Learning Terminological Naïve Bayesian Classifiers

Under Different Assumptions on Missing Knowledge

Pasquale Minervini, Claudia d'Amato, and Nicola Fanizzi

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

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

1 Introduction

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

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

Motivation

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

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

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

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

Related Work

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

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

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

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

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

2 Bayesian Networks and Robust Bayesian Estimation

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

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

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

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

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

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

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

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

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

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

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

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

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

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

3 Terminological Naïve Bayesian Classifiers

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

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

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

A terminological Bayesian classifier can be defined as follows:

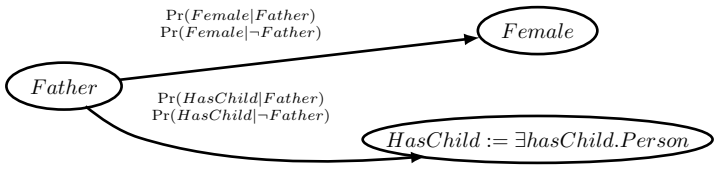
Definition 1 (Terminological Bayesian Classifier). *A terminological Bayesian classifier $\mathcal{N}_{\mathcal{K}}$, with respect to a DL KB $\mathcal{K}$, is defined as a pair $\langle \mathcal{G}, \Theta_{\mathcal{G}} \rangle$, where:*

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

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

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

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



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

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

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

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

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

Given

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

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

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

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

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

Different Assumptions on the Ignorance Model

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

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

Algorithm 1 Algorithm for Learning Terminological Bayesian Classifiers

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

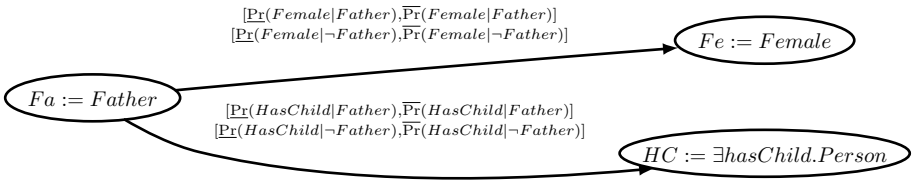
$\mathcal{N}_{\mathcal{K}} = \langle \mathcal{G}, \Theta_{\mathcal{G}} \rangle, \mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$;
for $X_i \in \mathcal{V}$ **do**
  for $\langle x_i, \pi_{x_i} \rangle \in vals(X_i, parents(X_i))$ **do**
    $\{\bar{n}(x_i, \pi_{x_i})$ contains the expected count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

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

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

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

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



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

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

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

4 Conclusions and Future Work

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

References

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

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