Instance-based Probabilistic Reasoning in the Semantic Web

Pedro Oliveira and Paulo Gomes
CISUC-DEI-FCTUC
University of Coimbra
Pólo II, 3030 Coimbra, Portugal
pcoliv@student.dei.uc.pt, pgomes@dei.uc.pt

ABSTRACT
Most of the approaches for dealing with uncertainty in the Semantic Web rely on the principle that this uncertainty is already asserted. In this paper, we propose a new approach to learn and reason about uncertainty in the Semantic Web. Using instance data, we learn the uncertainty of an OWL ontology, and use that information to perform probabilistic reasoning on it. For this purpose, we use Markov logic, a new representation formalism that combines logic with probabilistic graphical models.

Categories and Subject Descriptors
I.2.3 [**Artificial Intelligence**]: Deduction and Theorem Proving – *uncertainty, "fuzzy", and probabilistic reasoning*.

General Terms
Algorithms, Experimentation.

Keywords
Semantic Web, Probabilistic Reasoning, Markov Logic.

1. INTRODUCTION
The Semantic Web [1] envisions a world where agents share and transfer structured knowledge in an open and semi-automatic way. This knowledge, in most of the cases, is characterized by uncertainty. However, Semantic Web languages like RDF and OWL don't provide any means of dealing with this uncertainty. They are mainly based on crisp logic, being incapacitated of dealing with partial and incomplete knowledge. Reasoning in the Semantic Web resigns to a deterministic process of verifying if statements are true or false.

In the last years, some efforts have been made in representing and reasoning with uncertainty in the Semantic Web (see [2] for a complete overview about the subject). These works mainly focused on extending the logics behind Semantic Web languages to the probabilistic/possibilistic/fuzzy logics, or on combining these languages with probabilistic formalisms like Bayesian Networks. In all of these approaches, this is achieved by annotating the ontologies with some kind of uncertainty information about its axioms and use this information to perform uncertainty reasoning. However, a question arises: how are these uncertainties asserted?

The most obvious answer is that users are responsible for this task. However, this assumption is fairly fallible. It is studied that humans are not good at either producing or perceiving concepts like probability [3]. And even if humans were capable of doing that, creating and maintaining large annotated ontologies can be a cumbersome and difficult task, invalidating all the gains that could arise from the annotation.

WWW 2009, April 20–24, 2009, Madrid, Spain.
ACM 978-1-60558-487-4/09/04.

In fact, uncertainty is a common characteristic of the current Web. When we create a webpage, for example, search engines are responsible to assert what is the probabilistic relevance of it, compared to other pages, to certain topics. We don't have to explicitly refer that information: we just create its content, and search engines do the rest. So, we must develop similar automatic mechanisms to perform reasoning in the Semantic Web.

In this work, we study how we can make probabilistic reasoning on OWL ontologies without any kind of uncertainty annotation. To assert the uncertainty of its axioms, we use solely the information of its instances. For this purpose, we use Markov logic [4], a novel approach that combines logic and probability in the same representation.

2. MARKOV LOGIC
Markov logic combines first-order logic and probabilistic graphical models, namely Markov networks, in a unifying representation. The main idea behind Markov logic is that, unlike first-order logic, a world that violates a formula is not invalid, but only less probable. This is done by attaching weights to first-order logic formulas: the higher the weight, the bigger is the difference between a world that satisfies the formula and one that does not, other things been equal. These weighted formulas represent a Markov logic network.

A Markov logic network (MLN) [4] L is a set of pairs (F_i,w_i), where F_i is a formula in first-order logic and w_i is a real value representing the weight of the formula. If a set of constants $C = \{c_1,...,c_n\}$ is provided, we can construct a Markov network $M_{L,C}$, called a ground Markov network, as follows:

- A binary node is created for each possible grounding of each atom in L, being its value 1 if the ground atom is true, 0 otherwise.
- Each possible grounding of each formula F_i in L will generate a distinct feature, being its value 1 if the ground formula is true, 0 otherwise. The weight of the feature is the w_i associated with the formula.

This way, it is created a node for each ground atom and an edge if two ground atoms appear in the same formula. Suppose we have a simple MLN with two formulas.

Table 1. Markov logic network example

Formula	Weight
$\forall x\ Steal(x) \Rightarrow \Pr ison(x)$	3
$\forall x \forall y\ CrimePartners(x,y) \wedge Steal(x) \Rightarrow \Pr ison(y)$	1.5

Using the previous algorithm, if we have two constants, Anna and Bob, the resulting ground Markov network will have eight variables, corresponding to eight grounded atoms.

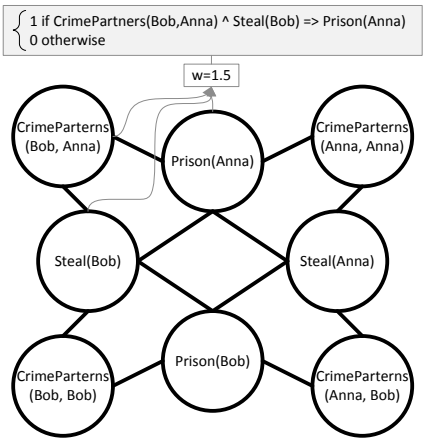

Figure 1. Ground Markov network built from the previous Markov logic network with one example feature.

We can define the probability distribution of a ground Markov Network as:

$$P(X = x) = \frac{1}{Z}\exp(\sum_{i=1}^{F} w_i n_i(x))$$

In this formula, F is the number of formulas in the MLN, $n_i(x)$ is the number of true groundings of F_i in the world x, w_i is the weight of F_i, and Z is a normalizing constant obtained by summing the formula to all the possible worlds.

This probability distribution can be used to answer probabilistic queries about the world. In this work, we are interested in finding marginal and conditional probabilities of some events. However, this task can be intractable in very large domains, so these probabilities are usually obtained by approximation methods, mainly those based on sampling-based techniques like Markov Chain Monte Carlo (MCMC) [4].

3. FROM OWL TO MLN

For the purpose of this work, we will use OWL-DL, a Web Ontology language proposed by the W3C. This language is based on the very expressive description logic $\mathcal{SHOIN(D)}$. Like other description logics, $\mathcal{SHOIN(D)}$ provides a model-theoretic semantics [5]. This means that descriptions can be identified with formulas in first-order logic. The main idea behind this identification is that concepts correspond to unary predicates, roles to binary predicates, and individuals correspond to constants.

So, given an OWL-DL ontology, we can interpret its semantics as a set of first-order formulas. Now we need to find the weights to those formulas. One way of learning those weights is through example data, by generatively maximizing the pseudo-likelihood of that data [4] (i.e., approximate the distribution of the features given the example data). In this work, we will use instances as example data to learn those weights.

4. EXPERIMENT

As experimentation of our approach, we choose GoldDLP[1], an ontology describing a financial domain. In this ontology, there is information about a bank that offers services like loans and credit cards to private persons. One of the most interesting tasks in this domain is to determine if a given loan is a problematic loan. There is an OWL class responsible for that information, named *ProblemLoan*, and some simple rules about that class (for example, *ProblemLoan* is the complement of *OkLoan*). So, the main task in this experiment is to use Markov logic to determine each loan's probability of being a *ProblemLoan*. For this purpose, we translated the OWL ontology to first-order logic and divided the formulas in two sets: one corresponding to the intensional knowledge (i.e., the structure) of the ontology, which will be the base of our MLN, and another corresponding to the extensional knowledge (i.e., the instances), which will be our evidence data. Next, we used Alchemy[2] to generatively learn the weights of our base MLN using the evidence data. Using MCMC, we queried for the probability of an individual being a *ProblemLoan*.

Nine loans have a probability >90% of being a *ProblemLoan*. If we compare the results with a non-probabilistic reasoner, like Pellet [3], these are the same nine instances identified deterministically by it. However, our approach returns some more interesting results that were not identified by Pellet. All the other loans have a probability between 35-39% of being a *ProblemLoan*. This information is valuable because, roughly speaking, it demonstrates that any loan has an associated probability of being a problematic loan. This kind of results cannot be achieved using non-probabilistic reasoning, and therefore demonstrates the necessity of probabilistic reasoning to have a more profound understanding about the domain.

5. CONCLUSION AND FUTURE WORK

In this work we used Markov logic to learn and reason about uncertainty in OWL-DL ontologies. Our preliminary experimentation shows interesting results, and we will continue to explore this approach by experimenting with more ontologies. Since there are many ontologies with no instances, we are also studying techniques to learn evidence data from textual corpus about the domain of the ontology, and use it to learn the weights. We are also exploring the possibility of learn the weights collectively from multiple ontologies about the same domain.

6. REFERENCES

[1] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," Scientific American, vol. 284, 2001, pp. 28-37.

[2] T. Lukasiewicz and U. Straccia, "Managing Uncertainty and Vagueness in Description Logics for the Semantic Web," Web Semantics Sci Serv Agents World Wide Web, 2008.

[3] A. Tversky and D. Kahneman, "Judgment under Uncertainty: Heuristics and Biases," Science, vol. 185, Sep. 1974, pp. 1124-1131.

[4] P. Domingos, S. Kok, D. Lowd, H. Poon, M. Richardson, and P. Singla, "Markov Logic," Probabilistic Inductive Logic Programming, 2008, pp. 92-117.

[5] 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, 2007.

[1] http://www.cs.put.poznan.pl/alawrynowicz/semintec.htm

[2] http://alchemy.cs.washington.edu/

[3] http://pellet.owldl.com/