8th International Workshop on Uncertainty Reasoning for the Semantic Web

Proceedings

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

Boston, USA, November 11, 2012

collocated with
the 11th International Semantic Web Conference
– ISWC 2012 –

Foreword

This volume contains the papers presented at the 8th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2012), held as a part of the 11th International Semantic Web Conference (ISWC 2012) at Boston, USA, November 11, 2012. It contains 7 technical papers and 2 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.

November 2012

Fernando Bobillo
Rommel Carvalho
Paulo C. G. da Costa
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)
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 (Technical University of Munich, Germany)
Michael Pool (Goldman Sachs, USA)

Program Committee

Fernando Bobillo (University of Zaragoza, Spain)
Silvia Calegari (University of Milano-Bicocca, Italy)
Rommel Carvalho (George Mason University, USA)
Davide Ceolin (Vrije Universiteit Amsterdam, Netherlands)
Paulo C. G. da Costa (George Mason University, USA)
Fabio Gagliardi Cozman (Universidade de São Paulo, Brazil)
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 (Technical University of Munich, Germany)
Jeff Z. Pan (University of Aberdeen, UK)
Rafael Peñaloza (TU Dresden, Germany)
Michael Pool (Goldman Sachs, 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)
Sergej Sizov (University of Koblenz-Landau, Germany)
Giorgos Stoilos (University of Oxford, UK)
Umberto Straccia (ISTI-CNR, Italy)
Matthias Thimm (Universität Koblenz-Landau, Germany)
Andreas Tolk (Old Dominion University, USA)
Peter Vojtáš (Charles University Prague, Czech Republic)

Table of Contents

Technical Papers

Position Papers

Technical Papers

Epistemic and Statistical Probabilistic Ontologies

Fabrizio Riguzzi, Elena Bellodi, Evelina Lamma, and Riccardo Zese

ENDIF – University of Ferrara, Via Saragat 1, I-44122, Ferrara, Italy
{fabrizio.riguzzi,elena.bellodi,evelina.lamma}@unife.it
riccardo.zese@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 the axioms of a probabilistic ontology can be annotated with an epistemic or a statistical probability. The epistemic probability represents a degree of confidence in the axiom, while the statistical probability considers the populations to which the axiom is applied.

1 Introduction

Uncertainty has been recognized as an important feature for the Semantic Web [36, 24]. In order to be able to represent and reason with probabilistic knowledge, various authors have advocated the use of probabilistic ontologies, see e.g. [26], and many proposals have been put forward for allowing ontology languages, and OWL in particular, to represent uncertainty [30, 10, 22, 23].

The integration of probability into logic has been much studied lately, with many different proposals. In the field of logic programming, the distribution semantics [33] has emerged as one of the most effective approaches [33, 28, 8].

In [3] we applied this approach to ontological languages and, in particular, to the OWL DL fragment, that is based on the description logic $\mathcal{SHOIN}(\mathbf{D})$. We called the approach DISPONTE for "DIstribution Semantics for Probabilistic ONTologiEs" (Spanish for "get ready"). The idea is to annotate axioms of a theory with a probability and assume that each axiom is independent of the others. In this paper we extend DISPONTE by considering two types of probabilistic annotations, an epistemic type, that represents a degree of belief in the axiom as a whole as in [3], and a new statistical type, that considers the populations to which the axiom is applied. Statistical probabilities allow to represent partial concept overlapping and knowledge on random individuals of populations. The two types of probability can be used separately or jointly in the same OWL DL knowledge base. The probability of a query is computed from a covering set of explanations by solving the disjoint sum problem.

The paper is organized as follows. Section 2 introduces Description Logics. Section 3 presents DISPONTE and explains the principles on which it is based, while Section 4 illustrates approaches for performing reasoning. Section 5 describes related work and, finally, Section 6 concludes the paper.

2 Description Logics

Description Logics (DLs) are knowledge representation formalisms that possess nice computational properties such as decidability and/or low complexity, see [1, 2] for excellent introductions. DLs are particularly useful for representing ontologies and have been adopted as the basis of the Semantic Web. For example, the OWL DL sublanguage of OWL is based on the $\mathcal{SHOIN}(\mathbf{D})$ DL.

While DLs can be translated into predicate logic, they are usually represented using a syntax based on concepts and roles. A concept corresponds to a set of individuals of the domain while a role corresponds to a set of couples of individuals of the domain. In order to illustrate DLs, we now describe $\mathcal{SHOIN}$ following [24].

Let $\mathbf{A}$, $\mathbf{R}$ and $\mathbf{I}$ be sets of *atomic concepts*, *roles* and *individuals*, respectively. A *role* is either an atomic role $R \in \mathbf{R}$ or the inverse R^- of an atomic role $R \in \mathbf{R}$. We use $\mathbf{R}^-$ to denote the set of all inverses of roles in $\mathbf{R}$. An RBox $\mathcal{R}$ consists of a finite set of *transitivity axioms* $Trans(R)$, where $R \in \mathbf{R}$, and *role inclusion axioms* $R \sqsubseteq S$, where $R, S \in \mathbf{R} \cup \mathbf{R}^-$.

Concepts are defined by induction as follows. Each $A \in \mathbf{A}$ is a concept, $\bot$ and $\top$ are concepts, and if $a \in \mathbf{I}$, then $\{a\}$ is a concept. If C, $C1$ and $C2$ are concepts and $R \in \mathbf{R} \cup \mathbf{R}^-$, then $(C_1 \sqcap C_2)$, $(C_1 \sqcup C_2)$, and $\neg C$ are concepts, as well as $\exists R.C$, $\forall R.C$, $n \geq R$ and $n \leq R$ for an integer $n \geq 0$.

A *TBox* $\mathcal{T}$ is a finite set of *concept inclusion axioms* $C \sqsubseteq D$, where C and D are concepts. We use $C \equiv D$ to abbreviate $C \sqsubseteq D$ and $D \sqsubseteq C$. An *ABox* $\mathcal{A}$ is a finite set of *concept membership axioms* $a : C$, *role membership axioms* $(a, b) : R$, *equality axioms* $a = b$, and *inequality axioms* $a \neq b$, where C is a concept, $R \in \mathbf{R}$ and $a, b \in \mathbf{I}$. A *knowledge base* $\mathcal{K} = (\mathcal{T}, \mathcal{R}, \mathcal{A})$ consists of a TBox $\mathcal{T}$, an RBox $\mathcal{R}$ and an ABox $\mathcal{A}$.

$\mathcal{SHOIN}$ is decidable iff there are no number restrictions on non-simple roles. A role is non-simple iff it is transitive or it has transitive subroles.

A knowledge base $\mathcal{K}$ is usually assigned a semantics in terms of set-theoretic interpretations and models of the form $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$. The semantics of DLs can be given equivalently by transforming a DL knowledge base into a predicate logic theory and then using the model-theoretic semantics of the resulting theory. A translation of $\mathcal{SHOIN}$ into First Order Logic with Counting Quantifiers is given in the following as an extension of the one given in [34]. The translation uses two functions π_x and π_y that map concept expressions to logical formulas, where π_x is given by

$$\begin{aligned}
\pi_x(A) &= A(x) \\
\pi_x(\neg C) &= \neg \pi_x(C) \\
\pi_x(\{a\}) &= (x = a) \\
\pi_x(C \sqcap D) &= \pi_x(C) \wedge \pi_x(D) \\
\pi_x(C \sqcup D) &= \pi_x(C) \wedge \pi_x(D)
\end{aligned}$$

$$
\begin{aligned}
\pi_x(\exists R.C) &= \exists y.R(x,y) \land \pi_y(C) \\
\pi_x(\exists R^-.C) &= \exists y.R(y,x) \land \pi_y(C) \\
\pi_x(\forall R.C) &= \forall y.R(x,y) \to \pi_y(C) \\
\pi_x(\forall R^-.C) &= \forall y.R(y,x) \to \pi_y(C) \\
\pi_x(\geq nR) &= \exists^{\geq n} y.R(x,y) \\
\pi_x(\geq nR^-) &= \exists^{\geq n} y.R(y,x) \\
\pi_x(\leq nR) &= \exists^{\leq n} y.R(x,y) \\
\pi_x(\leq nR^-) &= \exists^{\leq n} y.R(y,x)
\end{aligned}
$$

and π_y is obtained from π_x by replacing x with y and vice-versa.

Table 1 shows the translation of each axiom of $\mathcal{SHOIN}$ knowledge bases.

Axiom	Translation
$C \sqsubseteq D$	$\forall x.\pi_x(C) \to \pi_x(D)$
$R \sqsubseteq S$	$\forall x,y.R(x,y) \to S(x,y)$
$Trans(R)$	$\forall x,y,z.R(x,z) \land R(z,y) \to S(x,y)$
$a : C$	$C(a)$
$(a,b) : R$	$R(a,b)$
$a = b$	$a = b$
$a \neq b$	$a \neq b$

Table 1. Translation of $\mathcal{SHOIN}$ axioms into predicate logic.

$\mathcal{SHOIN}(\mathbf{D})$ adds to $\mathcal{SHOIN}$ datatype roles, i.e., roles that map an individual to an element of a datatype such as integers, floats, etc. Then new concept definitions, involving datatype roles, are added, that mirror those involving roles introduced above. We also assume that we have predicates over the datatypes.

A query over a knowledge base is usually an axiom for which we want to test the entailment from the knowledge base. The entailment test may be reduced to checking the satisfiability of a concept in the knowledge base, i.e., the non-emptiness of the concept. For example, the entailment of the axiom $C \sqsubseteq D$ may be tested by checking the satisfiability of the concept $C \sqcap \neg D$.

3 The DISPONTE Semantics for Probabilistic Ontologies

A *probabilistic knowledge base* is a set of *certain axioms*, that take the form of DL axioms, of *epistemic probabilistic axioms* of the form

$$p ::_e E \tag{1}$$

where p is a real number in $[0,1]$ and E is a TBox, RBox or ABox axiom, and of *statistical probabilistic axioms* of the form

$$p ::_s E \tag{2}$$

where p is a real number in $[0, 1]$ and E is a TBox or RBox axiom.

In axioms of the form (1), the notation $::_e$ introduces probability p as an epistemic probability, i.e., as the degree of our belief in axiom E, while in axioms of the form (2) the notation $::_s$ interprets p as a statistical probability, i.e., as information regarding random individuals from certain populations. For example, an epistemic probabilistic concept inclusion axiom of the form

$$p ::_e C \sqsubseteq D \tag{3}$$

represents the fact that we believe in the truth of $C \sqsubseteq D$ with probability p. A statistical probabilistic concept inclusion axiom of the form

$$p ::_s C \sqsubseteq D \tag{4}$$

instead means that a random individual of class C has probability p of belonging to D, thus representing the statistical information that a fraction p of the individuals of C belong to D. In this way, the overlap between C and D is quantified by the statistical probability p. The difference between the two axioms is that, if two individuals belong to class C, the probability that they both belong to D according to (3) is p, since p represents the truth of the formula as a whole, while according to (4) is $p \cdot p$, since each individual has probability p of belonging to class D and the two events are independent. Therefore, statistical probability regards the knowledge we have about population of a given domain.

In order to give a semantics to such probabilistic knowledge bases, we consider their translation into predicate logic. The idea of DISPONTE is to associate independent Boolean random variables to (instantiations of) the formulas in predicate logic that are obtained from the translation of the axioms. By assigning values to every random variable we obtain a *world*, the set of predicate logic formulas whose random variable is assigned to 1.

To clarify what we mean by instantiations, we assume a fixed interpretation domain $\Delta^{\mathcal{I}}$ that is non-empty and possibly infinite. Given a predicate logic formula F and a domain $\Delta^{\mathcal{I}}$, we define a *substitution* θ as a set of couples x/i where x is a variable universally quantified in the outermost quantifier in F and $i \in \Delta^{\mathcal{I}}$. The application of θ to F, indicated by $F\theta$, is called an *instantiation of* F and is obtained by replacing x with i in F and by removing x from the external quantification for every couple x/i in θ. By instantiating the universally qualified variables we are able to separately represent each individual to which the axiom is applied.

To obtain a world w of a probabilistic knowledge base T, we translate every axiom into a predicate logic formula and we replace each individual a appearing in the knowledge base $\mathcal{K}$ with $a^{\mathcal{I}}$. Every formula obtained from an axiom without a probability annotation is included in w. For each axiom of the form (1), we decide whether or not to include it in w. For each axiom of the form (2), we generate all the substitutions for the variables of the equivalent predicate logic formula indicated in Table 2.

There may be an infinite number of instantiations. For each instantiated formula we decide whether or not to include it in w. In this way we obtain

Axiom	Variables to be instantiated
$p ::_s C \sqsubseteq D$	x
$p ::_s R \sqsubseteq S$	x, y
$p ::_s Trans(R)$	x, y, z

Table 2. Axiom variables to be instantiated.

a predicate logic theory which can be assigned a model-theoretic semantics. A query is entailed by a world if it is true in every model of the world.

To formally define the semantics we follow the approach of [28]. An *atomic choice* in this context is a triple (F_i, θ_j, k) where F_i is the formula obtained by translating the ith axiom, θ_j is a substitution and $k \in \{0, 1\}$. k indicates whether (F_i, θ_j, k) is chosen to be included in a world ($k = 1$) or not ($k = 0$). If F_i is obtained from an unannotated axiom, then $\theta_j = \emptyset$ and $k = 1$. If F_i is obtained from an axiom of the form (1), then $\theta_j = \emptyset$. If F_i is obtained from an axiom of the form (2), then θ_j instantiates the variables indicated in Table 2. Note that, differently from [28], substitutions do not ground formulas but this is not a core requirement of [28].

A *composite choice* κ is a consistent set of atomic choices, i.e., $(F_i, \theta_j, k) \in \kappa, (F_i, \theta_j, m) \in \kappa \Rightarrow k = m$ (only one decision for each formula). 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, i.e., it contains an atomic choice (F_i, θ_j, k) for every instantiation $F_i\theta_j$ of every formula of the theory. Since the domain may be infinite, selections may, too. Let us indicate with $\mathcal{S}_T$ the set of all selections. A selection σ identifies a theory w_σ called a *world* in this way: $w_\sigma = \{F_i\theta_j | (F_i, \theta_j, 1) \in \sigma\}$. Let us indicate with $\mathcal{W}_T$ the set of all worlds. A composite choice κ identifies a set of worlds $\omega_\kappa = \{w_\sigma | \sigma \in \mathcal{S}_T, \sigma \supseteq \kappa\}$. We define the set of worlds identified by a set of composite choices K as $\omega_K = \bigcup_{\kappa \in K} \omega_\kappa$.

A composite choice κ is an *explanation* for a query Q if Q is entailed by every world of ω_κ. A set of composite choices K is *covering* with respect to Q if every world w_σ in which Q is entailed is such that $w_\sigma \in \omega_K$. Two composite choices κ_1 and κ_2 are *incompatible* if their union is inconsistent. A set K of composite choices is *mutually incompatible* if for all $\kappa_1 \in K, \kappa_2 \in K, \kappa_1 \neq \kappa_2 \Rightarrow \kappa_1$ and κ_2 are incompatible. We define *the probability of a mutually incompatible set of composite choices* K as $P(K) = \sum_{\kappa \in K} P(\kappa)$. Two sets of composite choices K_1 and K_2 are *equivalent* if $\omega_{K_1} = \omega_{K_2}$, i.e., if they identify the same set of worlds.

Kolmogorov defined probability functions (or measures) as real-valued functions over an algebra Ω of subsets of a set $\mathcal{W}$ called the *sample space*. The set Ω is an algebra of $\mathcal{W}$ iff (1) $\mathcal{W} \in \Omega$, (2) Ω is closed under complementation, i.e., $\omega \in \Omega \rightarrow (\mathcal{W} \setminus \omega) \in \Omega$ and (3) Ω is closed under finite union, i.e., $\omega_1 \in \Omega, \omega_2 \in \Omega \rightarrow (\omega_1 \cup \omega_2) \in \Omega$. The elements of Ω are called *measurable sets*. Not every subset of $\mathcal{W}$ need be present in Ω.

Given a sample space $\mathcal{W}$ and an algebra Ω of subsets of $\mathcal{W}$, a probability measure is a function $\mu : \Omega \rightarrow R$ that satisfies the following axioms: (1) $\mu(\omega) \geq 0$

for all $\omega \in \Omega$, (2) $\mu(\mathcal{W}) = 1$, (3) $\omega_1 \cap \omega_2 = \emptyset \to \mu(\omega_1 \cup \omega_2) = \mu(\omega_1) + \mu(\omega_2)$ for all $\omega_1 \in \Omega, \omega_2 \in \Omega$.

If $F\theta$ is an instantiated formula and κ is a composite choice such that $\kappa \cap \{(F, \theta, 0), (F, \theta, 1)\} = \emptyset$, the *split* of κ on $F\theta$ is the set of composite choices $S_{F\theta,\kappa} = \{\kappa \cup \{(F, \theta, 0)\}, \kappa \cup \{(F, \theta, 1)\}\}$. It is easy to see that κ and $S_{F\theta,\kappa}$ identify the same set of possible worlds, i.e., that $\omega_\kappa = \omega_{S_{F\theta,\kappa}}$.

Following [28], we can prove the following results.

Theorem 1 (Existence of a mutually incompatible set of composite choices). *Given a finite set K of finite composite choices, there exists a finite set K' of mutually incompatible finite composite choices such that $\omega_K = \omega_{K'}$.*

Proof. Given a finite set of finite composite choices K, there are two possibilities to form a new set K' of composite choices so that K and K' describe the same set of possible worlds:

1. **removing dominated elements**: if $\kappa_1, \kappa_2 \in K$ and $\kappa_1 \subset \kappa_2$, let $K' = K \setminus \{\kappa_2\}$.
2. **splitting elements**: if $\kappa_1, \kappa_2 \in K$ and $\kappa_1 \cup \kappa_2$ is consistent (and neither is a superset of the other), there is an $(F, \theta, k) \in \kappa_1 \setminus \kappa_2$. We replace κ_2 by the split of κ_2 on $F\theta$. Let $K' = K \setminus \{\kappa_2\} \cup S_{F\theta,\kappa_2}$.

In both cases $\omega_K = \omega_{K'}$. If we repeat this two operations until neither is applicable we obtain a splitting algorithm (see Figure 1) that terminates because K is a finite set of finite composite choices. The resulting set K' is mutually incompatible and is equivalent to the original set. □

Fig. 1. Splitting Algorithm.

```
procedure SPLIT($K$)
    Input: set of composite choices $K$
    Output: mutually incompatible set of composite choices equivalent to $K$
    loop
        if $\exists \kappa_1, \kappa_2 \in K$ and $\kappa_1 \subset \kappa_2$ then
            $K := K \setminus \{\kappa_2\}$
        else
            if $\exists \kappa_1, \kappa_2 \in K$, such that $\kappa_1 \cup \kappa_2$ is inconsistent then
                choose $(F, \theta, k) \in \kappa_1 \setminus \kappa_2$
                $K := K \setminus \{\kappa_2\} \cup S_{F\theta,\kappa_2}$
            else
                exit and return $K$
            end if
        end if
    end loop
end procedure
```

Theorem 2 (Equivalence of the probability of two equivalent mutually incompatible finite set of finite composite choices). *If K_1 and K_2 are both mutually incompatible finite sets of finite composite choices such that they are equivalent then $P(K_1) = P(K_2)$.*

Proof. The theorem is the same as Lemma A.8 in [29]. We report here the proof for the sake of clarity.

Consider the set D of all instantiated formulas $F\theta$ that appear in an atomic choice in either K_1 and K_2. This set is finite. Each composite choice in K_1 and K_2 has atomic choices for a subset of D. For both K_1 and K_2, we repeatedly replace each composite choice κ of K_1 and K_2 with its split K' on an $F_i\theta_j$ from D that does not appear in κ. This procedure does not change the total probability as the probabilities of $(F_i, \theta_j, 0)$ and $(F_i, \theta_j, 1)$ sum to 1.

At the end of this procedure the two sets of composite choices will be identical. In fact, any difference can be extended into a possible world belonging to ω_{K_1} but not to ω_{K_2} or vice versa. □

We can thus define a unique probability measure $\mu : \Omega_T \rightarrow [0, 1]$ where Ω_T is defined as the set of sets of worlds identified by finite sets of finite composite choices: $\Omega_T = \{\omega_K | K \text{ is a finite set of finite composite choices}\}$. It is easy to see that Ω_T is an algebra over $\mathcal{W}_T$.

Then μ is defined by $\mu(\omega_K) = P(K')$ where K' is a finite mutually incompatible set of finite composite choices equivalent to K. $\langle \mathcal{W}_T, \Omega_T, \mu \rangle$ is a probability space according to Kolmogorov's definition.

The probability of a query Q is given by $P(Q) = \mu(\{w|w \in \mathcal{W}_T \wedge w \models Q\})$. If Q has a finite set K of finite explanations such that K is covering then $\{w|w \in \mathcal{W}_T \wedge w \models Q\} = \omega_K \in \Omega_T$ and $P(Q)$ is well-defined.

Example 1. Let us consider the following ontology, inspired by the `people+pets` ontology proposed in [27]:

$$
\begin{aligned}
&\exists hasAnimal.Pet \sqsubseteq PetOwner \\
&(kevin, fluffy) : hasAnimal \\
&(kevin, tom) : hasAnimal \\
0.4 ::_e\ &\mathit{fluffy} : Cat \\
0.3 ::_e\ &tom : Cat \\
0.6 ::_e\ &Cat \sqsubseteq Pet
\end{aligned}
$$

The predicate logic formulas (without external quantifiers) equivalent to the probabilistic axioms are

$$
\begin{aligned}
F_1 &= Cat(\mathit{fluffy}) \\
F_2 &= Cat(tom) \\
F_3 &= Cat(x) \rightarrow Pet(x)
\end{aligned}
$$

A covering set of explanations for the query axiom $Q = kevin : PetOwner$ is $K = \{\kappa_1, \kappa_2\}$ where $\kappa_1 = \{(F_1, \emptyset, 1), (F_3, \emptyset, 1)\}$ and $\kappa_2 = \{(F_2, \emptyset, 1), (F_3, \emptyset, 1)\}$.

An equivalent mutually exclusive set K' of explanations can be obtained by applying the splitting algorithm. In this case $K' = \{\kappa'_1, \kappa'_2\}$ where $\kappa'_1 = \{(F_1, \emptyset, 1), (F_3, \emptyset, 1), (F_2, \emptyset, 0)\}$ and $\kappa'_2 = \{(F_2, \emptyset, 1), (F_3, \emptyset, 1)\}$. So $P(Q) = 0.4 \cdot 0.6 \cdot 0.7 + 0.3 \cdot 0.6 = 0.348$.

Example 2. If the axiom $0.6 ::_e Cat \sqsubseteq Pet$ in Example 1 is replaced by $0.6 ::_s Cat \sqsubseteq Pet$ then the query would have the explanations $K = \{\kappa_1, \kappa_2\}$ where $\kappa_1 = \{(F_1, \emptyset, 1), (F_3, \{x/\mathit{fluffy}\}, 1)\}$ and $\kappa_2 = \{(F_2, \emptyset, 1), (F_3, \{x/\mathit{tom}\}, 1)\}$.
An equivalent mutually exclusive set K' of explanations obtained by applying the splitting algorithm is $K' = \{\kappa'_1, \kappa'_2, \kappa'_3\}$ with $\kappa'_1 = \{(F_1, \emptyset, 1), (F_3, \{x/\mathit{fluffy}\}, 1), (F_2, \emptyset, 0)\}$, $\kappa'_2 = \{(F_1, \emptyset, 1), (F_3, \{x/\mathit{fluffy}\}, 1), (F_2, \emptyset, 1), (F_3, \{x/\mathit{tom}\}, 0)\}$ and $\kappa'_3 = \{(F_2, \emptyset, 1), (F_3, \{x/\mathit{tom}\}, 1)\}$. So $P(Q) = 0.4 \cdot 0.6 \cdot 0.7 + 0.4 \cdot 0.6 \cdot 0.3 \cdot 0.4 + 0.3 \cdot 0.6 = 0.3768$.

Example 3. Let us consider a slightly different ontology:

$$\begin{aligned}
0.5 ::_s \; & \exists hasAnimal.Pet \sqsubseteq PetOwner \\
& (kevin, \mathit{fluffy}) : hasAnimal \\
& (kevin, tom) : hasAnimal \\
& \mathit{fluffy} : Cat \\
& tom : Cat \\
0.6 ::_s \; & Cat \sqsubseteq Pet
\end{aligned}$$

The predicate logic formulas without external quantifiers equivalent to the probabilistic axioms are

$$\begin{aligned}
F_1 &= \exists y.hasAnimal(x, y) \land Pet(y) \rightarrow PetOwner(x) \\
F_2 &= Cat(x) \rightarrow Pet(x)
\end{aligned}$$

A covering set of explanations for the query axiom $Q = kevin : PetOwner$ is $K = \{\kappa_1, \kappa_2\}$ where $\kappa_1 = \{(F_1, \{x/kevin\}, 1), (F_2, \{x/\mathit{fluffy}\}, 1)\}$ and $\kappa_2 = \{(F_1, \{x/kevin\}, 1), (F_2, \{x/tom\}, 1)\}$.

An equivalent mutually exclusive set K' of explanations obtained by applying the splitting algorithm is $K' = \{\kappa'_1, \kappa'_2\}$ where $\kappa'_1 = \{(F_1, \{x/kevin\}, 1), (F_2, \{x/\mathit{fluffy}\}, 1), (F_2, \{x/tom\}, 0)\}$ and $\kappa'_2 = \{(F_1, \{x/kevin\}, 1), (F_2, \{x/tom\}, 1)\}$. So $P(Q) = 0.5 \cdot 0.6 \cdot 0.4 + 0.5 \cdot 0.6 = 0.42$.

Example 4. Let us consider the ontology:

$$\begin{aligned}
0.7 &::_s Schoolchild \sqsubseteq European \\
0.4 &::_s Schoolchild \sqsubseteq OnlyChild \\
0.6 &::_s European \sqsubseteq GoodInMath \\
0.5 &::_s OnlyChild \sqsubseteq GoodInMath
\end{aligned}$$

The predicate logic formulas without the external quantifiers equivalent to the probabilistic axioms are:

$$F_1 = Schoolchild(x) \rightarrow European(x)$$

$$F_2 = \mathit{Schoolchild}(x) \rightarrow \mathit{OnlyChild}(x)$$
$$F_3 = \mathit{European}(x) \rightarrow \mathit{GoodInMath}(x)$$
$$F_4 = \mathit{OnlyChild}(x) \rightarrow \mathit{GoodInMath}(x)$$

A covering set of explanations for the query axiom $Q = \mathit{Schoolchild} \sqsubseteq \mathit{GoodInMath}$ is $K = \{\kappa_1, \kappa_2\}$ where $\kappa_1 = \{(F_1, \{x/i\}, 1), (F_3, \{x/i\}, 1)\}$ and $\kappa_2 = \{(F_2, \{x/i\}, 1), (F_4, \{x/i\}, 1)\}$, where i is an anonymous member of $\Delta^{\mathcal{I}}$. After splitting we get $K' = \{\kappa'_1, \kappa'_2, \kappa'_3\}$ where $\kappa'_1 = \{(F_1, \{x/i\}, 1), (F_3, \{x/i\}, 1)\}$, $\kappa'_2 = \{(F_1, \{x/i\}, 0), (F_2, \{x/i\}, 1), (F_4, \{x/i\}, 1)\}$ and $\kappa'_3 = \{(F_1, \{x/i\}, 1), (F_3, \{x/i\}, 0), (F_2, \{x/i\}, 1), (F_4, \{x/i\}, 1)\}$. So $P(Q) = 0.7 \cdot 0.6 + 0.3 \cdot 0.4 \cdot 0.5 + 0.7 \cdot 0.4 \cdot 0.4 \cdot 0.5 = 0.536$.

4 Reasoning under the DISPONTE semantics

The BUNDLE algorithm presented in [3] computes the probability of queries from a probabilistic ontology that follows the DISPONTE semantics with only epistemic probabilities. BUNDLE uses an underlying DL reasoner that is able to return explanations for queries such as Pellet [35]. The *explain* function of Pellet is used for this purpose [15, 17, 12, 16]. BUNDLE makes the explanations mutually incompatible by using Binary Decision Diagrams (BDD)[4] because this approach has been demonstrated to be faster than using the splitting algorithm [18, 31]. BDDs are used to compute the probability using the dynamic programming algorithm of [8].

In order to also deal with statistical probabilities, Pellet needs to be modified so that it records, besides the axioms that have been used to answer the query, also the individuals to which they are applied. We are currently working on modifying the tableau expansion rules of Pellet so that the individuals on which they operate are recorded together with the axioms. The first results on this activity are reported in [32], which presents a performance evaluation of inference over the real probabilistic ontology for breast cancer risk assesment both for BUNDLE and PRONTO.

5 Related Work

[13] 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, [20] presented a probabilistic description logic based on Bayesian networks that deals with statistical terminological knowledge. [13, 20] do not allow probabilistic assertional knowledge about concept and role instances. [14] allows assertional knowledge about concept and role instances together with statistical terminological knowledge and combines the resulting probability distributions using cross-entropy minimization but does not allow epistemic statements.

[9] 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 measure to sets of worlds. PR-OWL [6, 5] is an upper ontology that provides a framework for building probabilistic ontologies. It allows to use the first-order probabilistic logic MEBN [21] for representing uncertainty in ontologies. The use of a full fledged first-order probabilistic logic distinguishes 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 [10, 22, 23] 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 [19] is one of the systems that allows to perform inference in this semantics. These works are based on Nilsson's probabilistic logic [25], where a probabilistic interpretation Pr defines a probability distribution over the set of interpretations Int. The probability of a logical formula F according to Pr, denoted $Pr(F)$, is the sum of all $Pr(I)$ such that $I \in Int$ and $I \models F$. A probabilistic knowledge base K is a set of probabilistic formulas of the form $F \geq p$. A probabilistic interpretation Pr satisfies $F \geq p$ iff $Pr(F) \geq p$. Pr satisfies K, or Pr is a model of K, iff Pr satisfies all $F \geq p \in K$. $Pr(F) \geq p$ is a tight logical consequence of K iff p is the infimum of $Pr(F)$ subject to all models Pr of K.

Nilsson's probabilistic logic differs from the distribution semantics: while the first computes the lowest p such that $Pr(F) \geq p$ holds for all Pr, the latter computes p such that $P(F) = p$. Thus Nilsson's logic allows weaker conclusions: consider a probabilistic ontology composed of the axioms $0.4 ::_e a : C.$ and $0.5 ::_e b : C.$ and a probabilistic knowledge base composed of $C(a) \geq 0.4$ and $C(b) \geq 0.5$. The distribution semantics allows to say that $P(a : C \vee b : C) = 0.7$, while with Nilsson's logic the lowest p such that $Pr(C(a) \vee C(b)) \geq p$ holds is 0.5. This is due to the fact that in the distribution semantics the probabilistic axioms are considered as independent, which allows to make stronger conclusions.

Other approaches, such as [7, 11], combine a liteweight ontology language, *DL-Lite* and Datalog+/- respectively, with graphical models, Bayesian networks and Markov networks respectively. In both cases, an ontology is composed of a set of annotated axioms and a graphical model and the annotations are sets of assignments of random variables from the graphical model. The semantics is assigned by considering the possible worlds of the graphical model and by stating that an axiom holds in a possible world if the assignments in its annotation hold. The probability of a conclusion is then the sum of the probabilities of the possible worlds where the conclusion holds. Our approach provides a tighter integration of probability in ontologies as we do not rely on an additional graphical model.

In summary, we allow to extend Description Logics languages with more complete treatment of probabilistic knowledge that includes subjective and statistical statement.

6 Conclusions

We extended the DISPONTE semantics by allowing the representation of statistical information on concepts and populations. In this way both epistemic and partial overlap information can be expressed in a seamless way.

References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
2. Baader, F., Horrocks, I., Sattler, U.: Description logics. In: Handbook of knowledge representation, chap. 3, pp. 135–179. Elsevier (2008)
3. Bellodi, E., Lamma, E., Riguzzi, F., Albani, S.: A distribution semantics for probabilistic ontologies. In: International Workshop on Uncertainty Reasoning for the Semantic Web. No. 778 in CEUR Workshop Proceedings, Sun SITE Central Europe (2011)
4. Bryant, R.E.: Graph-based algorithms for boolean function manipulation. IEEE Trans. on Comput. 35(8), 677–691 (1986)
5. Carvalho, R.N., Laskey, K.B., Costa, P.C.G.: PR-OWL 2.0 - bridging the gap to OWL semantics. In: International Workshop on Uncertainty Reasoning for the Semantic Web (2010)
6. Costa, P.C.G., Laskey, K.B., Laskey, K.J.: PR-OWL: A Bayesian ontology language for the semantic web. In: International Workshop on Uncertainty Reasoning for the Semantic Web. LNCS, vol. 5327, pp. 88–107. Springer (2008)
7. d'Amato, C., Fanizzi, N., Lukasiewicz, T.: Tractable reasoning with Bayesian description logics. In: International Conference on Scalable Uncertainty Management. LNCS, vol. 5291, pp. 146–159. Springer (2008)
8. 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)
9. Ding, Z., Peng, Y.: A probabilistic extension to ontology language OWL. In: Hawaii International Conference On System Sciences. IEEE (2004)
10. 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)
11. Gottlob, G., Lukasiewicz, T., Simari, G.I.: Conjunctive query answering in probabilistic Datalog+/- ontologies. In: International Conference on Web Reasoning and Rule Systems. LNCS, vol. 6902, pp. 77–92. Springer (2011)
12. Halaschek-Wiener, C., Kalyanpur, A., Parsia, B.: Extending tableau tracing for ABox updates. Tech. rep., University of Maryland (2006)
13. Heinsohn, J.: Probabilistic description logics. In: Conference on Uncertainty in Artificial Intelligence. pp. 311–318. Morgan Kaufmann (1994)
14. Jaeger, M.: Probabilistic reasoning in terminological logics. In: International Conference on Principles of Knowledge Representation and Reasoning. pp. 305–316 (1994)
15. Kalyanpur, A., Parsia, B., Cuenca-Grau, B., Sirin, E.: Tableaux tracing in SHOIN. Tech. Rep. 2005-66, University of Maryland (2005)

16. Kalyanpur, A., Parsia, B., Horridge, M., Sirin, E.: Finding all justifications of OWL DL entailments. In: International Semantic Web Conference. LNCS, vol. 4825, pp. 267–280. Springer (2007)
17. Kalyanpur, A., Parsia, B., Sirin, E., Hendler, J.A.: Debugging unsatisfiable classes in OWL ontologies. J. Web Sem. 3(4), 268–293 (2005)
18. Kimmig, A., Demoen, B., Raedt, L.D., Costa, V.S., Rocha, R.: On the implementation of the probabilistic logic programming language problog. Theor. Prac. Log. Prog. 11(2-3), 235–262 (2011)
19. Klinov, P.: Pronto: A non-monotonic probabilistic description logic reasoner. In: European Semantic Web Conference. LNCS, vol. 5021, pp. 822–826. Springer (2008)
20. Koller, D., Levy, A.Y., Pfeffer, A.: P-CLASSIC: A tractable probablistic description logic. In: National Conference on Artificial Intelligence. pp. 390–397 (1997)
21. Laskey, K.B., 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)
22. Lukasiewicz, T.: Probabilistic default reasoning with conditional constraints. Ann. Math. Artif. Int. 34(1-3), 35–88 (2002)
23. Lukasiewicz, T.: Expressive probabilistic description logics. Artif. Int. 172(6-7), 852–883 (2008)
24. Lukasiewicz, T., Straccia, U.: Managing uncertainty and vagueness in description logics for the semantic web. J. Web Sem. 6(4), 291–308 (2008)
25. Nilsson, N.J.: Probabilistic logic. Artif. Intell. 28(1), 71–87 (1986)
26. 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)
27. Patel-Schneider, P, F., Horrocks, I., Bechhofer, S.: Tutorial on OWL (2003)
28. Poole, D.: Abducing through negation as failure: stable models within the independent choice logic. J. Log. Program. 44(1-3), 5–35 (2000)
29. Poole, D.: Probabilistic horn abduction and Bayesian networks. Artif. Intell. 64(1) (1993)
30. Predoiu, L., Stuckenschmidt, H.: Probabilistic extensions of semantic web languages - a survey. In: The Semantic Web for Knowledge and Data Management: Technologies and Practices. Idea Group Inc (2008)
31. Riguzzi, F.: Extended semantics and inference for the Independent Choice Logic. Log. J. IGPL 17(6), 589–629 (2009)
32. Riguzzi, F., Lamma, E., Bellodi, E., Zese, R.: Semantics and inference for probabilistic ontologies. In: Popularize Artificial Intelligence Workshop. pp. 41–46. No. 860 in CEUR Workshop Proceedings, Sun SITE Central Europe (2012)
33. Sato, T.: A statistical learning method for logic programs with distribution semantics. In: International Conference on Logic Programming. pp. 715–729. MIT Press (1995)
34. Sattler, U., Calvanese, D., Molitor, R.: Relationships with other formalisms. In: Description Logic Handbook. pp. 137–177 (2003)
35. Sirin, E., Parsia, B., Cuenca-Grau, B., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. J. Web Sem. 5(2), 51–53 (2007)
36. URW3-XG: Uncertainty reasoning for the World Wide Web, final report (2005)

Trust Evaluation through User Reputation and Provenance Analysis

Davide Ceolin, Paul Groth, Willem Robert van Hage,
Archana Nottamkandath, and Wan Fokkink
{d.ceolin,p.t.groth,w.r.van.hage,
a.nottamkandath,w.j.fokkink}@vu.nl

VU University, Amsterdam, The Netherlands

Abstract. Trust is a broad concept which, in many systems, is reduced to reputation estimation. However, reputation is just one way of determining trust. The estimation of trust can be tackled from other perspectives as well, including by looking at provenance. In this work, we look at the combination of reputation and provenance to determine trust values. Concretely, the first contribution of this paper is a standard procedure for computing reputation-based trust assessments. The second is a procedure for computing trust values based on provenance information, represented by means of the W3C standard model PROV. Finally, we demonstrate how merging the results of these two procedures can be beneficial for the reliability of the estimated trust value.
We evaluate our procedures and hypothesis by estimating and verifying the trustworthiness of the tags created within the *Waisda?* video tagging game, launched by the Netherlands Institute for Sound and Vision. Within *Waisda?*, tag trustworthiness is estimated on the basis of user consensus. Hence, we first provide a means to represent user consensus in terms of trust values, and then we predict the trustworthiness of tags based on reputation, provenance and a combination of the two. Through a quantitative analysis of the results, we demonstrate that using provenance information is beneficial for the accuracy of trust assessments.

Keywords: Trust, Provenance, Subjective Logic, Machine Learning, Uncertainty Reasoning, Tags

1 Introduction

From deciding the next book to read to selecting the best movie review, we often use the reputation of the author to ascertain the trust in the thing itself. Reputation is an important mechanism in our set of strategies to determine trust. However, we may base our assessment on a variety of other factors as well, including prior performance, a guarantee, or knowledge of how something was produced. Nevertheless, many systems, especially on the Web, choose to reduce trust to reputation estimation and analysis alone. In this work, we take a multifaceted approach. We look at trust assessment of Web data based on reputation, provenance (i.e., how data has been produced), and the combination of the two. We use the term "trust" for the trust in information resources and "reputation" for the trust in agents (see the work of Artz and Gil [1] for complete definitions).

We know that over the Web "anyone can say anything about any topic" [24], and this constitutes one of the strengths of the Semantic Web (and of the Web in general), since it brings democracy in it (everybody has the same right to contribute) and does not prevent a priori any possible useful contribution. However, this principle brings along trust concerns, since the variety of the contributors can affect both the quality and the trustworthiness of the data. On the other hand, the fact that the Semantic Web itself offers the means to, and is putting more effort in recording provenance information, is beneficial to solve this issue. Our contribution is therefore important for two reasons: first, we propose procedures for computing trust assessments of (Semantic) Web data, and some of these procedures have provenance information already available over the Web. Second, by showing that trust assessments based on combinations of reputation and provenance are more accurate than those based only on reputation, we show how a solution to trust issues can be found on the Web itself.

We first propose a procedure for computing reputation that uses basic evidential reasoning principles and is implemented by means of subjective logic opinions [13]. Secondly, we propose a procedure for computing trust assessments based on provenance information represented in the W3C PROV model [23]. Here, PROV plays a key role, both because of the availability of provenance data over the Web recorded by using this standard, and because of its role of interchange format: having modeled our procedure on PROV, then any other different input format can be easily treated after having mapped it to PROV. We implement this procedure by discretizing the trust values and applying support vector machine classification. Finally, we combine these two procedures in order to maximize the benefit of both. The procedures are evaluated on data provided by the *Waisda?*[8] tagging game[1], where users challenge each other in tagging videos. If the tags of two or more users regarding the same video are matched within a given time frame, they both get points. User consensus about tags correlates with tag trustworthiness: the more users agree on a given tag, the more likely it is that the tag is correct. We show how it is possible to predict tag consensus based on who created the tag, how it was created and a combination of the two. In particular, we show that a reputation-based prediction is not significantly different from a provenance-based prediction and, by combining the two, we obtain a small but statistically significant improvement in our predictions. We also show that reputation- and provenance-based assessments correlate.

The rest of the paper is organized as follows: Section 2 describes related work, Section 3 describes the dataset used for our evaluations, Section 4, 5, 6 introduce respectively the trust assessment procedures based on reputation, provenance and their combination, including example associated experiments. Section 7 provides final conclusions.

[1] A zip file containing the R and Python procedures used, together with the dataset, is retrievable at `http://d.pr/f/YXoS`

2 Related work

Trust is a widely explored topic within a variety of computer science areas. Here, we focus on those works directly touching upon the intersection of trust, provenance, Semantic Web and Web. We refer the reader to the work of Sabater and Sierra [21], Artz and Gil [1], and Golbeck [10] for comprehensive reviews about trust in respectively artificial intelligence, Semantic Web and Web. The first part of our work focuses on reputation estimation and is inspired by the works collected by Masum and Tovey [15]. Pantola et al. [16] present reputation systems that measure the overall reputation of the authors based respectively on the quality of their contribution and the "seriousness" of their ratings; Javanmardi et al. [12] measure reputation based on user edit patterns and statistics. Their approaches are similar to ours, but these contributions are particularly tailored for wikis. The second part of our work focuses on the usage of provenance information for estimating trust assessments. In their works, Bizer and Cyganiak [2], Hartig and Zhao [11] and Zaihrayeu et al. [27], use provenance and background information expressed as annotated or named graphs [4] to produce trust values. We do not make use of annotated or named graph, but we use provenance graphs as features for classifying the trustworthiness of artifacts. The same difference is valid also with respect to two works of Rajbhandari et al. [20,19], where they quantify the trustworthiness of scientific workflows and they evaluate it by means of probabilistic and fuzzy models. The use of provenance information for computing trust assessments has also been investigated in a previous work of ours [5] where we determined the trustworthiness of event descriptions based on provenance information by applying subjective logic [13] to provenance traces of event descriptions. In the current paper, we still represent trust values by means of subjective opinions, but trust assessments are made by means of support vector machines, eventually combined with reputations, again represented by means of subjective opinions. Finally, the procedure introduced in Section 4 is a generalization of the procedure that we implemented in a precedent work [6], where we evaluated the trustworthiness of tags of the Steve.Museum [22] artifact collection.

3 The *Waisda?* dataset

Waisda? is a video tagging gaming platform launched by the Netherlands Institute for Sound and Vision in collaboration with the public Dutch broadcaster KRO. The game's logic is simple: users watch video and tag the content. Whenever two or more players insert the same tag about the same video in the same time frame (10 sec., relative to the video), they are both rewarded. The number of matches for a tag is used as an estimate of its trustworthiness. When a tag which is not matched by others is not considered to be untrustworthy, because, for instance, it can refer to an element of the video not noticed so far by any user, or it can belong to a niche vocabulary, so it is not necessarily wrong. In the game, when counting matching tags, typos or synonymity are not taken into consideration.

We validate our procedures by using them to estimate the trustworthiness of tag entries produced within the game. Our total corpus contains 37850 tag entries corresponding to 115 tags randomly chosen. These tag entries correspond to about 9% of the total population. We have checked their representativity of the entire dataset. First, we compared the distribution of each relevant feature that we will use in Section 5 in our sample with the distribution of the same feature in the entire dataset. A 95% confidence level Chi-squared test [18] confirmed that the hour of the day and the day of the week distribute similarly in our sample and in the entire dataset. The typing duration distributions, instead, are significantly different according to a 95% confidence level Wilcoxon signed-rank test [26]. However, the mode of the two distributions are the same, and the mean differs only 0.1 sec. which, according to the KLM-GOMS model [3], corresponds, at most, to a keystroke. So we conclude that the used sample is representative of the entire data set. A second analysis showed that, by randomly selecting other sets of 115 tags, the corresponding tag entries are not statistically different from the sample that we used. We used 26495 tag entries (70%) as a training set, and the remaining 11355 (30%) as a test set.

4 Computing user reputation

Reputation is an abstraction of a user identity that quantifies his reliability as artifact author. Here, we use it to estimate the trustworthiness of the artifact.

4.1 Procedure

We present a generic procedure for computing the reputation of a user with respect to a given artifact produced by him or her.

```
proc reputation($user$, $artifact$) $\equiv$
    $evidence := evidence\_selection(user, artifact)$
    $weighted\_evidence := weigh\_evidence(user, artifact, evidence)$
    $reputation := aggregate\_evidence(weighted\_evidence)$
```

Evidence Selection Reputation is based on historical evidence, hence the first step is to gather all pieces of evidence regarding a given person and select those relevant for trust computation. Typical constraints include temporal (evidence is only considered within a particular time-frame) or semantics based (evidence is only considered when is semantically related to the given artifact). *evidence* is the set of all evidence regarding *user* about *artifact.*

```
proc evidence_selection($user$, $artifact$) $\equiv$
    for $i$ :=1 to $length(observations)$ do
        if $observations[i].user = user$ then $evidence.add(observation[i])$ fi
```

Evidence Weighing Given the set of evidence considered, we can decide if and how to weigh its elements, that is, whether to count all the pieces of evidence as equally important, or whether to consider some of them as more relevant. This step might be considered as overlapping with the previous one since they are both about weighing evidence: evidence selection gives a boolean weight, while here a fuzzy or probabilistic weight is given. However, keeping this division produces an efficiency gain, since it allows computation to be performed only on relevant items.

```
proc weigh_evidence($user$, $artifact$, $evidence$) $\equiv$
  for $i := 1$ to $length(evidence)$ do
      $weighted\_evidence.add(weigh(evidence[i], artifact))$
```

Aggregate evidence Once the pieces of evidence (or observations) have been selected and weighed, these are aggregated to provide a value for the user reputation that can be used for evaluation. We can apply several different aggregation functions, depending on the domain. Typical functions are: *count*, *sum*, *average*. Subjective logic [13], a probabilistic logic that we use in the application of this procedure, aggregates the observations in subjective opinions about artifacts being trustworthy based on the reputation of their authors are represented as follows: $\omega(b, d, u)$ where

$$b = \frac{p}{p+n+2} \quad d = \frac{n}{p+n+2} \quad u = \frac{2}{p+n+2}$$

where b, d and u indicate respectively how much we believe that the artifact is trustworthy, non-trustworthy, and how uncertain our opinion is. p and n are the amounts of positive and negative evidence respectively. Subjective opinions are equivalent to Beta probability distributions (Fig. 1), which range over the trust levels interval $[0 \ldots 1]$ and are shaped by the available evidence.

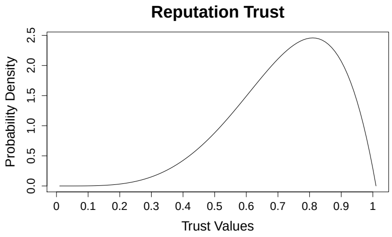


Fig. 1. Example of a Beta probability distribution aggregating 4 positive and 1 negative evidence. The most likely trust value is 0.8 (which is the ratio among the evidence). The variance of the distribution represents the uncertainty about the evaluation.

4.2 Application Evaluation

First, we convert the number of matches that each tag entry has into trust values:

tag selection For each tag inserted by the user, we select all the matching tags belonging to the same video. In other contexts, the number of matching tags can be substituted by the number of “likes”, “retweets”, etc..

tag entries weighing For each matching entry, we weigh the entry contribution on the time distance between the evaluated entry and the matched entry. The weight is determined from an exponential probability distribution, which is a "memory-less" probability distribution used to describe the time between events. If two entries are close in time, we consider it highly likely that they match. If they match but appear in distant temporal moments, then we presume they refer to different elements of the same video. Instead of choosing a threshold, we give a probabilistic weight to the matching entry. 85% of probability mass is assigned to tags inserted in a 10 sec. range.

tag entries aggregation In this step, we determine the trustworthiness of every tag. We aggregate the weighed evidence in a subjective opinion about the tag trustworthiness. We have at our disposal only positive evidence (the number of matching entries). The more evidence we have at disposal for the same tag entry, the less uncertain our estimate of its trustworthiness will be. Non-matched tag entries have equal probability to be correct or not;

We repeat this for each entry created by the user to compute his reputation.

user tag entries selection Select all the tag entries inserted by *user*.

user tag entries weighing Tag entries are weighed by the corresponding trust value previously computed. If an entry is not matched, it is considered as half positive (trust value 0.5) and half negative (1-0.5 = 0.5) evidence (it has 50% probability to be incorrect), as computed by means of subjective opinions. The other entries are also weighed according to their trust value. So, user reputation can either rise or decrease as we collect evidence.

user tag entries aggregation In turn, to compute the reputation of a user with respect to a given tag, we use all the previously computed evidence to build a subjective opinion about the user. This opinion represents the user reputation and can be summarized even more by the corresponding expected value or trust value (a particular average over the evidence count).

4.3 Results

We implement the abstract procedure for reputation computation and we evaluate its performance by measuring its ability to make use of the available evidence to compute the best possible trust assessment. Our evaluation does not focus on the ability to predict the exact trust value of the artifact by computing the user reputation, because these two values belong to a continuous space, and they are computed on a different basis. What we expect is that these two values hint at trustworthiness in a similar fashion: when a tag is trustworthy, then both trust value and reputation should be higher than a certain threshold and vice-versa. The validation, then, depends upon the choice of the threshold. We run the procedure with different thresholds as presented in Fig.3. Low thresholds correspond to low accuracy in our predictions. However, as the threshold increases, the accuracy of the prediction rises. Moreover, we should consider that: (1) It is preferable to obtain "false negatives" (reject correct tags) rather than "false positives" (accept wrong tags), so high thresholds are more likely to be chosen (e.g., see [9]), in order to reduce risks; (2) A Wilcoxon signed-rank test at 95% confidence level proved that the reputation-based estimates outperform blind guess estimates (having average probability of accuracy 50%). The average improvement is 8%, the maximum is 49%.

We previously adopted this same procedure to compute the trustworthiness of tags on the Steve.Museum artifacts [6]. Having to adapt it to the *Waisda?* case, we could understand the prominent features of it, hence this helped us in formulating the general procedure above.

5 Computing provenance-based trust

We focus on the "how" part of provenance, i.e., the modality of production of an artifact. (For simplicity, in the rest of the paper, we will use the word "provenance" to refer to the "how" part of it). We learn the relationships between PROV and trust values through machine learning algorithms. This procedure allows to process PROV data and, on the basis of previous trust evaluations, predict the trust level of artifacts. PROV is suitable for modeling the user behavior and provenance information in general.

5.1 Procedure

We present the procedure for computing trust estimates based on provenance.

```
proc provenance_prediction(artifact_provenance, artifact) ≡
  attribute_set := attribute_selection(artifact_provenance)
  attributes := attribute_extraction(attribute_set)
  trust_levels_aggregation
  classified_testset := classify(testset, trainingset)
```

attribute_selection Among all the provenance information, the first step of our procedure chooses the most significant ones: agent, processes, temporal annotations and input artifacts can all hint at the trustworthiness of the output artifact. This selection can lead to an optimization of the computation.

attribute_extraction Some attributes need to be manipulated to be used for our classifications, e.g., temporal attributes may be useful for our estimates because one particular date may be particularly prolific for the trustworthiness of artifacts. However, to ease the recognition of patterns within these provenance data, we extract the day of the week or the hour of the day of production, rather than the precise timestamp. In this way we can distinguish, e.g., between day and night hours (when the user might be less reliable). Similarly, we might refer to process types or patterns instead of specific process instances.

trust_level_aggregation To ease the learning process, we aggregate trust levels in n classes. Hence we apply classification algorithms operating on a nominal scale without compromising accuracy.

classification Machine learning algorithms (or any other kind of classification algorithm) can be adopted at this stage. The choice can be constrained either from the data or by other limitations.

5.2 Application evaluation

We apply the procedure to the tag entries from the *Waisda?* game as follows.

attribute selection and extraction The provenance information available in *Waisda?* is represented in Fig. 2, using the W3C PROV ontology. First, for each tag entry we extract: *typing duration*, *day of the week*, *hour of the day*, *game_id (to which the tag entry belongs)*, *video_id*. This the "how" provenance information at our disposal. Here we want to determine the trustworthiness of a tag given the modality with which it was produced, rather than the author reputation. Some videos may be easier to annotate than others, or, as we mentioned earlier, user reliability can decrease during the night. For similar reasons we use all the other available features.

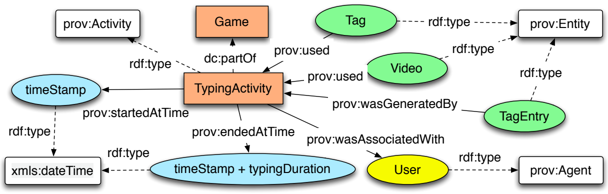


Fig. 2. Graph representation of the provenance information about each tag entry.

trust level classes computation In our procedure we are not interested in predicting the exact trust value of a tag entry. Rather we want to predict the range of trust within which the entry locates. Given the range of trust values $[0 \dots 1]$, we split it into 20 classes of length 0.5: from $[0 \dots 0.5]$ to $[9.5 \dots 10]$. This allows us to increase the accuracy of our classification algorithm without compromising the accuracy of the predicted value or the computation cost. The values in each class were approximated by the middle value of the class itself. For instance, the class $[0.5 \dots 0.55]$ are approximated as 0.525.

regression/classification algorithm We use a regression algorithm to predict the trustworthiness of the tags. Having at our disposal five different features (in principle, we might have more), and given that we are not interested in predicting the "right" trust value, but the class of trustworthiness, we adopt the "regression-by-discretization" approach [14], that allows us to use Support Vector Machines algorithm (SVM) [7] to classify our data. The training set is composed by 70% of our data, and then we predict the trust level of the test set. We used the SVM version implemented in the e1071 R library [25]. In the future, we will consider alternative learning techniques.

5.3 Results

The accuracy of our predictions depends on the choice of the thresholds. If we look at the ability to predict the right (class of) trust values, then the accuracy is of about 32% (which still is twice as much as the average result that we would have with a blind guess), but it is more relevant to focus on the ability to predict the trustworthiness of tags within some range, rather than the exact trust value. Depending on the choice of the threshold, the accuracy in this case varies in the range of 40% - 90%, as we can see in Fig. 3. For thresholds higher than 0.85 (the most likely choices), the accuracy is at least 70%. We also compared the provenance-based estimates with the reputation-based ones, with a 95% confidence level Wilcoxon signed-rank test that proved that the estimates of the two algorithms is not statistically different. *For the Waisda? case study, reputation- and provenance-based estimates are equivalent: when reputation is not available or it is not possible to compute it, we can substitute it with provenance-based estimates.* This is particularly important, since the ever growing availability of PROV data will increase the ease for computing less uncertain trust values.

If we apply the "regression-by-discretization" approach for making provenance-based assessments, then we approximate our trust values. This is not necessary with the reputation approach. Had we applied the same approximation to the reputations as well, then provenance-based trust would have performed better, as proven with a 95% confidence level Wilcoxon signed-ranked test, because reputation can rely only on evidence regarding the user, while provenance-based models can rely on larger data sets. Anyway, we have no need to discretize the reputation and, in general, we prefer it for its lightweight computational burden.

6 Combining reputation and provenance-based trust

We combine reputation- and provenance-based estimates to improve our predictions. If a certain user has been reliable so far, we can reasonably expect him/her to behave similarly in the near future. So we use reputation and we also constantly update it, to reduce the risk on relying on over-optimistic assumptions (if a user that showed to be reliable once, will maintain his/her status forever). However, reputation has an important limitation. To be reliable, a reputation has to be based on a large amount of evidence, which is not always possible. So, both in case the reputation is uncertain, or in case the user is anonymous, other sources of information should be used in order to correctly predict a trust value. The trust estimate based on provenance information, as described in Section 5, is based on behavioral patterns which have a high probability to be shared among several users. Hence, if a reputation is not reliable enough, we substitute it with the provenance-based prediction.

6.1 Procedure

The algorithm looks like the following:

```
proc provenance_prediction($user$, $artifact$) $\equiv$
  $q\_ev = evaluate\_user\_evidence(user, artifact)$
  if $q\_ev > min\_evidence$ then $predict\_reputation$ else $predict\_provenance$ fi
```

evaluate_user_evidence This function quantifies the evidence. Some implementation examples: (1) *count*; (2) compute a subjective opinion and check if the uncertainty is low enough. As future work we plan to investigate how to automatically determine *q_ev* and *evaluate_user_evidence*.

6.2 Application evaluation

We adopted the predictions obtained with each of the two previous procedures. The results are combined as follows: if the reputation is based on a minimum number of observations, then we use it, otherwise we substitute it with the prediction based on provenance. We run this procedure with different values for both the threshold and the minimum number of observations per reputation. We instantiate the *evaluate_user_evidence(user,artifact)* function as a *count* function of the evidence of *user* with respect to a given *tag*.

6.3 Results

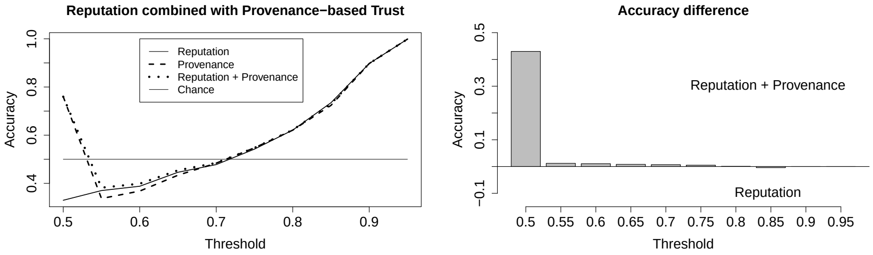


Fig. 3. Absolute and relative (Reputation+Provenance vs. Reputation) accuracy. The gap between the prediction (provenance-based) and the real value of some items explains the shape between 0.5 and 0.55: only very low or high thresholds cover it.

The performance of this algorithm depends both on the choice of the threshold for the decision and on the number of pieces of evidence that make a reputation reliable, so we ran the algorithm with several combinations of these two parameters (Fig. 3). The results converge immediately, after having set the minimum number of observations at two. We compared these results with those obtained before. Two Wilcoxon signed-rank tests (at 90% and 95% confidence level with respect to respectively reputation and provenance-based assessments) showed that *the procedure which combines reputation and provenance evaluations* *in this case performs better than each of them applied alone.* The improvement is, on average, about 5%. Despite the fact that most of the improvement regards the lower thresholds, which are less likely to be chosen (as we saw in Section 4), even at 0.85 threshold there is a 0.5% improvement. Moreover, we would like to stress how the combination of the two procedures performs better than (in a few cases, equal to) each of them applied alone, regardless of the threshold chosen.

Combining the two procedures allows us to go beyond the limitation of reputation-based approaches. Substituting estimates based on poorly reliable reputations with provenance-based ones improves our results without significantly increasing our risks, since we have previously proven that the two estimates are (on average) equivalent. Hence, when a user is new in a system (and so his/her history is limited) or anonymous, we can refer to the provenance-based estimate to determine the trustworthiness of his/her work, without running higher risks. This improvement is at least partly due to the existing correlation between the reputation and provenance-based trust assessments. A little positive correlation (0.16) has been proved by a Pearson's correlation test [17] with a confidence level of 99%. Thanks to this, we can safely enough substitute uncertain reputations with the corresponding provenance-based assessments. This explains also the similarity among the results shown in Fig. 3.

7 Conclusion

This paper explores two important components of trust assessments: reputation and provenance information. We propose and evaluate a procedure for computing reputation and one for computing trust assessments based on provenance information represented with the W3C standard PROV. We show that it is important to use reputation estimation for trust assessment, because it is simple, computationally light and accurate. We also show the potential of provenance-based trust assessments: these can be at least as accurate as reputation-based ones and can be used to overcome the limitations of a reputation based approach. In *Waisda?* the combination of the two methods revealed to be more powerful than each of the two alone. In the future we will investigate the possibility of automatically extracting provenance patterns usable for trust assessment, to automate, optimize and adapt the process to other case studies. We will also focus on the use of trust assessments as a basis for information retrieval.

Acknowledgements We thank the Netherlands Institute for Sound and Vision for launching and guiding the *Waisda?* project, and our colleagues Michiel, Riste and Valentina for their support. This research was partially supported by the PrestoPRIME project, in the EC ICT FP7 program, and by the Data2Semantics and SEALINC Media projects in the Dutch national program COMMIT.

References

1. D. Artz and Y. Gil. A survey of trust in computer science and the semantic web. *Journal of Semantic Web*, 5(2):131–197, 2007.

2. C. Bizer and R. Cyganiak. Quality-driven information filtering using the WIQA policy framework. *Journal Web Semantics*, 7(1):1–10, Jan. 2009.
3. S. Card, T. P. Moran, and A. Newell. *The Psychology of Human Computer Interaction.* Lawrence Erlbaum Associates, 1983.
4. J. Carroll, C. Bizer, P. Hayes, and P. Stickler. Named graphs, provenance and trust. In *WWW '05*, pages 613–622. ACM, 2005.
5. D. Ceolin, P. Groth, and W. R. V. Hage. Calculating the trust of event descriptions using provenance. In *SWPM 2010.* CEUR-WS, 2010.
6. D. Ceolin, A. Nottamkandath, and W. Fokkink. Automated evaluation of annotators for museum collections using subjective logic. In *Trust Management VI*, IFIP AICT 374. Springer, May 2012.
7. C. Cortes and V. Vapnik. Support-vector networks. *M. Learn.*, 20:273–297, 1995.
8. D. Gambetta. *Can We Trust Trust?* Basil Blackwell, 1988.
9. J. Golbeck. Trust on the World Wide Web: A Survey. *Foundations and Trends in Web Science*, 1(2):131–197, 2006.
10. O. Hartig and J. Zhao. Using web data provenance for quality assessment. In *SWPM 2009.* CEUR-WS, 2009.
11. S. Javanmardi, C. Lopes, and P. Baldi. Modeling user reputation in wikis. *Stat. Anal. Data Min.*, 3(2):126–139, Apr. 2010.
12. A. Jøsang. A logic for uncertain probabilities. *Intl. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–212, 2001.
13. I. Kononenko. Naive bayesian classifier and continuous attributes. *Informatica*, 16(1):1–8, 1992.
14. H. Masum and M. Tovey, editors. *The reputation society.* MIT Press, Feb. 2012.
15. Netherlands Inst. for Sound and Vision. Waisda? `http://waisda.nl`, June 2012.
16. A. V. Pantola, S. Pancho-Festin, and F. Salvador. Rating the raters: a reputation system for wiki-like domains. In *SIN '10*, pages 71–80. ACM, 2010.
17. K. Pearson. Mathematical Contributions to the Theory of Evolution. In *Proceedings of the Royal Society of London*, pages 489–498, 1896.
18. K. Pearson. On the Criterion that a Given System of Deviations from the Probable in the Case of Correlated System of Variables is such that it can be Reasonably Supposed to have Arisen from Random Sampling. *Phil. Mag.*, 50:157–175, 1900.
19. S. Rajbhandari, O. F. Rana, and I. Wootten. A fuzzy model for calculating workflow trust using provenance data. In *MG'08.* ACM, 2008.
20. S. Rajbhandari, I. Wootten, A. S. Ali, and O. F. Rana. Evaluating Provenance-based Trust for Scientific Workflows. In *CCGRID 06*, volume 1, pages 365–372. IEEE, 2006.
21. J. Sabater and C. Sierra. Review on computational trust and reputation models. *Artificial Intelligence Review*, 24:33–60, 2005.
22. U.S. Institute of Museum and Library Service. Steve Social Tagging Project. `http://www.steve.museum/`, June 2012.
23. W3C. PROV-O. `http://www.w3.org/TR/prov-o//`, June 2012.
24. W3C. Resource description framework (rdf): Concepts and abstract data model. `www.w3.org/TR/2002/WD-rdf-concepts-20020829/`, June 2012.
25. T. Wien. e1071: Misc functions of the department of statistics (e1071). `http://cran.r-project.org/web/packages/e1071/`, June 2012.
26. F. Wilcoxon. Individual comparisons by ranking methods. *Biom. Bull.*, 1:80–83, 1945.
27. I. Zaihrayeu, P. da Silva, and D. L. McGuinness. IWTrust: Improving User Trust in Answers from the Web. In *iTrust2005*, volume 3477, pages 384–392. Springer, 2005.

Subjective Logic Extensions for the Semantic Web

Davide Ceolin, Archana Nottamkandath, and Wan Fokkink
{d.ceolin, a.nottamkandath, w.j.fokkink}@vu.nl

VU University, Amsterdam, The Netherlands

Abstract. Subjective logic is a powerful probabilistic logic which is useful to handle data in case of uncertainty. Subjective logic and the Semantic Web can mutually benefit from each other, since subjective logic is useful to handle the inner noisiness of the Semantic Web data, while the Semantic Web offers a mean to obtain evidence useful for performing evidential reasoning based on subjective logic. In this paper we propose three extensions and applications of subjective logic in the Semantic Web, namely: the use of semantic similarity measures for weighing subjective opinions, a way for accounting for partial observations, and the new concept of "open world opinion", i.e. subjective opinions based on Dirichlet Processes, which extend multinomial opinions. For each of these extensions, we provide examples and applications to prove their validity.

Keywords: Subjective Logic, Semantic Similarity, Dirichlet Process, Partial Observations

1 Introduction

Subjective logic [7] is a probabilistic logic widely adopted in the trust management domain, based on evidential reasoning and statistical principles. This logic focuses on the representation and the reasoning on assertions of which truth value is not fully determined, but estimated on the basis of the observed evidence. The logic comes with a variety of operators that allow to combine such assertions and to derive the truth values of the consequences.

Subjective logic is well-suited for the management of uncertainty within the Semantic Web. For instance, the incremental access to these data (as a consequence of crawling) can give rise to uncertainty issues which can be dealt with using this logic. Furthermore, the fact that the fulcrum of this logic is the concept of "subjective opinion" (which represent an assertion, its corresponding evidence and the source of this evidence), allows to correctly represent how the estimated truth value of an assertion is bound to the source of the corresponding evidence and allows to easily keep lightweight provenance information. Finally, evidential reasoning allows to limit the typical noisiness of Semantic Web data. On the other hand, we also believe that the Semantic Web can be beneficial to this logic, as an immeasurably important source of information: since the truth value of assertions is based on availability of observations, the more data is available (hopefully of high enough quality), the closer we can get to the correct truth value for our assertions. We believe that this mutual relationship can be improved. This paper proposes extensions and applications of subjective logic that aim at the Semantic Web.

The rest of the paper is organized as follows: Section 2 describes related work, Section 3 proposes a combination of subjective logic and semantic similarity measures, Section 4 introduces a method for dealing with partial observations of evidence, Section 5 introduces the concept of Open World Opinion. Section 6 provides a final discussion about the work presented.

2 Related work

The development of subjective logic's operators has been investigated. Remarkably, the averaging and cumulative fusion [8,9] and the discounting [11] operators are among the most generic and useful operators for this logic. These operators provide the foundations for the work proposed in this paper. The connections among subjective logic and the (Semantic) Web are increasing. Ceolin et al. [4] adopt this logic for computing trust values of annotations provided by experts, using DBpedia and other Web sources as evidence. Unlike this work, they do not use semantic similarity measures. Ceolin et al. [3] and Bellenger et al. [1] provide applications of the combination of evidential reasoning with semantic similarity measures and Semantic Web technologies. In the current paper we provide the theoretical foundations for this kind of approaches, and we generalize them. Sensoy et al. [15] use semantic similarity in combination with subjective logic to import knowledge from one context to another. They use the semantic similarity measure to compute a prior value for the imported data, while we use it to weigh all the available evidence. Kaplan et al. [12] focus on the exploration of uncertain partial observations used for building subjective opinions. Unlike their work, we restrict our focus on partial observations of Web-like data and evaluations, which comprise the number of "likes", links and other similar indicators related to a given Web item. The weighing and discounting based on semantic similarity measures can resemble the work of Jøsang et al. [8], although the additional information that we include in our reasoning (which is semantic similarity) is related only to the frame of discernment in subjective logic, and not to the belief assignment function.

3 Combining Subjective Logic with Semantic Similarity

3.1 Preliminaries

Subjective Logic In subjective logic, so-called "subjective opinions" express the belief that source x owns with respect to the value of assertion y chosen among the elements of the set Θ ("frame of discernment"). The belief is assigned to the elements of the set $X = 2^{\Theta} \setminus \Theta$ ("frame"), according to the evidence. In symbols, this is represented as $\omega(b, d, u, a)$ when $|\Theta| = 2$ (binomial opinion) or as $\omega(\overrightarrow{b}, u, \overrightarrow{a})$ when $|\Theta| > 2$ (multinomial opinion). The positive and negative evidence is represented as p and n respectively. The belief (b), disbelief (d), uncertainty (u), and a priori values (a) for binomial opinions are computed as:

$$b = \frac{p}{p+n+2} \quad d = \frac{n}{p+n+2} \quad u = \frac{2}{p+n+2} \quad a = \frac{1}{2} \tag{1}$$

A subjective opinion is equivalent to a Beta probability distribution (binomial opinion) or to a Dirichlet distribution (multinomial opinion). The expected value (E) for the Beta distribution is computed as in equation (2).

Opinions are computed based on contexts. For example source x provides an observation about assertion y in context c (e.g. about an agent's expertise). The trustworthiness of assertion y in context c, represented as $t(x, y : c)$, is the expected value of the Beta distribution corresponding to the opinion and computed as:

$$E = t(x, y : c) = b + a \cdot u \tag{2}$$

Base Rate Discounting Operator in Subjective Logic In subjective logic, the base rate sensitive discounting of opinion of source B on y by opinion of source A on B ω_B^A, $\omega_y^B = (b_y^B, d_y^B, u_y^B, a_y^B)$ by opinion $\omega_B^A = (b_B^A, d_B^A, u_B^A, a_B^A)$ of source A produces transitive belief $\omega_y^{A:B} = (b_y^{A:B}, d_y^{A:B}, u_y^{A:B}, a_y^{A:B})$ where

$$\begin{array}{rlrl} b_y^{A:B} &= E(\omega_B^A) b_y^B & d_y^{A:B} &= E(\omega_B^A) d_y^B \\ u_y^{A:B} &= 1 - E(\omega_B^A)(b_y^B + d_y^B) & a_y^{A:B} &= a_y^B \end{array} \tag{3}$$

Wu & Palmer Semantic Similarity Measure Many semantic similarity measures have been developed (see the work of Budanitsky and Hirst [2]). We focus on those computed from *WordNet*. *WordNet* groups words into sets of synonyms called synsets that describes semantic relationships between them. It is a directed and acyclic graph with each vertex v, an integer that represents a synset, and each directed edge from v to w represents that w is a hypernym of v. We focus on the Wu & Palmer metric [18], which calculates semantic relatedness in a deterministic way by considering the depths between two synsets in the WordNet taxonomies, along with the depth of the Least Common Subsumer (lcs) as follows:

$$score(s1, s2) = \frac{2 \cdot depth(lcs)}{depth(s1) + depth(s2)} \tag{4}$$

This means that $score \in]0 \ldots 1]$. For deriving the opinions about a concept where no evidence is available, we incorporate *score*, which represents the semantic similarity ($sim(c, c')$) in our trust assessment, where c and c' are concepts belonging to synset $s1$ and $s2$ respectively which represent two contexts.

3.2 Using Semantic Similarity Measures within Subjective Logic

Deriving Opinion about a New or Unknown Context Since we compute opinions based on contexts, it is possible that evidence required to compute the opinion for a particular context is unavailable. For example, suppose that source x owns observations about an assertion in a certain context (e.g. the expertise of an agent about tulips), but needs to evaluate them in a new context (e.g. the agent's expertise about sunflowers), of which it owns no observations. The semantic similarity measure between two contexts, $sim(c, c')$ can be used for obtaining the opinion about an agent y on an unknown or new context through two different methods. In order to derive an opinion about a new or unknown context we can use either the weighing (on the evidence) or the discounting operation (on the opinion) and both the approaches are described below. We will show that the discounting and the weighing are theoretically but not statistically different.

Weighing the Evidence We weigh the positive and negative evidence belonging to a certain context (e.g. *Tulips*) on the corresponding semantic similarity to the new context (e.g. *Sunflowers*), *sim(Tulips, Sunflowers)*. We then perform this for all the contexts for which source x has already provided an opinion, $\forall c' \in C$, by weighing all the positive (p) and negative (n) evidence of c' with the similarity measure $sim(c, c')$ to obtain an opinion about y in c (see the work of Ceolin et al. [3]).

Discounting the Opinion In the second approach, every opinion source x has about other related contexts c', where $c' \in C$ is discounted with the corresponding semantic similarity measure $sim(c, c')$ using the Discounting operator in subjective logic. The discounted opinions are then aggregated to form the final opinion of x about y in the new context c.

Discounting Operator and Semantic Similarity Subjective logic offers a variety of operators for "discounting", i.e. for smoothing opinions given by third parties, provided that we have at disposal an opinion about the source itself. "Smoothing" is meant as reducing the belief provided by the third party, depending on the opinion on the source (the worse the opinion, the higher the reduction). Moreover, since the components of the opinion always sum to one, reducing the belief implies an increase of (one) of the other components: hence there exists a discounting operator favoring uncertainty and one favoring disbelief. Finally, there exists a discounting operator that makes use of the expected value E of the opinion. Following this line of thought, we can use the semantic similarity as a discount factor for opinions imported from contexts related to the one of interest, in case of a lack of opinions in it, to handle possible variations in the validity of the statements due to the change of context.

Choosing the Appropriate Discounting Operator We need to choose the appropriate discounting operator that allows us to use the semantic similarity value as a discounting factor for opinions. The disbelief favoring discounting is an operator that is employed whenever one believes that the source considered might be malicious. This is not our case, since the discounting is used to import opinions own by ourselves but computed in different contexts than the one of interest. Hence we do not make use of the disbelief favoring operator.

In principle, we would have no specific reason to choose one between the uncertainty favoring discounting and the base rate discounting. Basically, having that only rarely the belief (and hence the expected value) is equal to 1, the two discounting operators decrease the belief of the provided opinion, one by multiplying it by the belief in the source, the other one by the expected value of the opinion about the source. In practice, we will see that, thanks to Theorem 1 these two operators are equivalent in this context.

Theorem 1 (Semantic Relatedness Measure is a Dogmatic Opinion). *Let $sim(c, c')$ be the semantic similarity between two contexts c and c' obtained by computing the semantic distance between the contexts in a graph through deterministic measurements (e.g. [18]). Then, $\forall$ $sim(c, c') \in [0,1]$, $\omega_{c=c'}^{measure} = (b_{c=c'}^{measure}, d_{c=c'}^{measure}, u_{c=c'}^{measure}, a_{c=c'}^{measure})$ is equivalent to a dogmatic opinion in subjective logic.*

Proof. A binomial opinion is a dogmatic opinion if the value of *uncertainty is 0.* The semantic similarity measure can be represented as an opinion about the similarity of two contexts c and c'. However, since we restrict our focus on *WordNet*-based measures, the similarity is inferred by graph measurements, and not by probabilistic means. This means that, according to the source, this is a “dogmatic” opinion, since it does not provide any indication of uncertainty: $u_{c=c'}^{measure} = 0$. The opinion is not based on evidence observation, rather on actual deterministic measurements.

$$E(\omega_{c=c'}^{measure}) = b_{c=c'}^{measure} + u_{c=c'}^{measure} \cdot a = sim(c, c') \tag{5}$$

where *measure* indicates the procedure used to obtain the semantic distance, e.g. Wu and Palmer Measure. The values of belief and disbelief are obtained as:

$$b_{c=c'}^{measure} = sim(c, c') \qquad d_{c=c'}^{measure} = 1 - b_{c=c'}^{measure} \quad \square \tag{6}$$

Corollary 1 (Discounting an Opinion with a Dogmatic Opinion). *Let A be a source who has an opinion about y in context c' expressed as $\omega_{y:c'}^{A} = (b_{y:c'}^{A}, d_{y:c'}^{A}, u_{y:c'}^{A}, a_{y:c'}^{A})$ and let the semantic similarity between the contexts c and c' be represented as a dogmatic opinion $\omega_{c=c'}^{measure} = (b_{c=c'}^{measure}, d_{c=c'}^{measure}, 0, a_{c=c'}^{c'})$. Since, the source A does not have any prior opinion about the context c, we derive the opinion of A about c represented as $\omega_{c}^{A:c'} = (b_{c}^{A:c'}, d_{c}^{A:c'}, u_{c}^{A:c'}, a_{c}^{A:c'})$ using the base rate discounting operator on the dogmatic opinion.*

$$\begin{array}{cc} a_y^{A:B} = a_y^B & b_y^{A:B} = sim(c, c') \cdot b_y^B \\ u_y^{A:B} = 1 - sim(c, c') \cdot (b_y^B + d_y^B) & d_y^{A:B} = sim(c, c') \cdot d_y^B \end{array} \tag{7}$$

Definition 1 (Weighing Operator). *Let C be the set of contexts c' of which a source A has an opinion derived from the positive and negative evidence in the* *past. Let c be a new context for which A has no opinion yet. We can derive the opinion of A about facts in c, by weighing the relevant evidences in set C with the semantic similarity measure $sim(c, c')$ $\forall c' \in C$. The belief, disbelief, uncertainty and a priori obtained through the weighing operation are expressed below.*

$$\begin{array}{ll} b_c^A = \frac{sim(c,c')\cdot p_{c'}^A}{sim(c,c')(p_{c'}^A+n_{c'}^A)+2} & d_c^A = \frac{sim(c,c')\cdot n_{c'}^A}{sim(c,c')(p_{c'}^A+n_{c'}^A)+2} \\ u_c^A = 1 - \frac{sim(c,c')\cdot(p_{c'}^A+n_{c'}^A)}{sim(c,c')(p_{c'}^A+n_{c'}^A)+2} & a_c^A = a_{c'}^A \end{array} \tag{8}$$

Theorem 2 (Approximation of the Weighing and Discounting Operators). *Let $\omega_{y:c}^{A:c'} = (b_{y:c}^{A:c'}, d_{y:c}^{A:c'}, u_{y:c}^{A:c'}, a_{y:c}^{A:c'})$ be a discounted opinion which source A has about y in a new or unknown context c, derived by discounting A's opinion on known contexts c' $\in$ C represented as $\omega_{c'}^A = (b_{c'}^A, d_{c'}^A, u_{c'}^A, a_{c'}^A)$ with the corresponding dogmatic opinions (e.g. sim(c,c')). Let source A also obtain an opinion about the unknown context c based on the evidence available from the earlier contexts c', by weighing the evidence (positive and negative) with semantic similarity between c and c', sim(c,c') $\forall c' \in C$. Then the difference between the results from the weighing and from the discount operator in subjective logic are statistically insignificant.*

Proof. We substitute the values of belief, disbelief, uncertainty values in equation (9) for Base Rate Discounting with the values from equation (1) and expectation value from equation (5). We obtain the new value of the discounted base rate opinion as follows:

$$\begin{array}{ll} b_c^{A:c'} = \frac{sim(c,c')\cdot p_{c'}^A}{(p_{c'}^A+n_{c'}^A+2)} & d_c^{A:c'} = \frac{sim(c,c')\cdot n_{c'}^A}{(p_{c'}^A+n_{c'}^A+2)} \\ u_c^{A:c'} = 1 - \frac{sim(c,c')\cdot(p_{c'}^A+n_{c'}^A)}{(p_{c'}^A+n_{c'}^A+2)} & a_c^{A:c'} = a_{c'}^A \end{array} \tag{9}$$

Equation (9) and (8) are pretty similar, except for the $sim(c, c').(p_{c'}^A + n_{c'}^A)$ factor in the weighing operator. In the following section we use a 95% t-student and Wilcoxon signed-rank statistical test to prove that the difference due to that factor is not statistically significant for large values of $sim(c, c')$ (at least 0.5).

3.3 Evaluations

We prove statistically the similarity between the weighing and the discounting.[1]

First Validation: Discounting and Weighing in a Real-Life Case

Steve Social Tagging Project Dataset For the purpose of our evaluations, we use the "Steve Social Tagging Project" [16] data (in particular, the "Researching social tagging and folksonomy in the ArtMuseum"), which is a collaboration of museum professionals and others aimed at enhancing social

[1] Complete results are available at `http://tinyurl.com/bp43k5d`

tagging. In our experiments, we used a sample of tags which the users of the system provided for the 1784 images of the museum available online. Most of the tags were evaluated by the museum professionals to assess their trustworthiness. We used only the evaluated tags for our experiments. The tags can be single words or a string of words provided by the user regarding any objective aspect of the image displayed to them for the tagging.

Gathering Evidence for Evaluation We select a set of tags highly semantically related, by using a Web-based *WordNet* interface [14]. We then gather the list of users who provided the tags regarding the chosen words and count the number of positive and the negative evidence.

The opinions are calculated using two different methods. First by weighing the evidence with the semantic distance using equation (8) and the second method is by discounting the evidence with the semantic distance using equation (9). We consider the *Chinese-Asian* pair (semantic similarity 0.933) and the *Chinese-Buddhist* pair (semantic similarity 0.6667).

Results We employ the Student's t-test and the Wilcoxon signed-rank test to assess the statistical significance of the difference between two sample means. At 95% confidence level, both tests show a statistically significant difference between the two means. This difference, for the *Chinese-Asian* pair is 0.025, while for the *Chinese-Buddhist* pair is 0.11, thanks also to the high similarity (higher than 0.5) between the considered topics. Having removed the average difference from the results obtained from discounting (which, on average, are higher than those from weighing), both the tests assure that the results of the two methods distribute equally.

Second Validation: Discounting and Weighing on a Large Simulated Dataset In order to validate our hypothesis that weighing with semantic distance produces results that are highly similar to those obtained with the discounting operator of subjective logic, we perform the Student's t-test and the Wilcoxon signed-rank test on a larger dataset consisting of 1000 samples. For semantic distance values $sim(c, c') > 0.7$, the mean difference between the belief values obtained by weighing and discounting is 0.092. Thus with 95% confidence interval, both tests assure that both the weighing operator and the discounting operator produce similar results. The semantic similarity threshold $sim(c, c') > 0.7$ is relevant and reasonable, because it becomes more meaningful to compute opinions for a new context based on the opinions provided earlier for the most semantically related contexts, while also in case of lack of evidence for a given context, evidence about a very diverse context can not be much significant.

4 Partial Evidence Observation

The Web and the Semantic Web are pervaded of data that can be used as evidence for a given purpose, but that constitute partially positive/negative evidence for others. Think about the *Waisda?* tagging game [13]. Here, users challenge each other about video tagging. The more users insert the same tag about the same video within the same time frame, the more the tag is believed to be correct. Matching tags can be seen as positive observations for a specific tag to be correct. However, consider the orthogonal issue of the user reputation. User reputation is based on past behavior, hence on the trustworthiness of the tags previously inserted by him/her. Now, the trustworthiness of each tag is not deterministically computed, since it is roughly estimated from the number of matching tags for each tag inserted by the user. The expected value of each tag, which is less than one, can be considered as a partial observation of the trustworthiness of the tag itself. Vice-versa, the remainder can be seen as a negative partial observation. After having considered tag trustworthiness, one can use each evaluation as partial evidence with respect to the user reliability: no tag (or other kind of observation) is used as a fully positive or fully negative evidence, unless its correctness has been proven by an authority or by another source of validation. However, since only rarely the belief (and therefore, the expected value) is equal to one, these observations almost never count as a fully positive or fully negative evidence. We propose an operator for building opinions based on indirect observations, i.e., on observations used to build these opinions, each of which counts as an evidence.

Theorem 3 (Partial Evidence-Based Opinions). *Let p be a vector of positive observations (e.g. a list of "like" counts) about distinct facts related to a given subject s. Let l be the length of p. Let each opinion based on each entry of p have an a priori value of $\frac{1}{2}$. Then we can derive an opinion about the reliability of the subject in one of these two manners.*

- *By cumulating the expected values (counted as partial positive evidence) of each opinion based on each element of p:*

$$b = \frac{1}{l+2}\Sigma_{i=1}^{l}\frac{p_i+1}{p_i+2} \quad d = \frac{1}{l+2}\Sigma_{i=1}^{l}\frac{1}{p_i+2} \quad u = \frac{2}{l+2} \tag{10}$$

- *By averaging the expected values of the opinions computed on each of the elements of p:*

$$b = \frac{1}{l(l+2)}\Sigma_{i=1}^{l}\frac{p_i+1}{p_i+2} \quad d = \frac{1}{l(l+2)}\Sigma_{i=1}^{l}\frac{1}{p_i+2} \quad u = \frac{2}{l(l+2)} \tag{11}$$

Proof. The expected value of each opinion is computed as:

$$E = b + a \cdot u = \frac{p}{p+2} + \frac{1}{2} \cdot \frac{2}{p+2} = \frac{p+1}{p+2} \tag{12}$$

E is considered as partial positive evidence. Hence $1 - E$ is considered as partial negative evidence. Given that we have l pieces of partial evidence (because we have l distinct elements in $\overrightarrow{p}$), we compute the opinion about s following equations (1). Having that p (positive evidence of ω_s) is equal to $\frac{p'+1}{p'+2}$, we obtain equation (10). If we choose to average the evidence (and hence, the expected values) instead of cumulate them, what we obtain is $p = \frac{1}{l}\Sigma\frac{p_i+1}{p_i+2}$, hence $b = \frac{\frac{1}{l}\Sigma\frac{p_i+1}{p_i+2}}{l+2}$ and therefore we obtain equation (11). □

5 Dirichlet Process-Based Opinions: Open World Opinions

5.1 Preliminaries: Dirichlet Process

The Dirichlet Process [6] is a stochastic process representing a probability distribution whose domain is a random probability distribution. As we previously saw, the binomial and multinomial opinions are equivalent to Beta and Dirichlet probability distributions. The Dirichlet distribution represents an extension of the Beta distribution from a two-category situation to a situation where one among n possible categories has to be chosen. A Dirichlet process over a set S is a stochastic process whose sample path (i.e. an infinite-dimensional set of random variables drawn from the process) is a probability distribution on S. The finite dimensional distributions are from the Dirichlet distribution: if H is a finite measure on S, α is a positive real number and X is a sample path drawn from a Dirichlet process, written as

$$X \sim DP(\alpha, H) \tag{13}$$

then for any partition of S of cardinality m, say $\{B_i\}_{i=1}^m$

$$(X(B_1), \ldots, X(B_m)) \sim Dirichlet(\alpha H(B_1), \ldots, \alpha H(B_m)). \tag{14}$$

Moreover, given n draws from X, we can predict the next observation as:

$$obs_{n+1} = \begin{cases} x_i^*(i \in [1 \ldots k]) & \text{with probability } \frac{n(x_i^*)}{n+\alpha} \\ H & \text{with probability } \frac{\alpha}{n+\alpha} \end{cases} \tag{15}$$

where x_i^* is one of the k unique value among the observations gathered.

5.2 Open World Opinions

Having to deal with real data coming from the Web, which are accessed incrementally, the possibility to update the relative probabilities of possible outcomes might not be enough to deal with them. We may need to handle unknown categories of data which should be accounted and manageable anyway. Ceolin et al. [5] show how it is important to account for unseen categories, when dealing with Web data. Here, we propose a particular subjective opinion called “open world opinion” which accounts for partial knowledge about the possible outcomes. A subjective opinion resemble personal opinion provided by sources with respect to facts. Open world opinions represent the case when something about a given fact has been observed, but the evidence allow also for some other (not yet observed) outcome to be considered as plausible. With this extension we allow the frame of discernment to have infinite cardinality. In practice, open world opinions allow to represent situations when the unknown outcome of an event can be equal to one among a list of already observed values (proportionally to the amount of observations for each of them), but it is also possible that (and so some probability mass is reserved to) the outcome is different from what has been observed so far, and is drawn from an infinitely large domain.

Definition 2 (Open World Opinion). *Let: X be a frame of infinite cardinality, $\alpha \in \mathbb{R}^+$, k be the number of categories observed, $\overrightarrow{p}$ be the array of evidence per category, $\overrightarrow{B}$ be a belief function over X. We define the open world opinion ω_x:*

$$\omega_x(\overrightarrow{B}, U, H) \quad B_{x_i} = \frac{p_{x_i}}{\alpha + \Sigma_{x=1}^{k} p_{x_i}} \quad U = \frac{\alpha}{\alpha + \Sigma_{x=1}^{k} p_{x_i}} \quad 1 = U + \Sigma_{x_i} B_{x_i} \tag{16}$$

Definition 3 (Expected Value of Open World Opinion).
The expected value of an open world opinion is computed as follows:

$$E(p(x_i)|r, H) = \frac{p_{x_i} + H(x_i)}{\alpha + \Sigma p_{x_t}} = \frac{p_{x_i}}{\alpha + \Sigma p_{x_t}} \tag{17}$$

Theorem 4 (Equivalence between the Subjective and Dirichlet Process Notation). *Let $\omega_X^{bn} = (\overrightarrow{B}, U, H)$ be an opinion expressed in belief notation, and $\omega_X^{pn} = (E, \alpha, H)$ be an opinion expressed in probabilistic notation, both over the same frame X. ω_X^{bn} and ω_X^{pn} are equivalent when the following mappings holds:*

$$\begin{cases} B_{x_i} = \frac{p_{x_i}}{\alpha + \Sigma_{x=1}^{k} p_{x_i}} \\ \\ U = \frac{\alpha}{\alpha + \Sigma_{x=1}^{k} p_{x_i}} \end{cases} \Leftrightarrow \begin{cases} p_{x_i} = \frac{\alpha B_{x_i}}{U} \\ \\ 1 = U + \Sigma B_{x_i} \end{cases} \tag{18}$$

Proof. Each step of the Dirichlet Process can be seen as a Dirichlet Distribution. Hence the mapping between Dirichlet Distributions and multinomial opinions [9] holds also here. □

Theorem 5 (Mapping between Open World Opinion and Multinomial Opinion). *Let $\omega 1_y^x(\overrightarrow{B}, U, H)$ be an open world opinion and let $\omega 2_y^x(\overrightarrow{B}, U, \overrightarrow{a})$ be a multinomial opinion. Let X_2 and Θ_2 be the frame and the frame of discernment of $\omega 2_y^x$. Let $\{B_i\}_{i=1}^k$ be the result of the partition of $dom(H)$ such that:*

1. $|\Theta_2| = |\{B_i\}|$
2. $\bigcup \{B_i\}_{i=1}^k = dom(H)$
3. $\forall \{x_i\}[(\{x_i\} \in X_2 \wedge |\{x_i\}| = 1 \wedge x_i \in B_j) \Rightarrow \nexists x_{k \neq j} \in B_i]$
4. *$W = k$, where W is the non-informative constant of multinomial opinions*

Then there exists a function $D : Dom(H) \rightarrow \{B_i\}$ such that $D(\omega 1_y^x) = \omega 2_y^x$.

Proof. The equivalence between the discretized open world opinion and the multinomial opinion is proven by showing that:

- given equation (14), since the partition $\{B_i\}_{i=1}^k$ covers the entire $dom(H)$, then the partition distributes like the corresponding Dirichlet distribution;
- to each category of $\omega 2_y^x$ corresponds one and only one partition of $\{B_i\}$ as per item 2 of Theorem 5. □

In other words, open world opinions extend multinomial opinions by allowing the frame of discernment Θ to be infinite. However, by properly discretizing an open world opinion, what we obtain is an equivalent multinomial opinion.

5.3 Example: Using Open World Opinions

Piracy at sea is a well know problem. Every year, several ships are attacked, hijacked, etc. by pirates. The International Chamber of Commerce has created a repository of reports about ship attacks.[2] Van Hage et al. [17] have created an enriched Semantic Web version of such a repository, the Linked Open Piracy (LOP).[3] On the basis of LOP, one might think to be able to predict the frequency of attacks from one year based on the previously available data. However, a problem arises in this situation, since new attack types appear every year and this makes that frequencies vary. Ceolin et al. [5] have shown how the Dirichlet process can be employed to model such situations. Having the possibility to represent this information by means of an open world opinion adds the power of subjective logic to the Dirichlet process based representation. We can merge contributions from different sources, taking into account their reliability. Moreover, we can combine these facts with others in a logical way and then estimate the opinion (and the corresponding probability to be true) of the consequent facts. By using open world opinions, we can easily apply usual subjective operators to these data and easily represent them in a way that takes into account basic provenance information (e.g. data source) when applying fusing or discounting operators. For instance, if according to LOP, in Asia in 2010 we had 10 hijacking events and 10 attempted boarding, then we would represent this as:

$$\omega^{LOP}_{Attacks\ in\ Asia\ in\ 2010}([0.48, 0.48], 0.04, U(0,1))$$

If our opinion about LOP is that is a reliable but not fully accountable source (e.g. $\omega^{us}_{LOP}(0.8, 0.1, 0.1)$), then we can take this information into account by weighing the opinion given by LOP as follows:

$$\omega^{us}_{LOP}(0.8, 0.1, 0.1) \otimes \omega^{LOP}_{Attacks\ in\ Asia\ in\ 2010}([0.48, 0.48], 0.04, U(0,1)) =$$

$$= \omega^{us:LOP}_{Attacks\ in\ Asia\ in\ 2010}([0.384, 0.384], 0.232, U(0,1))$$

The resulting weighted opinion is more uncertain than the initial one, because, even though the two observed types are more likely to happen, the small uncertainty about the source reliability makes the other probabilities to rise.

A difference with respect to multinomial opinions arises in case of fusion, because the fusion operator requires that the *a priori values* have to be merged (averaged). Since the a priori values in the case of the open world opinions are represented by the distribution H (supposedly, H_1 and H_2 for two opinions to be merged). The averaging is still performed, and in this case the averaged distribution corresponds to the distribution Z having $E(Z) = b \cdot E(X_1) + a \cdot E(X_2)$ and $VAR(X) = b^2 \cdot (VAR(X_1)) + a^2 \cdot (VAR(X_2))$, where a, b are the two weights (e.g. u_1 and u_2 in case of cumulative fusion).

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

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

6 Discussion

We have shown the potential for employing subjective Logic as a basis for reasoning on Web and Semantic Web data. We have shown how it can be really powerful for handling uncertainty and how little extensions can help in improving the mutual benefit that Semantic Web and subjective logic obtain from cooperating together. Part of this work is based on previously mentioned practical applications that show the usefulness of it, and here we provide theoretical foundations for it. We foresee that other extensions will be possible as well like, for instance, the usage of hyperopinions [10] to handle subsumption reasoning about uncertain data.

References

1. A. Bellenger, S. Gatepaille, H. Abdulrab, and J.-P. Kotowicz. An Evidential Approach for Modeling and Reasoning on Uncertainty in Semantic Applications. In *URSW*, volume 778, pages 27–38. CEUR-WS.org, 2011.
2. A. Budanitsky and G. Hirst. Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*, 32(1):13–47, Mar. 2006.
3. D. Ceolin, A. Nottamkandath, and W. Fokkink. Automated Evaluation of Annotators for Museum Collections using Subjective Logic. In *IFIPTM*, pages 232–239. Springer, May 2012.
4. D. Ceolin, W. Van Hage, and W. Fokkink. A Trust Model to Estimate the Quality of Annotations Using the Web. In *WebSci10*. Online, 2010.
5. D. Ceolin, W. van Hage, and W. Fokkink. Estimating the Uncertainty of Categorical Web Data. In *URSW*, volume 778, pages 15–26. CEUR-WS.org, 2011.
6. T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *Annals of Statistics*, 2:209–230, 1973.
7. A. Jøsang. A Logic for Uncertain Probabilities. *Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–212, 2001.
8. A. Jøsang, M. Daniel, and P. Vannoorenberghe. Strategies for combining conflicting dogmatic beliefs. In *FUSION*. IEEE, 2003.
9. A. Jøsang, J. Diaz, and M. Rifqi. Cumulative and averaging fusion of beliefs. *Information Fusion*, 11(2):192–200, 2010.
10. A. Jøsang and R. Hankin. Interpretation and fusion of hyper opinions in subjective logic. In *FUSION*. IEEE, 2012.
11. A. Jøsang, S. Marsh, and S. Pope. Exploring Different Types of Trust Propagation. In *iTrust*, pages 179–192. Springer, 2006.
12. L. Kaplan, S. Chakraborty, and C. Bisdikian. Subjective Logic with Uncertain Partial Observations. In *FUSION*. IEEE, 2012.
13. Netherlands Inst. for Sound and Vision. Waisda? `http://wasida.nl`, Aug. 2012.
14. Princeton University. Wordnet::Similarity. `http://marimba.d.umn.edu/cgi-bin/similarity/similarity.cgi`, Feb. 2012.
15. M. Sensoy, J. Pan, A. Fokoue, M. Srivatsa, and F. Meneguzzi. Using Subjective Logic to Handle Uncertainty and Conflicts. In *TrustCom*. IEEE, 2012.
16. U.S Inst. of Museum and Library Service. Steve social tagging project, Jan. 2012.
17. W. van Hage, V. Malaisé, and M. van Erp. Linked Open Piracy: A story about e-Science, Linked Data, and statistics. *Journal of Data Semantics*, 2012.
18. Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *ACL '94*, pages 133–138. ACL, 1994.

A Graph Regularization Based Approach to Transductive Class-Membership Prediction

Pasquale Minervini, Claudia d'Amato, and Nicola Fanizzi

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

Abstract. Considering the increasing availability of structured machine processable knowledge in the context of the Semantic Web, only relying on purely deductive inference may be limiting. This work proposes a new method for similarity-based class-membership prediction in Description Logic knowledge bases. The underlying idea is based on the concept of *propagating* class-membership information among similar individuals; it is non-parametric in nature and characterised by interesting complexity properties, making it a potential candidate for large-scale transductive inference. We also evaluate its effectiveness with respect to other approaches based on inductive inference in SW literature.

1 Introduction

Standard Semantic Web (SW) reasoning services rely on purely deductive inference. However, this may be limiting, e.g. due to the complexity of reasoning tasks, availability and correctness of structured knowledge. Approximate deductive and inductive inference were discussed as a possible approach to try to overcome such limitations [19]. Various proposals to extend inductive inference methods towards SW formalisms have been discussed in SW literature: inductive methods can perform some sort of approximate and uncertain reasoning and derive conclusions which are not derivable or refutable from the knowledge base [19].

This work proposes a novel method for transductive inference on Description Logic representations. In the class-membership prediction task, discriminative methods proposed so far ignore unlabelled problem instances (individuals for which the value of such class-membership is unknown); however, accounting for unlabelled instances during learning can provide more accurate results if some conditions are met [6, 27]. Generative methods, on the other hand, try to model a joint probability distribution on both instances and labels, thus facing a possibly harder learning problem than only predicting the most probable label for any given instance.

In section 2 we will first shortly survey related works, and introduce a variant to the classic class-membership prediction problem. In section 3 we will introduce the proposed method: the assumptions it relies on, and how it can be used for class-membership prediction on large and Web scale ontological knowledge bases. In section 4, we will provide empirical evidence for the effectiveness of the proposed method with respect to other methods in SW literature.

2 Preliminaries

A variety of approaches have been proposed in the literature for class-membership prediction, either *discriminative* or *generative* [17]. Assuming instances are sampled i.i.d. from a distribution P ranging over a space $X \times Y$ (where X is the space of instances and Y a set of labels), *generative* prediction methods first build an estimate $\hat{P}$ of the joint probability distribution $P(X, Y)$, and then use it to infer $\hat{P}(Y \mid x) = \hat{P}(Y, x)/\hat{P}(x)$ for a given, unlabelled instance $x \in X$. On the other hand, *discriminative* methods simply aim at estimating when $P(y \mid x) \geq 0.5$, for any given $(x, y) \in X \times Y$ (thus facing a possibly easier problem than estimating a joint probability distribution over $X \times Y$). The following shortly surveys class-membership prediction methods proposed so far.

2.1 Discriminative Methods

Some of the approaches proposed for solving the class-membership prediction problem are similarity-based. For instance, methods relying on the k-Nearest Neighbours (k-NN) algorithm are discussed in [7, 19]. A variety of (dis-)similarity measures between either individuals or concepts have been proposed: according to [5], they can be based on *features* (where objects are characterised by a set of features, such as in [15]), on the *semantic-network* structure (where background information is provided in the form of a semantic network, such as in [9, 16]) or on the *information content* (where both the semantic network structure and population are considered, such as in [8]). Kernel-based algorithms [21] have been proposed for various learning tasks from DL-based representations. This is made possible by the existence of a variety of kernel functions, either for concepts or individuals (such as [10, 4, 12]). By (implicitly) projecting instances into an high-dimensional feature space, kernel functions allow to adapt a multitude of machine learning algorithms to structured representations. SW literature includes methods for inducing robust classifiers [11] or learning to rank [13] from DL knowledge bases using kernel methods.

2.2 Generative Methods

For learning from formal ontologies, a generative approach has been discussed in [20]. In this work, each individual is associated to a *latent variable* which influences its attributes and the relations it participates in. It proposes using Bayesian non-parametrics to avoid setting the number of possible values for such latent variables (which can be seen as *cluster indicators*); and an inferencing scheme based on Markov Chain Monte Carlo, where posterior sampling is constrained by a pre-defined set of DL axioms. A quite different approach is discussed in [18]: this work focuses on learning theories in a probabilistic extension of the $\mathcal{ALC}$ DL named CR$\mathcal{ALC}$, using DL refinement operators to efficiently explore the space of concepts. It is inspired by literature on Bayesian Logic Programs.

2.3 Semi-Supervised and Transductive Learning

Classic discriminative learning methods ignore unlabelled instances. However, real life scenarios are usually characterized by an abundance of unlabelled instances and a few labelled ones [27]. This may also be the case for class-membership prediction from formal ontologies: class-membership relations may be difficult to obtain during ontology engineering tasks (e.g. due to availability of domain experts) and inference (e.g. since deciding instance-membership may have an intractable time complexity in some languages).

Using unlabelled instances during learning is generally known in the machine learning community as *Semi-Supervised Learning* [6, 27] (SSL). A variant to this setting is known as *Transductive Learning* [23] and refers to finding a labelling only to unlabelled instances provided in the training phase, without necessarily generalizing to unseen instances (and thus resulting into a possibly *simpler* learning problem). If the marginal distribution of instances P_X is informative with respect to the conditional probability distribution $P(Y \mid x)$, accounting for unlabelled instances during learning can provide more accurate results [6, 27]. A possible approach is including terms dependent from P_X into the objective function. This results in the two fundamental assumptions [6]:

- **Cluster assumption** – The joint probability distribution $P(X, Y)$ is structured in such a way that points in the same *cluster* are likely to have the same label.
- **Manifold assumption** – Assume that P_X is supported on a low-dimensional manifold: then, $P(Y \mid x)$ *varies smoothly*, as a function of x, with respect to the underlying structure of the manifold.

In the following sections, we discuss a similarity-based, non-parametric and computationally efficient method for predicting missing class-membership relations. This method is discriminative in nature, but also accounts for unknown class-membership during learning.

We will face a slightly different version of the classic class-membership prediction problem, namely *transductive class-membership prediction*. It is inspired to the *Main Principle* in [23]: "If you possess a restricted amount of information for solving some problem, try to solve the problem directly and never solve a more general problem as an intermediate step. It is possible that the available information is sufficient for a direct solution but is insufficient for solving a more general intermediate problem". In this setting, the learning algorithm only aims at estimating the class-membership relation of interest for a given training set of individuals, without necessarily being able to generalise to individuals outside such set.

In this work, we formalise the transductive class-membership prediction problem as a cost minimisation problem: given a set of training individuals $\text{Ind}_C(\mathcal{K})$ whose class-membership relation to a target concept C is either known or unknown, find a function $f^* : \text{Ind}_C(\mathcal{K}) \to \{+1, -1\}$ defined over training individuals and returning a value $+1$ (resp. -1) if the individual likely to be a member of C (resp. $\neg C$), minimizing a given cost function. More formally:

Definition 1. *(Transductive Class-Membership Prediction) The Transductive Class-Membership Prediction problem can be formalised as follows:*

- **Given:**
  - *a* target *concept C;*

- *a set of training individuals* $\text{Ind}_C(\mathcal{K})$ *in a knowledge base* $\mathcal{K}$ *partitioned in positive, negative and neutral examples or, more formally, such that:*
  $\text{Ind}^+_C(\mathcal{K}) = \{a \in \text{Ind}_C(\mathcal{K}) \mid \mathcal{K} \models C(a)\}$ *positive examples,*
  $\text{Ind}^-_C(\mathcal{K}) = \{a \in \text{Ind}_C(\mathcal{K}) \mid \mathcal{K} \models \neg C(a)\}$ *negative examples,*
  $\text{Ind}^0_C(\mathcal{K}) = \{a \in \text{Ind}_C(\mathcal{K}) \mid \mathcal{K} \not\models C(a) \wedge \mathcal{K} \not\models \neg C(a)\}$ *neutral examples;*
- *A* cost function $cost(\cdot) : \mathcal{F} \mapsto \mathbb{R}$*, specifying the* cost *associated to a set of class-membership relations assigned to training individuals by* $f \in \mathcal{F}$*, where* $\mathcal{F}$ *is a space of labelling functions of the form* $f : \text{Ind}_C(\mathcal{K}) \mapsto \{+1, -1\}$;

– **Find** *a labelling function* $f^* \in \mathcal{F}$ *minimizing the given cost function with respect to training individuals* $\text{Ind}_C(\mathcal{K})$*:*

$$f^* \leftarrow \arg\min_{f \in \mathcal{F}} cost(f).$$

The function f^* can then be used to estimate the class-membership relation with respect to the target concept C for all training individuals $a \in \text{Ind}_C(\mathcal{K})$: it will return $+1$ (resp. -1) if an individual is likely to be a member of C (resp. $\neg C$). Note that the function is defined on the whole set of training individuals; therefore it can possibly contradict already known class-membership relations (thus being able to handle noisy knowledge). If $\text{Ind}_C(\mathcal{K})$ is finite, the space of labelling functions $\mathcal{F}$ is also finite, and each function $f \in \mathcal{F}$ can be equivalently expressed as a vector in $\{-1, +1\}^n$, where $n = |\text{Ind}_C(\mathcal{K})|$.

3 Propagating Class-Membership Information Among Individuals

This section discusses a *graph-based semi-supervised* [27] method for class-membership prediction from DL representations. The proposed method relies on a weighted *semantic similarity graph*, where nodes represent positive, negative and neutral examples of the transductive class-membership prediction problem, and weighted edges define similarity relations among such individuals.

More formally, let $\mathcal{K}$ be a knowledge base, $\text{Ind}_C(\mathcal{K})$ a set of training individuals with respect to a target concept C in $\mathcal{K}$, and $Y = \{-1, +1\}$ a space of labels each corresponding to a type of class-membership relation with respect to C. Each training individual $a \in \text{Ind}_C(\mathcal{K})$ is associated to a label, which will be $+1$ (resp. -1) if $\mathcal{K} \models C(a)$ (resp. $\mathcal{K} \models \neg C(a)$), and will be unknown otherwise, thus representing an unlabelled instance. For defining a cost over functions $f \in \mathcal{F}$, the proposed method relies on *regularization by graph*: the learning process aims at finding a labelling function that is both consistent with given labels, and changes smoothly between similar instances (where similarity relations are encoded in the semantic similarity graph). This can be formalised through a *regularization framework*, using a measure of the consistency to the given labels as a loss function, and a measure of smoothness among the similarity graph as a regulariser. Several cost functions have been proposed in SSL literature. An appealing class of functions, from the side of computational cost, relies on the *quadratic cost criterion* framework [6, ch. 11]: for this class of functions, a closed form solution to the cost minimisation problem can be found efficiently (subsection 3.2).

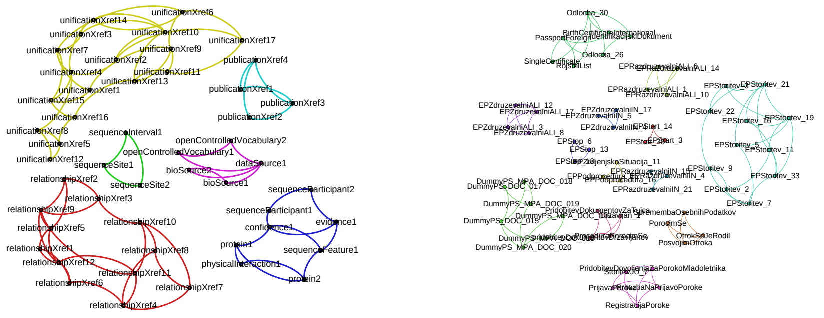
3.1 Semantic Similarity Graph

A similarity graph can be represented with a weight matrix $\mathbf{W}$, where the value of $\mathbf{W}_{ij}$ represents the strength of the similarity relation between two training examples x_i and x_j. In graph-based SSL literature, $\mathbf{W}$ is often obtained either as a Nearest Neighbour (NN) graph (where each instance is connected to the k most similar instances in the graph, or to those with a distance under a radius ϵ); or using a kernel function, such as the Gaussian kernel. Finding the best way to construct $\mathbf{W}$ is an active area of research; for example, in [6, ch. 20] authors discuss a method to combine multiple similarity measures in the context of protein function prediction, while [1] proposes a method for data-driven similarity graph construction.

When empirically evaluating the proposed method, we employ the family of dissimilarity measures between individuals in a DL knowledge base defined in [19], since it does not constrain to any particular family of DLs; we refer to the resulting similarity graph among individuals in a formal ontology as the *semantic similarity graph*. Given a set of concept descriptions $F = \{F_1, \ldots, F_n\}$ and a weight vector $\mathbf{w}$, such family of dissimilarity measures $d_p^F : Ind(\mathcal{A}) \times Ind(\mathcal{A}) \mapsto [0, 1]$ is defined as:

$$\delta_i(x, y) = \begin{cases} 0 \text{ if } (\mathcal{K} \models F_i(x) \wedge \mathcal{K} \models F_i(y)) \vee (\mathcal{K} \models \neg F_i(x) \wedge \mathcal{K} \models \neg F_i(y)) \\ 1 \text{ if } (\mathcal{K} \models F_i(x) \wedge \mathcal{K} \models \neg F_i(y)) \vee (\mathcal{K} \models \neg F_i(x) \wedge \mathcal{K} \models F_i(y)) \\ u_i \text{ otherwise} \end{cases} \quad (1)$$

where $x, y \in Ind(\mathcal{A})$ and $p > 0$.



(a) 3-NN graph, BioPAX (Proteomics)

(b) 3-NN graph, Leo

Fig. 1: k-Nearest Neighbour Semantic Similarity graphs for individuals BioPAX (Proteomics) ontology (left) and for the Leo ontology (right), obtained using the dissimilarity measure in [19]: F was defined as the set of atomic concepts in the ontology (each weighted with its normalized entropy [19]) and $p = 2$.

Two examples of (k-NN) semantic similarity graphs among all individuals in the ontologies BIOPAX (PROTEOMICS) and LEO, obtained using the aforementioned dissimilarity measure, are provided in Fig. 1.

3.2 Quadratic Cost Criteria

In quadratic cost criteria [6, ch. 11], the original label space $\{-1,+1\}$ (binary classification case) is relaxed to $[-1,+1]$. This allows to express the confidence associated to a labelling (and may give an indication about $P(Y \mid x)$). For such a reason, in the proposed method, the labelling functions space $\mathcal{F}$ will be relaxed to functions of the form $f : \text{Ind}_C(\mathcal{K}) \mapsto [-1,+1]$. As in subsection 2.3, labelling functions can be equivalently represented as vectors $\mathbf{y} \in [-1,+1]^n$. Let $\hat{\mathbf{y}} \in [-1,+1]^n$ be a possible labelling for n instances. We can see $\hat{\mathbf{y}}$ as a $(l+u) = n$ dimensional vector, where the first l indices refer to already labelled instances, and the last u to unlabelled instances: $\hat{\mathbf{y}} = [\hat{\mathbf{y}}_l, \hat{\mathbf{y}}_u]$.

Consistency of $\hat{\mathbf{y}}$ with respect to original labels can be formulated in the form of a quadratic cost: $\sum_{i=1}^{l}(\hat{y_i} - y_i)^2 = ||\hat{\mathbf{y}}_l - \mathbf{y}_l||^2$.

Similarly, labellings can be regularised with respect to the graph structure: as in [2], such consistency with respect to the geometry of instances can be estimated as $0.5\sum_{i,j=1}\mathbf{W}_{ij}(\hat{y_i} - \hat{y_j})^2 = \hat{\mathbf{y}}^T\mathbf{L}\hat{\mathbf{y}}$, where $\mathbf{W}$ is the semantic similarity graph and $\mathbf{L} = \mathbf{D} - \mathbf{W}$, $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$ ad 0 otherwise, is the unnormalized graph Laplacian. A different criterion, discussed in [24, 25], measures it as $(\mathbf{D}^{-0.5}\hat{\mathbf{y}})^T\mathbf{L}(\mathbf{D}^{-0.5}\hat{\mathbf{y}})$.

Another regularization term in the form of $||\hat{\mathbf{y}}||^2$ (or $||\hat{\mathbf{y}}_u||^2$, as in [24]) can be added to the final cost function to prefer smaller values in $\hat{\mathbf{y}}$. This is useful e.g. to prevent arbitrary labellings in a connected component of the semantic similarity graph containing no labelled instances.

Putting the pieces together, we obtain two quadratic cost criteria discussed in the literature, namely Regression on Graph [2] (RG) and the Consistency Method [24] (CM):

RG: $cost(\hat{\mathbf{y}}) = ||\hat{\mathbf{y}}_l - \mathbf{y}_l||^2 + \mu\hat{\mathbf{y}}^T\mathbf{L}\hat{\mathbf{y}} + \mu\epsilon||\hat{\mathbf{y}}||^2$;
CM: $cost(\hat{\mathbf{y}}) = ||\hat{\mathbf{y}}_l - \mathbf{y}_l||^2 + \mu(\mathbf{D}^{-0.5}\hat{\mathbf{y}})^T\mathbf{L}(\mathbf{D}^{-0.5}\hat{\mathbf{y}}) + ||\hat{\mathbf{y}}_u||^2$.

As a title of example, we will now derive a closed form solution for the problem of finding a (global) minimum for the quadratic cost criterion in RG. Its first order derivative is defined as follows:

$$\frac{1}{2}\frac{\partial cost(\hat{\mathbf{y}})}{\partial \hat{\mathbf{y}}} = (\mathbf{S} + \mu\mathbf{L} + \mu\epsilon\mathbf{I})\hat{\mathbf{y}} - \mathbf{S}\mathbf{y},$$

where $\mathbf{S} = diag(\mathbf{s}_1, \ldots, \mathbf{s}_n)$, with $\mathbf{s}_i = 1$ iff $i \leq l$ and 0 otherwise. Its second order derivative is a positive definite matrix if $\epsilon > 0$, since $\mathbf{L}$ is positive semi-definite. Therefore, setting the first order derivative to 0 leads to a global minimum:

$$\hat{\mathbf{y}} = (\mathbf{S} + \mu\mathbf{L} + \mu\epsilon\mathbf{I})^{-1}\mathbf{S}\mathbf{y},$$

showing that $\hat{\mathbf{y}}$ can be obtained either by matrix inversion or by solving a (possibly sparse) linear system.

This work leverages quadratic cost criteria to efficiently solve the transductive class-membership prediction problem. Finding a minimum $\hat{\mathbf{y}}$ for a predefined cost criterion is equivalent to finding a labelling function f^* in the form $f^* : \text{Ind}_C(\mathcal{K}) \mapsto [-1,+1]$, where the labelling returned for a generic training individual $a \in Ind_C(\mathcal{K})$ correspond to the value in $\hat{\mathbf{y}}$ in the position mapped to a. This can be done by representing the set of training individuals $Ind_C(\mathcal{K})$ as a partially labelled vector $\mathbf{y}$ of length $|Ind_C(\mathcal{K})| = n$, such that the first l (resp. last u) components correspond to positive and negative (resp. neutral) examples in $Ind_C(\mathcal{K})$. Such $\mathbf{y}$ can be then used to measure the consistency with original labels in a quadratic cost criterion; while the semantic similarity graph can be employed to enforce smoothness in class-membership predictions among similar training individuals.

An advantage of quadratic cost criteria is that their minimization ultimately reduces to solving a large sparse linear system [24, 6], a well-known problem in the literature whose time complexity is nearly linear in the number of non-zero entries in the coefficient matrix [22]. For large-scale datasets, a subset selection method is described in [6, ch. 18], which allows to greatly reduce the size of the original linear system.

4 Preliminary Empirical Evaluations

In this section, we evaluate several (inductive and transductive) methods for class-membership prediction, with the aim of comparing the methods discussed in section 3 with respect to other methods in SW literature. We are reporting evaluations for the Regularization on Graph [2] (RG) and the Consistency Method [24] (CM); Label Propagation [26] (LP); three kinds of Support Vector Machines [21] (SVM), namely Hard-Margin SVM (HM-SVM), Soft-Margin SVM with L_1 norm (SM-SVM) and Laplacian SVM [3] (LapSVM); and $\sqrt{l}$-Nearest Neighbors for class-membership prediction [19].

4.1 Description of Evaluated Methods

LP is a graph-based SSL algorithm relying on the idea of propagating labelling information among similar instances through an iterative process involving matrix operations. It can be equivalently formulated under the quadratic criterion framework [6, ch. 11]. More formally it associates, to each unlabelled instance in the graph, the probability of performing a random walk until a positively (resp. negatively) example is found.

We also evaluated Support Vector Machines (SVM), which have been proposed for inducing robust classifiers from ontological knowledge bases [12, 19]. SVM classifiers come in different flavours: the classic HM-SVM binary classifier aims at finding the hyperplane in the feature space separating the instances belonging to different classes, which maximises the *geometric margin* between the hyperplane and nearest training points. The SM-SVM classifier is a relaxation of HM-SVM, which allows for some misclassification in training instances (by relaxing the need of having perfectly linearly separable training instances in the feature space). LapSVM is a semi-supervised extension of the SM-SVM classifier: given a set of labelled instances and a set of unlabelled instances, it aims at finding an hyperplane that is also smooth with respect to the (estimated) geometry of instances. More formally, let $(\mathbf{x}_l, \mathbf{y}_l)$ (resp. $\mathbf{x}_u$) be a set of labelled (resp. unlabelled) instances. LapSVM finds a function f in a space of functions $\mathcal{H}_K$ determined by the kernel K (called *Reproducing Kernel Hilbert Space* [21]) minimizing $\frac{1}{l}\sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_L ||f||^2_{\mathcal{H}_K} + \gamma_{\mathcal{M}} ||f||^2_{\mathcal{M}}$, where V represents a costs function of errors committed by f on labeled samples (typically the hinge loss function $\max\{0, 1 - y_i f(x_i)\}$), $|| \cdot ||_{\mathcal{H}_K}$ imposes smoothness conditions on

Ontology	Expressivity	#Axioms	#Individuals	#Classes	#ObjectProperties
BioPax (Proteomics)	$\mathcal{ALCHN}(\mathcal{D})$	773	49	55	47
Family-Tree	$\mathcal{SROIF}(\mathcal{D})$	2059	368	22	52
Leo	$\mathcal{ALCHIF}(\mathcal{D})$	430	61	32	26
MDM0.73	$\mathcal{ALCHOF}(\mathcal{D})$	1098	112	196	22
Wine	$\mathcal{SHOIN}(\mathcal{D})$	1046	218	142	21

Table 1: Ontologies considered in the experiments.

possible solutions [21] and $||\cdot||^2_{\mathcal{M}}$, intuitively, penalizes rapid changes in the classification function between close instances in the similarity graph. It generalizes HM-SVM ($\gamma_L \to 0, \gamma_{\mathcal{M}} = 0$) and SM-SVM ($\gamma_{\mathcal{M}} = 0$). Our implementation of LapSVM follows the algorithm proposed in [3]; for HM-SVM, SM-SVM and LapSVM, we solve the underlying convex optimization problems using the Gurobi optimizer [14].

RG, CM, LP and LapSVM all rely on a semantic similarity graph $\mathbf{W}$ as a representation of the geometry of instances. We first calculate distances employing the dissimilarity measure defined in [19] and outlined in eq. 1, with $p = 2$; then we obtain $\mathbf{W}$ by building a k-Nearest Neighbour graph using such distances (since sparsity in $\mathbf{W}$ influences the scalability of quadratic cost criteria, as written in subsection 3.2). When building the neighbourhood of a node, we handled the cases in which nodes had the same distance by introducing a random ordering between such nodes. The Kernel function used for Hard-Margin SVM, Soft-Margin SVM and Laplacian SVM are also defined in [19], and directly correlated with the aforementioned dissimilarity measure in eq. 1 (given a committee of concepts F and the parameters $\mathbf{w}$ and p, the dissimilarity was originally obtained as $1 - k(a, b)$, where $k(a, b)$ is the value of the kernel function on a pair of individuals (a, b) in the knowledge base). We also provide a first evaluation for the k-NN algorithm (with $k = \sqrt{l}$, where l is the number of labelled instances, as discussed in [19]): we simply choose the majority class among the $\sqrt{l}$ most similar individuals to label each unlabelled instance.

4.2 Evaluations

Starting from a set of real ontologies [1] (outlined in Table 1), we generated a set of 20 random query concepts for each ontology [2], so that the number of individuals belonging to the target query concept C (resp. $\neg C$) was at least of 10 elements and the number of individuals in C and $\neg C$ was in the same order of magnitude. A DL reasoner [3] was employed to decide on the theoretical concept-membership of individuals to query concepts. We employ the evaluation metrics in [7], which take into account the peculiarities deriving by the presence of missing knowledge:

[1] From TONES Repository: `http://owl.cs.manchester.ac.uk/repository/`

[2] Using the methods available at `http://lacam.di.uniba.it/~nico/research/ontologymining.html`

[3] Pellet v2.3.0 – `http://clarkparsia.com/pellet/`

Leo	Match	Omission	Commission	Induction
RG	1 ± 0	0 ± 0	0 ± 0	0 ± 0
CM	1 ± 0	0 ± 0	0 ± 0	0 ± 0
LP	0.942 ± 0.099	0.007 ± 0.047	0.052 ± 0.091	0 ± 0
SM-SVM	0.963 ± 0.1	0 ± 0	0.037 ± 0.1	0 ± 0
LapSVM	0.978 ± 0.068	0 ± 0	0.022 ± 0.068	0 ± 0
$\sqrt{l}$-NN	0.971 ± 0.063	0 ± 0	0.029 ± 0.063	0 ± 0

BioPAX (Proteomics)	Match	Omission	Commission	Induction
RG	0.986 ± 0.051	0.004 ± 0.028	0.008 ± 0.039	0.002 ± 0.02
CM	0.986 ± 0.051	0.002 ± 0.02	0.01 ± 0.044	0.002 ± 0.02
LP	0.982 ± 0.058	0.002 ± 0.02	0.014 ± 0.051	0.002 ± 0.02
SM-SVM	0.972 ± 0.075	0 ± 0	0.026 ± 0.068	0.002 ± 0.02
LapSVM	0.972 ± 0.075	0 ± 0	0.026 ± 0.068	0.002 ± 0.02
$\sqrt{l}$-NN	0.972 ± 0.075	0 ± 0	0.026 ± 0.068	0.002 ± 0.02

MDM0.73	Match	Omission	Commission	Induction
RG	0.953 ± 0.063	0.003 ± 0.016	0.011 ± 0.032	0.015 ± 0.039
CM	0.953 ± 0.063	0.001 ± 0.009	0.013 ± 0.036	0.018 ± 0.04
LP	0.942 ± 0.065	0 ± 0	0.026 ± 0.046	0.033 ± 0.054
SM-SVM	0.793 ± 0.252	0 ± 0	0.174 ± 0.255	0.033 ± 0.054
LapSVM	0.915 ± 0.086	0 ± 0	0.052 ± 0.065	0.033 ± 0.054
$\sqrt{l}$-NN	0.944 ± 0.069	0 ± 0	0.023 ± 0.051	0.033 ± 0.054

Wine	Match	Omission	Commission	Induction
RG	0.24 ± 0.03	0 ± 0.005	0.007 ± 0.017	0.5 ± 0.176
CM	0.242 ± 0.028	0 ± 0.005	0.005 ± 0.015	0.326 ± 0.121
LP	0.239 ± 0.035	0 ± 0.005	0.008 ± 0.021	0.656 ± 0.142
SM-SVM	0.235 ± 0.036	0 ± 0	0.012 ± 0.024	0.753 ± 0.024
LapSVM	0.238 ± 0.033	0 ± 0	0.009 ± 0.021	0.753 ± 0.024
$\sqrt{l}$-NN	0.241 ± 0.031	0 ± 0	0.006 ± 0.018	0.753 ± 0.024

Table 2: Match, Omission, Commission and Induction [19] results for a k-Fold Cross Validation ($k = 10$) on 20 randomly generated queries. For each experiment, the best parameters within the training were found using a k-Fold Cross Validation ($k = 10$).

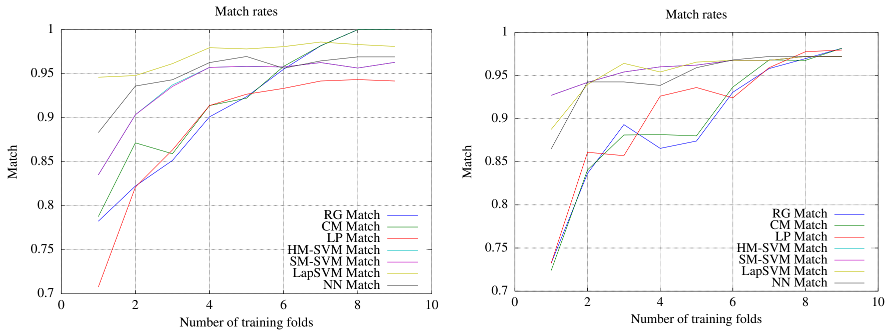
Match Case of an individual that got the same label by the reasoner and the inductive classifier.

Omission Error Case of an individual for which the inductive method could not determine whether it was relevant to the query concept or not while it was found relevant by the reasoner.

Commission Error Case of an individual found to be relevant to the query concept while it logically belongs to its negation or vice-versa.

Induction Case of an individual found to be relevant to the query concept or to its negation, while either case is not logically derivable from the knowledge base.

Before evaluating on the test set, parameter tuning was performed for each of the methods via a k-Fold Cross Validation ($k = 10$) within the training set, for finding the parameters with lower classification error in cross-validation. For LapSVM, the



(a) Leo ontology

(b) BioPAX (Proteomics) ontology

Fig. 2: Variation of average Match Rates with respect to the number of folds used in the training step, during a k-Fold Cross Validation (with $k = 10$).

$(\gamma_L, \gamma_{\mathcal{M}})$ parameters were varied in $\{10^{-4}, 10^{-3}, \ldots, 10^4\}$, while for SM-SVM, which follows the implementation in [21, pg. 223], the C parameter was allowed to vary in $\{10^{-4}, 10^{-3}, \ldots, 10^4\}$. Similarly, the (μ, ϵ) parameters in RG and CM where varied in $\{10^{-4}, 10^{-3}, \ldots, 10^4\}$. The parameter k for building the k-NN semantic similarity graph, used by LapSVM, RG, CM and LP, was varied in $\{2, 4, 8, 16\}$. We did not carefully choose the concept committee F defining the dissimilarity measure: we simply used the set of atomic concepts in the ontology, thus ignoring any prior knowledge about the structure of the target concept C or the presence of statistical correlations in the knowledge base. Each concept in the committee F was weighted with its normalized entropy [19]. RG, CM and LP give an indication of the uncertainty associated to a specific labelling by associating values in the set $[-1, +1]$ to each node; when such values are ≈ 0 (specifically, when the label was in the set $[-10^{-4}, 10^{-4}]$ we decided to leave the node unlabelled, so to try to provide more robust estimates of labels (and thus a possibly lower commission error and match rates and higher omission error rates). This may happen e.g. when there are no labelled examples within a connected component of the semantic similarity graph.

In Tab. 2 we report average index rates and standard deviations for each of the ontologies in Tab. 1; the only exceptions is for the FAMILY-TREE ontology, which provided 0.76 ± 0.13 match rates and 0.24 ± 0.13 induction rates for all methods (with the exception of LP, where the induction rates were 0.21 ± 0.14. In general, LapSVM outperformed the other two non-SSL SVM classification methods. This happened with varying quantities of unlabelled data; this is shown for example in the behavior of match rates in subfigure 2a, where results obtained in a k-Fold Cross Validation using a varying quantity of labelled instances. However, standard SVM training is $O(m^3)$ in general, where m is the number of training instances; therefore, some extra effort may be necessary to make SVM methods scale on SW knowledge bases. Such results may provide some empirical evidence that inductive methods for formal ontologies may take benefit from also accounting for unlabelled instances during learning.

5 Conclusion and Future Works

This work proposes a method for transductive class-membership prediction based on graph-based regularisation from DL representations. It leverages neutral examples by propagating class-membership information among similar individuals in the training set. The proposed method relies on quadratic cost criteria, whose optimization can be reduced to solving a (possibly sparse) linear system; this is a well-known problem in the literature, with a nearly linear time complexity in the number of non-zero entries in the coefficient matrix.

We did not analyse carefully the impact of different choices in the (dis-)similarity measure for building the semantic similarity graph. However, the similarity graph has a strong influence on the effectiveness of the methods used [27]. The construction of the similarity graph for class-membership learning tasks can be influenced by factors such as the structure of the target concept C, or by finding statistical correlation within the knowledge base. Also, it is not clear whether continuous labels assigned by the proposed methods may correspond to posterior probability estimates from the statistical point of view. In future work, we aim at investigating the aforementioned two aspects of graph-based transductive and semi-supervised class-membership prediction from DL representations.

References

[1] Alexandrescu, A., Kirchhoff, K.: Data-driven graph construction for semi-supervised graph-based learning in nlp. In: Sidner, C., et al. (eds.) HLT-NAACL. pp. 204–211. The Association for Computational Linguistics (2007)

[2] Belkin, M., Matveeva, I., Niyogi, P.: Regularization and semi-supervised learning on large graphs. In: Shawe-Taylor, J., et al. (eds.) COLT. LNCS, vol. 3120, pp. 624–638. Springer (2004)

[3] Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. Journal of Machine Learning Research 7, 2399–2434 (2006)

[4] Bloehdorn, S., Sure, Y.: Kernel methods for mining instance data in ontologies. In: Proceedings of the 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference. pp. 58–71. ISWC'07/ASWC'07, Springer (2007)

[5] Borgida, A., Walsh, T., Hirsh, H.: Towards measuring similarity in description logics. In: Horrocks, I., et al. (eds.) Description Logics. CEUR Workshop Proceedings, vol. 147. CEUR-WS.org (2005)

[6] Chapelle, O., Schölkopf, B., Zien, A. (eds.): Semi-Supervised Learning. MIT Press, Cambridge, MA (2006)

[7] d'Amato, C., Fanizzi, N., Esposito, F.: Query answering and ontology population: an inductive approach. In: Hauswirth, M., et al. (eds.) Proceedings of the 5th European Semantic Web Conference (ESWC'08). Springer, Tenerife, Spain (2008)

[8] d'Amato, C., Fanizzi, N., Esposito, F.: A semantic similarity measure for expressive description logics. CoRR abs/0911.5043 (2009)

[9] d'Amato, C., Staab, S., Fanizzi, N.: On the influence of description logics ontologies on conceptual similarity. In: Proceedings of the 16th international conference on Knowledge Engineering: Practice and Patterns. pp. 48–63. EKAW '08, Springer, Berlin, Heidelberg (2008)

[10] Fanizzi, N., d'Amato, C.: Inductive concept retrieval and query answering with semantic knowledge bases through kernel methods. In: Proceedings of the 11th international conferenceon Knowledge-based intelligent information and engineering systems: Part I. pp. 148–155. KES'07, Springer (2007)

[11] Fanizzi, N., d'Amato, C., Esposito, F.: Reduce: A reduced coulomb energy network method for approximate classification. In: Proceedings of the 6th European Semantic Web Conference (ESWC'09). pp. 323–337. Springer

[12] Fanizzi, N., d'Amato, C., Esposito, F.: Statistical Learning for Inductive Query Answering on OWL Ontologies. In: Proceedings of the 7th International Conference on The Semantic Web. pp. 195–212. ISWC '08, Springer (2008)

[13] Fanizzi, N., d'Amato, C., Esposito, F.: Towards learning to rank in description logics. In: Coelho, H., et al. (eds.) ECAI. Frontiers in Artificial Intelligence and Applications, vol. 215, pp. 985–986. IOS Press (2010)

[14] Gurobi Optimization, I.: Gurobi optimizer reference manual (2012)

[15] Hu, B., Dasmahapatra, S., Lewis, P.: Semantic metrics. Int. J. Metadata Semant. Ontologies 2(4), 242–258 (Jul 2007)

[16] Janowicz, K., Wilkes, M.: Sim-dla: A novel semantic similarity measure for description logics reducing inter-concept to inter-instance similarity. In: Aroyo, L., et al. (eds.) ESWC. LNCS, vol. 5554, pp. 353–367. Springer (2009)

[17] Lasserre, J., Bishop, C.M.: Generative or discriminative? getting the best of both worlds. BAYESIAN STATISTICS 8, 3–24 (2007)

[18] Ochoa-Luna, J.E., Cozman, F.G.: An algorithm for learning with probabilistic description logics. In: Bobillo, F., et al. (eds.) URSW. pp. 63–74 (2009)

[19] Rettinger, A., Lösch, U., Tresp, V., d'Amato, C., Fanizzi, N.: Mining the semantic web - statistical learning for next generation knowledge bases. Data Mining and Knowledge Discovery - Special Issue on Web Mining (2012)

[20] Rettinger, A., Nickles, M., Tresp, V.: Statistical relational learning with formal ontologies. In: Buntine, W.L., et al. (eds.) ECML/PKDD (2). LNCS, vol. 5782, pp. 286–301. Springer (2009)

[21] Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, New York, NY, USA (2004)

[22] Spielman, D.A., Teng, S.H.: Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In: Proceedings of the thirty-sixth annual ACM symposium on Theory of computing. pp. 81–90. STOC '04, ACM, New York, NY, USA (2004)

[23] Vapnik, V.N.: Statistical learning theory. Wiley, 1 edn. (Sep 1998)

[24] Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: Advances in Neural Information Processing Systems 16. pp. 321–328. MIT Press (2004)

[25] Zhou, D., Huang, J., Schölkopf, B.: Learning from labeled and unlabeled data on a directed graph. In: Proceedings of the 22nd international conference on Machine learning. pp. 1036–1043. ICML '05, ACM (2005)

[26] Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation. Tech. rep., CMU CALD tech report CMU-CALD-02 (2002)

[27] Zhu, X.: Semi-supervised learning literature survey. Tech. Rep. 1530, Computer Sciences, University of Wisconsin-Madison (2005)

Data-Driven Logical Reasoning

Claudia d'Amato[1], Volha Bryl[2], Luciano Serafini[2]

[1] Department of Computer Science - University of Bari, Italy
claudia.damato@di.uniba.it
[2] Data & Knowledge Management Unit - Fondazione Bruno Kessler, Italy
{bryl|serafini}@fbk.eu

Abstract. The co-existence of heterogeneous but complementary data sources, such as ontologies and databases describing the same domain, is the reality of the Web today. In this paper we argue that this complementarity could be exploited both for discovering the knowledge not captured in the ontology but learnable from the data, and for enhancing the process of ontological reasoning by relying on the combination of formal domain models and evidence coming from data. We build upon our previous work on knowledge discovery from heterogeneous sources of information via association rules mining, and propose a method for automated reasoning on grounded knowledge bases (i.e. knowledge bases linked to data) based on the standard Tableaux algorithm. The proposed approach combines logical reasoning and statistical inference thus making sense of heterogeneous data sources.

1 Introduction

From the introduction of the Semantic Web view [3], many domain ontologies have been developed and stored in open access repositories. However, still huge amounts of data are stored in relational databases (DBs) and managed by RDBMSs (relational database management systems). The seamless integration of these two knowledge representation paradigms is becoming a crucial research challenge. Most of the work in this area concerns what is addressed as *ontology based data access (OBDA)* [4]. In OBDA the ontology "replicates" at a higher conceptual level the physical schema of the DBMS and provides a "lens" under which the data can be viewed, and possibly adds additional semantic knowledge on the data. The connection between the ontology and the data is represented as conjunctive queries. Roughly speaking, every concept/relation of the ontology is associated to a conjunctive query which retrieves from the DB all and only the instances of such a concept/relation.

Another common situation is when existing ontologies describe domain aspects that (partially) complement data in a database. In this case the concepts of the ontologies are linkable to views of the DB. Also in this case it would be very useful to be able to combine the knowledge contained in the two information sources, for example for enriching the existing ontologies. Due to the heterogeneity of the information a crisp representation of the correspondence between the DB data and the classes and relations of the ontologies (such as the one adopted in OBDA) is not possible. A more flexible connection between the two sources of knowledge should be adopted. An option could be to exploit rules that are able to express a statistical evidence of the connection between the data in a DB and the knowledge in the ontology. For giving the intuition for this solution, let us consider the following scenario. Given an existing ontology describing people gender, family status and their interrelations with Italian urban areas[1] and a demographic DB describing Italian occupations, average salaries, etc., a possible connection between the two information sources can be described with rules as follows:

"*clerks between 35 and 45 years old* ***living in a big city*** *are* ***male*** *and earn between 40 and 50 €*" with a *confidence* value of 0.75. (1)

where bold face terms correspond to classes and relations in the ontology, non bold face terms correspond to data in the DB. The *confidence* value can be interpreted as the probability that the specified connection between the two sources occurs. Rules of the form (1) are called *semantically enriched association rules*. They have been introduced in our previous work [5] where an inductive approach for discovering new knowledge in the form of *association rules* [1] from heterogeneous data sources is proposed.

In this paper, we revise the approach introduced in [5] by taking into account the *Open World Assumption* adopted in description logics (DLs), while in [5] association rules are extracted from the hybrid data sources by adopting an implicit *Closed Word Assumption* that is not fully compliant with the theory of ontological representation. We also make a further step towards the framework for knowledge representation and reasoning in which knowledge can be represented by a mix of logical formulas and sets of data, linked together. Specifically, we introduce a concept of a *grounded knowledge base* and the notion of *mixed model* that integrates logical knowledge expressed in terms of a description logic knowledge base, and a statistical data mining model that expresses the statistical regularities of the properties associated to a set of individuals. Finally and most importantly, we propose a method for automated reasoning on *grounded knowledge bases*, which is the result of combining logical reasoning and statistical inductive inference and learning. In particular, we propose an extension of the standard Tableaux algorithm grounded on the adoption of an heuristic to be used when random choices (i.e. the processing of a disjunction, namely when we need to decide whether an object x belongs to concept C or to concept D) have to be made during the reasoning process. The heuristic exploits the evidence coming from the data. Assume, for example, that for a given object x, which is a $Person$, a high school student, and has the property x is 15 years old, we need to decide whether x is a $Parent$ or not, and there is no statements in the knowledge base from which it is possible to infer neither x is a $Parent$ nor x is $\neg Parent$. The following association rule learned from the data (with high degree of confidence)

$$\text{AGE} = [0, 16] \Rightarrow \neg Parent \quad 0.99$$

can be exploited to conclude that, with high probability, x is not a $Parent$.

The rest of the paper is structured as follows. In Section 2 we give basic definitions necessary to set up the framework. In Section 3 we summarize and extend the approach for learning association rules from heterogeneous sources of information pre-

[1] The concepts *"male", "parent", "big city", "medium-sized town",* and the relations *"lives_in"* are used in the ontology.

sented in [5]. Section 4 presents the data-driven Tableaux reasoning algorithm, followed by discussions and conclusions in Section 5.

2 Basic definitions

Let $\mathbf{D}$ be a non empty set of objects and $f_1, \ldots, f_n$ be n feature functions defined on every element of $\mathbf{D}$, with $f_i : \mathbf{D} \rightarrow D_i$. $\mathbf{D}$ is called the set of observed objects and $f_i(d)$ for every $d \in \mathbf{D}$ is the i-th feature observed on d. Notationally we use $\mathbf{d}$ for the elements of $\mathbf{D}$ and $\mathbf{d}_1, \ldots, \mathbf{d}_n$ to denote the values of $f_1(\mathbf{d}), \ldots, f_n(\mathbf{d})$.

Let Σ be a DL alphabet composed of three disjoint sets of symbols, Σ_C, Σ_R and Σ_I, the set of concepts symbols, the set of role symbols and the set of individual symbols. A knowledge base on Σ, is a set $\mathcal{K}$ of DL inclusion axioms and DL assertions (we assume $\mathcal{ALC}$ as DL language here). The elements of $\mathcal{K}$ are called axioms of $\mathcal{K}$. An axiom can be of the form $X \sqsubseteq Y$, where X and Y are $\mathcal{ALC}$ (complex) concepts, or $X(a)$ where X is a (complex) concept and a an individual symbol, or $R(a, b)$, and $a = b$, where R is a role symbol and a and b are individual symbols. We call $X \sqsubseteq Y$ a subsumption, and $X(a)$, $R(a, b)$, and $a = b$ assertions. $\mathcal{K}$ is formally defined as a couple $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ where $\mathcal{T}$ contains the inclusion axioms ($\mathcal{T}$ stands for Terminological part) and $\mathcal{A}$ contains the assertional axioms ($\mathcal{A}$ stands for Assertional part).

An interpretation of a DL alphabet Σ is a pair $\mathcal{I} = \left\langle \Delta_{\mathcal{I}}, \cdot^{\mathcal{I}} \right\rangle$ such that $\Delta^{\mathcal{I}}$ is a non empty set, and $\cdot^{\mathcal{I}}$ is a function that assigns to each concepts name a subset of $\Delta_{\mathcal{I}}$, to each role name a binary relation on $\Delta_{\mathcal{I}}$, and to each individual an element of $\Delta_{\mathcal{I}}$. The interpretation function can be extended to complex concepts in the usual way [2]. Satisfiability $\models$ of statements is also defined as usual [2]. $\mathcal{I} \models \mathcal{K}$ if $\mathcal{I} \models \phi$ for every axiom of $\mathcal{K}$. An interpretation $\mathcal{I}$ satisfies a knowledge base $\mathcal{K}$, (in symbols $\mathcal{I} \models \mathcal{K}$) if $\mathcal{I} \models \phi$ for every axiom of $\mathcal{K}$.

The "glue" between a dataset and a knowledge base is the so called *grounding*, which is a relation that connects the objects of the knowledge base with the data of the database. More formally: a *grounding* g of Σ on $\mathbf{D}$ is a total function $g : \mathbf{D} \rightarrow \Sigma_I$. This implies that for every $\mathbf{d} \in \mathbf{D}$ there is at least an element $a \in \Sigma_I$ with $g(\mathbf{d}) = a$. Intuitively $g(\mathbf{d}) = a$ represents the fact that the data $\mathbf{d}$ are about/correspond to object a of the knowledge base. Please note that the grounding g refers to objects that are explicitly mentioned in $\mathbf{D}$ and $\mathcal{K}$ respectively. In our framework (see Sect. 3) we assume that the grounding between $\mathbf{D}$ and $\mathcal{K}$ is already given.

3 Semantically enriched association rules

Association rules (ARs), originally introduced in [1], make it possible to represent in a rule based form some statistical regularities of the tuples in a relational database. Roughly speaking, ARs allow one to state conditional probabilities among the values of the attributes of the tuples of a database. Learning ARs is one of the fundamental tasks in data-mining.

In this section we recall how ARs can be extended to include information coming from an ontological knowledge base and how they can be used to bridge the knowledge contained in an ontology with that contained in a relational database. These rules are called *semantically enriched* ARs [5]. Hence, we revise the approach introduced in [5] by taking into account the *Open World Assumption* adopted in description logics (DL), while in [5] association rules are extracted from the hybrid data sources by adopting an implicit *Closed Word Assumption* that is not fully compliant with the theory of ontological representation. At the end of the section, the process of learning semantically enriched ARs [5] is also briefly recalled.

3.1 Association rules: an overview

Association rules [1] provide a form of rule patterns for data mining. Let $\mathbf{D}$ be a dataset made by a set of attributes $\{A_1, \ldots, A_n\}$ with domains $\mathcal{D}_i : i \in \{1, \ldots, n\}$. The basic components of an AR for the dataset $\mathbf{D}$ are itemsets. An itemset ϕ is a finite set of assignments of the form $A = a$ with $a \in \mathcal{D}(A)$. An itemset $\{A_{i_1} = a_1, \ldots, A_{i_m} = a_m\}$ can be denoted by the expression

$$A_{i_1} = a_1 \wedge \ldots \wedge A_{i_m} = a_m$$

An AR has the general form

$$\theta \Rightarrow \varphi \tag{2}$$

where θ and φ are itemsets. The *frequency* of an itemsets θ, denoted by $\mathit{freq}(\theta)$, is the number of cases in $\mathbf{D}$ that match θ, i.e.

$$\mathit{freq}(\theta) = |\{\mathbf{d} \in \mathbf{D} \mid \forall (A = a) \in \theta \; : \; f_A(\mathbf{d}) = a\}|$$

where f_A is the feature function for $\mathbf{d}$ w.r.t. the attribute A (see beginning of Sect.2).

The *support* of a rule $\theta \Rightarrow \varphi$ is equal to $\mathit{freq}(\theta \wedge \varphi)$. The *confidence* of a rule $\theta \Rightarrow \varphi$ is the fraction of items in $\mathbf{D}$ that match φ among those matching θ:

$$\mathit{conf}(\theta \Rightarrow \varphi) = \frac{\mathit{freq}(\theta \wedge \varphi)}{\mathit{freq}(\theta)}$$

A frequent itemset expresses the variables and the corresponding values that occur reasonably often together.

In terms of conditional probability, the confidence of a rule $\theta \Rightarrow \varphi$, can be seen as the maximum likelihood (frequency-based) estimate of the conditional probability that φ is true given that θ is true [8].

3.2 Semantically enriched association rules

Let $\mathcal{K}$ be a knowledge base on Σ, $\mathbf{D}$ a dataset and g a grounding of Σ on $\mathbf{D}$.

A *semantically enriched itemset* is a set containing statements of the form $f_i = a$, $C = tv$, $R = tv$ where, f_i is an attribute of $\mathbf{D}$, a a value in the range of f_i, C is a concept name of Σ_C and R is a role name of Σ_R and tv is a truth value in $\{\mathit{true}, \mathit{false}, \mathit{unknown}\}$. The elements of the itemset of the form $f_i = a$ are called *data items*, the elements of the form $C = tv$ and $R = tv$ are called *semantic items*.

A *semantically enriched* AR is an association rules made by *semantically enriched itemsets*. This means that for a certain set of individuals, both knowledge coming from the ontology and information coming from the database are available (see Sect. 3.3 for more details).

Coherently with ARs, it is possible to define the frequency of a *semantically enriched itemset* and the support of a *semantically enriched* AR. Given a grounding g of Σ on $\mathbf{D}$, the *frequency* of a *semantically enriched itemset* $\theta = \theta_d \wedge \theta_k$ (in the following also called *mixed itemset*) is the following generalization of the definition of frequency given for a standard itemset.

$$\mathit{freq}(\theta_d \wedge \theta_k) = |F|$$

where F is the following set:

$$F = \left\{ \mathbf{d} \in \mathbf{D} \;\middle|\; \begin{array}{l} \forall (f_i = a) \in \theta_d,\ f_i(\mathbf{d}) = a \\ \forall (C = \mathit{true}) \in \theta_k,\ \mathcal{K} \models C(g(\mathbf{d})) \\ \forall (C = \mathit{false}) \in \theta_k,\ \mathcal{K} \models \neg C(g(\mathbf{d})) \\ \forall (C = \mathit{unknown}) \in \theta_k,\ \mathcal{K} \not\models C(g(\mathbf{d}))\ \&\ \mathcal{K} \not\models \neg C(g(\mathbf{d})) \\ \forall (R = \mathit{true}) \in \theta_k,\ \mathcal{K} \models \exists R.\top(g(\mathbf{d})) \\ \forall (R = \mathit{false}) \in \theta_k,\ \mathcal{K} \models \neg\exists R.\top(g(\mathbf{d})) \\ \forall (R = \mathit{unknown}) \in \theta_k,\ \mathcal{K} \not\models \exists R.\top(g(\mathbf{d}))\ \&\ \mathcal{K} \not\models \neg\exists R.\top(g(\mathbf{d})) \end{array} \right\}$$

The support and confidence of a *semantically enriched* AR can be defined similarly.

3.3 Learning semantically enriched association rules

In [5] we proposed a framework for learning *semantically enriched* ARs from heterogeneous sources of information (namely an ontology and a relational database) grounded on the underlying *Closed World Assumption* that is not fully compliant with the theory of ontological representation. Here we extend the framework by taking into account the *Open World Assumption* usually made in DLs that is the theoretical framework underlying the OWL[2] language, namely the standard representation language in the Semantic Web [3]. With regard to this aspect it is important to note that the notions of frequency, confidence and support given in Sect. 3.2 are compliant with the *Open World Semantics*.

The approach for learning *semantically enriched* ARs is grounded on the assumption that a dataset $\mathbf{D}$ and an ontological knowledge base $\mathcal{K}$ share (a subset of) common individuals, and a grounding g of Σ on $\mathbf{D}$ is already available (see the end of Sect. 2 for more details on the grounding function). This assumption is reasonable in practice since, in the real world, there are several cases in which different information aspects concerning the same entities come from different data sources. An example is given by the public administration, where different administrative organizations have information about the same persons but concerning complementary aspects such as: personal data, income data, ownership data. Another example is given by the biological domain where research organizations have their own databases that could be complemented with existing domain ontologies.

The proposed framework is sketched in the following. To learn *semantically enriched* ARs from a dataset $\mathbf{D}$ and a knowledge base $\mathcal{K}$ grounded by g to $\mathbf{D}$, all the

[2] `http://en.wikipedia.org/wiki/Web_Ontology_Language`

information about the common domain of $\mathcal{K}$ and $\mathbf{D}$ are summarized (proposizionalized) in a tabular representation constructed as follows:

1. choose the primary entity of interest in $\mathbf{D}$ or $\mathcal{K}$ for extracting association rules and set this entity as the first attribute A_1 in the table $\mathbf{T}$ to be built; A_1 will be the primary key of the table
2. choose (a subset of) the attributes in $\mathbf{D}$ that are of interest for A_1 and set them as additional attributes in $\mathbf{T}$; the corresponding values are be obtained as a result of an SQL query involving the selected attributes and A_1
3. choose (a subset of) concept names $\{C_1, \ldots, C_m\}$ in $\mathcal{K}$ that are of interest for A_1 and set their names as additional attribute names in $\mathbf{T}$
4. for each $C_k \in \{C_1, \ldots, C_m\}$ and for each value a_i of A_1, if $\mathcal{K} \models C_k(a_i)$ then set to 1 the corresponding value of C_k in $\mathbf{T}$, else if $\mathcal{K} \models \neg C_k(a_i)$ then set the value to 0, otherwise set to $1/2$ the corresponding value of C_k in $\mathbf{T}$
5. choose (a subset of) role names $\{R_1, \ldots, R_t\}$ in $\mathcal{K}$ that are of interest for A_1 and set their names as additional attribute names in $\mathbf{T}$
6. for each $R_l \in \{R_1, \ldots, R_t\}$ and for each value a_i of A_1, if $\exists y \in \mathcal{K}\ s.t.\ \mathcal{K} \models R_l(a_i, y)$ then set to 1 the value of R_l in $\mathbf{T}$, else if $\forall y \in \mathcal{K}\ \mathcal{K} \models \neg R_l(a_i, y)$ then set the value of R_l in $\mathbf{T}$ to 0, otherwise set the value of R_l in $\mathbf{T}$ to $1/2$
7. choose (a subset of) the datatype property names $\{T_1, \ldots, T_v\}$ in $\mathcal{K}$ that are of interest for A_1 and set their names as additional attribute names in $\mathbf{T}$
8. for each $T_j \in \{T_1, \ldots, T_v\}$ and for each value a_i of A_1, if $\mathcal{K} \models T_j(a_i, \mathit{dataValue}_j)$ then set to $\mathit{dataValue}_j$ the corresponding value of T_j in $\mathbf{T}$, set 0 otherwise.

It is straightforward to note that for all but the datatype properties, the *Open World Assumption* is considered during the process for building the tabular representation. Numeric attributes are processed (as usual in data mining) for performing data discretization [10] namely for transforming numerical values in corresponding range of values (categorical values). An example of a unique tabular representation in the demographic domain is reported in Tab. 1 where $Person$, $Parent$, $Male$ and $Female$ are concepts of an ontological knowledge base $\mathcal{K}$, and JOB and AGE are attributes of a relational dataset $\mathbf{D}$. The numeric attribute (AGE) has been discretized.

The choice of representing the integrated source of information within tables allows for directly applying state of the art algorithms for learning association rules. Indeed, once a unique tabular representation is obtained, the well known APRIORI algorithm [1] is applied for discovering *semantically enriched* ARs from the integrated source of information[3] (see [5] for additional details and examples). Specifically, given a certain confidence threshold, ARs having a confidence value equal or greater than the fixed confidence threshold are learnt. This ensures that only significant ARs are considered while the others are discarded. As highlighted in sect. 3.1, the confidence value of the extracted *semantically enriched* ARs is interpreted as the conditional probability on the values of items in the consequence of the rule given that the left hand side of the rule is satisfied in (a model of) the available knowledge. Examples of *semantically enriched* ARs that could be learned from a table like Tab. 1 are reported in Tab. 2.

[3] Since a state of the art algorithm is adopted it is not reported in the paper. The novelty of the proposed approach consists in the way the integrated source of knowledge is built. Once this is obtained, the state of the art APRIORI algorithm is straightforwardly applied.

Table 1. Demographic example: a unique tabular representation $\mathbf{T}$

OBJECT	JOB	AGE	$Person$	$Parent$	$Male$	$Female$
x_1	Engineer	[36,45]	$true$	$true$	$true$	$false$
x_2	Policeman	[26,35]	$true$	$false$	$true$	$unknown$
x_3	Student	[16,25]	$true$	$false$	$true$	$false$
x_4	Student	[16,25]	$true$	$false$	$false$	$true$
x_5	Housewife	[26,35]	$true$	$true$	$false$	$true$
x_6	Clerk	[26,35]	$true$	$false$	$unknown$	$unknown$
...	...	...	...	...	...	...

Table 2. Demographic example: association rules

#	Rule	Confidence
1	(AGE=$[16, 25]$) $\wedge$ (JOB $= Student$) $\Rightarrow \neg Parent$	0.98
2	(JOB=$Policeman$) $\Rightarrow Male$	0.75
3	(AGE=$[16, 25]$) $\wedge$ $Parent \Rightarrow Female$	0.75
4	(JOB=$Primary\ school\ teacher$) $\Rightarrow Female$	0.78
5	(JOB=$Housewife$) $\wedge$ (AGE $= [26, 35]$) $\Rightarrow Parent \wedge Female$	0.85

4 Data-driven inference

We want to exploit the *semantically enriched* ARs (see Sect. 3.3) when performing deductive reasoning given DLs (namely ontological) representations. Since almost all DL inferences can be reduced to concept satisfiability [2], we focus on this inference procedure. For most expressive DL (such as $\mathcal{ALC}$) the Tableaux algorithm is employed. Its goal is to built a possible model, namely an interpretation, for the concept whose satisfiability has to be shown. If, building such a model, all clashes (namely contradictions) are found, the model does not exist and the concept is declared to be unsatisfiable.

Our goal is to set up a modified version of the Tableaux algorithm whose output, if any, is the **most plausible model**, namely the model that best fits the available data. This means to set up a data driven heuristic that should allow reducing the computational effort in finding a model for a given concept and should be also able to supply the model that is most coherent with/match the available knowledge. In this way the *variance due to intended diversity and incomplete knowledge* is reduced, namely, the number of possible models that could be built (see [7] for formal definitions). The inference problem we want to solve is formally defined as follows:

Definition 1 (Inference Problem).

Given: $\mathbf{D}$*, $\mathcal{K}$, the set R of ARs, a (possibly complex) concept E of $\mathcal{K}$, the individuals $x_1, \ldots, x_k \in \mathcal{K}$ that are instances of E, the grounding g of Σ on* $\mathbf{D}$

Determine: *the model $\mathcal{I}_r$ for E representing the* ***most plausible model*** *given the $\mathcal{K}$,* $\mathbf{D}$*, g and R.*

Intuitively, the most plausible model for E is the one on top of the ranking of the possible models $\mathcal{I}_i$ for E. The ranking of the possible models is built according to the degree up to which the models respect the ARs. The detailed procedure for building the *most plausible model* is illustrated in the following.

In order to find (or not find) a model, the standard Tableaux algorithm exploits a set of transformation rules that are applied to the considered concept. A transformation rule for each constructor of the considered language exists. In the following, the transformation rules for $\mathcal{ALC}$ logic are briefly recalled (see [2] for more details).

$\sqcap$-rule: **IF** the ABox $\mathcal{A}$ contains $(C_1 \sqcap C_2)(x)$, but it does not contain both $C_1(x)$ and $C_2(x)$ **THEN** $\mathcal{A} = \mathcal{A} \cup \{C_1(x), C_2(x)\}$

$\sqcup$-rule: **IF** $\mathcal{A}$ contains $(C_1 \sqcup C_2)(x)$, but it does not contain neither $C_1(x)$ nor $C_2(x)$ **THEN** $\mathcal{A}_1 = \mathcal{A} \cup \{C_1(x)\}$, $\mathcal{A}_2 = \mathcal{A} \cup \{C_2(x)\}$

$\exists$-rule: **IF** $\mathcal{A}$ contains $(\exists R.C)(x)$, but there is no individual name z s.t. $C(z)$ and $R(x,z)$ are in $\mathcal{A}$ **THEN** $\mathcal{A} = \mathcal{A} \cup \{C(y), R(x,y)\}$ where y is an individual name not occurring in $\mathcal{A}$.

$\forall$-rule: **IF** $\mathcal{A}$ contains $(\forall R.C)(x)$ and $R(x,y)$, but it does not contain $C(y)$ **THEN** $\mathcal{A} = \mathcal{A} \cup \{C(y)\}$

To test the satisfiability of a concept E, the algorithm starts with the ABox $\mathcal{A} = E(x_0)$ (with x_0 being a new individual) and applies to the ABox the consistency preserving transformation rules reported above until no more rules apply. The result could be all clashes, which means the concept is unsatisfiable, or an ABox containing a model for the concept E that means the concept is satisfiable.

The transformation rule for the disjunction ($\sqcup$-rule) is non-deterministic, that is, a given ABox is transformed into finitely many new ABoxes. The original ABox is consistent if and only if one of the new ABoxes is so. In order to save the computational complexity, the ideal solution (for the case of a consistent concept) should be to choose the ABox containing a model directly. Moving from this observation, in the following we propose an alternative version of the Tableaux algorithm. The main differences with respect to the standard Tableaux algorithm summarized above are:

1. the starting model for the inference process is given by the set of all attributes (and corresponding values) of $\mathbf{D}$ that are related to individuals $x_1, \ldots, x_k$ that are instances of E differently from the standard Tableaux algorithm where the initial model is simply given by the assertion concerning the concept of which the satisfiability (or unsatisfiability) has to be shown,
2. a heuristic is adopted in performing the $\sqcup$-rule, differently from the standard case where no heuristic is given,
3. the most plausible model for the concept E and individuals $x_1, \ldots, x_k$ is built with respect to the available knowledge $\mathcal{K}$, $\mathbf{D}$ and R. The obtained model is a *mixed model*, namely a model containing both information from R and $\mathcal{K}$. Differently, in the standard Tableaux algorithm the model that is built only refers to $\mathcal{K}$ and does not take into account the (assertional) available knowledge.

In the following these three characteristics are analyzed and the way for accomplish each of them is illustrated. First of all, the way in which the starting model $\mathcal{I}_r$ is built is illustrated. For each $x_i \in \{x_1, \ldots, x_k\}$, all attribute names A_i related to x_i are selected[4]

[4] As an example the following query may be performed: SELECT * FROM ⟨TABLE_NAME⟩ WHERE $A_i = x_i$. Alternatively, a subset of the attributes in $\mathbf{D}$ may be considered.

jointly with the corresponding attribute values a_i. The assertions $A_i(a_i)$ are added to $\mathcal{I}_r$. For simplicity and without loss of generality, a single individual x will be considered in the following. The generalization to multiple individuals is straightforward by simply applying the same procedure to all individuals that are (or assumed to be) instances of the considered concept.

Once the initial model $\mathcal{I}_r$ is built, all deterministic expansion rules, namely all but $\sqcup$-rule, are applied following the standard Tableaux algorithm as reported above. Instead, for the case of the $\sqcup$-rule, a heuristic is adopted. The goal of such a heuristic is twofold: a) choosing a new consistent ABox almost in one step to save computational complexity if $E(x)$ is consistent (see discussion above concerning the $\sqcup$-rule); b) driving the construction of the most plausible model given $\mathcal{K}$ and R. The approach for the assessing the heuristic is illustrated in the following.

Let $C \sqcup D$ be the disjunctive concept to be processed by $\sqcup$-rule. The choice on C rather than D (or vice versa) will be driven by the following process.

- The ARs (see Sect. 3.2) containing C (resp. D) or its negation in the *semantic items* of the right hand side of the rules are selected.
- Given the model under construction $\mathcal{I}_r$, the left hand side of each selected rule is considered and the degree of match is computed. This is done by counting the number of (both data and semantic) items in the left hand side of a rule that are contained in $\mathcal{I}_r$, and averaging this number w.r.t. the length of the left hand side of the rule. Items with uncertain (*unknown*) values are not taken into account. The degree of match for the rules whose (part of the) left hand side is contradictory w.r.t. to the model is set to 0.
- After the degree of match is computed, the rules having the degree of match equal to 0 are discarded.
- For each of the remaining rules the weighted confidence value is computed as $weightedConf = ruleConfidence * degreeOfMatch$.
- Rules that have the degree of match below a given threshold (e.g. 0.75) are discarded.
- The rule having the highest weighted confidence value is selected; in case of equal weighted confidence value of different rules, a random choice is performed.
- If the chosen rule contains $C = 1$ (resp. $D = 1$) in the right hand side, the model under construction $\mathcal{I}_r$ is enriched with $C(x)$ (resp. $D(x)$), where x is the individual under consideration.
- If the chosen rule contains $C = 0$ (resp. $D = 0$) in the right hand side, the model under construction $\mathcal{I}_r$ is enriched with $D(x)$ (resp. $C(x)$).
- In the general case the right hand side of the selected AR may contain additional items besides that involving C or D. Assertions concerning such additional items will be also added in $\mathcal{I}_r$ accordingly[5].

If there are no extracted ARs (satisfying a fixed confidence threshold) containing neither C or D in the right hand side, the following approach may be adopted.

[5] If a most conservative behavior of the heuristic has to be considered only the assertion concerning the disjunct C (resp. D) will be added in $\mathcal{I}_r$ while the additional items in the right hand side of the selected rules are not taken into account.

Given $\mathcal{I}_r$, a corresponding item set is created by transforming each assertion $A_i(a_i)$ referring to an attribute in $\mathbf{D}$ as a data item $A_i = a_i$, each concept and role assertion to a knowledge item. Specifically, each positive (not negated) assertion is transformed in $concept/role\ name = 1$, each negative assertion is transformed in $concept/role\ name = 0$. Let θ be the conventional name of such a built itemset. Four rules, $\theta \Rightarrow C = 1$, $\theta \Rightarrow C = 0$, $\theta \Rightarrow D = 1$ and $\theta \Rightarrow D = 0$ are created and their confidence value is computed (see Sect. 3.2). Then, the rule having the highest confidence (satisfying a given confidence threshold) value is selected and the corresponding right hand side will be used as a guideline for expanding $\mathcal{I}_r$.

The presented approach for the case in which no rules are available could result to be computationally expensive. As an alternative, the following criterion, grounded in the exploitation of the prior probability of C (resp. D) could be used. Specifically, the prior probability is computed, by adopting a frequency-based approach, as: $P(C) = |ext(C)|/|\mathcal{A}|$ where $ext(C)$ is the extension of C, namely the number of individuals that are instances (asserted or derived) of C and $|\cdot|$ returns the cardinality of the set extension. Similarly $P(D)$ can be defined for D. The concept to be chosen for extending $\mathcal{I}_r$ will be the one having the highest prior probability.

In the cases discussed above, the disjunctive expression is assumed to be made by atomic concept names. However, in $\mathcal{ALC}$, more complex expressions may occur as part of a disjunctive expression as: existential concept restrictions (i.e. $\exists R.A \sqcup \exists R.B$), universal concept restrictions (i.e. $\forall R.A \sqcup \forall S.B$), nested concept expression (i.e. $\exists R.\exists S.A$ or $\exists R.(A \sqcap B)$). To cope with these cases a straightforward solution is envisioned: new concept names are created for naming the cases listed above. In this way, a disjunction of atomic concept names is finally obtained. These new artificial concept names have to be added in the table representing the heterogeneous source of information (see Sect. 3.2) and the process for discovering ARs has to be run (see Sect. 3.2). This is because potentially useful ARs for treating the disjuncts may be found. It is important to note that the artificial concept names are not used for the process of discovering new knowledge in itself (as illustrated in Sect. 3.2) but only for the reasoning purpose presented in this section.

Now let us consider the following example concerning the demographic domain where the starting point for the inference is given in Tab. 3. Note that it is assumed that $Parent$ is $true$ for x_2. In the following the expansion of $(Male \sqcup Female)(x)$ for each of the objects in Tab. 3 is illustrated:

Table 3. Demographic example: data given at the inference stage

OBJECT	JOB	AGE	$Parent$	$Male$	$Female$
x_1	Primary school teacher	47	$unknown$	$unknown$	$unknown$
x_2	Policeman	25	$true$	$unknown$	$unknown$
x_3	Student	20	$unknown$	$unknown$	$unknown$

- x_1:
  $degreeOfMatch(rule_2)$ = 0,

$degreeOfMatch(rule_3)$ = 0,
$degreeOfMatch(rule_4)$ = 1,
$degreeOfMatch(rule_5)$ = 0,
thus the $Female$ decision is taken with $weightedConf = 0.78$.

- x_2:
  $degreeOfMatch(rule_2)$ = 1,
  $degreeOfMatch(rule_3)$ = 1,
  $degreeOfMatch(rule_4)$ = 0,
  $degreeOfMatch(rule_5)$ = 0,
  for both $rule_2$ and $rule_3$ $weightedConf = 0.75$ so we have a conflict here, and a random decision is taken.
- x_3:
  $degreeOfMatch(rule_2)$ = 0,
  $degreeOfMatch(rule_3)$ = 0,
  $degreeOfMatch(rule_4)$ = 0.5,
  $degreeOfMatch(rule_5)$ = 0,
  thus for $rule_3$ $weightedConf = 0.75 * 0.5 = 0.375$, which is below a given threshold (let it be 0.75), and so a random decision is taken.

As processing a disjunct expansion we always add assertions coming from the evidence of the available knowledge, the proposed approach should ensure that the model built is the one mostly compliant with the statistical regularities learned from data.

5 Discussion and concluding remarks

To summarize, in this paper we make a step towards the framework for knowledge representation and reasoning in which knowledge is specified by a mix of logical formulas and data linked together. We revise the preliminary results we presented in [5] by explicitly taking into account the *Open World Assumption* made in DLs. Differently from [9], where federated DBs are considered with the goal of removing structural conflicts automatically while maintaining unchanged the views of the different DBs, we focus on building a new knowledge base that is able to collect the complementary knowledge that is contained in heterogeneous sources of information. Eventually, we propose a method for data-driven logical reasoning, which is the result of combining logical reasoning and data mining methods embedded in a Tableaux algorithm. Differently from [6], where an integrated system ($\mathcal{AL}$-*log*) for knowledge representation based on DL and the deductive database language Datalog is presented, here purely relational databases are considered. Additionally, while in [6] a method for performing query answering based on constrained resolution is proposed, where the usual deduction procedure defined for Datalog is integrated with a method for reasoning on the structural knowledge, here a more expressive DL is considered and *semantically enriched* ARs are also exploited.

Our proposed mixed inference imitates in a way the cognitive reasoning process performed by humans. Indeed a human usually performs a logic reasoning process when he/she has knowledge that is assumed to be complete for a certain domain, for example, the medical domain. This step is represented in our case by the standard deductive approach. If some degrees of uncertainty occur, for instance there are strange symptoms that do not allow for a straightforward diagnosis, then *existing cases* are analyzed to support a given diagnosis or an alternative one. The existing cases would represent our external source of information (DBs and/or ARs). The process for determining a diagnosis is now driven by the integration of the logic reasoning process and inductive reasoning process that takes into account the additional cases and tries to produce a reasonable diagnosis given the additional available evidence.

The *most plausible model* that we build may be enriched with additional knowledge coming from the external source of information. Specifically, given the selected rule for resolving a disjunction (see Sect. 3.3), the information on the left hand side concerning only the external source of information could be added as part of the model under construction thus applying a sort of abductive inference. Alternatively, this additional knowledge may be exploited during the matching process for preferring a rule rather than another one (besides of the condition concerning the confidence of the rule). Particularly, as an additional criterion the level of match between the rule and the model under construction may be considered. The same would be done for additional information on the right hand side of a rule even if this case appears to be a bit more problematic.

For the future we aim at implementing the extended Tableaux algorithm for experimental purpose.

References

1. R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In P. Buneman and S. Jajodia, editors, *Proc. of the 1993 ACM SIGMOD Int. Conf. on Management of Data*, pages 207–216. ACM Press, 1993.
2. F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
3. T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 2001.
4. Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Mariano Rodriguez-Muro, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. The mastro system for ontology-based data access. *Semantic Web Journal*, 2(1):43–53, 2011.
5. C. d'Amato, V. Bryl, and L. Serafini. Semantic knowledge discovery from heterogeneous data sources. In H. Stuckenschmidt et al., editor, *Proc. of 18th Int. Conf. on Knowledge Engineering and Knowledge Management, EKAW 2012*, page To appear, 2012.
6. F. M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. $\mathcal{AL}$-log: Integrating datalog and description logics. *Journal of Intelligent Information Systems*, 10:227–252, 1998.
7. S. Grimm, B. Motik, and C. Preist. Variance in e-business service discovery. In *Proceedings of the ISWC Workshop on Semantic Web Services*, 2004.
8. David J. Hand, Padhraic Smyth, and Heikki Mannila. *Principles of data mining*. MIT Press, Cambridge, MA, USA, 2001.
9. S. Spaccapietra and C. Parent. View integration: A step forward in solving structural conflicts. *IEEE TRANS. On Knowledge and Data Engineering*, 6(2):258–274, 1994.
10. Ian Witten, Eibe Frank, and Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques, 3rd Edition*. Morgan Kaufman, 2011.

An Experimental Evaluation of a Scalable Probabilistic Description Logic Approach for Semantic Link Prediction

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

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

Abstract. In previous work, we presented an approach for link prediction using a probabilistic description logic, named CR$\mathcal{ALC}$. Inference in CR$\mathcal{ALC}$, considering all the social network individuals, was used for suggesting or not a link. Despite the preliminary experiments have shown the potential of the approach, it seems unsuitable for real world scenarios, since in the presence of a social network with many individuals and evidences about them, the inference was unfeasible. Therefore, we extended our approach through the consideration of graph-based features to reduce the space of individuals used in inference. In this paper, we evaluate empirically this modification comparing it with standard proposals. It was possible to verify that this strategy does not decrease the quality of the results and makes the approach scalable.

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 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 [13] aims at predicting whether two nodes (i.e. people) should be connected given that we know previous information about their relationships or interests.

In [13] a survey with some representative link prediction methods, categorized in three groups, was presented. In the first group, feature-based methods construct pair-wise features to use in a classification task. The majority of the features are extracted from the graph topology computing the similarity based on the neighborhood of the pair of nodes or based on ensembles of paths between the pair of nodes [10]. Recently, semantic informations have also being used as features [21, 17]. The second group includes probabilistic approaches which model the joint-probability among the entities in a network by Bayesian graphical models [20]. And, finally the third group concerns linear algebraic approaches which computes the similarity between the nodes in a network by rank-reduced similarity matrices [9].

In [15] we presented an approach that uses a Bayesian graphical model together with semantic-based features for semantic link prediction. Therefore, our proposal lies on the first two categories described previously. To model the domain and thus consider semantic-based features, the proposal adopted a probabilistic description logic called Credal $\mathcal{ALC}$ (CR$\mathcal{ALC}$) [5], that extends the popular logic $\mathcal{ALC}$ [3] with *probabilistic inclusions*. These are 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. Exact and approximate inference algorithms have been proposed [5], using ideas inherited from the theory of Relational Bayesian Networks (RBN) [8]. In [14], we extended our proposal to also consider graph-based approaches in order to scale for large social network. In this paper we conduct some experimental analisis in order to verify the benefits of our proposal.

The paper is organized as follows. Section 2.1 reviews basic concepts of probabilistic description logics, CR$\mathcal{ALC}$ and our proposal for a scalable semantic link prediction approach. Section 3 describes the experiments we conducted bringing some discussions. Section 4 concludes the paper.

2 Background

In this section, probabilistic description logic CR$\mathcal{ALC}$ and our former proposal for semantic link prediction are reviewed.

2.1 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) [3]. Knowledge is expressed in terms of *individuals*, *concepts*, and *roles*. The semantics 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 have appeared in the literature [11]. Heinsohn [7] and Sebastiani [18] 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: probabilities 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* semantic.

The probabilistic description logic 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. The graph $\mathcal{G}(\mathcal{T})$ is a Relational Bayesian Network (RBN) [8].

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 in the left depicts $\mathcal{G}(\mathcal{T})$, while the graph in the right illustrates the grounding of $\mathcal{G}(\mathcal{T})$ for a domain with two individuals ($\mathcal{D} = \{a, b\}$).

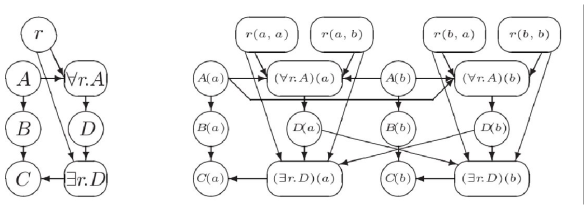


Fig. 1. $\mathcal{G}(\mathcal{T})$ for terminology $\mathcal{T}$ in Example 1 and its grounding for domain$\mathcal{D} = \{a, b\}$.

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, generating a grounding RBN, where one slice is built for each individual. Therefore, not always exact probabilistic inference is possible. In [5], a first order loopy propagation algorithm was proposed for approximate calculations.

2.2 Link Prediction with CR$\mathcal{ALC}$

Given a social network $\mathcal{N}$, where nodes are entities (represented by letters $a, b, c, \ldots$), one is interested in defining whether a link between a and b is suitable given that there is no link between these nodes in $\mathcal{N}$. In [15], interests, i.e., semantics between the nodes were modeled through a probabilistic ontology represented by the probabilistic description logic CR$\mathcal{ALC}$. In addition, in [14] graph path information was used to improve probabilistic inference. In summary, the semantic link prediction task proposed in [15] (and improved in [14]) can be described as:

Given:

- a network $\mathcal{N}$ defining relationship between objects;
- an ontology $\mathcal{O}$ in CR$\mathcal{ALC}$ describing the domain of the objects;
- the ontology concept $\mathcal{C}$ that defines the semantics of the network objects;
- the ontology role $r(\_,\_)$ that defines the semantics of the relationship between network objects;

Find:

- a revised network $\mathcal{N}_f$ with new relationship between objects.

The proposed algorithm for link prediction receives a network of a specific domain. For instance, in a co-authorship network the nodes represent researchers and the relationship can have the semantics ”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 and Publication and roles like hasPublication, hasSameInstitution and sharePublication. This ontology can be learned automatically through a learning algorithm as the ones proposed in [16]. Thus, the nodes represent instances of one of the concepts described in the probabilistic description logic CR$\mathcal{ALC}$ and the semantics of the links is described by one of the roles in the probabilistic description logic 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 $\mathcal{C}$ defined as the concept that provides the semantics for the network nodes — this is a general setting, as a rule the set of possible pairs is restricted. For each pair, it checks whether a link between the corresponding nodes exist in the network. If not the probability of the link is calculated through the probability of the defined role conditioned on evidences (step 5). The evidences are provided by the instances of the ontology. The number of instances in an ontology has a great impact in inference. Usually one considers that more instances better inference. However, evidences for different individuals can turn out the inference process computationally expensive, since in a RBN a slice is created for each individual,

```
Require: a network $\mathcal{N}$, an ontology $\mathcal{O}$, the role $r(\_,\_)$ representing the semantics of
    the network link, the concept $\mathcal{C}$ describing the objects of the network and a
    $threshold$.
Ensure: a revised network $\mathcal{N}_f$
 1: define $\mathcal{N}_f$ as $\mathcal{N}$;
 2: for all pair of instances $(a, b)$ of concept $\mathcal{C}$ do
 3:   if does not exist a link between nodes $a$ and $b$ in the network $\mathcal{N}$ then
 4:     compute evidence based on $a$, $b$ and nodes in their path;
 5:     infer probability $P(r(a, b)|evidence)$ using the RBN created through the
        ontology $\mathcal{O}$;
 6:     if $P(r(a, b)|evidence) > threshold$ then
 7:       add a link between $a$ and $b$ in network $\mathcal{N}_f$;
 8:     end if
 9:   end if
10: end for
```

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

and then inference should be done for each slice. In [5], an approximate inference algorithm was proposed where all slices without evidence are consolidated in a unique slice, thus making inference feasible in real domains. Therefore, less individuals with evidence faster inference is. From another perspective we are interested in predicting a relationship between two individuals, a and b. Therefore, evidences for these two individuals and other individuals strongly related to them are more relevant for link prediction than evidences from other individuals in the network. Thus, in [14] we extended our semantic link prediction approach in order to consider evidences about a, b and the individuals in their path, which makes the link prediction problem scalable for large networks. Therefore, in step 4 the nodes (individuals) belonging to the path between a and b are found. The inference is then performed through CR$\mathcal{ALC}$ lifted variational method on ontology $\mathcal{O}$. If the probability inferred is greater than a threshold then the corresponding link 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.

3 Experiments

In order to evaluate our previously proposed approach for semantic link prediction empirical experiments were performed. To do so, a real world dataset was used and our algorithm was combined with state-of-the-art measures on a classification model for link prediction. This section reports on steps involved in this process.

3.1 Scenario Description

The Lattes Platform is the public repository of Brazilian scientific curriculum which is comprised by approximately a million of registered researchers. Information is given in HTML format, and ranges from personal information such as name and address to a list of publications, examination board participations, research areas, research projects and advising/advisor information. There is implicit relational information in these web pages, for instance collaboration networks are given by advising/adviser links, shared publications and so on. We have randomly selected a set of 1100 researchers from engineering and math backgrounds and based on assertional data about these researchers a probabilistic ontology has been learned. To perform link prediction, this ontology has also been extended with some probabilistic roles — learning is mainly addressed to probabilistic inclusions and concepts. Part of the revised ontology is as follows.

$$\begin{array}{ll}
 & P(\mathsf{Publication}) = 0.3 \\
 & P(\mathsf{Board}) = 0.33 \\
 & P(\mathsf{sharePublication}) = 0.22 \\
 & P(\mathsf{wasAdvised}) = 0.05 \\
 & P(\mathsf{hasSameInstitution}) = 0.14 \\
 & P(\mathsf{sameExaminationBoard}) = 0.31 \\
\mathsf{ResearcherLattes} \equiv & \mathsf{Person} \\
 & \sqcap(\exists \mathsf{hasPublication.Publication} \\
 & \sqcap\exists \mathsf{advises.Person} \sqcap \exists \mathsf{participate.Board}) \\
P(\mathsf{PublicationCollaborator} & \mid \mathsf{Researcher} \sqcap \exists \mathsf{sharePublication.Researcher}) = 0.91 \\
P(\mathsf{SupervisionCollaborator} & \mid \mathsf{Researcher} \sqcap \exists \mathsf{wasAdvised.Researcher}) = 0.94 \\
P(\mathsf{SameInstitution} & \mid \mathsf{Researcher} \sqcap \exists \mathsf{hasSameInstitution.Researcher}) = 0.92 \\
P(\mathsf{SameBoard} & \mid \mathsf{Researcher} \sqcap \\
 & \exists \mathsf{sameExaminationBoard.Researcher}) = 0.95 \\
P(\mathsf{NearCollaborator} & \mid \mathsf{Researcher} \sqcap \exists \mathsf{sharePublication}.\exists \mathsf{hasSameInstitution}. \\
 & \exists \mathsf{sharePublication.Researcher}) = 0.95 \\
\mathsf{FacultyNearCollaborator} \equiv & \mathsf{NearCollaborator} \\
 & \sqcap\ \exists \mathsf{sameExaminationBoard.Researcher} \\
P(\mathsf{NullMobilityResearcher} & \mid \mathsf{Researcher} \sqcap \exists \mathsf{wasAdvised}. \\
 & \exists \mathsf{hasSameInstitution.Researcher}) = 0.98 \\
\mathsf{StrongRelatedResearcher} \equiv & \mathsf{Researcher} \\
 & \sqcap\ (\exists \mathsf{sharePublication.Researcher} \sqcap \\
 & \exists \mathsf{wasAdvised.Researcher}) \\
\mathsf{InheritedResearcher} \equiv & \mathsf{Researcher} \\
 & \sqcap\ (\exists \mathsf{sameExaminationBoard.Researcher} \sqcap \\
 & \exists \mathsf{wasAdvised.Researcher})
\end{array}$$

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 (instances of concept Researcher) in a collaboration graph. Therefore, the concept Researcher and the role sharePublication are inputs to the algorithm we proposed in Algorithm 1. Moreover, their instances were used to define a collaboration network, which was also provided to the algorithm. Topological graph information was computed accordingly. Figure 2 depicts a subset of collaborations among researchers. To perform inferences and therefore to obtain link predictions we resort to the lifted algorithm in CR$\mathcal{ALC}$.

If we carefully inspect this collaboration graph we could be interested, for instance, in predicting links among researchers from different groups. Since filling form is prone to errors, there is uncertainty regarding real collaborations. Thus, in Figure 2 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 B is calculated, $P(link(R, B)|E)$, where E denotes evidence about researchers such as

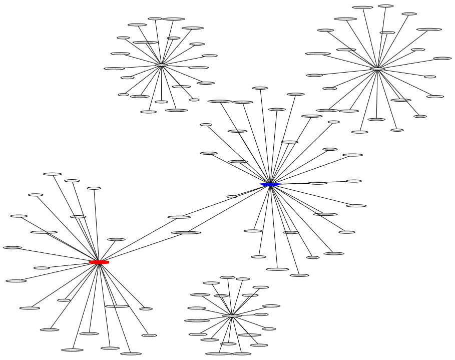


Fig. 2. Lattes collaboration network.

publications, institution, examination board participations and so on. Since the role sharePublication defines the semantics of the links in the graph, it is through it that we must calculate $P(link(R,B)|E)$. 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:

$$
\begin{aligned}
P(\mathsf{PublicationCollaborator(R)}\ |&\mathsf{Researcher(R)} \\
&\sqcap\exists\mathsf{hasSameInstitution}.\mathsf{Researcher}(B)) = 0.57.
\end{aligned}
$$

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 2), denoted by I_1, I_2. The inference would be

$$
\begin{aligned}
P(\mathsf{PublicationCollaborator(R)}\ |&\mathsf{Researcher(R)} \\
&\sqcap\exists\mathsf{sharePublication}(I_1).\exists\mathsf{sharePublication(B)} \\
&\sqcap\exists\mathsf{sharePublication}(I_2).\exists\mathsf{sharePublication(B)}) = 0.65.
\end{aligned}
$$

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

In order to compare with existing baseline algorithms, topological and semantic features have also been defined. Further details are given as follows.

3.2 Methodology

In this section we describe our main design choices to run experiments. According to cross validation principles, our dataset (1100 researchers which give rise to

1400 true co-authoring links) has been divided in training and validation sets. To avoid skeweness (due to unbalanced classes), every fold is comprised by balanced negative and positive instances, where positive instances correspond to a link between two nodes while negative instance means that there is not a link between these two nodes.

In order to classify possible co-authoring links and therefore to perform comparisons with previous approaches we resort to the Logistic regression classification algorithm.

In a classification approach for link prediction, features are commonly extracted from topological graph properties such as neighborhood and paths between nodes. In addition, numerical features are also computed from joint probability distributions and semantics.

Two baseline graph-based numerical features have been used in our experiments. First, the Katz measure [10] is a weighted sum of the number of paths in the graph that connect two nodes, with higher weight for shorter paths. This leads to the following formula:

$$Katz(x,y) = \sum_{i=1}^{\infty} \beta^i p_i$$

where p_i is the number of paths of length i connecting x and y, while β (≤ 1) parameter is used to regularize this feature. A small value of β considers only the shorter paths.

Since computing all paths (∞) is expensive we only consider paths of length at most four ($i \leq 4$).

The second numerical feature is the Adamic-Adar measure [1] which computes the similarity between two nodes in a graph. Let $\mathit{\Gamma}(x)$ be the set of all neighbors of node x. Then the similarity between two nodes x, y is given by

$$\text{Adamic-Adar}(x,y) = \sum_{z \in \mathit{\Gamma}(x) \cap \mathit{\Gamma}(y)} \frac{1}{\log |\mathit{\Gamma}(z)|}$$

The intuition behind the score is that instead of simply counting the number of neighbors shared by two nodes, we should weight the hub nodes less and rarer nodes more. In this way, Adamic-Adar weighs the common neighbors with smaller degree more heavily.

We have also considered semantic features. The degree of semantic similarity among entities is something that can be useful to predict links that might not be captured by either topological or frequency-based features [20]. In this work, for each author a document with the words appearing in the title of his publications (removing stop words) is considered. Thus, an author is represented as a set of words, which allow us to compute two features based on semantic similarity:

i The keyword match count between two authors [6].
ii The cosine between the TFIDF features vectors of two authors [20].

To compute (ii), we derive a bag of words representation for each author, weighting each word by its TFIDF (Term Frequency - Inverse Document Frequency) measure. The TFIDF weighting scheme assigns to term t a weight in document d given by

$$\mathrm{TFIDF}_{t,d} = \mathrm{TF}_{t,d} \times \mathrm{IDF}_t$$

$\mathrm{TF}_{t,d}$ is the term frequency in d, and IDF_t denotes the inverse document frequency of t which is given by $\mathrm{IDF}_t = \log \frac{\mathrm{N}}{\mathrm{DF}_t}$, where N is the total number of documents and DF_t is the number of documents containing the term.

The standard way of quantifying the similarity between two documents d_1 and d_2 is to compute the cosine similarity of their vector representations $\overrightarrow{V}(d_1)$ and $\overrightarrow{V}(d_2)$

$$\mathrm{cosine}(d_1, d_2) = \frac{\overrightarrow{V}(d_1) \cdot \overrightarrow{V}(d_2)}{|\overrightarrow{V}(d_1)||\overrightarrow{V}(d_2)|}$$

where the numerator represents the dot product (also known as the inner product) of the vectors $\overrightarrow{V}(d_1)$ and $\overrightarrow{V}(d_2)$, while the denominator is the product of their Euclidean lengths.

Finally, we also use the probability, $P(r(x,y)|evidence)$, given by our probabilistic description logic model, as a numerical feature in the classification model. We wish to investigate whether this probabilistic logic measure can improve the classification approach for link prediction.

3.3 Results

In order to evaluate suitability of our approach in predicting co-authorships in the Lattes dataset, three experiments were run. In the first experiment two baseline scores, Katz and Adamic-Adar, have been used as features in the logistic regression algorithm. After a ten-fold cross validation process the classification algorithm yielded results on accuracy which are depicted in Table 1.

Given the Lattes dataset, one can see that the Katz feature yields the best accuracy (75.49%) when the two topological features are used in isolation. Katz has been shown to be among the most effective topological measures for the link prediction task [10]. Furthermore, when we combine the Katz and the Adamic-Adar features, we improve the accuracy to 76.44%.

Table 1. Classification results on accuracy (%) for baseline features: Adamic-Adar (Adamic), Katz and a combined one (Adamic+Katz)

	Adamic	Katz	Adamic+Katz
Lattes dataset	72.75 ± 1.87	75.49 ± 2.07	**76.44 ± 2.03**

In the second experiment, we evalute two features based on semantic similarity and their combination with topological features. Results on accuracy for these semantic features are depicted in Table 2. The cosine similarity feature performs better than matching keyword feature and outerperforms the two former topological features. This feature alone yields 82.45% on accuracy. When we combine all the four features together, there is an improvement in accuracy to 85.63%.

Table 2. Classification results on accuracy(%) for semantic similarity features: matching keyword (match) and cosine similarity (cosine) and topological features.

	match	cosine	Adamic+Katz+match+cosine
Lattes dataset	69.42 ± 2.66	82.45 ± 1.37	**85.63 ± 1.23**

In the third experiment, a probabilistic feature based on our probabilistic description logic approach was introduced into the model. Results on accuracy for this feature are depicted in Table 3. The probabilistic description logic feature performs better than the other features. This feature yields 87.72% on accuracy. When we combine all the five features together, there is an improvement in accuracy to 89.48%.

Table 3. Classification results on accuracy(%) for probabilistic description logics and baseline features: CR$\mathcal{ALC}$ based (cralc) and Adamic-Adar, Katz, match, cosine, CR$\mathcal{ALC}$ (Adamic+Katz+match+cosine+cralc).

	cralc	Adamic+Katz+match+cosine+cralc
Lattes dataset	87.72 ± 0.52	**89.48 ± 0.96**

It is worth noting that the probabilistic logic feature probability outer performs all other features and allow us to improve the classification model for link prediction on accuracy.

Nothing prevent us to define ad-hoc probabilistic networks to estimate link probabilities. However, by doing so we are expected to define a large propositionalized network (a relational Bayesian network) [15] or estimate local probabilistic networks [20]. These approaches do not scale well since computing probabilistic inference for large networks is expensive.

To overcome these performance and scalability issues, we resort to lifted inference in CR$\mathcal{ALC}$ which is based on variational methods — tunned by evidence defined according nodes's neighborhood. Thus, for a ten thousand network, if evidence is given for 5 nodes, then there is only 6 slices which have messages interchanged.

In our experiments, the average runtime for inference in CR$\mathcal{ALC}$ (1100 nodes network) was 135 milliseconds. Table 4 depicts some runtime results for larger networks which demonstrates the scalability of our approach.

Table 4. Average runtime for inference.

nodes	runtime(milliseconds)
1100	135
10000	168
100000	175
1000000	185

On the other hand, a propositionalized relational Bayesian network fails to run inference due to out of memory issues.

4 Conclusion

In [15, 14] we have presented an approach for predicting links that resorts to both graph-based and ontological information. Given a collaborative network, we encode interests and graph features through a CR$\mathcal{ALC}$ probabilistic ontology. In order to predict links we resort to probabilistic inference, where only information about two nodes being analyzed and the nodes in their path are used as evidence. Thus, making the proposal scalable. In this paper, we evaluated our proposal focused on an academic domain, and we aimed at predicting links among researchers. The approach was successfully compared with graph-based and semantic-based features. As future work we intend to consider other datasets.

Previous combined approaches for link prediction [4, 2] have focused on machine learning algorithms [12]. 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. Unless 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. Moreover, in [19], a Probabilistic Relational Model is used for link prediction task. This is one of the approaches more closed to ours, since uses semantic features considering a probabilistic graphical model. However, inference is done in a propositionalized network that can not scale for large networks.

Acknowledgements

The third author is partially supported by CNPq. The work reported here has received substantial support by FAPESP grant 2008/03995-5 and FAPERJ grant E-26/111484/2010.

References

1. L.A. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25:211–230, 2001.

2. 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.
3. F. Baader and W. Nutt. Basic description logics. In *Description Logic Handbook*, pages 47–100. Cambridge University Press, 2007.
4. D. Caragea, V. Bahirwani, W. Aljandal, and W.H. Hsu. Ontology-based link prediction in the livejournal social network. In *SARA'09*, pages 1–1, 2009.
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. Mohammad Al Hasan, Vineet Chaoji, Saeed Salem, and Mohammed Zaki. Link prediction using supervised learning. In *In Proc. of SDM 06 workshop on Link Analysis, Counterterrorism and Security*, 2006.
7. J. Heinsohn. Probabilistic description logics. In *International Conf. on Uncertainty in Artificial Intelligence*, pages 311–318, 1994.
8. M. Jaeger. Relational Bayesian networks: a survey. *Linkoping Electronic Articles in Computer and Information Science*, 6, 2002.
9. J. Kunegis and A. Lommatzsch. Learning spectral graph transformations for link prediction. In *Proceedings of the ICML*, pages 561–568, 2009.
10. D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. *American Society for Information Science and Technology*, 7(58):1019–1031, 2007.
11. T. Lukasiewicz and U. Straccia. Managing uncertainty and vagueness in description logics for the semantic web. *Semantic Web Journal*, 6(4):291–308, 2008.
12. T. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.
13. A.H. Mohammad and J.Z. Mohammed. A survey of link prediction in social networks. In *Social Network Data Analytics*, pages 243–275. 2011.
14. J. Ochoa-Luna, K. Revoredo, and F.G. Cozman. A scalable semantic link prediction approach through probabilistic description logics. In *In proceeding of 9th Artificial Intelligence National Meeting (ENIA) - to appear*, 2012.
15. K. Revoredo, J. Ochoa-Luna, and F.G. Cozman. Semantic link prediction through probabilistic description logics. In *Bobillo, F., et al. (eds.) Proceedings of the 7th International Workshop on URSW*, volume 778, pages 87–97.
16. K. Revoredo, J. Ochoa-Luna, and F.G. Cozman. Learning probabilistic description logics: A framework and algorithms. In *In proceedings of the MICAI*, volume 7094 of *LNCS*, pages 28–39. Springer, 2011.
17. 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.
18. 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.
19. B. Taskar, M. faiWong, P. Abbeel, and D. Koller. Link prediction in relational data. In *Proceedings of the 17th NIPS*, 2003.
20. C. Wang, V. Satuluri, and S. Parthasarathy. Local probabilistic models for link prediction. In *Proceedings of the 2007 Seventh IEEE ICDM*, pages 322–331, Washington, DC, USA, 2007. IEEE Computer Society.
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.

Graph Summarization in Annotated Data Using Probabilistic Soft Logic

Alex Memory[1], Angelika Kimmig[1,2], Stephen H. Bach[1], Louiqa Raschid[1] and Lise Getoor[1]

[1] University of Maryland
[2] KU Leuven

Abstract. Annotation graphs, made available through the Linked Data initiative and Semantic Web, have significant scientific value. However, their increasing complexity makes it difficult to fully exploit this value. Graph summaries, which group similar entities and relations for a more abstract view on the data, can help alleviate this problem, but new methods for graph summarization are needed that handle uncertainty present within and across these sources. Here, we propose the use of probabilistic soft logic (PSL) [1] as a general framework for reasoning about annotation graphs, similarities, and the possibly confounding evidence arising from these. We show preliminary results using two simple graph summarization heuristics in PSL for a plant biology domain.

1 Introduction

The Linked Data initiative and Semantic Web technologies have been very successful in providing access to a diversity of data collections. Of particular interest are *annotation graphs*, where scientific concepts are tagged with controlled vocabulary terms from ontologies or thesauri. As these collections grow, tools and techniques to analyze, explore and inspect such data become ever more important. In this paper we consider the problem of mining the richly curated annotation graphs, in conjunction with the wealth of semantic knowledge encoded within ontologies, to create graph summaries. Graph summaries group entities and relations based on similarity as well as local graph structure, thus creating a graph at a higher level of abstraction that can be easier to analyze. This can help the scientist to understand the underlying evidence, to find patterns, and to make predictions.

Linked Data can provide multiple rich and possibly confounding sources of evidence about concepts. As a motivating example, we consider an annotation graph from the domain of plant biology. The nodes in this graph are genes from the model organism Arabidopsis thaliana (these are the concepts) as well as terms from both the Gene Ontology (GO) and the Plant Ontology (PO) (these are the annotations). Edges represent annotations of genes with such terms. Other sources of information of interest include sequence-based similarity between pairs of genes, co-occurrence frequencies of pairs of GO terms, taxonomic distances between pairs of PO or pairs of GO terms, etc. This evidence may be confounding; for example, genes can have high sequence based similarity to other genes in their same family. However, more useful evidence may be that they share high GO functional similarity with genes in unrelated families (with or without high sequence similarity).

We propose the use of probabilistic soft logic (PSL) [1] as a general framework for reasoning about annotation graphs, similarities, and the possibly confounding evidence arising from these. PSL is a framework for collective, probabilistic reasoning in relational domains that directly exploits available similarities. It uses rules to capture the dependency structure of the domain, based on which it builds a joint probabilistic model over the data. This allows us to easily encode the annotation graph, similarity information for nodes, and a number of graph summarization heuristics, and to explore the effect of these heuristics on the resulting graph summaries. In this work, we show preliminary results from two simple heuristics.

2 Motivating Example and Problem Setting

We use an example annotation graph from the plant biology domain to present our goals for graph summarization. We also use this domain for our experimental evaluation in Section 6. The graph represents gene annotation data for the model organism Arabidopsis thaliana, which originates in The Arabidopsis Information Resource (TAIR).[3] Each gene in TAIR is *annotated* with terms from the Plant Ontology (PO) and from the Gene Ontology (GO). A fragment of the resulting annotation graph is illustrated in Figure 1, with PO terms on the left, genes in the center, and GO terms on the right.

For a scientist exploring a set of genes of interest within a biological context, e.g., genes related to light-mediated development, finding regularities in such a graph can provide useful information. Our goal is to facilitate this process by providing *summaries* of the graph, that is, by grouping together nodes (and edges). The grouping can exploit multiple sources of evidence including explicit similarity between pairs of nodes, or shared annotations. For ease of illustration, we drastically simplify the graph to the topmost part of Figure 1, shown on the left in Figure 2. On the right, Figure 2 shows a possible graph summary, where three pairs of nodes have been grouped into three supernodes or clusters, and sets of edges between all pairs of nodes in adjacent clusters are represented by single edges between clusters. However, for real-world graphs, many clusterings are possible, and so different heuristics and combinations of heuristics may be appropriate for different graphs. In this work, we show how two such heuristics can be easily incorporated into a probabilistic framework, but others are certainly possible. Future work can extend this approach by incorporating additional heuristics and adapting heuristics to different graph-summarization tasks.

[3] `http://www.arabidopsis.org`

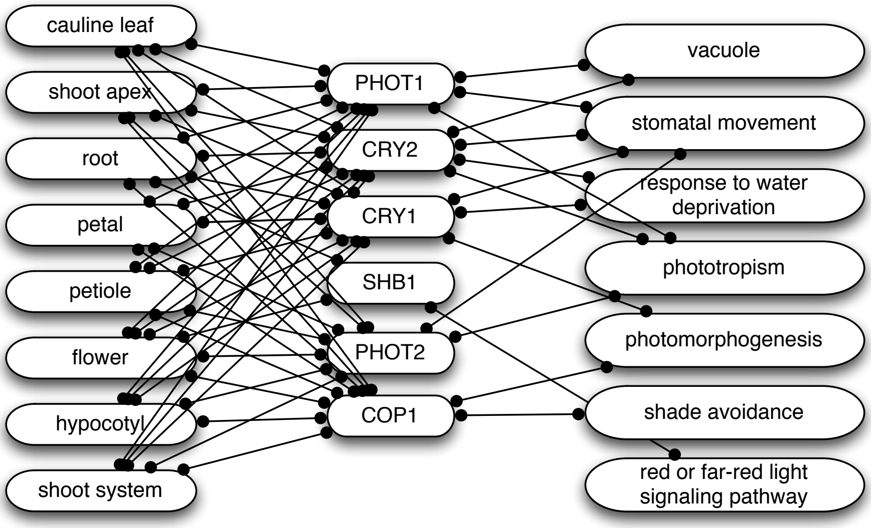


Fig. 1. Part of the annotation graph: PO terms (left), genes (middle), and GO terms (right).

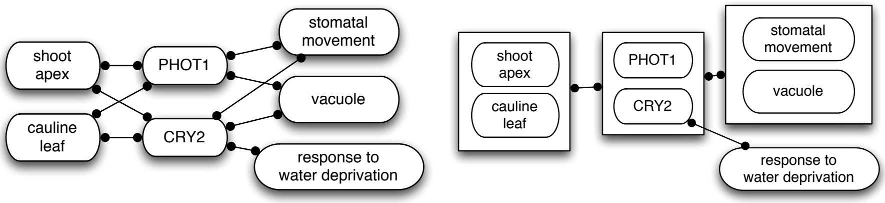


Fig. 2. Example subgraph (left) and a possible summary for it (right).

The first is an *annotation link heuristic*: We would like to cluster nodes that share a large fraction of their neighbors in the annotation graph. For instance, the two PO terms "cauline leaf" and "shoot apex" both annotate genes PHOT1 and CRY2 in our example, and there are no genes that are annotated with only one of these terms. The terms are thus similar in terms of the link structure they participate in, which supports clustering them. On the GO side, the same argument holds for "vacuole" and "stomatal movement", but not for "response to water deprivation", which only annotates CRY2. Clearly, the direct link structure alone does not provide sufficient evidence to decide whether the latter term should be added to the GO cluster or not. Choosing to include the term in the cluster would correspond to implicitly assuming that the term should actually annotate PHOT1 as well, and thus allow one to *predict* a new link, whereas the latter would tend more towards accepting the absence of such a link. Finally, we observe that the two gene nodes share four out of their five neighbors, which can still be viewed as a relatively strong indication to cluster them.

Next, we consider explicit similarities between pairs of nodes. Such additional information could help deciding whether the third GO term should be included in the cluster. For this we use the sequence based similarity between pairs of genes and information retrieval based metrics between pairs of annotation terms. For instance, amongst the extensive statistics published by the GO Consortium, the *annotation co-occurrence* of pairs of GO terms has significant biological meaning and is a good predictor of new function. For the GO term `response to water deprivation`, `stomatal movement` is the 5th highest co-occurring term; for the reverse case the rank is 11. Incorporating a similarity measure between GO terms into the graph summarization process might thus provide additional evidence in favor of clustering all three terms in the example and further predicting the "response to water deprivation" annotation on PHOT1. This would help the biologist understand the new functional annotation of PHOT1 and also understand that there is a posssible interaction between CRY2 and PHOT1.

To be able to exploit this similarity information, we introduce a *similarity heuristic*: we prefer to cluster nodes that are similar according to some available similarity measure. Recall however that this may also introduce conflicting evidence. For instance, the two genes in our example belong to different groups of blue light receptors and are therefore dissimilar in terms of sequence similarity, but similar in terms of their annotations in the graph.

We integrate the multiple types of evidence from the annotation links, the various similarity metrics, and the two graph summarization heuristics within a probabilistic model using PSL. We discuss this model in more detail in Section 5, after an introduction to PSL in Section 4.

3 Related Work

Graph summarization as broadly considered in this paper is a form of multi-relational clustering that exploits attributes of the nodes or objects to be clustered, as well as additional relational features or properties in which these nodes participate [2–4]. Multi-relational clustering aims at grouping nodes in heterogeneous, multi-relational networks, i.e., networks with both multiple types of nodes and multiple types of relationships between nodes. The clusters group nodes based on their similarities, where the value is either given explicitly, or derived from node attributes or relations between nodes of the same or different type(s). There is a large body of work on multi-relational clustering, and methods include matrix factorization approaches, generative models, and other optimization methods. Other work on graph summarization explores summarization techniques that can be tailored to user needs and which scale to large graphs with minimal loss [5–8]. Our proposed approach makes use of the notion of *examplars*, used in methods such as affinity propagation [9], to denote the elements which are chosen as the canonical representation for nodes in each cluster. Besides multi-relational clustering and graph summarization, there is a broad range of other mining and analysis techniques for heterogeneous information networks, cf. for instance [10].

Probabilistic soft logic (PSL) [1] combines ideas from fuzzy logic [11] and graphical models. Similar to Markov Logic [12], it uses first order logic as a template language for a graphical model. However, its use of soft truth values turns inference from a discrete into a continuous optimization task, which can be solved efficiently.

4 Probabilistic Soft Logic

Probabilistic soft logic (PSL) [1] is a framework for collective, probabilistic reasoning in relational domains. PSL uses rules to capture the dependency structure of the domain, based on which it builds a joint probabilistic model over all atoms. Each rule has an associated non-negative weight that captures the rule's relative importance. Furthermore, PSL uses soft truth values in the interval $[0, 1]$, which allows one to directly incorporate similarity functions into the logical model. We refer to Broecheler et al. [1] for full technical details and instead illustrate the key concepts in the context of the following example program:

$$w_1 : \mathit{exemplar}(A, B) \rightarrow \mathit{similar}(A, B) \tag{1}$$

$$w_2 : \mathit{link}(A, B) \wedge \mathit{exemplar}(A, C) \wedge \mathit{exemplar}(D, C) \rightarrow \mathit{link}(D, B) \tag{2}$$

Here, for simplicity of presentation, we assume $w_1 = w_2 = 1$. Consider any concrete nodes a, b, c, and d instantiating logical variables A, B, C, and D respectively. The first rule states that if a is in the cluster exemplified by b, they should be similar (similarity heuristic), whereas the second states that if a and d are both in the cluster exemplified by c, and a has a link to b, then d should also have a link to b (link heuristic). While PSL shares the syntax of its rules with first order logic, PSL uses *soft truth values* from the interval $[0, 1]$ instead of its extremes 0 (false) and 1 (true) only. Given a set of atoms $\ell = \{\ell_1, \dots, \ell_n\}$, we call the mapping $I : \ell \rightarrow [0, 1]^n$ from atoms to soft truth values an *interpretation*. PSL defines a probability distribution over interpretations that makes those satisfying more ground rule instances more probable.

To determine the degree to which a ground rule is satisfied, PSL uses the *Lukasiewicz t-norm* and its corresponding *co-norm* as the relaxation of the logical AND and OR, respectively. These relaxations are exact at the extremes, but provide a consistent mapping for values in-between. Given an interpretation I, the formulas for the relaxation of the logical conjunction ($\wedge$), disjunction ($\vee$), and negation ($\neg$) are as follows:

$$\begin{aligned} \ell_1 \,\tilde{\wedge}\, \ell_2 &= \max\{0, I(\ell_1) + I(\ell_2) - 1\}, \\ \ell_1 \,\tilde{\vee}\, \ell_2 &= \min\{I(\ell_1) + I(\ell_2), 1\}, \\ \tilde{\neg}\, l_1 &= 1 - I(\ell_1), \end{aligned}$$

where we use $\tilde{}$ to indicate the relaxation from the Boolean domain. For a ground rule $r \equiv r_{body} \rightarrow r_{head} \equiv \tilde{\neg}\, r_{body} \,\tilde{\vee}\, r_{head}$, where r_{body} and r_{head} are logical formulas composed of atoms and the logical operators defined above, an interpretation

I over the atoms in r determines whether r is satisfied, and, if not, its distance to satisfaction. Abusing notation, we can expand the usage of I to also denote the truth assignments to logical formulas induced by assignments to atoms and applying the definitions of the logical operators in the formula, i.e., $I(r)$ is the truth value that results from applying the logical operators in r to the truth values of atoms in r given by I. Then, given I, r is satisfied, i.e., $I(r) = 1$, if and only if $I(r_{body}) \leq I(r_{head})$, that is, the head has at least the same truth value as the body. Again, this coincides with the usual definition of satisfaction of a rule when truth values are restricted to 0 and 1. The rule's *distance to satisfaction* under interpretation I then measures the degree to which this condition is violated:

$$d_r(I) = \max\{0, I(r_{body}) - I(r_{head})\} \tag{3}$$

For instance, consider the interpretation $I = \{link(a,b) \mapsto 1, exemplar(a,c) \mapsto 0.9, exemplar(d,c) \mapsto 0.8, link(d,b) \mapsto 0\}$ and let r be the corresponding ground instance of Rule (2) above. We get $I(r_{body}) = \max\{0, 1 + 0.8 + 0.9 - 2\} = 0.7$ and thus $d_r(I) = \max\{0, 0.7 - 0\} = 0.7$, whereas the distance would be 0 if the head had truth value 0.7 or greater.

Given a set of atoms ℓ of interest, a PSL program induces a distribution over possible interpretations I. ℓ first induces a set of ground rules R, which contains every possible ground rule r such that r can be obtained by performing variable substitution on one of the rules in the program and each atom mentioned in r is in ℓ. The probability density function f over I is:

$$f(I) = \frac{1}{Z}\exp[-\sum_{r\in R}\lambda_r(d_r(I))^p] \quad ; \quad Z = \int_I \exp[-\sum_{r\in R}\lambda_r(d_r(I))^p] \tag{4}$$

where λ_r is the weight of the rule r, Z is the continuous version of the normalization constant used in discrete Markov random fields, and $p \in \{1,2\}$ determines the loss function for minimizing the distance from satisfaction. If $p = 2$ the loss function is quadratic and the distance from satisfaction for each ground rule is squared. Constraints can be imposed on interpretations and the domain updated accordingly, for instance, requiring a predicate to be functional. Also, the density function can be conditioned on a partial interpretation and the domain and definitions of distances to satisfaction updated accordingly.

Finding the most probable interpretation in PSL is an instance of MPE inference. Maximizing the density function $f(I)$ is equivalent to minimizing the summation in the exponent. This optimization problem, if subject only to linear equality and inequality constraints on the interpretation, can be solved efficiently by casting it as a second-order cone program [1].

5 A PSL Model for Graph Summarization

Figure 3 lists the set of PSL rules used in this work for graph summarization in annotation data. Different subsets of these rules are experimentally evaluated and compared in Section 6. We model similarity of pairs of nodes of the same type

$$\begin{aligned} exemplar(A,B) &\rightarrow similar(A,B) && (5)\\ exemplar(A,B) &\rightarrow exemplar(B,B) && (6)\\ link(A,B) \wedge link(C,B) \wedge exemplar(A,D) &\rightarrow exemplar(C,D) && (7)\\ link(A,B) \wedge exemplar(A,C) \wedge exemplar(D,C) &\rightarrow link(D,B) && (8) \end{aligned}$$

Fig. 3. PSL rules for graph summarization as discussed in Section 5 and experimentally evaluated in Section 6. Labels refer to the introduction of rules in the text.

with predicate $similar/2$ and relations between pairs of nodes of different types with predicate $link/2$. Both predicates are symmetric. Note that while these predicates allow us to easily write general rules for all types of links and nodes appearing in the data, the inference engine takes into account the node types during grounding and thus ensures that clustering respects the types. Given truth values for all relevant atoms of these two predicates, the task of inference is to infer truth values of the remaining predicate $exemplar/2$, which encodes clusters. More specifically, the truth value of an atom $exemplar(a, b)$ indicates whether node a is a member of the cluster that has node b as its exemplar. We constrain $exemplar/2$ to be a functional predicate, that is, the truth values of all its groundings using a given node a as first argument have to sum to one. We also set a small prior on $exemplar/2$, further limiting its groundings.

In the following, we discuss the individual rules in more detail, showing how they encode the clustering heuristics introduced in Section 2 as probabilistic dependencies.

5.1 Similarity Heuristic

We start with the similarity heuristic, which indicates that pairs of similar nodes of the same type should probably be clustered. It is modeled by the first PSL rule:

$$exemplar(A,B) \rightarrow similar(A,B) \qquad (5)$$

This rule connects truth values of $similar/2$, which are given, to those of $exemplar/2$, which are inferred. For a pair of nodes (a, b) with low similarity, the rule is only satisfied for low truth values of $exemplar(a, b)$. In other words, it encourages node a to choose a different, more similar exemplar. If a and b are highly similar, on the other hand, a wider range of truth values for $exemplar(a, b)$ will satisfy the rule, making it possible for a to choose b or another node as its exemplar without penalty.

We further encourage clusters with a single exemplar, which is modeled by the second PSL rule:

$$exemplar(A,B) \rightarrow exemplar(B,B) \qquad (6)$$

This rule breaks chains of exemplar choices by penalizing situations where a node that is chosen as exemplar by some node in the cluster does not choose itself as exemplar. As truth values of $exemplar/2$ atoms are inferred during clustering, this rule can propagate information in both directions. If the truth value of $exemplar(b, b)$ is low for a given node b, it will encourage low truth values for all atoms $exemplar(a, b)$ with other nodes a as first argument. Conversely, each atom $exemplar(a, b)$ with high truth value encourages a high truth value for $exemplar(b, b)$.

5.2 Annotation Link Heuristic

The following two PSL rules model the annotation link heuristic:

$$link(A, B) \wedge link(C, B) \wedge exemplar(A, D) \rightarrow exemplar(C, D) \tag{7}$$

$$link(A, B) \wedge exemplar(A, C) \wedge exemplar(D, C) \rightarrow link(D, B) \tag{8}$$

Rule (7) states that a shared neighbor is an indication that two nodes should be clustered. Consider a pair of candidate nodes a and c for clustering, and keep the exemplar d and the node b on the other side fixed. Due to symmetry, we get two groundings of the rule, one replacing A with a and C with c, the other replacing A with c and C with a:

$$link(a, b) \wedge link(c, b) \wedge exemplar(a, d) \rightarrow exemplar(c, d) \tag{9}$$

$$link(c, b) \wedge link(a, b) \wedge exemplar(c, d) \rightarrow exemplar(a, d) \tag{10}$$

During clustering, the truth values of $link/2$ atoms are fixed to either 1 (link exists) or 0 (link does not exist). If one of the two links does not exist, both groundings are trivially satisfied, as their bodies will have the minimal truth value 0. If they both exist, the rules simplify to

$$exemplar(a, d) \rightarrow exemplar(c, d) \tag{11}$$

$$exemplar(c, d) \rightarrow exemplar(a, d) \tag{12}$$

and thus encourage the truth value of $exemplar(a, d)$ to be at most and at least that of $exemplar(c, d)$, respectively. In other words, the two nodes should agree in the degree to which they choose that specific exemplar and its corresponding cluster. Note that the influence of this rule grows with the number of joint neighbors the two candidates for clustering share, as those will produce individual groundings.

While Rule (8) again involves a pair of nodes that are candidates for clustering (a neighboring node and an exemplar), due to its different form, it encodes a different dependency. Consider the grounding

$$link(a, b) \wedge exemplar(a, c) \wedge exemplar(d, c) \rightarrow link(d, b) \tag{13}$$

As truth values of $link/2$ are fixed to either 0 or 1, this grounding is trivially satisfied if there is no link between a and b (in which case the truth value of the body is minimal) or if there is a link between d and b (in which case the truth value of the head is maximal). The interesting case is thus the one where there is a link between a and b, but no link between d and b.[4] In this case, the rule increases the probability that

$$exemplar(a, c) \,\tilde{\wedge}\, exemplar(d, c) \leq 0 \tag{14}$$

In words, the rule will be satisfied in this case if and only if the truth values of the two *exemplar*/2 atoms sum to at most 1, thus encouraging the two nodes not to strongly agree on a joint exemplar. The influence of this rule grows with the number of neighbors on which a and d disagree. Together, the two rules thus allow one to take into account both shared and unshared neighbors during clustering.

6 Evaluation

The goals of graph summarization include identifying patterns and making predictions in the annotation graph. We use prediction, specifically the task of predicting missing links, to explore the utility of the simple heuristics from Section 5. In our experimental setting, the missing links are links between genes and GO terms in the annotation graph for the model organism Arabidopsis thaliana, as described in Section 2. We begin by generating graph summaries using the rules described ealier. Next, we use this model to predict gene-GO annotations.

In addition to the GO, gene and PO annotation graph, we consider gene-gene sequence-based similarity,[5] as well as PO-PO path based distances from the PO ontology and GO-GO path based distances from the GO ontology. These are represented as *similar*/2 atoms between nodes of each of the three types within the graph: PO terms, genes and GO terms. In our model, all instances of *similar*/2 atoms are treated uniformly; however, they are computed using different similarity metrics. All other relations from the data are *link*/2 atoms between nodes of different types. Although the type of each node in the graph could be represented explicitly and used in the rules to control the graph summaries, e.g., to ensure that no cluster contains both PO and GO terms, in our implementation we only consider relations between nodes where a relation of that type might exist between nodes of those types. Further, in the grounding of the atoms in the data, we make each *similar*/2 atom symmetric by asserting its inverse, and we do the same for *link*/2 atoms.

Using each *graph summarization* program, we infer the *exemplar*/2 atoms forming clusters with soft membership, which is the input to our *link prediction* program. To then evaluate our link prediction using graph summaries, we perform leave-one-out evaluation. Specifically, for each link in the original annotation graph we first remove the link and compute the graph summary. We then

[4] Note that we get a symmetric grounding that affects the opposite case as well.

[5] We compute pair-wise sequence similarity between pairs of genes using the Nucleotide-Nucleotide BLAST 2.2.26+ package.

predict missing links. We sort the predicted links based on their truth values, and we interpret the truth values as the *confidence* in the prediction. We calculate the link prediction precision with recall of one, and we report on the mean average precision computed over all leave-one-out links.

Many combinations of heuristics could be explored for the summarization and prediction programs, so we chose the following four configurations to evaluate. The first configuration (**LINK1**) uses Rule (7) for graph summarization and link prediction. The second configuration (**LINK2**) is the same as **LINK1** but it uses Rule (8) in place of Rule (7). The third configuration (**SIM1**) uses only the similarity rules for graph summarization and adds Rule (7) for link prediction. The fourth configuration (**SIM2**) is the same as **SIM1** but uses Rule (8) in place of Rule (7). Each configuration also uses Rule (5) and Rule (6). All rules have weight one except Rule (6) which has a high weight, 1000, to encourage distinct clusters by breaking chains of $exemplar/2$ atoms. Learning individual weights for these rules may be beneficial, but we do not consider it in this work.

Finally, we choose the loss function for minimizing the distance from satisfaction, which is set by p in (4). In each configuration, we use the linear loss function for the graph summarization program and the quadratic loss function for the link prediction program. We use the quadratic loss function for the link prediction programs for two reasons. First, inference with quadratic loss is more expensive than with linear loss, so for the interest of time we only use it on link prediction, which is less expensive than graph summarization. Our link prediction programs are less expensive, in part, because inferring $link/2$ atoms involves a smaller number of rules than inferring $exemplar/2$ atoms. Second, quadratic loss tends to assign $link/2$ atom truth values between rather than at the extremes, 0 or 1, more often than linear loss, and this is helpful when ranking predicted links to calculate precision.

6.1 Results

Table 1 describes the TAIR annotation graph data sets we used for evaluation. The first two data sets (**DS1** and **DS2**) have fewer genes but more terms and annotations over all than the last data set (**DS3**). Table 2 reports mean average precision (MAP) on the evaluation data sets for each PSL model configuration.

Table 1. The evaluation data sets. The number of genes, Plant Ontology terms, Gene Ontology terms, PO-to-gene annotation links and GO-to-gene annotation links.

	DS1	**DS2**	**DS3**
Genes	10	10	18
PO Terms	53	48	40
GO Terms	44	31	19
PO-Gene	255	255	218
GO-Gene	157	157	92

Considering the **LINK1** and **LINK2** configurations across all data sets, we see that Rule (8), used in the latter, consistently has higher precision than Rule (7), used in the former. Rule (8) is also used in **SIM2** where it similarly has higher precision than the other link rule in **SIM1** across all data sets.

Now considering precision across data sets, on **DS3** both **LINK2** and **SIM2** perform well compared to previous link prediction results on similar annotation data [5]. However, none of the configurations perform well on **DS1** and **DS2**. To interpret this result we convert the clusters to hard membership, calculate the average number of clusters of size greater than one produced by each configuration and normalize by the number of nodes in the data set. This is shown in the right side of Table 2.

Using this information, we see a small number of clusters formed in **DS1** and **DS2** and a larger number formed in **DS3**[6] except where Rule (7) is used. This suggests that Rule (8) is helpful for link prediction on this data and may be helpful for clustering; on the other hand, Rule (7) is not helpful for clustering or link prediction on this data, and may interfere with the use of similarity attributes in clustering. Finally, this also suggests that neither annotation link heuristic rule works well for prediction on graphs where we find few clusters of size greater than one. Since, for example in Rule (8) the truth value of inferred *link*/2 atoms is bounded only when there are multiple nodes A and D in the same cluster, this result is not surprising.

Table 2. Mean average precision of link prediction for evaluated PSL model configurations on each data set. Also, the average number of clusters of size greater than one, divided by the number of nodes in the data set.

Configuration	MAP			Clusters		
	DS1	**DS2**	**DS3**	**DS1**	**DS2**	**DS3**
LINK1	0.01	0.00	0.01	0.01	0.01	0.01
LINK2	0.06	0.04	0.30	0.03	0.03	0.10
SIM1	0.01	0.00	0.01	0.01	0.01	0.12
SIM2	0.07	0.12	0.28	0.01	0.01	0.12

7 Conclusions and Future Work

In this work, we demonstrated an exploratory use of graph summarization heuristics in probabilistic soft logic (PSL) on annotation graph data, combining relational and similarity evidence from multiple, heterogeneous sources. The power of the approach is the ease in which a variety of clustering criteria can be declaratively expressed. Our work, which is ongoing, will continue to explore the space of graph summarization rules for combining data from rich sources, such as the

[6] A similar pattern of cluster sizes emerges when a version of these graphs is clustered using a separate method similar to [5].

gene sequences, annotations and term ontologies used in this work and other sources now made available through the Linked Data initiative and Semantic Web.

Acknowledgements

Angelika Kimmig is a postdoctoral fellow of the Research Foundation Flanders (FWO Vlaanderen). This work is partially supported by NSF under grants IIS0746930 and CCF0937094.

References

1. Broecheler, M., Mihalkova, L., Getoor, L.: Probabilistic similarity logic. In: Conference on Uncertainty in Artificial Intelligence (UAI). (2010)
2. Taskar, B., Segal, E., Koller, D.: Probabilistic classification and clustering in relational data. In: International Joint Conference on Artificial Intelligence (IJCAI). (2001)
3. Kemp, C., Tenenbaum, J.B., Griffiths, T.L., Yamada, T., Ueda, N.: Learning systems of concepts with an infinite relational model. In: National Conference on Artificial Intelligence (AAAI). (2006)
4. Long, B., Zhang, Z.M., Yu, P.S.: A probabilistic framework for relational clustering. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD). (2007)
5. Thor, A., Anderson, P., Raschid, L., Navlakha, S., Saha, B., Khuller, S., Zhang, X.N.: Link prediction for annotation graphs using graph summarization. In: International Semantic Web Conference (ISWC). (2011)
6. Zhang, N., Tian, Y., Patel, J.M.: Discovery-driven graph summarization. In: International Conference on Data Engineering (ICDE). (2010)
7. Tian, Y., Hankins, R.A., Patel, J.M.: Efficient aggregation for graph summarization. In: ACM SIGMOD International Conference on Management of Data (SIGMOD). (2008)
8. Navlakha, S., Rastogi, R., Shrivastava, N.: Graph summarization with bounded error. In: ACM SIGMOD International Conference on Management of Data (SIGMOD). (2008)
9. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. Science **315** (2007) 972–976
10. Berthold, M., ed.: Bisociative Knowledge Discovery. Volume 7250 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2012)
11. Lee, R.C.T.: Fuzzy logic and the resolution principle. J. ACM **19**(1) (1972) 109–119
12. Richardson, M., Domingos, P.: Markov logic networks. Machine Learning **62**(1-2) (2006) 107–136

Position Papers

Introducing Ontological CP-Nets

Tommaso Di Noia[1] and Thomas Lukasiewicz[2]

[1] Dipartimento di Elettrotecnica ed Elettronica, Politecnico di Bari, Italy
t.dinoia@poliba.it

[2] Department of Computer Science, University of Oxford, UK
thomas.lukasiewicz@cs.ox.ac.uk

Abstract. Preference representation and reasoning is a key issue in many real-world scenarios. Currently, there are many approaches allowing preferences to be assessed in a qualitative or quantitative way. The most prominent qualitative approach for representing preferences are CP-nets. Their clear graphical structure unifies an easy representation of user desires with nice computational properties when computing the best outcome. Here, we introduce ontological CP-nets, which allow the representation of preferences using a CP-net over an ontological domain, i.e., variable values are logical formulas constrained relative to a background domain ontology.

1 Motivation

During the last five years, we have seen two main phenomena emerging from the classical Web. On the one side, we have had the social revolution, where the users act as first-class citizens in the creation and delivery of contents over the Web. On the other side, we have seen an increasing interest in the so-called Web of Data as a special case of the Semantic Web vision. These two technological waves (Social Web and Semantic Web) led to what is known as Web 3.0, i.e., a Web where, on top of the classical Web of interlinked documents, we have the two layers represented by: (a) user contents, connections, interactions, reviews, tags, etc.; and (b) semantic data and tags constrained by ontologies. All this information may be exploited to create semantic user profiles containing a representation of users' preferences. Given a query, these latter play a fundamental role when a crisp yes/no answer is not enough to satisfy a user's needs, since there is a certain degree of uncertainty in possible answers [4]. We have two main ways of modeling preferences: (a) *quantitative preferences* are associated with a number representing their worth or they are represented as an ordered set of objects (e.g., "my preference for WiFi connection is 0.8" and "my preference for cable connection is 0.4"), while (b) *qualitative preferences* are related to each other via pairwise comparisons (e.g., "I prefer WiFi over cable connection"). The two approaches can also be combined (see, e.g., [6]).

As also stated in [4], the qualitative approach seems to be a more natural way of representing preferences, since humans are not very comfortable in expressing their "wishes" in terms of a numerical value. To have a quantitative representation of her preferences, the user needs to explicitly determine a value for a large number of alternatives usually described by more than one attribute. It is generally much easier to provide information about preferences as pairwise qualitative comparisons [4]. Among the diverse qualitative frameworks for preference representation and reasoning, one of the most powerful are CP-nets. They are a graphical language that unifies an easy representation of user desires with nice computational properties when computing the best outcome. In this paper, we propose an enhancement of CP-nets by adding ontological information associated to preferences. The rest of this paper is structured as follows. In Section 2, we briefly introduce CP-nets and constrained CP-nets. Section 3 then shows some of their limits, and introduces the notion of ontological CP-nets.

2 CP-Nets and Constrained CP-Nets

In this section, we introduce some necessary preliminary notions and formalisms. Given a set of variables $\mathcal{V}$, an *outcome* is an assignment to all the variables in $\mathcal{V}$. A preference relation $\succeq$

is a total pre-order over the set of outcomes. We write $o_1 \succ o_1$ iff o_1 is strictly preferred to o_2, and $o_1 \succeq o_2$ iff o_1 is strictly or equally preferred to o_2; we then also say that o_2 is *dominated* by o_1. If there is no outcome o such that $o \succ o_1$, we say that o_1 is *undominated.*

Conditional preference networks (CP-nets) [2] are a formalism to represent and reason with qualitative preferences. This compact but powerful language allows the specification of preferences based on the notion of conditional preferential independence. Fundamental for CP-nets is the notion of *conditionally preferentially independent* (CPI). Let $P, Q \in \mathcal{V}$ be two variables and $\mathcal{R} \subset \mathcal{V}$ be a set of variables such that P, Q, and $\mathcal{R}$ partition $\mathcal{V}$, and $Dom(P)$, $Dom(Q)$, and $Dom(\mathcal{R})$ represent all possible assignments for P, Q, and all the variables in $\mathcal{R}$, respectively. We say that P is *conditionally preferentially independent* (CPI) of Q given an assignment $r \in \mathcal{R}$ iff, for all $p_1, p_2 \in P$ and $q_1, q_2 \in Q$, we have that $p_1q_1r \succeq p_2q_1r$ iff $p_1q_2r \succeq p_2q_2r$. Here, $\succeq$ represents the preference order among assignments for sets of variables. CP-nets are a graphical language to model CPI statements. Formally, a CP-net N consists of a directed graph G representing preference relations among variables P_i and a set of conditional preference tables $CPT(P_i)$ (one for each variable). Given the set of variables $\mathcal{V} = \{P_i \,|\, i \in \{1, \ldots, n\}\} \cup \{P_{n+1}\}$ representing the nodes of G, such that P_i is a parent of P_{n+1} in G, the corresponding $CPT(P_{n+1})$ contains a preference for each pair of values of P_{n+1} conditioned to all possible assignments of variables P_i. The representation of a CP-net assumes that the user explicitly specifies her preferences over the values of P_{n+1} for each complete assignment of P_i, with $i \in \{1, \ldots, n\}$. Given a CP-net N, we denote by $\mathcal{CPT}_i$ the set of all conditional preferences represented in $CPT(P_i)$, and $\mathcal{CPT}_N = \{\mathcal{CPT}_i \,|\, i \in \{1, \ldots, n\}\}$.

Given a CP-net, the two main queries one may ask are:

- *Dominance query*: Given two outcomes o_1 and o_2, decide whether $o_1 \succeq o_2$.
- *Outcome optimization*: What is the optimal outcome given the preferences represented in the CP-net? That is, we look for one of the undominated outcomes.

Given an acyclic CP-net, one can compute the best outcome in linear time. The algorithm just follows the order among variables represented by the graph and assigns values to the variables P_i from top to bottom satisfying the preference order in the corresponding $CPT(P_i)$. Finding the optimal outcome in cyclic CP-nets is NP-hard.

In *constrained CP-nets* [5, 3], constraints among variables are added to the basic formalism of CP-nets. Adding constraints among variables may reduce the set of possible outcomes. The approach to finding the optimal outcome proposed in [5] relies on a reduction of the preferences represented in the CP-net to a set of hard constraints (which can be represented in clause form for binary variables) taking into account the variables occurring in the preferences. Given a CP-net N and a set of constraints $\mathcal{C}$, an outcome o is *feasible* if it satisfies all the constraints in $\mathcal{C}$. A feasible outcome is *Pareto optimal* [3] *iff* it is undominated. In [5], the authors present an algorithm to find the optimal outcome solving a constraint satisfaction problem. For binary variables, given a conditional preference $(p_{n+1} \succeq \neg p_{n+1} \mid \bigwedge_{i=1\ldots n} \tilde{p_i})$, where $\tilde{p_i} \in \{p_i, \neg p_i\}$, the corresponding constraint is the clause $\bigwedge_{i=1\ldots n} \tilde{p_i} \to p_{n+1}$ (analogously, for $(\neg p_{n+1} \succeq p_{n+1} \mid \bigwedge_{i=1\ldots n} \tilde{p_i})$, we have $\bigwedge_{i=1\ldots n} \tilde{p_i} \to \neg p_{n+1}$).

Given a CP-Net N and a set of constraints $\mathcal{C}$, a feasible Pareto optimal outcome is an assignment satisfying the corresponding set of clauses and all the constraints in $\mathcal{C}$ (and vice versa).

3 Ontological CP-Nets

We now introduce a framework for preference representation harnessing the technologies described in the previous section. The idea is to combine CP-nets and ontologies represented in description logics (DLs) [1]. In this combination, variable values are satisfiable DL concepts. For ease of presentation, we describe our approach using variables whose domain contains only two values, i.e., two concepts; the extension to more than two concepts is straightforward.

Two conditional preferences $(\alpha \succeq \beta \mid \gamma)$ and $(\alpha' \succeq \beta' \mid \gamma')$ are *equivalent* relative to an ontology $\mathcal{T}$ iff $\mathcal{T} \models \gamma \equiv \gamma'$, $\mathcal{T} \models \alpha \equiv \alpha'$, and $\mathcal{T} \models \beta \equiv \beta'$.

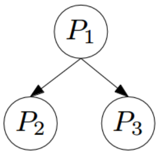
Definition 1. An *ontological CP-net* $(\mathcal{T}, N)$ consists of an ontology $\mathcal{T}$ and a CP-net N such that: (1) for each variable P in N, $Dom(P) = \{\alpha, \beta\}$, where α and β are DL concepts that are satisfiable relative to $\mathcal{T}$, and such that $\mathcal{T} \not\models \alpha \equiv \top$, $\mathcal{T} \not\models \beta \equiv \top$, and $\mathcal{T} \not\models \alpha \equiv \beta$; (2) any two conditional preferences in $\mathcal{CPT}_N$ are pairwise not equivalent.

Note that each variable $P \in \mathcal{V}$ with $Dom(P) = \{\alpha, \beta\}$ may have one of the four values α, $\neg\alpha$, β, and $\neg\beta$. That is, the variables are not strictly binary.

Example 1 (Hotel). Consider a simple ontology, describing the services offered by a hotel:

$$\begin{aligned} \mathtt{Scooter} &\sqsubseteq \mathtt{Motorcycle} \\ \mathtt{Motorcycle} &\sqsubseteq \neg\mathtt{Bike} \\ \exists\mathtt{rent}.\mathtt{Scooter} &\sqsubseteq \exists\mathtt{facilities}.(\mathtt{Parking} \sqcap \exists\mathtt{payment} \sqcap \forall\mathtt{payment}.\mathtt{Free})\,. \end{aligned}$$

A simple ontological CP-net is depicted in the following together with possible CPTs related to the nodes P_1 and P_2. The domains of P_1, P_2 and P_3 are:
$Dom(P_1) = \{\alpha_1 = \exists\mathtt{location}.\mathtt{OnTheSea},\ \beta_1 = \exists\mathtt{location}.\mathtt{NearTheAirport}\}$
$Dom(P_2) = \{\alpha_2 = \exists\mathtt{rent}.\mathtt{Bike},\ \beta_2 = \exists\mathtt{facilities}.(\mathtt{Parking} \sqcap \exists\mathtt{payment} \sqcap \forall\mathtt{payment}.\mathtt{Free})\}$
$Dom(P_3) = \{\alpha_3 = \exists\mathtt{rent} \sqcap \forall\mathtt{rent}.\mathtt{Scooter},\ \beta_3 = \exists\mathtt{facilities}.\mathtt{Shuttle}\}$



$$CPT(P_1) = \begin{cases} (\alpha_1 \succeq \neg\alpha_1) \\ (\alpha_1 \succeq \beta_1) \\ (\alpha_1 \succeq \neg\beta_1) \\ (\beta_1 \succeq \neg\beta_1) \\ (\beta_1 \succeq \neg\alpha_1) \\ (\neg\alpha_1 \succeq \neg\beta_1) \end{cases} \quad CPT(P_2) = \begin{cases} (\beta_2 \succeq \alpha_2 \mid \alpha_1) \\ (\alpha_2 \succeq \beta_2 \mid \neg\alpha_1) \\ (\beta_2 \succeq \alpha_2 \mid \beta_1) \\ (\alpha_2 \succeq \beta_2 \mid \neg\beta_1)\,. \blacksquare \end{cases}$$

Although we know how to reason with expressive DLs and with CP-nets, their combination leads to diverse issues both from the modeling and the reasoning perspective. We now sketch the main ideas behind our approach, using illustrative examples whenever possible.

Implicitly constrained variables. Even if we do not have any explicit hard constraint expressed among the variables of the CP-net, due to the background ontology, we have a set of implicit constraints among α and β values of the variables $\mathcal{V}$ in the CP-net.

Example 2 (Hotel cont'd). Consider the ontology $\mathcal{T}$ of Example 1 and the two variables P_2 and P_3. Because of $\mathcal{T}$, we have the implicit constraint $\mathcal{T} \models \alpha_3 \sqsubseteq \beta_2$. ■

One way to infer all possible constraints among variable values is to adopt the ontology compilation technique presented in [7]. There, the authors propose an algorithm to elicit all possible hidden constraints (represented in clausal form) occurring among a set of DL concepts.

Preference satisfiability. Following [5], for each preference $\Phi = (\tilde{\alpha} \succeq \tilde{\beta} \mid \gamma) \in \mathcal{CPT}_N$, we may write the clause: $\gamma \rightarrow \tilde{\alpha}$ (i.e., $\neg\gamma \sqcup \tilde{\alpha}$). In ontological CP-nets, this is not sufficient. Indeed, since α and β belong to the domain of the same variable, we have to explicitly state that they are disjoint with each other relative to $\mathcal{T}$. Hence, for each preference we have to add one more clause of the form $\tilde{\alpha} \rightarrow \neg\tilde{\beta}$. This may also lead to unsatisfiable clauses.

Example 3 (Hotel cont'd). Consider the preference (not allowed by the CP-net represented in Example 1) $\Phi = (\alpha_3 \succeq \beta_2 \mid \neg\alpha_1)$. If we imposed $\alpha_3 \rightarrow \neg\beta_2$ then we had $\mathcal{T} \models (\neg\alpha_1 \rightarrow \alpha_3) \sqcap (\alpha_3 \rightarrow \neg\beta_2) \sqsubseteq \bot$. In fact, we know that $\mathcal{T} \models \alpha_3 \sqsubseteq \beta_2$. That is, we are saying that Φ is never satisfied. ■

Hence, the preference Φ is *satisfiable* iff $\mathcal{T} \models (\gamma \rightarrow \tilde{\alpha}) \sqcap (\tilde{\alpha} \rightarrow \neg\tilde{\beta}) \not\sqsubseteq \bot$. The notion of satisfiability can be extended also to the whole CP-net.

Definition of outcome. In a constrained CP-net, if we had propositional `true`/`false` variables, an outcome would be a model, i.e., a `true`/`false` assignment that satisfies all the constraints and some of the clauses built starting from the preferences represented in $\mathcal{CPT}_N$. In ontological CP-nets, we deal with DL concepts, so a model satisfying the constraints cannot be explicitly represented. Actually, we have more than one equivalent outcome, i.e., all the models that satisfy the same preferences. A solution to this issue is to compute a formula whose models satisfy the same preferences. Also, in this case, such formula can be computed by adapting the techniques proposed in [7], where the preference satisfaction problem for DL concepts is solved via Integer Linear Programming encoding.

Dominance test and eligibility of CP-statements. A set of conditional preferences (CP-statements) is eligible iff it has an undominated outcome. In case of a set of eligible CP-statements, once we introduce an ontology to describe the background knowledge we can make the undominated outcome unsatisfiable. Then we have to be very careful when evaluating the dominance test among a set of possible outcomes of the CP-net. In fact, in case we have cycles in the dependency graph [2] associated to the CP-net, due to the presence of an ontology, we could be unable to find an undominated outcome.

Complexity of reasoning. As also argued in [5], having background knowledge may introduce implicit cycles in the graph representing the CP-net. This affects the computational complexity related to the computation of an outcome.

4 Conclusion

The availability of semantic information over the Web and the social revolution, pave the way to a new wave of personalized applications where ontological knowledge plays a fundamental role. User's preferences may act as a filter to the information accessed by the user in order to provide a personalized experience while interacting with the Semantic Web. Among the various formalisms proposed in the literature to represent preferences, a very promising one is that of CP-nets. They have a strong theoretical background and many results are already available in the literature both related to their computational properties and to modeling aspects. Nevertheless, to the best of our knowledge, almost nothing has been done to combine CP-nets with ontological modeling and reasoning. In this paper we introduce the notion of Ontological CP-nets and highlight some issues related to the ontological nature of the information we deal with when combined with conditional preferences arranged in a CP-net.

Acknowledgments. This work was supported by the EPSRC grant EP/J008346/1 ("PrOQAW"), the ERC under the EU's 7th Framework Programme (FP7/2007-2013/ERC) grant 246858 ("DIADEM"), a Google Research Award, and a Yahoo! Research Fellowship. Tommaso Di Noia also acknowledges support by HP IRP 2011, grant CW267313.

References

1. F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
2. C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, and D. Poole. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *JAIR*, 21:135–191, 2004.
3. C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, and D. Poole. Preference-based constrained optimization with CP-nets. *Comput. Intell.*, 20(2):137–157, 2004.
4. C. Domshlak, E. Hüllermeier, S. Kaci, and H. Prade. Preferences in AI: An overview. *Artif. Intell.*, 175(7/8):1037–1052, 2011.
5. C. Domshlak, S. Prestwich, F. Rossi, K. B. Venable, and T. Walsh. Hard and soft constraints for reasoning about qualitative conditional preferences. *J. Heuristics*, 12(4/5):263–285, 2006.
6. T. Lukasiewicz and J. Schellhase. Variable-strength conditional preferences for ranking objects in ontologies. *J. Web Sem.*, 5(3):180–194, 2007.
7. A. Ragone, T. D. Noia, F. M. Donini, E. D. Sciascio, and M. P. Wellman. Weighted description logics preference formulas for multiattribute negotiation. In *Proc. SUM-2009*, pp. 193–205, 2009.

Computing Inferences for Credal $\mathcal{ALC}$ Terminologies

Rodrigo B. Polastro[a], Fabio G. Cozman[a],
Felipe I. Takiyama[a], and Kate C. Revoredo[b]

[a]Univ. de São Paulo - Av. Prof. Mello Moraes 2231, São Paulo, SP - Brazil
[b]Dept. Informática Aplicada, Unirio - Av. Pasteur, 458, Rio de Janeiro, RJ - Brazil
rodrigopolastro@gmail.com, fgcozman@usp.br, felipe.takiyama@usp.br,
katerevoredo@gmail.com

Abstract. We describe a package that performs inferences for the probabilistic description logic CR$\mathcal{ALC}$: given a terminology consisting of a set of sentences in CR$\mathcal{ALC}$, and a set of assertions, the package computes the probability of additional assertions using an approximate variational method. We briefly review the essentials of CR$\mathcal{ALC}$, mention some recent applications, and describe the package. We then describe our current efforts to incorporate lifted inference into the package.

1 Introduction

This paper focuses on a particular probabilistic description logic, Credal $\mathcal{ALC}$ (referred to as CR$\mathcal{ALC}$). This logic adds some probabilistic operators to the popular logic $\mathcal{ALC}$ [1] and combines these operators with independence assumptions inspired by the theory of relational Bayesian networks [4]. One can see CR$\mathcal{ALC}$ as a language to express ontologies with probabilistic assessments, or simply as a language to describe relational Bayesian networks. Applications in mobile robotics [2], automatic construction of ontologies [7], and analysis of social networks [8] have benefited from the use of CR$\mathcal{ALC}$, often coupled with machine learning techniques. We summarize the main features of CR$\mathcal{ALC}$, and some of its applications, in Section 2. Alas, so far there has been no simple way to produce inferences in CR$\mathcal{ALC}$ — here an *inference* means the computation of a probability value for a given assertion conditional on other observed assertions, using a probabilistic terminology as background knowledge.

In Section 3 we introduce a package, coded by the first author, that accepts sentences and assertions in CR$\mathcal{ALC}$, and that produces inferences using an approximate variational algorithm. We then discuss our current efforts in developing exact *lifted* inference methods that can be added to the package.

2 cr$\mathcal{ALC}$: A summary, and applications

As usual with description logics, we have *individuals*, *concepts*, and *roles*. The semantics is given by a *domain* $\mathcal{D}$ (a set that we assume *finite* in this paper) and an *interpretation* $\cdot^{\mathcal{I}}$ (a functor). Each *concept* is interpreted as a subset of a domain $\mathcal{D}$. Each *role* is interpreted as a binary relation on the domain.

Many probabilistic descriptions logics have appeared in the literature [6]. Several consider probabilities over the interpretations. 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. The logic CR$\mathcal{ALC}$ is a probabilistic extension of the description logic $\mathcal{ALC}$ that adopts such an interpretation-based semantics [3]. 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 [3]) given by:

$$\forall x \in \mathcal{D} \; : \; P(C(x)|D(x)) = \alpha, \qquad \forall x \in \mathcal{D}, y \in \mathcal{D} \; : \; P(r(x,y)) = \beta.$$

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, 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. We then assume a Markov condition on this graph, similar to the Markov condition on Bayesian networks; with a few additional assumptions concerning uniqueness of names and values, this guarantees that any probability distribution over interpretations factorizes as a Bayesian network over grounded concepts and roles [3].

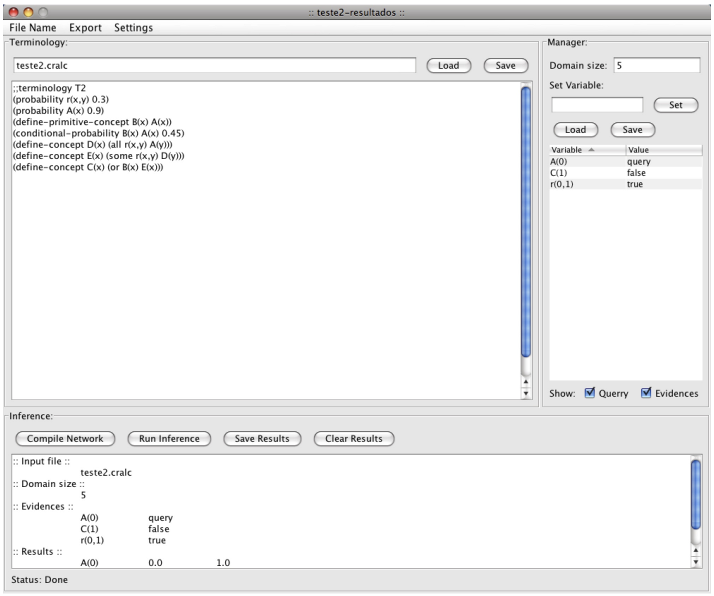
Inferences, such as $P(\mathsf{A}_o(\mathsf{a}_0)|\mathcal{A})$ for an ABox $\mathcal{A}$, can be computed by grounding a set of sentences into a possibly large Bayesian network. As this may be too complex in practice, an alternative is to run approximate schemes, for instance schemes based on approximate variational approximations [3].

Recent work has explored the use of probabilistic terminologies in CR$\mathcal{ALC}$ in several applications [2, 7, 8]. These applications require the computation of many inferences; thus it is important to have a package that can perform inference in CR$\mathcal{ALC}$ terminologies.

3 A package

This section describes a software package that handles CR$\mathcal{ALC}$ terminologies and assertions, and that produces inferences (either by producing relational Bayesian networks that can be further processed, or by running approximate variational inference). The package has been coded by the first author using the Java language, and can work either from the command prompt or through a graphical user interface (depicted in Figure 1).

The first design decision was the input language. We have chosen to adapt the Knowledge Representation System Specification (KRSS). The standard complete specification of KRSS can be found at http://dl.kr.org/krss-spec.ps. We use



teste2.cralc

```
;;terminology T2
(probability r(x,y) 0.3)
(probability A(x) 0.9)
(define-primitive-concept B(x) A(x))
(conditional-probability B(x) A(x) 0.45)
(define-concept D(x) (all r(x,y) A(y)))
(define-concept E(x) (some r(x,y) D(y)))
(define-concept C(x) (or B(x) E(x)))
```

Domain size: 5

Set Variable:

Variable	Value
A(0)	query
C(1)	false
r(0,1)	true

Show: ☑ Querry ☑ Evidences

```
:: Input file ::
            teste2.cralc
:: Domain size ::
            5
:: Evidences ::
            A(0)        query
            C(1)        false
            r(0,1)      true
:: Results ::
            A(0)        0.0         1.0
```

Status: Done

Fig. 1. Terminologies are written in the larger panel, while assertions are set in the right panel; the lower panel reports on inferences.

the following constructs: `(and C1...Cn)` for conjunction; `(or C1...Cn)` for disjunction; `(not C)` for complement; `(all r C)` to indicate the quantifier $\forall r.C$; `(some r C)` to indicate the quantifier $\exists r.C$; `(define-concept C D)` for $C \equiv D$; and `(define-primitive-concept C D)` for $C \sqsubseteq D$.

Probabilistic inclusions are specified as follows: (`probability B` α) denotes $P(B) = \alpha$; (`conditional-probability B A` α) for $P(A|B) = \alpha$. An example of valid input file is:

```
(probability A(x) 0.7)      (probability B(x) 0.4)
(define-concept C(x) (and A(x) (not B(x))))
```

Assertions can be represented through written files as well; inference results can be exported to files. Alternatively, the graphical user interface depicted in Figure 1 can be used to load/save files, to specify the size of the domain and the assertions, to ask for inferences, and to check results. The package is freely available at http://sites.poli.usp.br/pmr/ltd/Software/CRALC/index.html.

Approximate inferences are produced by generating a set of grounded Bayesian networks, one for each individual mentioned in the query and in the evidence, plus an additional Bayesian network for a "generic" individual [3]. Exact Bayesian network inference is performed in each one of these networks (the package assumes that such exact inference is feasible) and messages are exchanged between the networks using a loopy-propagation scheme. A relatively small number of message-passing iterations seems to generate good approximations; the cost of running an approximate inference is then the number of allowed iterations times the sum of inference costs for each one of the grounded networks plus the “generic” individual network.

4 Conclusion

Efficient inference for probabilistic description logics is a key enabler of technologies that must deal with uncertainty and semantic information. Currently there are many proposals for probabilistic description logics but relatively few implemented inference engines. In this short paper we have described our modest efforts in providing easier ways to represent and process sentences in probabilistic description logics. The software package we have presented still requires much development, but it is a step in a direction we feel has not received enough attention.

Our current effort is to implement exact *lifted* inference; that is, inference that does not require grounding concepts and roles for the entire domain. We are using recently developed methods for lifted inference in graphical models [5], and plan to report on the results soon.

Acknowledgements

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

References

1. F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, and P.F. Patel-Schneider. *Description Logic Handbook*. Cambridge University Press, 2003.
2. F. Correa, F. G. Cozman, and J. Okamoto Jr. Collective classification in semantic mapping with a probabilistic description logic. In *Int. Workshop on Description Logics*, pages 455–465, Barcelona, Spain, 2011.
3. F. G. Cozman and R. B. Polastro. Complexity analysis and variational inference for interpretation-based probabilistic description logics. In *Uncertainty in Artificial Intelligence*, pages 117–125, Corvallis, Oregon, 2009. AUAI Press.
4. M. Jaeger. Relational Bayesian networks. In *Uncertainty in Artificial Intelligence*, pages 266–273, San Francisco, California, 1997. Morgan Kaufmann.
5. J. Kisynski and D. Poole. Lifted aggregation in directed first-order probabilistic models. Int. Joint Conf. on Artifical Intelligence, pages 19221929, 2009.
6. T. Lukasiewicz and U. Straccia. Managing uncertainty and vagueness in description logics for the semantic web. *Journal of Web Semantics*, 6:291–308, 2008.
7. J. E. Ochoa Luna, K. Revoredo, and F. G. Cozman. Learning probabilistic description logics: A framework and algorithms. In *Mexican Int. Conf. on Artificial Intelligence, Lecture Notes in Artificial Intelligence*, volume 7094 Part I, pages 28–39. Springer, 2011.
8. K. Revoredo, J. E. Ochoa-Luna, and F. G. Cozman. Semantic link prediction through probabilistic description logic. In *Workshop on Uncertainty Reasoning for the Semantic Web*, pages 87–97, Bonn, Germany, 2011.