Learning Probabilistic Description Logic Concepts

Under Different Assumptions on Missing Knowledge

Pasquale Minervini
Dipartimento di Informatica
Università degli Studi di Bari
pasquale.minervini@uniba.it

Claudia d'Amato
Dipartimento di Informatica
Università degli Studi di Bari
claudia.damato@di.uniba.it

Nicola Fanizzi
Dipartimento di Informatica
Università degli Studi di Bari
fanizzi@di.uniba.it

ABSTRACT

Knowledge available through Semantic Web standards can be missing, generally because of the adoption of the Open World Assumption. We present a Statistical Relational Learning system for learning terminological naïve Bayesian classifiers, which estimate the probability that an individual belongs to a target concept given its membership to a set of Description Logic concepts. During the learning process, we consistently handle the lack of knowledge that may be introduced by the adoption of the Open World Assumption, depending on the varying nature of the missing knowledge itself.

1. INTRODUCTION

Real-world knowledge generally involves some degrees of uncertainty and imprecision; for this reason, on the Semantic Web (SW) [1] difficulties arise when trying to model real-world domains using purely logical formalisms. The World Wide Web Consortium (W3C), recognising the need of soundly represent such knowledge, in 2007 created the Uncertainty Reasoning for the World Wide Web Incubator Group [1] (URW3-XG), with the aim of identifying the requirements for reasoning with and representing the uncertain knowledge in Web-based information. A wide range of approaches to represent and infer with knowledge enriched with probabilistic information has been proposed, ranging from extensions of existing knowledge representation standards to probabilistic enrichment of Description Logics or logic programming formalisms.

Motivation

The main problem of applying these approaches in real settings is given by the fact that they almost always assume the availability of probabilistic information. However, except of seldom cases, this information would be hardly known in advance. Having a method that, exploiting available knowledge, (such as an already designed and populated ontology) is able to capture the necessary logic and probabilistic structure, would be of great benefit.

Also, *Open World Assumption* (OWA) is often employed when reasoning through SW knowledge bases (e.g. when OWL is considered as a syntactic variant of some Description Logic): under OWA, a statement is true or false only if its truth value can be formally derived. As a consequence, there can be some cases (e.g. determining if an individual is a member of a given concept) for which the truth value cannot be determined (it cannot be derived neither that the individual is instance of the considered concept nor that the individual is instance of the negated concept). This is opposed by the commonly employed *Closed World Assumption* (CWA), where every statement that cannot be proved to be true, is assumed to be false. In this paper, we face the problem of finding a set of logic features (in the form of Description Logic concepts) that, used within a probabilistic model, can be used to estimate the probability of a previously unknown concept membership between a generic individual and a target concept. For such reason, our method has to handle the case in which the membership relation between such individual and logic features cannot be inferred, consistently with the potential information contained in this lack of knowledge.

Related Work

Within the SW, Machine Learning (ML) is going to cover a relevant role in the analysis of distributed data sources described using SW standards [22], with the aim of discovering new and refining existing knowledge. A collection of ML approaches oriented to SW have already been proposed in literature, ranging from propositional and single-relational (e.g. SPARQL-ML [13], or based on low-rank matrix approximation techniques such as in [22, 21]) to multi-relational (e.g. distance-based [5, 10] or kernel-based [9, 2]).

In the class of multi-relational learning methods, *Statistical Relational Learning* [12] (SRL) ones seem particularly appealing, being designed to learn in domains with both a complex relational and a rich probabilistic structure; the URW3-XG provided in [15] a large group of situations in which knowledge on the SW needs to represent uncertainty. There have already been some proposals regarding the adaptation and application of SRL systems to the SW, e.g. [7] proposes to employ Markov Logic Networks [19] (MLN) for first-order probabilistic inference and learning within the SW, and [17] proposes to learn first-order probabilistic theories in a probabilistic extension of the $\mathcal{ALC}$ Description Logic named CR$\mathcal{ALC}$. However, ML techniques proposed so far for the SW do not explicitly consider the nature of the missing knowledge during learning – e.g. matrix completion methods in [21] inherently assume data is Missing At Random, CR$\mathcal{ALC}$ learning methods tend to learn theories sensitive to a specific ignorance model, and proposed MLN learning methods resort to Closed World Assumption (or the Missing At Random assumption) during learning.

Learning from incomplete knowledge bases by adopting methods not coherent with the nature of the missing knowledge itself (e.g. expecting the value of a random variable being missing to

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

SAC'12 March 25-29, 2012, Riva del Garda, Italy.

be non-informative on the actual value of such variable, when it is) can lead to misleading results with respect to the real model followed by the data [6]. In the rest of this paper, we will first describe Bayesian Networks (representation, inference and learning) and a proposed extension, called Robust Bayesian Estimator, making use of probability intervals; then we will describe our probabilistic-logic model, terminological Bayesian classifiers, and the problem of learning them from a set of training individuals and a Description Logic knowledge base. Also, we will describe our learning algorithm, and the adaptations to learn under different assumptions on the ignorance model. In the final part, we will give experimental evidence on the effectiveness of our method.

2. BAYESIAN NETWORKS AND ROBUST BAYESIAN ESTIMATION

Graphical models [14] (GMs) are a popular framework to compactly describe the joint probability distribution for a set of random variables, by representing the underlying structure through a series of modular factors. Depending on the underlying semantics, GMs can be grouped into two main classes: *directed graphical models*, which found on directed graphs, and *undirected graphical models*, founding on undirected graphs.

A Bayesian network (BN) is a directed GM which represents the conditional dependencies in a set of random variables by using a directed acyclic graph (DAG) $\mathcal{G}$ augmented with a set of conditional probability distributions $\theta_{\mathcal{G}}$ (also referred to as *parameters*) associated with $\mathcal{G}$'s vertices. In such graph, each vertex corresponds to a random variable X_i and each edge indicates a *direct influence* relation between the two random variables. A BN stipulates a set of *conditional independence assumptions* over its set of random variables: each vertex X_i in the DAG is conditionally independent of any subset $S \subseteq Nd(X_i)$ of vertices that are not descendants of X_i given a joint state of its parents, or formally: $\forall X_i : \Pr(X_i \mid S, parents(X_i)) = \Pr(X_i \mid parents(X_i))$, where the function $parents(X_i)$ returns the parent vertices of X_i in the DAG representing the BN. The conditional independence assumption allows to represent the *joint probability distribution* $\Pr(X_1, \ldots, X_n)$ defined by a Bayesian network over a set of random variables $\{X_1, \ldots, X_n\}$ as a production of the individual probability distributions, conditional on their parent variables:

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

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

Given a BN specifying a joint probability distribution over a set of variables, it is possible to evaluate inference queries by marginalization, like calculating the posterior probability distribution for a set of query variables given some observed event (i.e. assignment of values to the set of evidence variables). Exact inference for general BNs is an NP-hard problem, but algorithms exist to efficiently infer in restricted classes of networks, such as variable elimination, which has linear complexity in the number of vertices if the BN is a singly connected network [14]. Approximate inference methods also exist in literature, such as *Monte Carlo* algorithms, *belief propagation* or *variational methods* [14].

The compact parametrization in graphical models allows for effective learning, both model selection (structural learning) and parameter estimation. In the case of BNs, however, finding a model which is optimal with respect to a given scoring criterion (which measures how well the model fits observed data) may not be trivial: the number of possible structures for a BN is super-exponential in the size of its vertices, making it generally impractical to perform an exhaustive search through the space of its possible models. For such reason, in our approach, we tried to find an acceptable trade-off between efficiency and expressiveness, so to make our method suitable for a context like SW: we decided to focus on a particular subclass of Bayesian networks, i.e. *naïve Bayesian networks*, modelling the dependencies between a set of random variables $\mathcal{X} = \{X_1, \ldots, X_n\}$, also called *features*, and a random variable C, also called *class*, so that each pair of features are independent of each other given the class, i.e. $\forall X_i, X_j \in \mathcal{X} : i \neq j \Rightarrow (X_i \perp\!\!\!\perp X_j \mid C)$. This kind of models is especially interesting since they proved to be effective also in contexts in which the underlying independence assumptions are violated [8], even outperforming more current approaches [3]. It is relevant to note that BNs can be used as classifiers, by assigning each new, unclassified instance to the class C maximizing the probability value $\Pr(C \mid e)$, where e indicates the evidence available about the instance and Pr the probability distribution represented by the BN.

However, defining a BN requires a number of precise probability assessments which, as we will see, will not be always possible to obtain. A generalisation of naïve Bayesian networks to probability intervals is the *robust Bayesian estimator* [18] (RBE): each conditional probability in the network is a *probability interval* characterised by its *lower* and *upper bounds*, defined respectively as $\underline{\Pr}(A) = \min_{\Pr \in \mathcal{P}} \Pr(A)$ and $\overline{\Pr}(A) = \max_{\Pr \in \mathcal{P}} \Pr(A)$, where $\mathcal{P}$ is a convex set of probability distributions. An approach very similar to RBE is presented in [4] and proposes using *Credal networks* (which are structurally similar to a BN, but where the conditional probability densities belong to convex sets of mass functions) to represent uncertainty about network parameters.

A problem with this class of approaches arises when using such model for classification – in the case of binary classification with classes C_1 and C_2, given evidence e for a new, unclassified instance, two posterior intervals are obtained, i.e. $\mathcal{P}(C_1 \mid e)$ and $\mathcal{P}(C_2 \mid e)$. If such intervals do not overlap, the *stochastic dominance criterion* can be employed, which assigns a new unclassified instance to class C_1 iff $\underline{\mathcal{P}}(C_1 \mid e) > \overline{\mathcal{P}}(C_2 \mid e)$; otherwise, [18] proposes using a weaker criterion, called *weak dominance criterion*, which is based on representing each probability interval into a single probability value represented by its middle point.

3. TERMINOLOGICAL NAÏVE BAYESIAN CLASSIFIERS

The learning problem we intend to focus on consists in, given a set of training individuals, learning a terminological naïve Bayesian classifier $\mathcal{N}_{\mathcal{K}}$; this is defined as a naïve BN modelling the dependency relations between a set of Description Logic (DL) concepts (also referred to as *feature concepts*) and a target concept C. Training individuals are distinguished in *positive*, *negative* and *neutral*, belonging respectively to the target concept C, $\neg C$ and whose membership of C is unknown. A terminological Bayesian classifier can be defined as follows:

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

- *$\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ is a directed acyclic graph, in which:*
  - *$\mathcal{V} = \{F_1, \ldots, F_n, C\}$ is a set of vertices, each $V \in \mathcal{V}$ representing a random variable associated to a DL concept defined over $\mathcal{K}$; F_i's are associated to feature concepts, and C to the target (class) concept;*

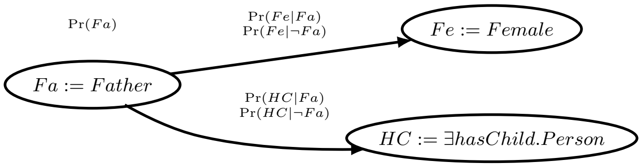
  - $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ *is a set of edges, modelling the dependence relations between the elements of* $\mathcal{V}$*.*

- $\Theta_{\mathcal{G}}$ *is a set of* conditional probability distributions *(CPD), one for each* $V \in \mathcal{V}$*, representing the conditional probability distribution of the feature concept given the state of its parents in the graph.*

Given a generic individual $a \in N_I$*, each variable* $V \in \mathcal{V}$ *in the network has value* $True$ *(resp.* $False$*) if* $\mathcal{K} \models V(a)$ *(resp.* $\mathcal{K} \models \neg V(a)$*)* [2]*, otherwise (i.e.* $\mathcal{K} \not\models C(a) \wedge \mathcal{K} \not\models \neg C(a)$*) its value is considered as not observable. If the concept-membership relation between* a *and* C *is not known, its probability can be estimated using BN inference algorithms.*

In the case of terminological naïve Bayesian Classifiers, $\mathcal{E} = \{\langle C, F_i \rangle \mid i \in \{1, \ldots, n\}\}$, i.e. each feature concept is independent on other feature concepts, given the value of the target concept.

EXAMPLE 1. *(Example of Terminological Naïve Bayesian Classifier) Given a set of DL feature concepts* $\mathcal{F} = \{Fe := Female, HC := \exists hasChild.Person\}$ [3] *and a target concept* $Father$*, a terminological naïve Bayesian classifier expressing the target concept in terms of the feature concepts is the following:*



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

$$\Pr(Fa(a)) = \frac{\Pr(Fa)\Pr(HC \mid Fa)}{\sum\limits_{Fa' \in \{Fa, \neg Fa\}} \Pr(Fa')\Pr(HC \mid Fa')};$$

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

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

Given *the following:*

- *a* target concept C*;*
- *a DL KB* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$*, so that:*
  - $\forall a \in Ind^+_C(\mathcal{A}) : \mathcal{K} \models C(a)$*,*
  - $\forall a \in Ind^-_C(\mathcal{A}) : \mathcal{K} \models \neg C(a)$*,*
  - $\forall a \in Ind^0_C(\mathcal{A}) : \mathcal{K} \not\models C(a) \wedge \mathcal{K} \not\models \neg C(a)$*;*
- *a* scoring function *specifying a measure of the quality of an induced terminological Bayesian classifier* $\mathcal{N}_{\mathcal{K}}$ *w.r.t. the samples in* $Ind_C(\mathcal{A}) = \bigcup_{v \in \{+,-,0\}} Ind^v_C(\mathcal{A})$*;*

Find *a network* $\mathcal{N}^*_{\mathcal{K}}$ *maximizing the score function with respect to the samples:* $\mathcal{N}^*_{\mathcal{K}} \leftarrow \underset{\mathcal{N}_{\mathcal{K}}}{\arg\max} score(\mathcal{N}_{\mathcal{K}}, Ind_C(\mathcal{A})))$*.*

The search space to find the optimal network $\mathcal{N}^*_{\mathcal{K}}$ may be too large to explore exhaustively; therefore our learning algorithm, outlined in Alg. 1, works by greedily searching the space of features (i.e. DL complex concepts) for the ones that maximize the score of the induced network, with respect to a scoring function, and incrementally building the resulting network. While the features are added one by one, the search in the space of DL complex concepts is made through a beam search, starting from a concept $Start$ and gradually specializing candidate feature concepts, by employing a DL refinement operator [16]. For each new complex concept being evaluated, the algorithm creates a new set of concepts/variables $\mathcal{V}'$ and finds the optimal structure, under a given set of constraints (which, in the case of terminological naïve Bayesian classifiers, is already fixed) and parameters (which may vary depending on the assumptions on the nature of the ignorance model). Then, the new network is scored, with respect to a given scoring criterion.

Algorithm 1 Algorithm for Learning Terminological Bayesian Classifiers

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

In our algorithm, the $extend$ function greedily searches for a new (complex) feature concept which can improve the whole network's score (determined by a scoring function $score$). The search through the space of concept definitions is performed through a beam search, using the $\rho^{cl}_{\downarrow}$ refinement operator [16] ($\rho^{cl}_{\downarrow}(C)$ returns a set of refinements D of C so that $D \sqsubset C$, which we consider only up to a given concept length n). The functions $optimalNetwork$ and $score$ are used, respectively, to find the optimal Bayesian network structure between the nodes in the network (eventually under a set of constraints, like in the naïve Bayes case or some of its extensions) and for scoring a classifier (to compare its effectiveness with others). However, those two functions are sensitive to the assumptions made about the ignorance model.

[2] Each node is named after its associated concept for brevity.

[3] Aliases for DL concepts are used for brevity.

Different Assumptions on the Ignorance Model

Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a DL KB; under OWA, it is not always possible to know if a generic DL assertion α is or is not entailed by $\mathcal{K}$ (i.e. there may be cases in which $\mathcal{K} \not\models \alpha \wedge \mathcal{K} \not\models \neg\alpha$). We characterize such a lack of knowledge about concept-memberships through the probability distribution of the ignorance model [20]. Given a concept D, a generic individual a, an *ignorance model* $\mathcal{IM}$ and a DL KB $\mathcal{K}^*$ such that we can extract, by using $\mathcal{IM}$, $\mathcal{K}$ as a fragment of $\mathcal{K}^*$, i.e. $\forall \alpha : \mathcal{K} \models \alpha \Rightarrow \mathcal{K}^* \models \alpha \wedge \mathcal{K}^* \models \alpha \not\Rightarrow \mathcal{K} \models \alpha$.

Given a probabilistic model that calculates the probability that the concept-membership between D and a is unknown in $\mathcal{K}$, we can say that the ignorance model underlying the concept-membership between a and D in $\mathcal{K}$ (given $\mathcal{K}^*$) is one of the following:

- **MCAR** (Missing Completely at Random) – when the probability for such concept-membership to be missing is independent on the knowledge on a available in $\mathcal{K}^*$: $\Pr(\mathcal{K} \not\models D(a) \wedge \mathcal{K} \not\models \neg D(a) \mid \mathcal{K}^*) = \Pr(\mathcal{K} \not\models D(a) \wedge \mathcal{K} \not\models \neg D(a))$.

- **MAR** (Missing At Random) – when the probability for such concept-membership to be missing depends on the knowledge on a available in $\mathcal{K}$: $\Pr(\mathcal{K} \not\models D(a) \wedge \mathcal{K} \not\models \neg D(a) \mid \mathcal{K}^*) = \Pr(\mathcal{K} \not\models D(a) \wedge \mathcal{K} \not\models \neg D(a) \mid \mathcal{K})$.

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

When the assumed ignorance model is **MCAR**, *Available Case Analysis* [14] can be used, which builds an unbiased estimator of the network parameters using only available knowledge. A scoring function that can be used for this case is network's class-conditional log-likelihood on positive and negative training individuals, defined as [4]:

$$\begin{aligned}\mathcal{L}(\mathcal{N}_{\mathcal{K}} \mid Ind_C(\mathcal{A})) = & \sum_{a \in Ind_C^+(\mathcal{A})} \log \Pr(C(a) \mid \mathcal{N}_{\mathcal{K}}) \\ & + \sum_{a \in Ind_C^-(\mathcal{A})} \log \Pr(\neg C(a) \mid \mathcal{N}_{\mathcal{K}}); \qquad (1)\end{aligned}$$

A problem with using simple log-likelihood for finding optimal feature set and structure, is that it grows monotonically with the number of edges and features. To avoid overfitting, it is possible to resort to log-likelihood-based scoring criteria, such as the *Bayesian Information Criterion* (BIC) or the *Akaike Information Criterion* (AIC) [14], which subtract a penalty score to the log-likelihood proportional to the complexity of the model (which, in case of BIC and AIC, are respectively $(|\Theta_{\mathcal{G}}|/2) \log n$ and $|\Theta_{\mathcal{G}}|$, where $|\Theta_{\mathcal{G}}|$ is the number of independent network parameters and n is the number of data points). Under the naïve Bayes assumption, there is no need to perform a search for finding the optimal network, since the structure is already fixed (each node except the target concept node has only one parent, i.e. the target concept node); otherwise, finding a network structure which is optimal under some criterion (e.g. the BIC score [14]) may require an exhaustive search in the space of possible structures. However, for an extension of naïve Bayesian networks (which allows for a tree structure among feature nodes), it is possible to efficiently compute the optimal structure employing the method in [11], making it appealing for real-life applications requiring efficiency and ability to scale.

In the **MAR** case, a possible solution for learning models accounting for missing knowledge is to use the Expectation - Maximization (EM) algorithm, MCMC sampling or gradient ascent [14]. We use EM to learn terminological naïve Bayesian classifiers from MAR data. In our approach, outlined in Alg. 2, we first heuristically estimate network's parameters by only using available data; then, we consider individuals whose membership to a generic concept D is not known as several fractional individuals belonging, with different weights (corresponding to the posterior probability of their class membership), to both the components D and $\neg D$; such fractional individuals are used to recalculate network parameters (obtaining the so-called *expected counts* and the process is repeated until convergence (e.g. when the improvement in log-likelihood is lower than a certain threshold).

Algorithm 2 Outline for our implementation of the EM algorithm for parameter learning in a terminological Bayesian classifier.

```
function $EM(\mathcal{N}_{\mathcal{K}}^0, Ind_C(\mathcal{A}))$
  $\mathcal{N}_{\mathcal{K}}^0 = \langle \mathcal{G}, \Theta_{\mathcal{G}}^0 \rangle, \mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle; t \leftarrow 0;$
  repeat
      $\{\bar{n}(x_i, \pi_{x_i})\} \leftarrow ExpCounts(\mathcal{N}_{\mathcal{K}}, Ind_C(\mathcal{A}));$
      for $X_i \in \mathcal{V}$ do
          for $\langle x_i, \pi_{x_i} \rangle \in vals(X_i, parents(X_i))$ do
              $\theta_{\mathcal{G}}^{t+1}(x_i, \pi_{x_i}) \leftarrow \frac{\bar{n}(x_i, \pi_{x_i})}{\sum_{x'_i \in vals(X_i)} \bar{n}(x'_i, \pi_{x_i})};$
          end for
      end for
      $t \leftarrow t + 1; \mathcal{N}_{\mathcal{K}}^t = \langle \mathcal{G}, \Theta_{\mathcal{G}}^t \rangle;$
  until $\mathcal{L}(\mathcal{N}_{\mathcal{K}}^t \mid Ind_C(\mathcal{A})) - \mathcal{L}(\mathcal{N}_{\mathcal{K}}^{t-1} \mid Ind_C(\mathcal{A})) \leq \tau;$
  return $\mathcal{N}_{\mathcal{K}}^t;$
function $ExpCounts(\mathcal{N}_{\mathcal{K}}, Ind_C(\mathcal{A}))$
  $\mathcal{N}_{\mathcal{K}} = \langle \mathcal{G}, \Theta_{\mathcal{G}} \rangle, \mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle;$
  for $X_i \in \mathcal{V}$ do
      for $\langle x_i, \pi_{x_i} \rangle \in vals(X_i, parents(X_i))$ do
          $\bar{n}(x_i, \pi_{x_i}) \leftarrow 0;$
      end for
  end for
  for $a \in Ind_C(\mathcal{A})$ do
      for $X_i \in \mathcal{V}$ do
          for $\langle x_i, \pi_{x_i} \rangle \in vals(X_i, parents(X_i))$ do
              $\bar{n}(x_i, \pi_{x_i}) \leftarrow \bar{n}(x_i, \pi_{x_i}) + \Pr(x_i, \pi_{x_i} \mid \mathcal{N}_{\mathcal{K}});$
          end for
      end for
  end for
  return $\{\bar{n}(x_i, \pi_{x_i})\};$
```

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

- **Expectation step** – using available data and the current network parameters, calculate a distribution over possible completions for the missing knowledge;

- **Maximization step** – considering each possible completion as a fully available data case (weighted by its probability), calculate next parameters using frequency counting.

About finding optimal structures for networks with less restrictions on their structure (such as tree-augmented naïve BNs or unrestricted BNs) from MAR data, it is possible to employ the Structural EM (SEM) algorithm [14]. In SEM, the maximization step is performed both in the space of structures $\mathcal{G}$ and in the space of parameters $\Theta_{\mathcal{G}}$, by first searching a better structure and then the best

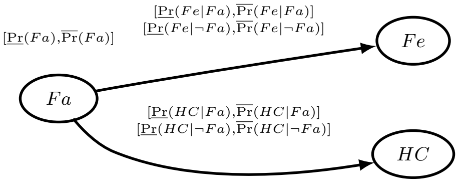
[4] Log-likelihoods here are calculated ignoring available knowledge about the membership between individuals and target concept.

parameters associated to the given structure; it can be proven that, if the search procedure finds a structure that is better than the one used in the previous iteration with respect to e.g. the BIC score, then the SEM algorithm will monotonically improve the score.

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

An alternate solution is *Robust Bayesian Estimation* [18] (RBE), which allows to learn (interval-valued) conditional probability distributions without making any sort of assumption about the nature of the missing data. RBE allows to infer posterior probability intervals instead of single posterior probability values, obtained by taking in account all the possible fillings of the missing knowledge. In [18], a method for efficiently calculate interval network parameters and posterior intervals [5] is provided. To score each induced network, we empirically chose to calculate posterior intervals, get their central point and then use them as probability values to calculate the log-likelihood as in Equation 1. Another evaluation approach has been proposed in [23] to compare credal classifiers, and proposes using a scoring criterion based on discounted accuracy and a function indicating risk-aversion.

EXAMPLE 2. *(Example of Terminological Naïve Bayesian Classifier using Robust Bayesian Estimation) Consider again the terminological naïve Bayesian classifier in Example 1: when learning in presence of NMAR knowledge, it can be extended with interval-valued network parameters for inferring posterior probability intervals instead of single posterior probability values through Robust Bayesian Estimation. In such class of networks, conditional probability tables associated to each node contain convex intervals of probability values instead of single probability values, each defined by its upper and lower bound.* [6]


$[\underline{\Pr}(Fa),\overline{\Pr}(Fa)]$
$[\underline{\Pr}(Fe|Fa),\overline{\Pr}(Fe|Fa)]$
$[\underline{\Pr}(Fe|\neg Fa),\overline{\Pr}(Fe|\neg Fa)]$
Fa
Fe
$[\underline{\Pr}(HC|Fa),\overline{\Pr}(HC|Fa)]$
$[\underline{\Pr}(HC|\neg Fa),\overline{\Pr}(HC|\neg Fa)]$
HC

Interval-valued network parameters can be calculated efficiently [18]. E.g. the parameters associated to the feature concept HC can be calculated as follows:

$$\overline{n}(HC \mid Fa)= n(? \mid Fa) + n(HC \mid ?) + n(? \mid ?);$$

$$\underline{n}(HC \mid Fa)= n(? \mid Fa) + n(\neg HC \mid ?) + n(? \mid ?);$$

$$\overline{\Pr}(HC \mid Fa)= \frac{n(HC|Fa)+\overline{n}(HC|Fa)}{n(Fa)+\overline{n}(HC|Fa)};$$

$$\underline{\Pr}(HC \mid Fa)= \frac{n(HC|Fa)}{n(Fa)+\underline{n}(HC|Fa)};$$

where $n(? \mid Fa) = |\{a \in Ind^{+}_{Fa}(\mathcal{A}) \mid \mathcal{K} \not\models HC(a) \wedge \mathcal{K} \not\models \neg HC(a)\}|$, $n(HC \mid ?) = |\{a \in Ind^{0}_{Fa}(\mathcal{A}) \mid \mathcal{K} \models HC(a)\}|$ *and* $n(? \mid ?) = |\{a \in Ind^{0}_{Fa}(\mathcal{A}) \mid \mathcal{K} \not\models HC(a) \wedge \mathcal{K} \not\models \neg HC(a)\}|$.

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

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

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

4. EXPERIMENTS

In this section we aim at empirically assess the impact of different missing knowledge handling methods when learning terminological naïve Bayesian classifiers from real world ontologies.

Ontology	DL Expressivity	#Axioms	#Individuals
MDM0.73	$\mathcal{ALCHOF}(\mathcal{D})$	1098	112
Leo	$\mathcal{ALCHIF}(\mathcal{D})$	430	61
Family-Tree	$\mathcal{SROIF}(\mathcal{D})$	2059	368
Wine	$\mathcal{SHOIN}(\mathcal{D})$	747	161

Table 1: Ontologies considered in the experiments.

Starting from a set of real ontologies [7] (outlined in table 1), we generated a set of 20 random query concepts for each ontology [8], 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 standard reasoner [9] was employed to decide on the theoretical class-membership (and non-membership) of the individuals with respect to the query concepts. In experiments, we re-learned such concept queries as terminological naïve Bayesian classifiers, using individuals retrieved by each query (resp. its complement) as positive (resp. negative) examples. The evaluated learning approaches were Available Case Analysis (ACA), EM algorithm (EM), Robust Bayesian Estimation (ROBUST) and two considering both observable and missing observations (IM^3 and IM^2). The last two approaches build networks which are dependant on the ignorance model: IM^3 (for Informatively Missing) makes use of three-valued feature variables take a value in $\{True, False, Unknown\}$ when the membership to the associated feature concept is respectively true, false or not known; and IM^2, in which two values feature variables take a value in $\{True, Other\}$, when the membership to the associated feature concept is respectively true or one of false or not known. During experiments, refinements were only allowed to contain conjunctions/disjunctions of concepts, complements and existential restrictions, and refinements started from concept $\top$. To avoid overfitting, the construction of each new network was driven by the BIC score. In experiments, each of the 20 generated query concepts generated, was used to obtain a pair of sets composed by positive and negative examples, selecting the individuals in the ontology belonging respectively to the query concept and its complement. On each of these pair of positive/negative examples, k-fold cross validation (with $k = 10$) was used to estimate k accuracy values (for ROBUST, discounted accuracy was used, to also consider

[5] A posterior interval estimate represents the range of probability values associated to the membership of an instance to a class.

[6] In the following part, feature concepts will be aliased with the labels as in Example 1 for brevity.

[7] Found on the TONES Ontology Repository – http://owl.cs.manchester.ac.uk/repository/

[8] Using the query concept generation method available at http://lacam.di.uniba.it:8000/~nico/research/ontologymining.html

[9] Pellet v2.2.2 – http://clarkparsia.com/pellet/

the cases in which an unique class label could not be identified). Results are summarised in Table 2.

	ACA	EM	IM^3	IM^2	ROBUST
Leo	$.97 \pm .08$	$.97 \pm .08$	$.94 \pm .12$	$.97 \pm .08$	$.93 \pm .14$
MDM	$.96 \pm .07$	$.96 \pm .07$	$.95 \pm .07$	$.97 \pm .05$	$.9 \pm .1$
Wine	$.91 \pm .12$	$.91 \pm .12$	$.92 \pm .13$	$.94 \pm .11$	$.88 \pm .12$
F-T	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0

Table 2: Cross-validated accuracy results on the generated data sets – for each ontology in Table 1, 20 query concepts were generated, and each was used to obtain a sample of positive/negative individuals, which were used to evaluate the methods using k-fold cross validation (with $k = 10$).

Both the parameters $depth$ and $maxLength$, indicating respectively the maximum depth of each refinement step and the maximum length of a feature concept, were both set to 3 (2 in the case of the more complex ontology Family-Tree). In two cases, IM^2 achieved better results than other methods; this is particularly true in Wine, where the 20 accuracy values obtained by IM^2 were greater than those obtained by ACA and EM (wit $p < 0.05$ measured with a paired Student's t-test). IM^3 was modelling both observability and ignorance too, but the higher number of parameters (feature variables were three-valued) caused it to have a higher number of parameters, which is penalized by the BIC score. In the Family-Tree ontology, all methods achieved nearly perfect accuracy; the reason is that generated query concepts, in this case, were summarised by single, shorter concepts that the system was able to learn as feature concepts (e.g. $GrandparentOfRobert$, $\neg Man$, or $Woman$ [10]). In Leo, ACA, EM and IM^2 achieved nearly the same results suggesting that, in this particular case, the missingness of the discriminant features could be ignored. In general, ROBUST was often overcautious during classification (thanks to its ability to find the cases in which changes in the missingness patterns can cause a different classification), which caused it to have a lower, discounted accuracy.

5. CONCLUSIONS AND FUTURE WORK

We presented a SRL system designed for learning terminological naïve Bayesian classifiers, which estimates the probability that a generic individual belongs to a certain target concept, given its membership relation to an induced set of complex Description Logic concepts. We gave a characterisation of the lack of knowledge that may be introduced by the OWA depending on the underlying ignorance model, and handled such missing knowledge, during learning, under different assumptions on the nature of missing knowledge. In our future work, we aim at estimating computationally the ignorance model followed by each feature concept, at developing new methods to exploit the potential information contained in missigness as well as new scoring and searching techniques, and evaluate our methods more extensively on real world ontologies.

6. REFERENCES

[1] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, May 2001.

[2] V. Bicer, T. Tran, and A. Gossen. Relational kernel machines for learning from graph-structured rdf data. In *ESWC2011*.

[3] R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *ICML2006*.

[4] G. Corani and M. Zaffalon. Learning reliable classifiers from small or incomplete data sets: The naive credal classifier 2. *Journal of Machine Learning Research*, 9:581–621, 2008.

[5] C. d'Amato, N. Fanizzi, and F. Esposito. Query answering and ontology population: an inductive approach. In *ESWC2008*.

[6] S. R. de Morais and A. Aussem. Exploiting data missingness in bayesian network modeling. In *IDA2009*.

[7] P. Domingos, D. Lowd, S. Kok, H. Poon, M. Richardson, and P. Singla. Just add weights: Markov logic for the semantic web. In *URSW2008*, pages 1–25.

[8] P. Domingos and M. J. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130, 1997.

[9] N. Fanizzi, C. D'Amato, and F. Esposito. Learning with kernels in description logics. In *ILP2008*.

[10] N. Fanizzi, C. D'Amato, and F. Esposito. Reduce: A reduced coulomb energy network method for approximate classification. In *ESWC2009*.

[11] N. Friedman et al. Bayesian network classifiers. In *Machine Learning*, pages 131–163, 1997.

[12] L. Getoor and B. Taskar. *Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2007.

[13] C. Kiefer, A. Bernstein, and A. Locher. Adding data mining support to sparql via statistical relational learning methods. In *ESWC2008*.

[14] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[15] K. J. Laskey and K. B. Laskey. Uncertainty reasoning for the world wide web: Report on the urw3-xg incubator group. In *URSW2008*.

[16] J. Lehmann et al. Concept learning in description logics using refinement operators. *Mach. Learn.*, 78:203–250.

[17] J. E. O. Luna and F. G. Cozman. An algorithm for learning with probabilistic description logics. In *URSW2009*.

[18] M. Ramoni and P. Sebastiani. Robust learning with missing data. *Mach. Learn.*, 45:147–170, October 2001.

[19] M. Richardson and P. Domingos. Markov logic networks. *Mach. Learn.*, 62:107–136, February 2006.

[20] D. B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.

[21] V. Tresp, M. Bundschus, Y. Huang, and A. Rettinger. Materializing and querying learned knowledge. In *IRMLeS2009*.

[22] V. Tresp, M. Bundschus, A. Rettinger, and Y. Huang. Towards machine learning on the semantic web. In *URSW2008*.

[23] M. Zaffalon, G. Corani, and D. Mauá. Utility-based accuracy measures to empirically evaluate credal classifiers. In *ISIPTA 2011*, pages 401–410, Innsbruck.

[10] Namespaces were omitted for brevity.