom F, indicated with $F\theta_j$, by replacing the variables in the substitution with constants. A substitution θ_j is grounding for logic atom F if $F\theta_j$ is ground. Suppose that a grounding is obtained with

the substitution θ_j: $F_i\theta_j$ corresponds to a Boolean random variable X_{ij} that is independent of the others.

Example 1. The following ProbLog program T encodes a very simple model of the development of an epidemic or pandemic:

$C_1 = epidemic : -flu(X), epid(X), cold.$
$C_2 = pandemic : -flu(X), \backslash + epid(X), pand(X), cold.$
$C_3 = flu(david).$
$C_4 = flu(robert).$
$F_1 = 0.7 :: cold.$
$F_2 = 0.6 :: epid(X).$
$F_3 = 0.3 :: pand(X).$

This program models the fact that if somebody has the flu and the climate is cold there is the possibility that an epidemic or a pandemic arises. We are uncertain whether the climate is cold but we know for sure that David and Robert have the flu. $epid(X)$ and $pand(X)$ can be considered as "probabilistic activators" of the effects in the head given that the causes ($flu(X)$ and $cold$) are present. $\backslash + epid(X)$ means the negation of $epid(X)$.

Fact F_1 has only one grounding so there is a single Boolean variable X_{11}. Fact F_2 has two groundings, $epid(david)$ and $epid(robert)$ so there are two Boolean random variables X_{21} and X_{22}. F_3 also has two groundings so there are two Boolean random variables X_{31} and X_{32}.

In order to present the distribution semantics, let us first give some definitions. An *atomic choice* is a selection of a value for a grounding of a probabilistic fact F and is represented by the triple (F_i, θ_j, k) where θ_j is a substitution grounding F_i and $k \in \{0, 1\}$. A set of atomic choices κ is *consistent* if $(F_i, \theta_j, k) \in \kappa, (F_i, \theta_j, m) \in \kappa \Rightarrow k = m$, i.e., only one truth value is selected for a ground fact. A *composite choice* κ is a consistent set of atomic choices. The probability of composite choice κ is $P(\kappa) = \prod_{(F_i,\theta_j,1)\in\kappa} p_i \prod_{(F_i,\theta_j,0)\in\kappa}(1 - p_i)$. A *selection* σ is a total composite choice (one atomic choice for every grounding of every probabilistic fact). A selection σ identifies a normal logic program w_σ called a *world* in this way: $w_\sigma = T_C \cup \{F_i\theta_j | (F_i, \theta_j, 1) \in \sigma\}$. The probability of w_σ is $P(w_\sigma) = P(\sigma) = \prod_{(F_i,\theta_j,1)\in\kappa} p_i \prod_{(F_i,\theta_j,0)\in\kappa}(1 - p_i)$. Since $ground(T_F)$ is finite the set of worlds is finite: $W_T = \{w_1, \ldots, w_m\}$ and $P(w)$ is a distribution over worlds: $\sum_{w\in W_T} P(w) = 1$. A world w_σ is *compatible* with a composite choice κ if $\kappa \subseteq \sigma$

We can define the conditional probability of a query Q given a world as $P(Q|w) = 1$ if $w \models Q$ and 0 otherwise. This allows to define a joint distribution of the query and the worlds $P(Q, w)$ by using the product rule of the theory of probability: $P(Q, W) = P(Q|w)P(w)$. The probability of Q can then be obtained from the joint distribution by the sum rule (marginalization over Q):

$$P(Q) = \sum_{w\in W_T} P(Q, w) = \sum_{w\in W_T} P(Q|w)P(w) = \sum_{w\in W_T : w\models Q} P(w) \qquad (1)$$

In Example 1, T has 5 Boolean random variables and thus 32 worlds. The query *epidemic* is true in 5 of them and its probability is $P(epidemic) = 0.588$.

It is often unfeasible to find all the worlds where the query is true so inference algorithms find instead *explanations* for the query [8,21] , i.e. composite choices such that the query is true in all the worlds that are compatible with them . For example, $\kappa_1 = \{(F_2, \{X/david\}, 1), (F_1, \{\}, 1)\}$ is an explanation for the query *epidemic* and so is $\kappa_2 = \{(F_2, \{X/robert\}, 1), (F_1, \{\}, 1)\}$.

Each explanation κ identifies a set of worlds, those that are compatible with it, and a set of explanations K identifies the set ω_K of worlds compatible with one of its explanations ($\omega_K = \{w_\sigma | \kappa \in K, \kappa \subseteq \sigma\}$). A set of explanations K is *covering* for a query Q if every world in which Q is true is in ω_K. For example, $K = \{\kappa_1, \kappa_2\}$ is covering for the query *epidemic*.

The probability of a query can thus be computed from a covering set of explanations for the query by computing the probability of the Boolean formula

$$B(Q) = \bigvee_{\kappa \in K} \bigwedge_{(F_i, \theta_j, 1) \in \kappa} X_{ij} \bigwedge_{(F_i, \theta_j, 0) \in \kappa} \neg X_{ij} \tag{2}$$

For Example 1, the formula is $B(epidemic) = X_{11} \wedge X_{21} \vee X_{11} \wedge X_{22}$.

Explanations however, differently from possible worlds, are not necessarily mutually exclusive with respect to each other, so the probability of the query can not be computed by a summation as in (1). In fact computing the probability of a DNF formula of independent Boolean random variables is a #P-complete problem [25]. The method that was found to be the most efficient up to now consists in building a Binary Decision Diagram for the formula and using a dynamic programming algorithm on the BDD [8,21]. A BDD is a rooted graph that has one level for each variable. Each node n has two children, a 0-child and a 1-child. The leaves store either 0 or 1. Given values for all the variables, a BDD can be used to compute the value of the formula by traversing the graph starting from the root, following the edges corresponding to the variables values and returning the value associated to the leaf that is reached. The BDD for Example 1 is shown in Figure 1.

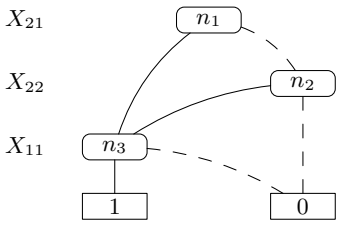


Fig. 1. BDD for Example 1.

A BDD performs a Shannon expansion of the Boolean formula $f(\mathbf{X})$, so that if X is the variable associated to the root level of a BDD, the formula $f(\mathbf{X})$ can be represented as $f(\mathbf{X}) = X \wedge f^X(\mathbf{X}) \vee \neg X \wedge f^{\neg X}(\mathbf{X})$ where $f^X(\mathbf{X})$

($f^{\neg X}(\mathbf{X})$) is the formula obtained by $f(\mathbf{X})$ by setting X to 1 (0). Now the two disjuncts are mutually exclusive and the probability of $f(\mathbf{X})$ can be computed as $P(f(\mathbf{X})) = P(X)P(f^X(\mathbf{X}))+(1-P(X))P(f^{\neg X}(\mathbf{X}))$ Figure 2 shows the function Prob that implements the dynamic programming algorithm for computing the probability of a formula encoded as a BDD.

Fig. 2. Function Prob: computation of the probability of a Boolean formula encoded as a BDD with root *node*.

```
1: function Prob(node)
2:     if node is a terminal then
3:         return value(node)                          ▷ value(node) is either 0 or 1
4:     else
5:         let X be v(node)                ▷ v(node) is the variable associated to node
6:         return Prob(child_0(node)) ·(1 − P(X))+Prob(child_1(node))·P(X)
7:     end if
8: end function
```

Languages with non-binary choices such as Logic Programs with Annotated Disjunctions can be handled by encoding the choices with binary variables [21].

3 The DISPONTE Semantics for Probabilistic Ontologies

DISPONTE assigns a semantics to probabilistic ontologies following the approach of the distribution semantics for probabilistic logic programs. It defines a probability distribution over non-probabilistic ontologies called worlds. This probability distribution is extended to a joint distribution of the worlds and a query and the probability of the query is obtained by marginalization.

The probabilistic ontologies we consider associate to each axiom of the ontology a Boolean random variable that indicates whether the axiom is present in a world. A *probabilistic ontology* is thus a set of annotated axioms of the form

$$p_i :: A_i \tag{3}$$

or of unannotated axioms of the form A_i, for $i = 1, \ldots, n$, where p_i is the probability with which axiom A_i is included in a world. Let us call O_A the set $\{A_1, \ldots, A_n\}$ and X_i the Boolean random variable associated to axiom A_i. Each X_i is independent of every X_j with $i \neq j$. The probability of each X_i of being true is p_i. If the $p_i ::$ annotation is omitted for an axiom, we assume that the axiom is certain, i.e., that it has probability 1.

A *world* w is obtained by sampling a value for X_i for every axiom A_i of O_A and by including A_i in w if $X_i = 1$. Since the random variables for the different axioms are independent, the probability $P(w)$ of w is obtained as:

$$P(w) = \prod_{A_i \in w} p_i \prod_{A_j \in O_A \setminus w} (1 - p_j)$$

Given a query Q to O, we can define its conditional probability of being true given a world $P(Q|w)$ in the following intuitive way: $P(Q|w) = 1$ if $w \models Q$ and $P(Q|w) = 0$ if $w \not\models Q$.

The probability $P(Q)$ can be obtained from the joint distribution of the query and the worlds by the sum rule:

$$P(Q) = \sum_w P(Q, w) = \sum_w P(Q|w)P(w) = \sum_{w:w\models Q} P(w)$$

Similarly to the case of probabilistic logic programming, the probability of a query Q given a probabilistic ontology O can be computed by first finding the explanations for Q in O. An explanation in this context is a subset of axioms of O that is sufficient for entailing Q. Typically minimal explanations are sought for efficiency reasons. All the explanations for Q must be found, corresponding to all ways of proving Q. Let E_Q be set of explanations and e be an explanation from E_Q. The probability of Q can be obtained by computing the probability of the DNF formula

$$F(Q) = \bigvee_{e\in E_Q} \bigwedge_{A_i\in e} p_i$$

Example 2. This example is inspired by Examples 3.1, 4.1, 4.2 and 4.3 of [15] that describe a probabilistic ontology about cars. We know for sure that a $SportCar$ is a Car to which a max_speed greater than 245Km/h is associated:

$$SportsCar \sqsubseteq Car \sqcap \exists max\_speed.\geq_{245Km/h} \tag{4}$$

We also know that a Car is a subset of the class of vehicles $HasFourWheels$ with probability 0.9:

$$0.9 :: Car \sqsubseteq HasFourWheels \tag{5}$$

Please note that this does not mean that a member of the class Car is a member of $HasFourWheels$ with probability 0.9, see Section 5. $johns\_car$ is an instance of $SportsCar$ with probability 0.8:

$$0.8 :: johns\_car : SportsCar \tag{6}$$

We want to know what is the probability $P(Q_1)$ of axiom $Q_1 = johnsCar : HasFourWheels$ being true. Q_1 has a single explanation containing the axioms (4), (5) and (6). Since (4) is certain, $P(Q_1)$ is $0.8 \times 0.9 = 0.72$.

Example 3. Let us consider another example, inspired by the `people+pets` ontology proposed in [18]. We know that $kevin$ is a $DogOwner$ with probability 0.6 and a $CatOwner$ with probability 0.6:

$$0.6 :: kevin : DogOwner; \tag{7}$$

$$0.6 :: kevin : CatOwner. \tag{8}$$

Moreover we know for sure that *DogOwner* and *CatOwner* are subclasses of *PetOwner*

$$DogOwner \sqsubseteq PetOwner \tag{9}$$
$$CatOwner \sqsubseteq PetOwner \tag{10}$$

Then the query axiom $Q_2 = kevin : PetOwner$ has two explanations, one composed of the axioms (7) and (9) and the other composed of the axioms (8) and (10). Since (9) is certain, the probability of the first explanation is 0.6. Similarly, the probability of the second explanation is again 0.6. If we associate the Boolean random variable X_1 to (7) and X_2 to (8), the query axiom is true if the formula $X_1 \vee X_2$ is true. Thus, $P(Q_2) = P(X_1 \vee X_2)$. Since X_1 and X_2 are independent, we get $P(Q_2) = 0.6 + 0.6 - 0.6 \times 0.6 = 0.84$. As you can see, the fact that *kevin* is an instance of both *DogOwner* and *CatOwner* increases the probability that he is an instance of *PetOwner*: if he were an instance of *DogOwner* only, its probability of being a *PetOwner* would be 0.6 and similarly if he were an instance of *CatOwner* only.

Now suppose that we known that *PetOwner* is a subclass of *Ecologist* with probability 0.7:

$$0.7 :: PetOwner \sqsubseteq Ecologist \tag{11}$$

The query axiom $Q_3 = kevin : Ecologist$ has again two explanations, one composed of axioms (7), (9) and (11) and the other composed of the axioms (8), (10) and (11). Since (9) is certain, the probability of the first explanation is $0.4 \times 0.6 = 0.24$. Similarly, the probability of the second explanation is $0.5 \times 0.6 = 0.3$. If we associate the Boolean random variable X_3 to (11), Q_3 is a consequence of the theory if $X_1 \wedge X_3 \vee X_2 \wedge X_3$ is true. A BDD that can be built for this formula is the one shown in Figure 1 after replacing variable X_{21} with X_1, variable X_{22} with X_2 and variable X_{11} with X_3.

The probability of node n_3 computed by PROB is $0.7 \times 1 + 0.3 \times 0 = 0.7$. The probability of node n_2 is $0.6 \times 0.7 + 0.4 \times 0 = 0.42$ and the probability of node n_1 (and of Q_3) is $0.6 \times 0.7 + 0.4 \times 0.42 = 0.588$.

4 The BUNDLE System

BUNDLE computes the probability of a query Q given a probabilistic ontology O that follows the DISPONTE semantics. BUNDLE exploits an underlying ontology reasoner that is able to return all explanations for a query. One of these system is Pellet [23] that is a complete OWL-DL reasoner. Pellet takes as input an OWL ontology in various formats, including the RDFXML language.

In order to assign probabilities to axioms, we exploit the possibility given by OWL1.1 of declaring an annotation property for axioms. We thus annotate the axioms with the XML tag `bundle:probability` whose value should be a real number in [0,1].

BUNDLE takes as input two RDFXML files, one containing the ontology and one containing the annotations. For Example 3, the ontology file contains the following definition of $PetOwner$:

```
<owl:Class rdf:about="#PetOwner">
    <rdfs:subClassOf>
        <owl:Class rdf:about="#Ecologist" />
    </rdfs:subClassOf>
</owl:Class>
```

The annotation file contains the annotation for the above axiom in the following form:

```
<owl11:Axiom>
    <rdf:subject rdf:resource="#PetOwner"/>
    <rdf:predicate rdf:resource="&rdfs;subClassOf"/>
    <rdf:object rdf:resource="#Ecologist"/>
    <bundle:probability>0.6</bundle:probability>
</owl11:Axiom>
```

BUNDLE first uses the annotation file for building a data structure $PMap$ that associates axioms with their probability. In order to do so, axioms are first converted to strings. We use the Manchester syntax to obtain a string representation of an axiom.

Then BUNDLE uses the EXPLAIN function of Pellet to compute explanations for a query axiom. BUNDLE thus accepts all the forms of query axioms that are accepted by Pellet's EXPLAIN function, namely subclass, instance, property value, theory inconsistency and class unsatisfiability.

Pellet returns the explanations for the query in the form of a set of sets of axioms. Then BUNDLE performs a double loop over the set of explanations and over the set of axioms in each explanation in which it builds a BDD representing the set of explanations. To manipulate BDDs we used the JavaBDD library[1] that provides a Java interface to the major BDD libraries such as CUDD[2].

Outside the outer loop, two data structures are initialized: $VarAxAnn$ is an array that maintains the association between Boolean random variables (whose index is the array index) and axioms together with their probability, and BDD represents the set of explanations. BDD is initialized to the BDD representing the zero Boolean function. Then the outer loop is entered in which $BDDE$ is initialized to the BDD representing the one Boolean function. In the inner loop the axioms of an explanation are considered one by one. Each axiom is first looked up in $PMap$ to get its probability. If NULL is returned this means that this is a certain axiom and it does not need to be considered anymore. Then the axiom is searched for in $VarAxAnn$ to see if it has already been assigned a random variable. If not, a cell is added to $VarAxAnn$ to store the axiom with its probability. At this point we know the axiom's position i in $VarAxAnn$

[1] http://javabdd.sourceforge.net/
[2] http://vlsi.colorado.edu/~fabio/CUDD/

and so the index of its Boolean variable X_i. We obtain a BDD representing $X_i = 1$ and we conjoin it with $BDDE$. At the end of the inner loop the BDD for the current explanation, $BDDE$, is disjoined with BDD. After the two cycles, function PROB of Figure 2 is called over BDD and its result is returned to the user.

BUNDLE has been implemented in Java and will be available for download from `http://sites.unife.it/bundle`. It has been successfully tested on various examples, including those of Section 3.

5 Discussion

The probabilistic knowledge that can be expressed with the DISPONTE semantics is epistemic by nature, namely it represents degrees of belief in the axioms rather that statistical information. While this is reasonable for many axioms, for subclass and subproperty axioms one may want to express statistical information, for example with a probabilistic subclass axiom $p :: A \sqsubseteq B$ one may want to express the fact that a random individual of A has probability p of belonging to B. The DISPONTE semantics, instead, interpret the axioms as stating that $A \sqsubseteq B$ is true with probability p. The difference is that, if two individuals i and j belong to class A, the probability that they both belong to B in the DISPONTE semantics is p while with a statistical interpretation is $p \times p$. Thus statistical information can be used to define a degree of partial overlap between classes. Extending DISPONTE to take account of this case is possible, it requires to define a probability distribution over models rather than over theories.

However, to reason with such knowledge, the inference engine must be modified in its inference procedure and cannot be used as a black box as in BUNDLE. In fact, BUNDLE assigns a single Boolean random variable to the axiom $A \sqsubseteq B$, while with a statistical interpretation a different Boolean random variable must be assigned to each assertion that an individual of class A belongs to class B. We leave this extension for future work.

Another limitation of BUNDLE is the use of the OWL 1.1 Axiom construct to specify probabilities. This seems to restrict the kind of axioms on which probabilities can be placed, since the object of the RDF triple does not allow complex class expressions. However this limitation can be overcome by defining a new class which is equivalent to the complex class expression and using the new class name in the RDF triple. In the future we plan to investigate the possibility of annotating the axioms directly in the ontology file.

As regards the complexity of reasoning on DISPONTE, it is equal to the complexity of the underlying description logic plus the #P complexity of computing the probability of a DNF formula of independent Boolean random variables, assuming the cost of keeping track of explanations during inference is negligible. Thus, the problem of inference in DISPONTE remains decidable if it was so in the underlying description logic.

6 Related Work

Our work differs from previous work in many respects. [6] proposed an extension of the description logic $\mathcal{ALC}$ that is able to express statistical information on the terminological knowledge such as partial concept overlapping. Similarly, [11] presents a probabilistic description logic based on Bayesian networks that deals with statistical terminological knowledge. As illustrated in Section 5, currently we are not able to express statistical terminological knowledge but it is possible to extend the semantics to do so. Differently from us, [6,11] do not allow probabilistic assertional knowledge about concept and role instances. [7] allows assertional knowledge about concept and role instances together with statistical terminological knowledge and combines the resulting probability distributions using cross-entropy minimization. In the future we plan to compare the DISPONTE semantics extended with statistical information with this approach.

[4] proposed a probabilistic extension of OWL that admits a translation into Bayesian networks. The semantics that is proposed assigns a probability distribution $P(i)$ over individuals, i.e. $\sum_i P(i) = 1$, and assigns a probability to a class C as $P(C) = \sum_{i \in C} P(i)$, while we assign a probability distribution over theories. PR-OWL [2,1] is an upper ontology that provides a framework for building probabilistic ontologies. It allows to use the first-order probabilistic logic MEBN [12] for representing uncertainty in ontologies. The use of a full fledged first-order probabilistic logic distringuishes this work from ours, where we tried to provide a minimal extension to description logics.

A different approach to the combination of description logic with probability is taken by [5,13,14] where the authors use probabilistic lexicographic entailment from probabilistic default reasoning. The logics proposed in these papers allow both terminological probabilistic knowledge as well as assertional probabilistic knowledge about instances of concepts and roles. PRONTO [9] is one of the systems that allows to perform inference in this semantics.

Similary to [7], the terminological knowledge is interpreted statistically while the assertional knowledge is interpreted epistemically by assigning degrees of beliefs to assertions, thus differing from our current treatment of terminological knowledge. Moreover it also allows to express default knowledge about concepts that can be overridden in subconcepts and whose semantics is given by Lehmann's lexicographic default entailment.

These works are based on Nilsson's probabilistic logic [16] where a probabilistic interpretation Pr defines a probability distribution over the set of interpretations $\mathcal{I}$. The probability of a logic formula ϕ according to Pr, denoted $Pr(\phi)$, is the sum of all $Pr(I)$ such that $I \in \mathcal{I}$ and $I \models \phi$.

A probabilistic knowledge base K is a set of probabilistic formulas of the form $\phi \geq p$. A probabilistic interpretation Pr satisfies $\phi \geq p$ iff $Pr(\phi) \geq p$. Pr satisfies K, or Pr is a model of K, iff Pr satisfies all $F \in K$. We say $\phi \geq p$ is a tight logical consequence of K iff p is the infimum of $Pr(\phi)$ subject to all models Pr of K. Computing tight logical consequences from probabilistic knowledge bases can be done by solving a linear optimization problem.

Nilsson's probabilistic logic differs from the distribution semantics: while a probabilistic knowledge base in Nilsson's logic may have multiple models that are probabilistic interpretations, a probabilistic program under the distribution semantics has a single model that defines a single distribution over interpretations. Also, while in Nilsson's logic we want to compute the lowest p such that $Pr(\phi) \geq p$ holds for all Pr, in the distribution semantics we want to compute p such that $P(\phi) = p$. Nilsson's logic complexity is lower than the #P complexity of the distribution semantics.

In fact Nilsson's logic allows weaker conclusions than the distribution semantics. For example, consider a probabilistic program composed of $0.4 :: a.$ and $0.5 :: b.$ and a probabilistic knowledge base composed of $a \geq 0.4$ and $b \geq 0.5$. The distribution semantics allows to say that $P(a \vee b) = 0.7$, while with Nilsson's logic the lowest p such that $Pr(a \vee b) \geq p$ holds is 0.5. This is due to the fact that in the distribution semantics the probabilistic atoms are considered independent, which allows to make stronger conclusions. However, note that this does not restrict expressiveness as you can specify with the distribution semantics any joint probability distribution over the atoms of the Herbrand base interpreted as Boolean random variables, possibly introducing new random facts if needed.

Alternative approaches to modeling imperfect and incomplete knowledge in ontologies are based on fuzzy logic. A good survey of these approaches is presented in [15].

7 Conclusions

We have presented the semantics DISPONTE for probabilistic ontologies that is inspired by the distribution semantics of probabilistic logic programming. We have also presented the system BUNDLE that is able to compute the probability of queries from an uncertain OWL DL ontology.

In the future, we plan to extend DISPONTE to take into account statistical terminological knowledge and improve the way in which the input to BUNDLE is specified.

References

1. Carvalho, R.N., Laskey, K.B., Costa, P.C.: PR-OWL 2.0 - bridging the gap to OWL semantics. In: International Workshops on Uncertainty Reasoning for the Semantic Web (2010)
2. Costa, P.C.G., Laskey, K.B., Laskey, K.J.: Pr-owl: A bayesian ontology language for the semantic web. In: International Workshops on Uncertainty Reasoning for the Semantic Web. vol. 5327, pp. 88–107. Springer (2008)
3. De Raedt, L., Kimmig, A., Toivonen, H.: ProbLog: A probabilistic Prolog and its application in link discovery. In: International Joint Conference on Artificial Intelligence. pp. 2462–2467 (2007)
4. Ding, Z., Peng, Y.: A probabilistic extension to ontology language OWL. In: Hawaii International Conference On System Sciences. IEEE (2004)

5. Giugno, R., Lukasiewicz, T.: P-SHOQ(D): A probabilistic extension of SHOQ(D) for probabilistic ontologies in the semantic web. In: European Conference on Logics in Artificial Intelligence. LNCS, vol. 2424, pp. 86–97. Springer (2002)
6. Heinsohn, J.: Probabilistic description logics. In: Conference on Uncertainty in Artificial Intelligence. pp. 311–318. Morgan Kaufmann (1994)
7. Jaeger, M.: Probabilistic reasoning in terminological logics. In: International Conference on Principles of Knowledge Representation and Reasoning. pp. 305–316 (1994)
8. Kimmig, A., Demoen, B., Raedt, L.D., Costa, V.S., Rocha, R.: On the implementation of the probabilistic logic programming language problog. Theory Pract. Log. Program. 11(2-3), 235–262 (2011)
9. Klinov, P.: Pronto: A non-monotonic probabilistic description logic reasoner. In: European Semantic Web Conference. LNCS, vol. 5021, pp. 822–826. Springer (2008)
10. Koller, D.: Probabilistic relational models. In: International Workshop on Inductive Logic Programming. LNCS, vol. 1634, pp. 3–13. Springer (1999)
11. Koller, D., Levy, A.Y., Pfeffer, A.: P-classic: A tractable probablistic description logic. In: National Conference on Artificial Intelligence. pp. 390–397 (1997)
12. Laskey, K.B., da Costa, P.C.G.: Of starships and klingons: Bayesian logic for the 23rd century. In: Conference in Uncertainty in Artificial Intelligence. pp. 346–353. AUAI Press (2005)
13. Lukasiewicz, T.: Probabilistic default reasoning with conditional constraints. Ann. Math. Artif. Intell. 34(1-3), 35–88 (2002)
14. Lukasiewicz, T.: Expressive probabilistic description logics. Artif. Intell. 172(6-7), 852–883 (2008)
15. Lukasiewicz, T., Straccia, U.: Managing uncertainty and vagueness in description logics for the semantic web. J. Web Sem. 6(4), 291–308 (2008)
16. Nilsson, N.J.: Probabilistic logic. Artif. Intell. 28(1), 71–87 (1986)
17. Obrst, L., McCandless, D., Stoutenburg, S., Fox, K., Nichols, D., Prausa, M., Sward, R.: Evolving use of distributed semantics to achieve net-centricity. In: AAAI Fall Symposium (2007)
18. Patel-Schneider, P.F., Horrocks, I., Bechhofer, S.: Tutorial on OWL (2003), `http://www.cs.man.ac.uk/~horrocks/ISWC2003/Tutorial/`
19. Poole, D.: Abducing through negation as failure: stable models within the independent choice logic. J. of Log. Program. 44(1-3), 5–35 (2000)
20. Richardson, M., Domingos, P.: Markov logic networks. Machine Learning 62(1-2), 107–136 (2006)
21. Riguzzi, F.: Extended semantics and inference for the Independent Choice Logic. Log. J. IGPL 17(6), 589–629 (2009)
22. Sato, T.: A statistical learning method for logic programs with distribution semantics. In: International Conference on Logic Programming. pp. 715–729. MIT Press (1995)
23. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. J. Web Sem. 5(2), 51–53 (2007)
24. URW3-XG: Uncertainty reasoning for the World Wide Web, final report, `http://www.w3.org/2005/Incubator/urw3/XGR-urw3/`
25. Valiant, L.G.: The complexity of enumeration and reliability problems. SIAM J. Comp. 8(3), 410–421 (1979)
26. Vennekens, J., Verbaeten, S., Bruynooghe, M.: Logic programs with annotated disjunctions. In: International Conference on Logic Programming. LNCS, vol. 3131, pp. 195–209. Springer (2004)

Semantic Link Prediction through Probabilistic Description Logics

Kate Revoredo[1], José Eduardo Ochoa Luna[2], and Fabio Gagliardi Cozman[2]

[1] Departamento de Informática Aplicada, Unirio
Av. Pasteur, 458, Rio de Janeiro, RJ, Brazil
[2] Escola Politécnica, Universidade de São Paulo,
Av. Prof. Mello Morais 2231, São Paulo - SP, Brazil
katerevoredo@uniriotec.br,eduardo.ol@gmail.com,fgcozman@usp.br

Abstract. Predicting potential links between nodes in a network is a problem of great practical interest. Link prediction is mostly based on graph-based features and, recently, on approaches that consider the semantics of the domain. However, there is uncertainty in these predictions; by modeling it, one can improve prediction results. In this paper, we propose an algorithm for link prediction that uses a probabilistic ontology described through the probabilistic description logic crALC. We use an academic domain in order to evaluate this proposal.

1 Introduction

Many social, biological, and information systems can be well described by networks, where nodes represent objects (individuals), and links denote the relations or interactions between nodes. Predicting a possible link in a network is an interesting issue that has recently gained attention, due to the growing interest in social networks. For instance, one may be interested on finding potential friendship between two persons in a social network, or a potential collaboration between two researchers. Thus link prediction [12, 20] aims at predicting whether two nodes (i.e. people) should be connected given that we know previous information about their relationships or interests. A common approach is to exploit the network structure, where numerical information about nodes is analyzed [12, 20, 9]. However, knowledge about the objects represented in the nodes can improve prediction results. For instance consider that the researchers *Joe* and *Mike* do not have a publication in common, thus they do not share a link in a collaboration network. Moreover, graph features do not indicate a potential link between them. However, they have published in the same journal and they both teach the same course in their respectively universities. This information can be an indication of a potential collaboration between them. Given this, approaches that are based on the semantics related to the domain of the objects represented by the nodes [21, 18] have been proposed. In some of them, an ontology modeling the domain and the object interests were used in the prediction task.

However, there is uncertainty in such predictions. Often, it is not possible to guarantee the relationship between two objects (nodes). This is maybe due

to the fact that information about the domain is incomplete. Thus, it would be interesting if link prediction approaches could handle the *probability* of a link conditioned on the information about the domain. In our example, knowing that the probability of the relationship between Joe and $Mike$ conditioned on the knowledge of them publishing in the same journal and teaching the same course is high implies a link between them in the network; otherwise, a link is not suggested. In graph-based approaches, probabilistic models learned through machine learning algorithms were used for link prediction. Some examples of probabilistic models are Probabilistic Relational Model (PRM) [6], Probabilistic Entity Relationship Model (PERM) [7] and Stochastic Relational Model (SRM) [22]. On approaches based on semantic we claim that ontologies must be used to model the domain. Therefore, to model uncertainty, probabilistic approaches, such as probabilistic ontologies, must be considered.

An ontology can be represented through a description logic [2], which is typically a decidable fragment of first-order logic that tries to reach a practical balance between expressivity and complexity. To encode uncertainty, a probabilistic description logic (PDL) must be contemplated. The literature contains a number of proposals for PDLs [8, 10, 19, 13]. In this paper we adopt a recently proposed PDL, called Credal $\mathcal{ALC}$ (CR$\mathcal{ALC}$) [4, 16, 5], that extends the popular logic $\mathcal{ALC}$ [2]. In CR$\mathcal{ALC}$ one can specify sentences such as $P(\mathsf{Professor}|\mathsf{Researcher}) = 0.4$, indicating the probability that an element of the domain is a Professor given that it is a Researcher. These sentences are called *probabilistic inclusions*. Exact and approximate inference algorithms that deal with probabilistic inclusions have been proposed [4, 5], using ideas inherited from the theory of Relational Bayesian Networks (RBN)[11].

In this paper, we propose to use a probabilistic ontology defined with the PDL CR$\mathcal{ALC}$ for semantic link prediction.

The paper is organized as follows. Section 2 reviews basic concepts of PDLs and CR$\mathcal{ALC}$. Section 3 presents our algorithm for semantic link prediction through the PDL CR$\mathcal{ALC}$. Experiments are discussed in Section 4, and Section 5 concludes the paper.

2 Probabilistic Description Logics and CR$\mathcal{ALC}$

Description logics (DLs) form a family of representation languages that are typically decidable fragments of first order logic (FOL) [2]. Knowledge is expressed in terms of *individuals*, *concepts*, and *roles*. The semantic of a description is given by a *domain* $\mathcal{D}$ (a set) and an *interpretation* $\cdot^{\mathcal{I}}$ (a functor). Individuals represent objects through names from a set $N_{\mathsf{I}} = \{a, b, \ldots\}$. Each *concept* in the set $N_{\mathsf{C}} = \{C, D, \ldots\}$ is interpreted as a subset of a domain $\mathcal{D}$. Each *role* in the set $N_{\mathsf{R}} = \{r, s, \ldots\}$ is interpreted as a binary relation on the domain.

Several probabilistic descriptions logics (PDLs) have appeared in the literature. Heinsohn [8], Jaeger [10] and Sebastiani [19] consider probabilistic inclusion axioms such as $P_{\mathcal{D}}(\mathsf{Professor}) = \alpha$, meaning that a randomly selected object is a Professor with probability α. This characterizes a *domain-based* semantics: prob-

abilities are assigned to subsets of the domain $\mathcal{D}$. Sebastiani also allows inclusions such as $P(\mathsf{Professor}(\mathsf{John})) = \alpha$, specifying probabilities over the interpretations themselves. For example, one interprets $P(\mathsf{Professor}(\mathsf{John})) = 0.001$ as assigning 0.001 to be the probability of the set of interpretations where John is a Professor. This characterizes an *interpretation-based* semantics.

The PDL CR$\mathcal{ALC}$ is a probabilistic extension of the DL $\mathcal{ALC}$ that adopts an interpretation-based semantics. It keeps all constructors of $\mathcal{ALC}$, but only allows concept names on the left hand side of inclusions/definitions. Additionally, in CR$\mathcal{ALC}$ one can have probabilistic inclusions such as $P(C|D) = \alpha$ or $P(r) = \beta$ for concepts C and D, and for role r. If the interpretation of D is the whole domain, then we simply write $P(C) = \alpha$. The semantics of these inclusions is roughly (a formal definition can be found in [5]) given by:

$$\forall x \in \mathcal{D} \ : \ P(C(x)|D(x)) = \alpha,$$

$$\forall x \in \mathcal{D}, y \in \mathcal{D} \ : \ P(r(x,y)) = \beta.$$

We assume that every terminology is acyclic; no concept uses itself. This assumption allows one to represent any terminology $\mathcal{T}$ through a directed acyclic graph. Such a graph, denoted by $\mathcal{G}(\mathcal{T})$, has each concept name and role name as a node, and if a concept C directly uses concept D, that is if C and D appear respectively in the left and right hand sides of an inclusion/definition, then D is a *parent* of C in $\mathcal{G}(\mathcal{T})$. Each existential restriction $\exists r.C$ and value restriction $\forall r.C$ is added to the graph $\mathcal{G}(\mathcal{T})$ as nodes, with an edge from r and C to each restriction directly using it. Each restriction node is a *deterministic* node in that its value is completely determined by its parents.

Example 1. Consider a terminology $\mathcal{T}_1$ with concepts $\mathsf{A}, \mathsf{B}, \mathsf{C}, \mathsf{D}$. Suppose $P(\mathsf{A}) = 0.9, \mathsf{B} \sqsubseteq \mathsf{A}, \mathsf{C} \sqsubseteq \mathsf{B} \sqcup \exists r.\mathsf{D}, P(\mathsf{B}|\mathsf{A}) = 0.45, P(\mathsf{C}|\mathsf{B} \sqcup \exists r.\mathsf{D}) = 0.5$, and $P(\mathsf{D}|\forall r.\mathsf{A}) = 0.6$. The last three assessments specify beliefs about partial overlap among concepts. Suppose also $P(\mathsf{D}|\neg\forall r.\mathsf{A}) = \epsilon \approx 0$ (conveying the existence of exceptions to the inclusion of D in $\forall r.\mathsf{A}$). Figure 1 depicts $\mathcal{G}(\mathcal{T})$.

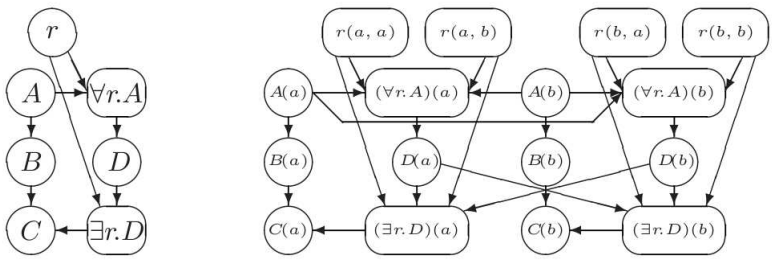


Fig. 1. $\mathcal{G}(\mathcal{T})$ for terminology $\mathcal{T}$ in Example 1 and its grounding for domain $\mathcal{D} = \{a, b\}$.

The semantics of CR$\mathcal{ALC}$ is based on probability measures over the space of interpretations, for a fixed domain. Inferences, such as $P(\mathsf{A}_o(\mathsf{a}_0)|\mathcal{A})$ for an ABox $\mathcal{A}$, can be computed by propositionalization and probabilistic inference (for exact calculations) or by a first order loopy propagation algorithm (for approximate calculations) [5].

3 Link Prediction by using CR$\mathcal{ALC}$

In this section we describe how to apply the PDL CR$\mathcal{ALC}$ for semantic link prediction. We borrowed some syntax from the graph-based approach where each node (a person in a social network) is represented by A, B, C, and we are interested in defining whether a link between A and B is suitable given there is no link between these nodes. Interests between the nodes are modeled through a probabilistic ontology represented by the PDL CR$\mathcal{ALC}$. The prediction link task can be described as:

Given:
- a network defining relationship between objects;
- an ontology represented by CR$\mathcal{ALC}$ describing the domain of the objects;
- the ontology role that defines the semantic of the relationship between objects;
- the ontology concept that describes the network objects.

Find:
- a revised network defining relationship between objects.

The proposed algorithm for link prediction receives a network of a specific domain. For instance, in a collaboration network the nodes represent researchers and the relationship can have the semantic "has a publication with" or "is advised by". Therefore, the ontology represented by CR$\mathcal{ALC}$ describes the domain of publications between researchers, having concepts like Researcher, Publication, StrongRelatedResearcher and NearCollaborator and roles like hasPublication, hasSameInstitution and sharePublication. This ontology can be learned automatically through a learning algorithm as the ones proposed in [15, 17]. Thus, the nodes represent instances of one of the concepts described in the PDL CR$\mathcal{ALC}$ and the semantic of the links is described by one of the roles in the PDL CR$\mathcal{ALC}$. These concept and role must be informed as inputs to the proposed algorithm. The link prediction algorithm is described in Algorithm 1.

The algorithm starts looking for all pairs of instances of the concept C defined as the concept that provides the semantic for the network nodes. For each pair it checks whether the corresponding nodes exist in the network (this can be improved by exploring graph-based properties). If not the probability of the link is calculated through the probability of the defined role conditioned on evidence. The evidence is provided by the instances of the ontology. As many instances the ontology have the better is the inference performed. The inference is performed through the Relational Bayesian network build from ontology O. If the probability inferred is greater than a threshold then the corresponding link

Require: a network N, an ontology O, the role $r(\_,\_)$ representing the semantic of the network link, the concept C describing the objects of the network and a *threshold*.
Ensure: a revised network N_f
1: define N_f as N
2: **for all** pair of instances (a, b) of concept C **do**
3: **if** does not exist a link between nodes a and b in the network N **then**
4: infer probability $P(r(a, b)|evidences)$ using the RBN created through the ontology O
5: **if** $P(r(a, b)|evidences) > threshold$ **then**
6: add a link between a and b in network N_f
7: **end if**
8: **end if**
9: **end for**

Algorithm 1: Algorithm for link prediction through CR$\mathcal{ALC}$.

is added to the network. Alternatively, when the threshold to be considered is not known a priori, a rank of the inferred links based on their probability is done and the top-k, where k would be a parameter, are chosen.

4 Preliminary Results

Experiments were run over a collaborative network of researchers. Data was gathered from the Lattes curriculum platform [3], the public repository for Brazilian curriculum researchers. In this platform, every researcher has a unique Lattes code that allows one to link to other researchers according to: shared publications, advising tasks, and examination board participations. Given this collaborative network we are interested in predicting further links among researchers in order to either promote further collaborations (suitable co-workers to research tasks would be suggested) or gather information about research groups. Due to form-filling errors there are many missing links among researchers; thus, we are unable to completely state co-working relationships using only the Lattes platform.

To tackle link prediction we firstly have collected information about 1200 researchers and learned a probabilistic ontology [15, 17], represented by the PDL CR$\mathcal{ALC}$, for modeling their research interests. A simplified probabilistic ontology

[3] http://lattes.cnpq.br/

is given by:

$P(\mathsf{Publication}) = 0.3$
$P(\mathsf{Board}) = 0.33$
$P(\mathsf{sharePublication}) = 0.22$
$P(\mathsf{wasAdvised}) = 0.05$
$P(\mathsf{hasSameInstitution}) = 0.14$
$P(\mathsf{sameExaminationBoard}) = 0.31$

$\mathsf{ResearcherLattes} \equiv \mathsf{Person} \sqcap (\exists \mathsf{hasPublication.Publication} \sqcap \exists \mathsf{advises.Person} \sqcap \exists \mathsf{participate.Board})$

$P(\mathsf{PublicationCollaborator} \mid \mathsf{Researcher} \sqcap \exists \mathsf{sharePublication.Researcher}) = 0.91$

$P(\mathsf{SupervisionCollaborator} \mid \mathsf{Researcher} \sqcap \exists \mathsf{wasAdvised.Researcher}) = 0.94$

$P(\mathsf{SameInstitution} \mid \mathsf{Researcher} \sqcap \exists \mathsf{hasSameInstitution.Researcher}) = 0.92$

$P(\mathsf{SameBoard} \mid \mathsf{Researcher} \sqcap \exists \mathsf{sameExaminationBoard.Researcher}) = 0.92$

$P(\mathsf{NearCollaborator} \mid \mathsf{Researcher} \sqcap \exists \mathsf{sharePublication}.\exists \mathsf{hasSameInstitution}.\exists \mathsf{sharePublication.Researcher}) = 0.95$

$\mathsf{FacultyNearCollaborator} \equiv \mathsf{NearCollaborator} \sqcap \exists \mathsf{sameExaminationBoard.Researcher}$

$P(\mathsf{NullMobilityResearcher} \mid \mathsf{Researcher} \sqcap \exists \mathsf{wasAdvised}.\exists \mathsf{hasSameInstitution.Researcher}) = 0.98$

$\mathsf{StrongRelatedResearcher} \equiv \mathsf{Researcher} \sqcap (\exists \mathsf{sharePublication.Researcher} \sqcap \exists \mathsf{wasAdvised.Researcher})$

$\mathsf{InheritedResearcher} \equiv \mathsf{Researcher} \sqcap (\exists \mathsf{sameExaminationBoard.Researcher} \sqcap \exists \mathsf{wasAdvised.Researcher})$

In this probabilistic ontology concepts and probabilistic inclusions denote mutual research interests. For instance, a PublicationCollaborator inclusion refers to Researchers who shares a Publication, thus relates two nodes (Researcher) in a collaboration graph. Therefore, the concept Researcher and the role sharePublication are inputs to the algorithm we proposed in Algorithm 1.

To perform inferences and therefore to obtain link predictions, a propositionalization step (a resulting relational Bayesian network) is required.

In addition, a collaboration graph, based on shared publications, was also defined. Statistical information was computed accordingly. Figure 2 depicts collaborations among 303 researchers. Several relationships and clusterings can also be observed.

If we carefully inspect this collaboration graph (Figure 3 shows a subgraph obtained from Figure 2) we could be interested, for instance, in predicting links among researchers from different groups.

Thus, in Figure 3 one could further investigate whether a link between researcher R (red octagon node) and the researcher B (blue polygon node) is suitable. In order to infer this, the probability of a possible link between R and

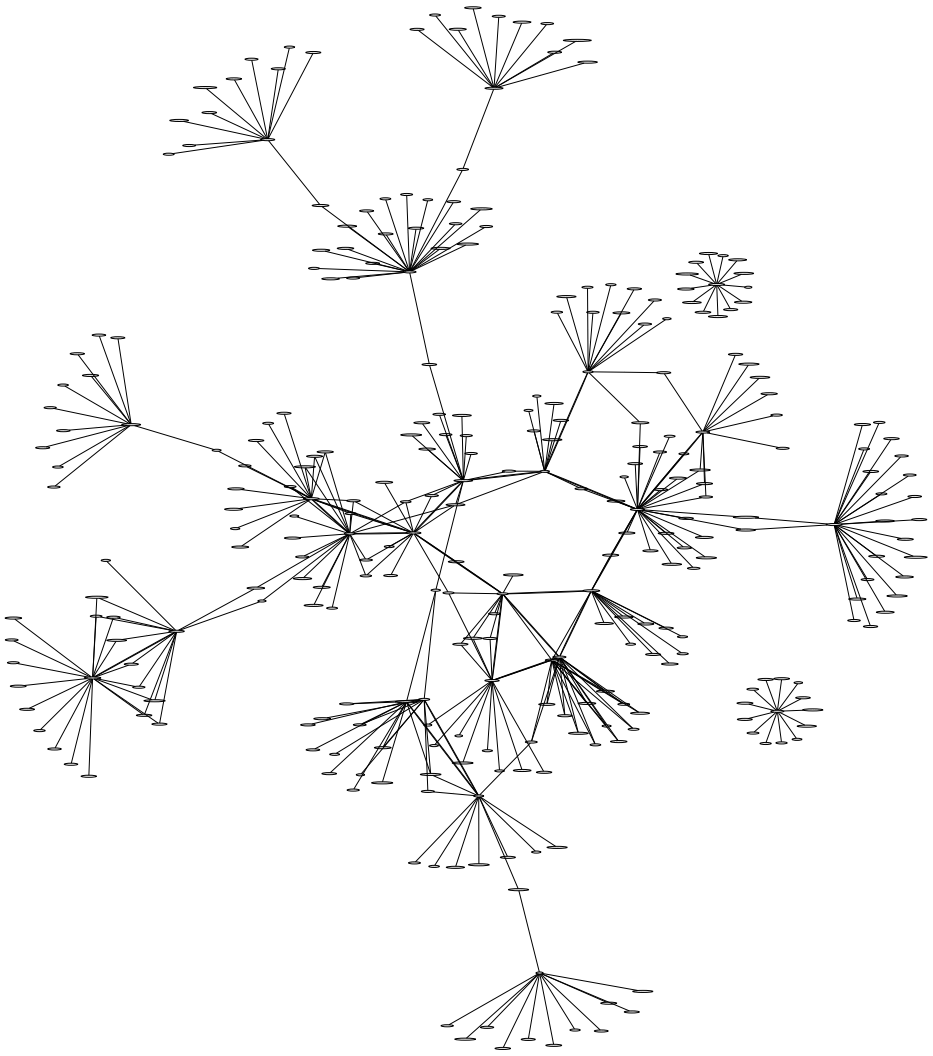


Fig. 2. Collaboration graph among researchers.

B is calculated, $P(link(R, B)|E)$, where E denotes evidence about researchers such as publications, institution, examination board participations and so on. The role sharePublication is the one defining the semantic of the links in the graph. Therefore, it is through it that we must calculate $P(link(R, B)|E)$. Since the concept PublicationCollaborator is defined by the role sharePublication and considering as evidence $\mathsf{Researcher}(R) \sqcap \exists \mathsf{hasSameInstitution}.\mathsf{Researcher}(B)$ one can infer $P(link(R, B)|E)$ through:

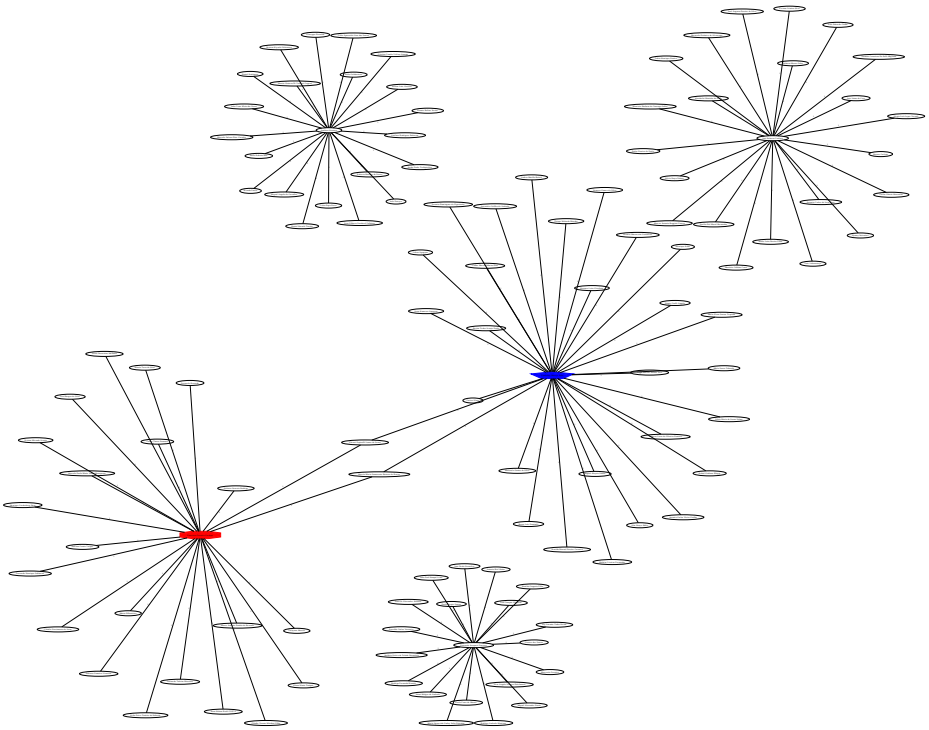


Fig. 3. Collaboration subgraph.

$$P(\mathsf{PublicationCollaborator(R)} \mid \mathsf{Researcher(R)} \sqcap \exists \mathsf{hasSameInstitution.Researcher}(B)) = 0.57.$$

If we took a threshold of 0.60, the link between R and B would not be included.

One could gain more evidence, such as information about nodes that indirectly connect these two groups (Figure 3), denoted by I_1, I_2. The inference would be

$$P(\mathsf{PublicationCollaborator(R)} \mid \mathsf{Researcher(R)} \sqcap \exists \mathsf{sharePublication}(I_1).\exists \mathsf{sharePublication(B)} \sqcap \exists \mathsf{sharePublication}(I_2).\exists \mathsf{sharePublication(B)}) = 0.65.$$

Because more information was provided the probability inferred was different. The same threshold now would preserve the link.

Other inferences are possible by considering the suggestion of links between surrounding nodes, i.e. nodes directly linked to the two nodes R and B , denoted by $R_1, \ldots, R_k$, and $B_1, \ldots, B_n$ respectively. For each $i = 1, ..., k$ and $j = 1, ..., n$, calculates $P(link(R_i, R_j)|E)$ and $P(link(B_i, B_j)|E)$.

As a rule, if we are interested in discovering whether A and B could be linked, probabilistic inference $P(link(A, B))$ should be performed.

In a more general framework, graph information could be useful to deal with a large number of link predictions. Note that graph adjacency allow us to address probabilistic inference for promising nodes. In a naive approach, each pair of nodes in the collaboration graph would be evaluated so, multiple probabilistic relational Bayesian inference calls would be required.

On the other hand, if graph-based information is used, such naive scheme could be improved. In our approach, two nodes are probabilistically evaluated if there is a path between them (number of incoming/outgoing edges, number of mutual friends, node distances are also considered). Thus, numerical graph-based information guides the inference process in the relational Bayesian network (linked to the probabilistic ontology). In addition, other candidates sharing any kind of evidence are also evaluated, i.e., interests based features (linked to ontological knowledge) allow us to further explore link prediction.

Alternatively, by completing an overall link predicting task we can devise further functionalities to the resulting collaboration network. The resulting graph can be considered as being a probabilistic network, i.e., probabilities inferred for each link could be denote strenght of the relationship.

5 Conclusion

We have presented an approach for predicting links that resorts both to graph-based and ontological information. Given a collaborative network, e.g., a social network, we encode interests and graph features through a CR-$\mathcal{ALC}$ probabilistic ontology. In order to predict links we resort to probabilistic inference. Preliminary results focused on an academic domain, and we aimed at predicting links among researchers. These preliminary results showed the potential of the idea.

Previous combined approaches for link prediction [3, 1] have focused on machine learning algorithms [14]. In such schemes, numerical graph-based features and ontology-based features are computed; then both features are input into a machine learning setting where prediction is performed. Differently from such approaches, in our work we adopt a generic ontology (instead of a hierarchical ontology, expressing only is-a relationships among interests). Therefore, our approach uses more information about the domain to help the prediction.

Acknowledgements

The third author is partially supported by CNPq. The work reported here has received substantial support by FAPESP grant 2008/03995-5.

References

1. W. Aljandal, V. Bahirwani, D. Caragea, and H.W. Hsu. Ontology-aware classification and association rule mining for interest and link prediction in social networks.

In *AAAI 2009 Spring Symposium on Social Semantic Web: Where Web 2.0 Meets Web 3.0*, Standford,CA, 2009.

2. F. Baader and W. Nutt. Basic description logics. In *Description Logic Handbook*, pages 47–100. Cambridge University Press, 2002.
3. D. Caragea, V. Bahirwani, W. Aljandal, and W. H. Hsu. Ontology-based link prediction in the livejournal social network. In *Symposium on Abstraction, Reformulation and Approximation*, 2009.
4. F. G. Cozman and R. B. Polastro. Loopy propagation in a probabilistic description logic. In Sergio Greco and Thomas Lukasiewicz, editors, *Second International Conference on Scalable Uncertainty Management*, Lecture Notes in Artificial Intelligence (LNAI 5291), pages 120–133. Springer, 2008.
5. F. G. Cozman and R. B. Polastro. Complexity analysis and variational inference for interpretation-based probabilistic description logics. In *Conference on Uncertainty in Artificial Intelligence*, 2009.
6. N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *Proc. 16th Int. Joint Conference on Artificial Intelligence*, pages 1300–1309, 1999.
7. D. Heckerman, C. Meek, and D. Koller. Probabilistic entity-relationship models, prms, and plate models. In *In Proceedings of the 21st International Conference on Machine Learning*, 2004.
8. J. Heinsohn. Probabilistic description logics. In *International Conf. on Uncertainty in Artificial Intelligence*, pages 311–318, 1994.
9. W.H. Hsu, A.L. King, M.S.R. Paradesi, T. Pydimarri, and T. Wneinger. Collaborative and structural recommendation of friends using weblog-based social network analysis. In *Proceedings of Computational Approaches to Analysing WebLogs (AAAI)*, 2006.
10. M. Jaeger. Probabilistic reasoning in terminological logics. In *Principals of Knowledge Representation (KR)*, pages 461–472, 1994.
11. M. Jaeger. Relational Bayesian networks: a survey. *Linkoping Electronic Articles in Computer and Information Science*, 6, 2002.
12. D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proceedings of the twelfth international conference on Information and Knowledge Management*, pages 556–559, New York, NY, USA, 2003. ACM.
13. T. Lukasiewicz and U. Straccia. Managing uncertainty and vagueness in description logics for the semantic web. *Web Semant.*, 6:291–308, November 2008.
14. T. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.
15. J. Ochoa-Luna, K. Revoredo, and F.G. Cozman. Learning sentences and assessments in probabilistic description logics. In *Bobillo, F., et al. (eds.) Proceedings of the 6th International Workshop on Uncertainty Reasoning for the Semantic Web*, volume 654, pages 85–96, Shangai, China, 2010. CEUR-WS.org.
16. R. B. Polastro and F. G. Cozman. Inference in probabilistic ontologies with attributive concept descriptions and nominals. In *4th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW) at the 7th International Semantic Web Conference (ISWC)*, Karlsruhe, Germany, 2008.
17. K. Revoredo, J. Ochoa-Luna, and F. Cozman. Learning terminologies in probabilistic description logics. In Antônio da Rocha Costa, Rosa Vicari, and Flavio Tonidandel, editors, *Advances in Artificial Intelligence SBIA 2010*, volume 6404 of *Lecture Notes in Computer Science*, pages 41–50. Springer / Heidelberg, Berlin, 2010.

18. M. Sachan and R. Ichise. Using semantic information to improve link prediction results in network datasets. *International Journal of Computer Theory and Engeneering*, 3:71–76, 2011.
19. F. Sebastiani. A probabilistic terminological logic for modelling information retrieval. In *ACM Conf. on Research and Development in Information Retrieval (SIGIR)*, pages 122–130, 1994.
20. B. Taskar, M. FaiWong, P. Abbeel, and D. Koller. Link prediction in relational data. In *Proceedings of the 17th Neural Information Processing Systems (NIPS)*, 2003.
21. T. Wohlfarth and R. Ichise. Semantic and event-based approach for link prediction. In *Proceedings of the 7th International Conference on Practical Aspects of Knowledge Management*, 2008.
22. K. Yu, W. Chu, S. Yu, V. Tresp, and Z. Xu. Stochastic relational models for discriminative link prediction. In *Proceedings of the Neural Information Processing Systems (NIPS)*, 2006.

Position Papers

Distributed Imprecise Design Knowledge on the Semantic Web

Julian R. Eichhoff[1] and Wolfgang Maass[2]

[1] Furtwangen University, Robert-Gerwig-Platz 1, 78120 Furtwangen, Germany
julian.eichhoff@hs-furtwangen.de
[2] Saarland University, P.O. 15 11 50, 66041 Saarbrücken, Germany
wolfgang.maass@iss.uni-saarland.de

Abstract. In this paper we outline a shared knowledge representation based on RDF. It can be used in a distributed multi-tenant environment to store design knowledge. These RDF-graphs incorporate all necessary information to instantiate Bayesian network representations of certain problem solving cases, which are used to support the conceptual design tasks carried out by a salesperson during lead qualification.

1 Introduction

In industries that offer customized goods and services, which meet their customer's individual business needs, vendors are often required to employ a consultative sales strategy called "solution selling" (cf. [7]). It comprises mainly four interdependent processes carried out on a per project basis: requirements definition, customization and integration, deployment, and post-deployment support. The groundwork for these processes is laid by the vendor's sales force screening for potential customers (leads) and assessing their willingness and ability to buy a solution. This task is termed "lead qualification". Lead qualification in solution selling industries is highly dependent on a salesperson's individual knowledge of a lead's (problem) situation, of goods and services offered by the vendor and its partners, and of how certain bundles of goods and services may be used for problem solving; we term this design knowledge. But especially external salespersons are not directly involved in product development at the employing vendor, and thus may have narrow insights on how their work affects downstream processes. Experiences from other salespersons may not be considered due to limited reporting or inconsequent knowledge reuse. And limited possibilities or rigid policies for inter-organizational communication may exclude design insights from partnering organizations. To overcome these shortcomings in intra- and inter-organizational design knowledge reuse, we've implemented a shared design knowledge repository based on the Function-Behavior-Structure (FBS) framework [4] and use it for services which support the design activities during lead qualification.

Xue and Xu [8] suggest a web-accessible distributed database to store design knowledge based on the FBS notation. Like other models that operationalize the FBS framework [2, 6], they follow an entity-relationship approach. However, it

would require a significant knowledge engineering effort to continuously maintain a design model that builds on a highly detailed and formal knowledge representation (KR), where innovative yet uncertain design beliefs may be left out. Probability theory can provide an adequate framework to model uncertainties in design decisions [5]. Encouraging approaches that represent probabilistic belief networks by means of semantic models exist in other domains [9, 10]. But to the author's knowledge, there exist no Semantic Web representations of the FBS framework that incorporate uncertainty information, and can be managed in a multi-tenant environment. In [3] we defined a Bayesian Network (BN) representation of the FBS model, termed FBS-BN, which encodes design knowledge as probability tables. In the following we outline its shared storage in a distributed RDF-Store.

2 Design Knowledge Representation and Storage

A FBS-BN represents a configurational design space for a specific problem-solving situation in form of a Bayesian Network. Discrete random variables are used to describe possible design object configurations in light of the customer's demands. Every variable is associated with a certain component of the design object to serve as characterizing attribute. There are three different variable types: Function variables (F) represent the purpose for which a solution is designed for, i.e. goals and constraints of the customer. Structure variables (S) represent possible offerings, i.e. product and service bundles that can be provided by the vendor. Behaviors are mediating concepts between Functions and Structures representing the actual solution, i.e. how products and services are meant to achieve goals and fulfill constraints. There are three subtypes of Behavior variables: Be variables describe the solution as expected by the customer; their value is derived from Function variables. Bs variables represent the solution as offered by the vendor; their value is derived from Structure variables. And Bc variables are used for comparing the match of Be and Bs. The design knowledge about how Functions, Behaviors and Structures affect each other is encoded in form of conditional probability distributions (CPDs). These CPDs represent a set of propositions of the form "if concept X is in state x then another concept Y is (or should be) in state y". The associated probabilities express the degree of belief that a proposition holds. Possible relations are $F \rightarrow Be$ (Function expects Behavior), $S \rightarrow Bs$ (Structure exhibits Behavior), and implications within a variable group ($F \rightarrow F$, $Be \rightarrow Be$, $Bs \rightarrow Bs$, and $S \rightarrow S$).

To support the assessment of information in lead qualification, a support service should highlight those concepts that are yet uncertain and thus need further investigation. Therefore we generate a case-specific FBS-BN to characterize the current problem-solving situation. Changes in a node's prior can be used to represent explicit design decisions (evidence), i.e. assigning a relatively high probability to a state would express its preference over other states. Implicit design decisions are then given by Bayesian inference in form of probability

estimates for the hidden nodes. Building on this, we highlight (yet) uncertain concepts by rating every hidden node with an uncertainty measure (e.g. [3]).

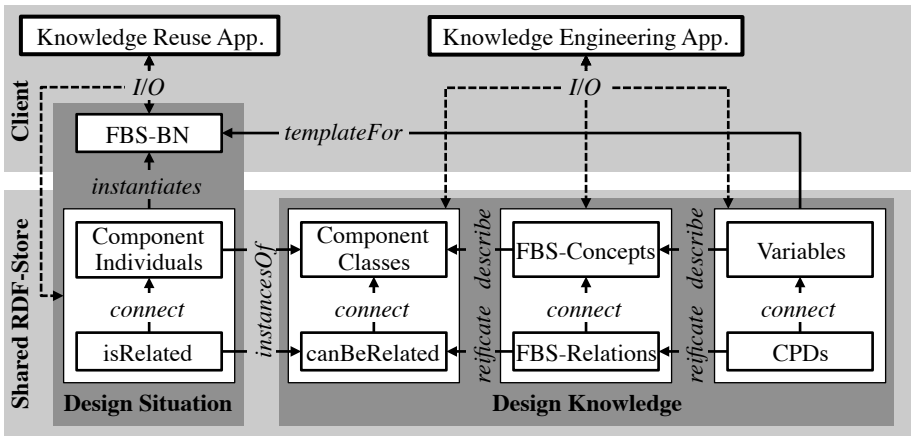


Fig. 1. Architecture of Proposed Distributed Knowledge Based System

For using FBS-BN representations across different participating organizations, we employ a RDF-based shared design KR to store the needed variable, CPD, and design object component definitions. All applications interoperate via this KR. Principally we consider two types of client-side application roles, namely knowledge engineering and knowledge reuse applications. While knowledge engineering applications provide an interface to manage the KR, knowledge reuse applications use it to support designing tasks in problem solving situations (cf. Fig. 1). The KR is stored in a distributed RDF-store is based on S3DB [1]. S3DB provides a sophisticated framework for graph-based permission management. Rather than using coarse all or nothing policies, S3DB allows an organization, department or individual to share certain parts of their design knowledge with designated users. Moreover, S3DB offers a meta-model for cooperatively defining TBoxes and ABoxes for RDF-graphs.

To facilitate the hierarchical formalization of design knowledge on different levels of complexity we employ a formalism for iterative reification: We start from a simple relational model for design object component classes and their individuals. Component classes can be linked with "canBeRelatedTo" relations to denote that they are dependent "somehow". These relations then frame possibilities for "isRelatedTo" relations on instance layer. The first step in clarifying these yet anonymous relations is done by providing FBS-concepts as characterizing attributes for component classes and connect them via expects, exhibits and implicates relations (FBS-relations). These associations determine how the design object components are actually interrelated with each other. In the second step, FBS-concepts are operationalized as discrete variables by specifying a set of possible variable states (or attribute values), which results in a description of

the attribute network as configurational variable space. Building on these variables we reificate attribute associations, i.e. we provide a detailed explication of expects, exhibits and implicates relations in form of conditional probability tables. Lastly variable and CPD definitions can be used as templates for FBS-BN instantiation.

3 Conclusion and Future Work

We have outlined a Semantic Web KR of the FBS framework based on RDF, which can be managed in a distributed multi-tenant environment. It is used to employ FBS-BN-based uncertainty reasoning for lead qualification support. Currently we are implementing two prototype applications, and look forward to test their impact on lead qualification performance empirically.

Acknowledgement This work was partially funded by the German Federal Ministry for Education and Research (BMBF, contract 17N0409). The authors would like to thank Sabine Janzen, Andreas Filler and Tobias Kowatsch for valuable discussions.

References

1. Almeida, J.S., Deus, H.F., Maass, W.: S3DB core: a framework for RDF generation and management in bioinformatics infrastructures. BMC Bioinformatics 11(387), pp. 1-10 (2010)
2. Christophe, F.,Bernard, A., Coatanéa, É.: RFBS: A model for knowledge representation of conceptual design. CIRP Ann.-Manuf. Techn. 59(1), pp. 155–158 (2010)
3. Eichhoff, J.R., Maass, W.: Representation and Reuse of Design Knowledge: An Application for Sales Call Support. In: König, A., Dengel, A., Hinkelmann, K., Kise, K., Howlett, R.J., Jain, L.C. (eds.) KES 2011, Part I. LNAI, vol. 6881, pp. 387–396. Springer, Heidelberg (2005)
4. Gero, J.S., Kannengiesser, U.: The situated function-behaviour-structure framework. Des. Stud. 25(4), pp. 373–391 (2004)
5. Nikolaidis, E., Mourelatos, Z.P., Pandey, V.: Design Decisions under Uncertainty with Limited Information. CRC Press/Balkema, Leiden (2011)
6. Szykman, S., Sriram, R.D., Bochenek, C., Racz, J.W., Senfaute, J.: Design Repositories: Next-Generation Engineering Design Databases. IEEE Intell. Syst. App. 15(3), pp. 48–55 (2000)
7. Tuli, K.R., Kohli, A.K., Bharadwaj, S.G.: Rethinking Customer Solutions: From Product Bundles to Relational Processes. J. Market. 71(3), pp. 1–17 (2007)
8. Xue, D., Xu, Y.: Web-based distributed system and database modeling for concurrent design. Computer-Aided Design 35(5), pp. 433–452 (2003)
9. Zhanga, W.Y., Caib, M., Qiuc, J., Yinb, J.W.: Managing distributed manufacturing knowledge through multi-perspective modelling for semantic web applications. Int. J. of Prod. Res. 47(23), pp. 6525–6542 (2009)
10. Zheng, H.-T., B.-Y. Kang, Kim, H.-G.: An ontology-based bayesian network approach for representing uncertainty in clinical practice guidelines. In: Goebel, R., Siekmann, J., Wahlster, W. (eds.) URSW 2005-2007. LNAI, vol. 5327, pp. 161–173. Springer, Heidelberg (2008)

Reasoning under Uncertainty with Log-Linear Description Logics

Mathias Niepert

KR & KM Research Group,
Universität Mannheim
Mannheim, Germany
{mathias}@informatik.uni-mannheim.de

Abstract. The position paper provides a brief summary of log-linear description logics and their applications. We compile a list of five requirements that we believe a probabilistic description logic should have to be useful in practice. We demonstrate the ways in which log-linear description logics answer to these requirements.

1 Introduction

Uncertainty is pervasive in the real world and reasoning in its presence one of the most pressing challenges in the development of intelligent systems. It is therefore hard to imagine how the Semantic Web could succeed without the ability to represent and reason under uncertainty. Nevertheless, purely logical approaches to knowledge representation and reasoning such as description logics have proven useful in providing the formal backbone of the Semantic Web. There is not only a large body of important work on the logical and algorithmic properties of such languages but also highly optimized tools that are successfully employed in meaningful applications. Still, the need to model uncertainty persists. Two prominent examples where the processing of uncertainty is crucial are (a) data integration (schema and instance alignment) and (b) ontology learning. In both cases, algorithms usually generate confidence values for particular axioms. In ontology matching, for instance, string similarity measures are often used to find confidence values for equivalence axioms between concepts and properties, respectively.

There have been attempts to combine logic and probability in various ways. Resulting approaches are probabilistic formalism for description logics [4, 5, 2, 6, 8] and, more generally, statistical relational languages [3]. The former are important theoretical contributions but have not been adopted by practitioners. We believe this is primarily due to the computational complexity of probabilistic inference, the rather involved way of expressing uncertainties syntactically, and the lack of implementations. Statistical relational approaches, on the other hand, have been successfully applied to numerous real-world problems but they do not explicitly take into account the notion of coherency and consistency which is crucial in the context of the Semantic Web.

Name	Syntax	Semantics
top	$\top$	$\Delta^{\mathcal{I}}$
bottom	$\bot$	$\emptyset$
nominal	$\{a\}$	$\{a^{\mathcal{I}}\}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
existential restriction	$\exists r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} : (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
GCI	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
RI	$r_1 \circ ... \circ r_k \sqsubseteq r$	$r_1^{\mathcal{I}} \circ ... \circ r_k^{\mathcal{I}} \subseteq r^{\mathcal{I}}$

Table 1. The description logic $\mathcal{EL}^{++}$ without nominals and concrete domains.

Based on these observations (and biases), and with the more concrete applications of ontology learning and matching in mind, we have compiled the following wish list for a probabilistic description logic.

1. The system must be usable by individuals knowledgeable only in Semantic Web languages and tools such as OWL and Protégé;
2. It must be possible to express uncertainty in form of *degrees of confidence* (real-valued weights) and not necessarily in form of precise probabilities. Real-world problems such as ontology matching and learning require this;
3. The user should not have to worry about inconsistent and incoherent input to the probabilistic reasoner. All types of inconsistencies are handled by the probabilistic reasoner and not the user;
4. Two types of queries should be supported under uncertainty: (a) The "most probable ontology" query and (b) the probability of (conjunctions) of axioms query; and
5. The worst-case complexity should not exceed that of probabilistic graphical models such as Markov and Bayesian networks. While inference in these models is generally NP-hard, numerous highly efficient algorithms exist and can be employed in the context of probabilistic DLs.

These five requirements are captured by log-linear description logics [7]. We provide a brief overview of log-linear description logics and discuss how this family of probabilistic logics answers to the outlined requirements.

2 Log-Linear Description Logics

Log-linear description logics integrate description logics with probabilistic log-linear models. Detailed technical and empirical results are available [7] and are mostly omitted in this position paper. The syntax of log-linear description logics is taken from the underlying description logic. However, it is possible to assign real-valued weights to axioms. Here, we focus on the log-linear description logic based on $\mathcal{EL}^{++}$ [1] without concrete domains (see Table 1) which we denote as

$\mathcal{EL}^{++}$-LL. $\mathcal{EL}^{++}$ captures the expressivity of numerous ontologies in the biomedical sciences and other domains, and it is the description logic on which the web ontology language profile OWL 2 EL is based. More formally, a $\mathcal{EL}^{++}$-LL ontology $\mathcal{C} = (\mathcal{C}^{\mathsf{D}}, \mathcal{C}^{\mathsf{U}})$ is a pair consisting of a *deterministic* $\mathcal{EL}^{++}$ CBox (set of axioms) $\mathcal{C}^{\mathsf{D}}$ and an *uncertain* CBox $\mathcal{C}^{\mathsf{U}} = \{(c, w_c)\}$ which is a set of pairs (c, w_c) with each c being a $\mathcal{EL}^{++}$ axiom and w a real-valued weight assigned to c. While the *deterministic* CBox contains axioms that are known to be true the *uncertain* CBox contains axioms for which we only have a *degree of confidence*. Every axiom can either be part of the deterministic or the uncertain CBox but not of both.

The semantics of log-linear DLs is based on joint probability distributions over *coherent* $\mathcal{EL}^{++}$ CBoxes and similar to that of Markov logic [9]. The weights of the axioms determine the log-linear probability distribution. For a $\mathcal{EL}^{++}$-LL CBox $(\mathcal{C}^{\mathsf{D}}, \mathcal{C}^{\mathsf{U}})$ and a $\mathcal{EL}^{++}$ CBox $\mathcal{C}'$ over the same set of basic concept descriptions and role names, we have that

$$P(\mathcal{C}') = \begin{cases} \frac{1}{Z} \exp\left(\sum_{\{(c,w_c) \in \mathcal{C}^{\mathsf{U}} : \mathcal{C}' \models c\}} w_c\right) & \text{if } \mathcal{C}' \text{ is } \textit{coherent} \\ & \text{and } \mathcal{C}' \models \mathcal{C}^{\mathsf{D}}; \\ 0 & \text{otherwise} \end{cases}$$

where Z is the normalization constant of the log-linear probability distribution.

The semantics of the log-linear description logic leads to probability distributions one would expect under the open world semantics of description logics.

Example 1. Let $\mathsf{Student}$ and $\mathsf{Professor}$ be two classes and let $\mathcal{C}^{\mathsf{D}} = \emptyset$ and $\mathcal{C}^{\mathsf{U}} = \{\langle \mathsf{Student} \sqsubseteq \mathsf{Professor}, 0.5\rangle, \langle \mathsf{Student} \sqcap \mathsf{Professor} \sqsubseteq \bot, 0.5\rangle\}$. Then[1], $P(\{\mathsf{Student} \sqsubseteq \mathsf{Professor}, \mathsf{Student} \sqcap \mathsf{Professor} \sqsubseteq \bot\}) = 0$, $P(\{\mathsf{Student} \sqsubseteq \mathsf{Professor}\}) = Z^{-1} \exp(0.5)$, $P(\{\mathsf{Student} \sqsubseteq \mathsf{Professor}, \mathsf{Professor} \sqsubseteq \mathsf{Student}\}) = Z^{-1} \exp(0.5)$, $P(\{\mathsf{Student} \sqcap \mathsf{Professor} \sqsubseteq \bot\}) = Z^{-1} \exp(0.5)$, $P(\{\mathsf{Professor} \sqsubseteq \mathsf{Student}\}) = Z^{-1} \exp(0)$, and $P(\emptyset) = Z^{-1} \exp(0)$ with $Z = 3 \exp(0.5) + 2 \exp(0)$.

We distinguish two types of probabilistic queries. The maximum a-posteriori (MAP) query: *"Given a $\mathcal{EL}^{++}$-LL CBox, what is a most probable coherent $\mathcal{EL}^{++}$ CBox over the same concept and role names?"*; and the conditional probability query: *"Given a $\mathcal{EL}^{++}$-LL CBox, what is the probability of a conjunction of axioms?"* We believe that the first type of query is useful since it infers the most probable *coherent* ontology from one that contains axioms with confidence values. The MAP query, therefore, has immediate applications in ontology learning and matching.

Probabilistic inference in log-linear description logics seems daunting at first, considering the combinatorial complexity of the problem. It turns out, however, that both the MAP and the conditional probability query can be computed efficiently for ontologies with thousands of known and uncertain axioms [7]. The worst-case complexity of both queries is equivalent to the worst-case complexity of the analogous queries in Markov and Bayesian networks (requirement 5).

[1] We omit trivial axioms that are present in every classified CBox such as $\mathsf{Student} \sqsubseteq \top$ and $\mathsf{Student} \sqsubseteq \mathsf{Student}$.

3 Log-Linear Description Logics in Practice

ELOG is a log-linear description logic reasoner developed at the University of Mannheim. A detailed description, the source code, and example ontologies are available at its webpage[2]. ELOG directly loads ontologies expressed in OWL 2 EL. The assignment of confidence values to axioms is made with the annotation property "confidence." Consider the following example ontology zoo.owl:

```
SubClassOf(
  Annotation(<http://URI/ontology#confidence> "0.5"^^xsd:double)
  <http://zoo/Penguin>
  <http://zoo/Bird>
)
DisjointClasses(
  <http://zoo/Bird>
  <http://zoo/Mammal>
)
```

Here, the subclass axiom is assigned the confidence value 0.5 and the disjointness axiom is considered true since it is not annotated. Therefore, the subclass axiom is part of the uncertain CBox and the disjointness axiom is part of the deterministic CBox. Considering that annotations can simply be added with popular ontology editors such as Protégé or using the OWL API[3], log-linear description logics fulfill requirements 1 and 2. In addition, the annotated axioms do not have to be consistent or coherent in any way because ELOG computes the probabilistic queries with respect to the joint probability distribution over *coherent* ontologies. Thus, ELOG also fulfills requirement 3.

References

1. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: Proceedings of the International Joint Conference on Artificial Intelligence (2005)
2. Costa, P.: Bayesian semantics for the Semantic Web. Ph.D. thesis, George Mason University (2005)
3. Getoor, L., Taskar, B.: Introduction to Statistical Relational Learning. MIT Press (2007)
4. Jaeger, M.: Probabilistic reasoning in terminological logics. In: Proceedings of the Conference on the Principles of Knowledge Representation and Reasoning (1994)
5. Koller, D., Levy, A., Pfeffer, A.: P-classic: A tractable probabilistic description logic. In: Proceedings of the 14th AAAI Conference on Artificial Intelligence (1997)
6. Lukasiewicz, T., Straccia, U.: Managing uncertainty and vagueness in description logics for the semantic web. J. of Web Sem. 6 (2008)
7. Niepert, M., Noessner, J., Stuckenschmidt, H.: Log-Linear Description Logics. In: Proceedings of the International Joint Conference on Artificial Intelligence (2011)
8. Predoiu, L., Stuckenschmidt, H.: Probabilistic models for the semantic web. In: The Semantic Web for Knowledge and Data Management (2008)
9. Richardson, M., Domingos, P.: Markov logic networks. Machine Learning (2006)

[2] http://code.google.com/p/elog-reasoner/
[3] http://owlapi.sourceforge.net/

Handling uncertainty in information extraction

Maurice van Keulen[1] and Mena B. Habib[1]

University of Twente, Faculty of EEMCS, Enschede, The Netherlands
{m.vankeulen,m.badiehhabibmorgan}@utwente.nl

Abstract. This position paper proposes an interactive approach for developing information extractors based on the ontology definition process with knowledge about possible (in)correctness of annotations. We discuss the problem of managing and manipulating probabilistic dependencies.

1 Introduction

(Too) much data is still inaccessible for data processing, because it is unstructured, textually embedded in documents, webpages, or text fields in databases. Information extraction (IE) is a technology capable of extracting entities, facts, and relations. IE helps to turn the web into a real 'web of data' [BHBL09].

In the Neogeography-project [HvK11b], we focus on named entity extraction (NEE) from database text fields and short messages. NEE typically consists of phases like recognition (which phrases are named entities), matching and enrichment (lookups in reference databases and dictionaries possibly adding information), and disambiguation (to which real-world object does a phrase refer).

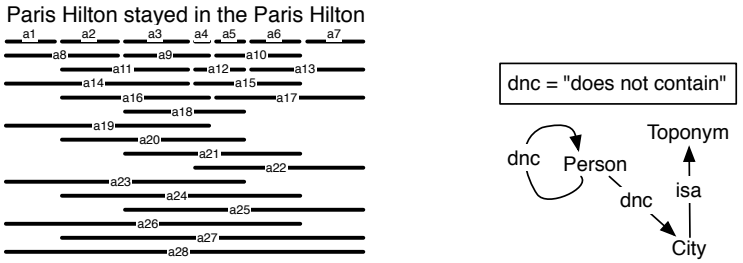
Because natural language is highly ambiguous and computers are still incapable of 'real' semantic understanding, NEE (and IE in general) is a highly imperfect process. For example, it is ambiguous how to interpret the word "Paris": it could be a first name, a city, etc. Even resolving it to a city, a lookup in GeoNames[1] learns that there are numerous other places called "Paris" besides the capital of France. In [HvK11a], we found that around 46% of toponyms[2] have two or more, 35% three or more, and 29% four or more references in GeoNames.

Although many probabilistic and fuzzy techniques abound, some aspects often remain absolute: extraction rules absolutely recognize and annotate a phrase or not, only a top item from a ranking is chosen for a next phase, etc. We envision an approach that *fundamentally* treats annotations and extracted information as uncertain throughout the process. We humans happily deal with doubt and misinterpretation every day, why shouldn't computers?

We envision developing information extractors 'Sherlock Holmes style' — *"when you have eliminated the impossible, whatever remains, however improbable, must be the truth"* — by adopting the principles and requirements below.

- Annotations are uncertain, hence we process both annotations as well as information about the uncertainty surrounding them.

[1] http://www.geonames.org
[2] A *toponym* is any name that refers to a location including, e.g., names of buildings.



(a) All possible annotations for the example sentence (b) Small example ontology

Fig. 1. Example sentence and NEE ontology

- We have an unconventional conceptual starting point, namely not "no annotations" but "there is no knowledge hence anything is possible". Fig.1(a) shows all possible annotations for an example sentence for one entity type.
- A developer gradually and interactively defines an ontology with positive and negative knowledge about the correctness of certain (combinations of) annotations. At each iteration, added knowledge is immediately applied improving the extraction result until the result is good enough (see also [vKdK09]).
- Storage, querying and manipulation of annotations should be scalable. Probabilistic databases are an attractive technology for this.

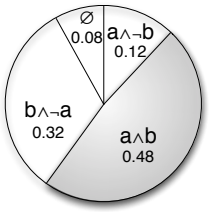
Basic forms of knowledge are the entity types one is interested in and declarations like τ_1 —*dnc*— τ_2 (no subphrase of a τ_1-phrase should be interpreted as τ_2, e.g, Person —*dnc*— City). See Fig.1(b) for a small example. We also envision application of background probability distributions, uncertain rules, etc. We hope these principles and forms of knowledge also allow for more effective handling of common problems (e.g., "you" is also the name of a place; should "Lake Como" or "Como" be annotated as a toponym).

2 Uncertain annotation model

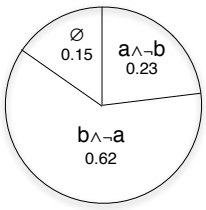
An *annotation* $a = (b, e, \tau)$ declares a phrase φ_e^b from b to e to be interpreted as entity type τ. For example, a_8 in Fig. 1(a) declares $\varphi =$ "Paris Hilton" from $b = 1$ to $e = 2$ to be interpreted as type $\tau =$ Person. An *interpretation* $I = (A, \mathcal{U})$ of a sentence s consists of an annotation set A and a structure $\mathcal{U}$ representing the uncertainty among the annotations. In the sequel, we discuss what $\mathcal{U}$ should be, but for now view it as a set of random variables (RVs) R with their dependencies.

Rather unconventionally, we don't start with an empty A, but with a 'no knowledge' point-of-view where any phrase can have any interpretation. So our initial A is $\{a \mid a = (b, e, \tau) \wedge \tau \in T \wedge \varphi_e^b \text{ is a phrase of } s\}$ where T is the set of possible types.

With T finite, A is also finite. More importantly, $|A| = O(klt)$ where $k = |s|$ is the length of s, l is the maximum length phrases considered, and $t = |T|$. Hence, A grows linearly in size with each. In the example of Fig.1(a), $T =$



(a) Annotations a and b independent with probabilities $P(a) = 0.6$ and $P(b) = 0.8$



(b) a and b conditioned to be mutually exclusive ($a \wedge b$ not possible)

Fig. 3. Defining a and b to be mutually exclusive means conditioning the probabilities.

$\{\mathsf{Person}, \mathsf{Toponym}, \mathsf{City}\}$ and we have $28 \cdot |T| = 84$ annotations. Even though we envision a more ingenious implementation, no probabilistic database would be severely challenged by a complete annotation set for a typical text field.

3 Knowledge application is conditioning

We explain how to 'apply knowledge' in our approach by means of the example of Fig.1, i.e., with our A with 84 (possible) annotations and an ontology only containing Person, Toponym, and City. Suppose we like to add the knowledge Person —*dnc*— City. The effect should be the removal of some annotations and adjustment of the probabilities of the remaining ones.

An initial promising idea is to store the annotations in an uncertain relation in a probabilistic database, such as MayBMS [HAKO09]. In MayBMS, the existence of each tuple is determined by an associated world set descriptor (wsd) containing a set of RV assignments from a world set table (see Fig.2). RVs are assumed independent. For example, the 3rd annotation tuple only exists when $x_8^1 = 1$ which is the case with a probability of 0.8. Each annotation can be seen as a probabilistic event, which are all independent in our starting point. Hence, we can store A by associating each annotation tuple a_i^j with one boolean RV x_i^j. Consequently, the database size is linear with $|A|$.

annotations

	b	**e**	**type**	...	**wsd**
a_1^1	1	1	Person	...	$\{x_1^1 = 1\}$
a_1^2	1	1	City	...	$\{x_1^2 = 1\}$
a_8^1	1	2	Person	...	$\{x_8^1 = 1\}$
	...	...	...	...	

world_set

x	v	P
x_1^1	0	0.4
x_1^1	1	0.6
x_1^2	0	0.7
x_1^2	1	0.3
x_8^1	0	0.2
x_8^1	1	0.8
...	...	...

Fig. 2. Initial annotation set stored in a probabilistic database (MayBMS-style)

Adding knowledge such as Person —*dnc*— City means that certain RVs become dependent and that certain combinations of RV assignments become impossible. Let us focus on two individual annotations a_1^2 ('"Paris" is a City) and a_8^1 ("Paris Hilton" is a Person). These two annotations become mutually exclusive. The process of adjusting the probabilities is called *conditioning* [KO08]. It boils down to redistributing the remaining probability mass. Fig.3 illustrates this for $a = a_1^2$ and $b = a_8^1$. The remaining probability mass is $1 - 0.48 = 0.52$. Hence, the

distribution of this mass over the remaining possibilities is $P(a \wedge \neg b) = \frac{0.12}{0.52} \approx 0.23$, $P(b \wedge \neg a) = \frac{0.32}{0.52} \approx 0.62$, and $P(\emptyset) = P(\neg a \wedge \neg b) = \frac{0.08}{0.52} \approx 0.15$.

A first attempt is to replace x_1^2 and x_8^1 with one fresh three-valued RV x' with the probabilities just calculated, i.e., $\text{wsd}(a_1^2) = \{x' = 1\}$ and $\text{wsd}(a_8^1) = \{x' = 2\}$ with $P(x' = 0) = 0.15$, $P(x' = 1) = 0.23$, and $P(x' = 2) = 0.62$. Unfortunately, since annotations massively overlap, we face a combinatorial explosion. For this rule, we end up with one RV with up to $2^{2 \cdot 28} = 2^{56} \approx 7 \cdot 10^{16}$ cases.

Solution directions What we are looking for in this paper is a structure that is expressive enough to capture all dependencies between RVs and at the same time allowing for scalable processing of conditioning operations. The work of [KO08] represents dependencies resulting from queries with a tree of RV assignments. We are also investigating the shared correlations work of [SDG08].

4 Conclusions

We envision an approach where information extractors are developed based on an ontology definition process for knowledge about possible (in)correctness of annotations. Main properties are treating annotations as fundamentally uncertain and interactive addition of knowledge starting from a 'no knowledge hence everything is possible' situation. The feasibility of the approach hinges on efficient storage and conditioning of probabilistic dependencies. We discuss this very problem, argue that a trivial approach doesn't work, and propose two solution directions: the conditioning approach of MayBMS and the shared correlations work of Getoor et al.

References

[BHBL09] C. Bizer, T. Heath, and T. Berners-Lee. Linked data: The story so far. *Int'l J. on Semantic Web and Information Systems (IJSWIS)*, 5(3):1–22, 2009.

[HAKO09] J. Huang, L. Antova, C. Koch, and D. Olteanu. MayBMS: a probabilistic database management system. In *Proc. of the 35th SIGMOD Int'l Conf. on Management Of Data, Providence, Rhode Island*, pages 1071–1074, 2009.

[HvK11a] M. B. Habib and M. van Keulen. Named entity extraction and disambiguation: The reinforcement effect. In *Proc. of the 5th Int'l Workshop on Management of Uncertain Data (MUD), Seatle, 29 Aug*, pages 9–16, 2011.

[HvK11b] M. B. Habib and M. van Keulen. Neogeography: The challenge of channelling large and ill-behaved data streams. Technical Report TR-CTIT-11-08, Enschede, 2011. ISSN 1381-3625, http://eprints.eemcs.utwente.nl/19854.

[KO08] C. Koch and D. Olteanu. Conditioning probabilistic databases. *Proc. VLDB Endow.*, 1(1):313–325, 2008.

[SDG08] P. Sen, A. Deshpande, and L. Getoor. Exploiting shared correlations in probabilistic databases. *Proc. VLDB Endow.*, 1(1):809–820, 2008.

[vKdK09] M. van Keulen and A. de Keijzer. Qualitative effects of knowledge rules and user feedback in probabilistic data integration. *The VLDB Journal*, 18(5):1191–1217, 2009.