Learning Terminological Bayesian Classifiers

A Comparison of Alternative Approaches to Dealing with Unknown Concept-Memberships

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 representation formalisms can be missing, i.e. it is not always possible to infer the truth value of an assertion (due to the Open World Assumption). We propose a method for incrementally inducing terminological (tree-augmented) naïve Bayesian classifiers, which aim at estimating the probability that an individual belongs to a target concept given its membership to a learned set of Description Logic concepts. We then evaluate the impact of employing different methods of handling assertions whose truth value is unknown, each consistent with a different assumption on the ignorance model.

1 Introduction

Real-world knowledge often involves various degrees of uncertainty. For such reason, in the context of Semantic Web (SW), 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; URW3-XG provided in [13] a number of situations in which there is a clear need of explicitly represent and reason in presence of uncertainty. A wide range of approaches to represent and infer with knowledge enriched with probabilistic information has been proposed: some of them extend knowledge representation formalisms actually used in the SW, while others rely on probabilistic enrichment of Description Logics or logic programming formalisms.

Motivation

The main problem of applying such approaches in real world settings is given by the fact that they almost always assume the availability of probabilistic information, while it is hardly known in advance. Having a method that, by exploiting available knowledge (such as an already designed and populated ontology) is able to extract both the needed logic and the probabilistic structure, would be of great benefit. During this process, the

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

Open World Assumption (OWA) must be taken into account: under OWA, an assertion is true or false only if its truth value can be formally derived. As a consequence, there may be reasoning tasks (such as instance checking) for which the truth value cannot be determined. 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. Machine Learning (ML) is already covering a relevant role in the analysis of SW knowledge bases, to overcome the limitations of purely deductive reasoning [17, 10]. In fact, purely deductive inference does not scale up easily to the size of the web, does not exploit regularities in data, the construction of a SW knowledge base can be an expensive process and commonly used SW inference and representation formalisms do not consider the inherent uncertainty characterizing the knowledge in various domains. In this paper, we face the problem of finding a (locally optimal) set of logic features (in the form of Description Logic concepts) that, used within a probabilistic graphical model, can be used to estimate the probability of a previously unknown concept membership relation between a generic individual and a target concept. Also, we evaluate different methods of dealing with missing concept-memberships, each coherent with a different assumption on the missingness mechanism. We will start by describing Bayesian Networks (representation, inference and learning) and their extensions towards probability intervals. Then we will describe our probabilistic-logic model, named *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.

Related Work

A variety of ML approaches specifically designed for SW knowledge bases have been proposed; the expressive power of such ontological knowledge representation formalisms may vary, ranging from languages such as RDF(S) to Description Logics (theoretical fundation of many OWL variants). A recent survey on this topic is in [17]. In the class of multi-relational learning techniques, *Statistical Relational Learning* [7] (SRL) methods seem particularly appealing, being designed to learn in domains with both a complex relational and a rich probabilistic structure. There have been proposals for employing SRL methods when learning from Description Logic knowledge bases: in [4], authors propose to employ Markov Logic Networks [19] (MLN) for first-order probabilistic inference and learning within the SW context; learning concepts in a probabilistic extension of the $\mathcal{ALC}$ Description Logic named CR$\mathcal{ALC}$ is proposed in [15]; in [18], the *Infinite Hidden Relational Models* [22] framework is extended to also take into account a set of constraints in the form of (even more expressive) Description Logic concepts (such as $\mathcal{SHOIN}(D)$). The aforementioned methods rely on probabilistic graphical models, which offer sound methods for both inferencing and learning in the presence of latent variables and missing values [11] (given some assumptions on the missingness pattern), providing a way for handling assertions whose truth value is not known due to the adoption of the OWA. However, in the literature it is not clear whether such assumptions hold in the SW context: this may be an issue, since from incomplete knowledge

bases by adopting methods not coherent with the nature of the missing knowledge itself can lead to misleading results with respect to the real model followed by the data [20].

2 Bayesian Networks and Robust Bayesian Estimation

Graphical models [11] (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*, which found 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 a 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 [11]. Approximate inference methods also exist in literature, such as *Monte Carlo* algorithms, *belief propagation* or *variational methods* [11]. 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 be not 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 this reason we tried to find an acceptable trade-off between efficiency and expressiveness, so to make our method suitable

for a context like SW: we focused on particular subclasses of Bayesian networks in which both inference and structure/parameters learning can be performed in polynomial time. The first is *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. $\forall X_i, X_j \in \mathcal{X} : i \neq j \Rightarrow (X_i \perp\!\!\!\perp X_j \,|\, C)$. This type of models is especially interesting since it proved to be effective also in contexts in which the underlying independence assumptions do not hold [5], even outperforming more current approaches [1]. However, the mutual conditional independence assumption behind naïve Bayesian networks can be quite strong: therefore, we also propose employing *tree-augmented naïve* (TAN) Bayesian networks, which also allow a tree structure to exist between feature variables [6]. 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 encoded by the BN. 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* [16] (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 [2] 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, [16] 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. Due to the low complexity of inferencing and learning in (tree-augmented) naïve Bayesian networks, we choose to employ such structures to represent dependency relations between variables in our probabilistic-logic model; also, we attempt to employ RBE to explicitly encode the uncertainty about parameters introduced by the adoption of the OWA, and empirically evaluate different approaches to handling missing attributes.

3 Terminological Bayesian Classifiers

We introduce a formalism, named *terminological Bayesian classifier* (TBC), consisting of a BN defined over a set of variables, each mapped to a (possibly complex) Description Logic (DL) concept defined over a DL knowledge base (KB). Each of such DL concepts can be considered as a feature (so we will refer to them as *feature concepts*) so that, given a generic individual a defined over a DL KB $\mathcal{K}$, inferring the membership relation to such concepts allows us, by means of a TBC defined over $\mathcal{K}$, to infer the

membership probability to a given target concept in $\mathcal{K}$ if it was previously unknown. This means that, within the TBC, each input individual is described by its concept-membership relation with respect to the feature concepts contained in it. Given a generic individual a in $\mathcal{K}$, a variable assigned to a DL feature concept F in a TBC defined over $\mathcal{K}$ takes value $True$ if $\mathcal{K} \models D(a)$, $False$ if $\mathcal{K} \models \neg D(a)$ and the variable is considered not observable otherwise. A more formal definition of TBC can be given 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 $\langle \mathcal{G}, \Theta_{\mathcal{G}} \rangle$, representing respectively the structure and parameters of a BN, in which:*

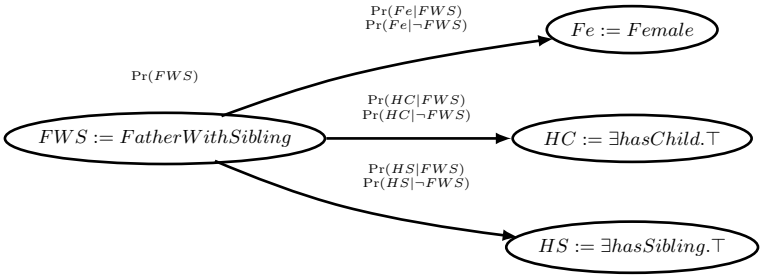
- *$\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ is an augmented directed acyclic graph, in which:*
  - *$\mathcal{V} = \{F_1, \ldots, F_n, C\}$ (vertices) is a set of random Boolean variables, each linked to a DL concept defined over $\mathcal{K}$. Each F_i ($i = 1, \ldots, n$) is associated to a feature concept, and C to the target (class) concept (we will use the names of variables in $\mathcal{V}$ to represent the corresponding DL concept for brevity);*
  - *$\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges, which model the (in)dependence relations among the variables in $\mathcal{V}$.*
- *$\Theta_{\mathcal{G}}$ is a set of* conditional probability distributions *(CPD), one for each variable $V \in \mathcal{V}$, representing the conditional probability distribution of the feature concept given the state of its parents in the graph.*

Given a generic individual a in $\mathcal{K}$, each variable $F_i \in \mathcal{V}$ in the TBC has value $True$ (resp. $False$) if $\mathcal{K} \models F_i(a)$ (resp. $\mathcal{K} \models \neg F_i(a)$); otherwise (i.e. when $\mathcal{K} \not\models F_i(a)$ and $\mathcal{K} \not\models \neg F_i(a)$) its value is considered as not observable (or missing*). If the concept-membership relation between a and the target concept C cannot be inferred from $\mathcal{K}$, the probability of such concept-membership can be estimated by calculating the conditional posterior probability using regular BN inference algorithms $\Pr(C \mid F_1, \ldots, F_n)$ (such as Variable Elimination).*

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 variable is independent on other feature variables, given the value of the target variable. TAN networks relax such independence assumptions by allowing a tree structure among feature variables: in terminological TAN Bayesian classifiers, $\mathcal{E} = \{\langle C, F_i \rangle \mid i \in \{1, \ldots, n\}\} \cup E_T$, where $E_T = \{\langle F_i, F_j \rangle \mid i, j \in \{1, \ldots, n\}, i \neq j\}$ is a set of directed edges defining a directed tree structure.

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

[2] Here DL concepts have been aliased for brevity.



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 concepts Fe and HS is not known, i.e. $\mathcal{K} \not\models Fe(a)$ and $\mathcal{K} \not\models \neg Fe(a)$. It is possible to infer, through the given network, the probability that the individual a is a member of the target concept FWS:

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

In the following we define the problem of learning a terminological Bayesian classifier $\mathcal{N}_\mathcal{K}$, given a DL KB $\mathcal{K}$ and a set of positive, negative and neutral training individuals $Ind_C(\mathcal{K}) = Ind_C^+(\mathcal{K}) \cup Ind_C^-(\mathcal{K}) \cup Ind_C^0(\mathcal{K})$.

Definition 2. *(Terminological Bayesian Classifier Learning Problem) The TBC learning problem consists in finding a TBC $\mathcal{N}_\mathcal{K}^*$ maximizing a TBC scoring function with respect of the training individuals $Ind_C(\mathcal{K})$ organised in positive, negative and neutral examples, given their concept-membership to the target concept C in $\mathcal{K}$. Formally:*

Given *the following:*
- *a* target concept C*;*
- *a set of training individuals $Ind_C(\mathcal{K})$ in a DL KB $\mathcal{K}$ such that:*
  - $\forall a \in Ind_C^+(\mathcal{K})$ *positive example:* $\mathcal{K} \models C(a)$*,*
  - $\forall a \in Ind_C^-(\mathcal{K})$ *negative example:* $\mathcal{K} \models \neg C(a)$*,*
  - $\forall a \in Ind_C^0(\mathcal{K})$ *neutral example:* $\mathcal{K} \not\models C(a) \wedge \mathcal{K} \not\models \neg C(a)$*;*
- *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 $Ind_C(\mathcal{K})$;*

Find *a network $\mathcal{N}_\mathcal{K}^*$ maximizing a given scoring function $Score$ wrt the samples:*

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

The search space to find the optimal network $\mathcal{N}_\mathcal{K}^*$ may be too large to explore exhaustively. For this reason the learning approach proposed here works by incrementally building the set of feature concepts, with the aim of obtaining a set of concepts maximizing the score of the induced network; each feature concepts is individually searched by an inner search process, guided by the scoring function itself, and the whole strategy of adding and removing feature concepts follows a forward selection/backward elimination strategy. This approach is motivated by the literature about *selective Bayesian classifiers* [12], where forward selection of attributes generally increases the classifier's

accuracy. The algorithm proposed here is organised in two nested loops: the inner loop is concerned with finding the best feature DL concept addition/removal operation, while the outer loop implements the abstract greedy feature selection strategy; both are guided by the network scoring function. In the inner loop, outlined in Alg. 1, the search through

Algorithm 1 Scoring function-driven beam search for a new concept to add to the terminological Bayesian network.

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

$BestNetwork \leftarrow \emptyset; Beam \leftarrow \{Start\}; \mathcal{N}_\mathcal{K} = \langle \mathcal{G}, \Theta_\mathcal{G} \rangle, \mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle;$
repeat
  $Candidates \leftarrow \emptyset;$
  for $c \in Beam$ **do**
    for $c' \in \{c' \in \rho^{cl}_{\downarrow}(c) \mid |c'| \leq min(|c| + depth, maxLength)\}$ **do**
      $\mathcal{V}' \leftarrow \mathcal{V} \cup \{c'\};$
      {$BuildOptimalNetwork$ finds the optimal Bayesian network structure and parameters for the set of concepts $\mathcal{V}'$ wrt a given scoring criterion, eventually under a set of constraints on the network structure and assumptions on the missingness pattern}
      $\mathcal{N}'_\mathcal{K} \leftarrow BuildOptimalNetwork(\mathcal{V}', Ind_C(\mathcal{K}));$
      $Candidates \leftarrow Candidates \cup \{\mathcal{N}'_\mathcal{K}, \};$
    end for
  end for
  $Beam \leftarrow NextBeam(Candidates, Score, width);$
  $BestNetwork \leftarrow \underset{\mathcal{N}^*_\mathcal{K} \in Beam \cup \{Best\}}{\arg\max} Score(\mathcal{N}^*_\mathcal{K}, Ind_C(\mathcal{K}));$
until stopping criterion on $BestNetwork$, $Beam$;
return $BestNetwork$;

the space of concept definitions is performed through a beam search, using the $\rho^{cl}_{\downarrow}$ refinement operator [14] ($\rho^{cl}_{\downarrow}(C)$ returns a set of refinements D of C so that $D \sqsubset C$, which we consider only up to a given concept length n). 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 (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. In the outer loop, a variety of feature selection strategies can be implemented [9]. In this particular case, a *Forward Selection Backward Elimination* approach is proposed, at each iteration considering to add a new concept to the network or removing at most a variable number of concepts. We experimented with two variants of such approach, both implemented through Alg. 2: *Forward Selection* (FS) which adds a single concept to the network at each iteration and *Fast Forward Selection Backward Elimination* (FFSBE) which at each iteration adds or removes one concept from the network. In the algorithm, such feature selection methods correspond to different values of the max parameter in Alg. 2 (representing the maximum number of concepts that can be removed from the network), i.e. 0 and 1 respectively.

Algorithm 2 Forward Selection Backward Elimination approach for the incremental construction of terminological Bayesian classifiers.

function $FSBE(\mathcal{K}, Ind_C(\mathcal{K}), max)$

1: $\mathcal{N}_{\mathcal{K}}^0 = \langle \mathcal{G}^0, \Theta_{\mathcal{G}^0} \rangle, \mathcal{G}^0 = \langle \mathcal{V}^0 \leftarrow \{C\}, \mathcal{E}^0 \leftarrow \emptyset \rangle; t \leftarrow 0;$
2: **repeat**
3: $\quad t \leftarrow t + 1;$
4: $\quad$ {A new network is selected among a set of possible candidates, obtained by either adding of removing a set of concepts to the structure, so to maximize the scoring criterion $Score$}
5: $\quad Candidates = \{Extend(\mathcal{N}_{\mathcal{K}}^{t-1}, Ind_C(\mathcal{K})), Remove(\mathcal{N}_{\mathcal{K}}^{t-1}, Ind_C(\mathcal{K}), max)\};$
6: $\quad \mathcal{N}_{\mathcal{K}}^t \leftarrow \arg\max_{\mathcal{N}_{\mathcal{K}}^* \in Candidates} Score(\mathcal{N}_{\mathcal{K}}^*, Ind_C(\mathcal{K}));$
7: **until** $Score(\mathcal{N}_{\mathcal{K}}^t, Ind_C(\mathcal{K})) \leq Score(\mathcal{N}_{\mathcal{K}}^{t-1}, Ind_C(\mathcal{K}));$
8: **return** $\mathcal{N}_{\mathcal{K}}^{t-1};$

function $Remove(\mathcal{N}_{\mathcal{K}}, Ind_C(\mathcal{K}), max)$

1: {Finds the best network that could be obtained by removing at most max feature concepts from the network structure, wrt a given scoring criterion $Score$}
2: $\mathcal{N}_{\mathcal{K}} = \langle \mathcal{G}, \Theta_{\mathcal{G}} \rangle, \mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle; BestNetwork \leftarrow \mathcal{N}_{\mathcal{K}};$
3: **for** $V' \subset V : |V| - |V'| \leq max$ **do**
4: $\quad \mathcal{N}'_{\mathcal{K}} \leftarrow BuildOptimalNetwork(\mathcal{V}', Ind_C(\mathcal{K}));$
5: $\quad$ **if** $Score(\mathcal{N}'_{\mathcal{K}}, Ind_C(\mathcal{K})) \geq Score(BestNetwork, Ind_C(\mathcal{K}))$ **then**
6: $\quad\quad BestNetwork \leftarrow \mathcal{N}'_{\mathcal{K}};$
7: $\quad$ **end if**
8: **end for**
9: **return** $BestNetwork;$

Different Assumptions on the Ignorance Model

However, during the learning process, it may happen that the concept membership between a training individual and some of the feature concepts may not be known. Depending on the reason of such missingness, Probabilistic Graphical Models offer a variety of approaches of handling this [11]. Formally, the missing data handling method depends on the probability distribution underlying the missingness pattern [21], which in turn can be classified on the basis of its behaviour with respect to the variable of interest.

- **Missing Completely At Random** (MCAR) – in this case, the variable of interest is independent from its observability, as any other variable in the probabilistic model. This is the precondition for case deletion to be valid, and missing data does not usually belong to such class [21].
- **Missing At Random** (MAR) – happens when the observability of the variable of interest depends on the value of some other variable in the probabilistic model.
- **Not Missing At Random/Informatively Missing** (NMAR, IM) – here, the actual value of the variable of interest influences the probability of its observability.

Example 2. (Different Ignorance Models in Terminological Bayesian Classifiers) Consider the network in Ex. 1: if the probability that the variable Fe is observable is independent on all other variables in the network, then it's missing completely at random; if it only depends, for example, on the value of FWS, then it's missing at random; if it

is dependent on the value Fe would have if it was not missing, then it is informatively missing.

Each of the aforementioned assumptions on the missingness pattern implies a different way of learning both network structure and parameters in presence of partially observed data. If **MCAR** holds, *Available Case Analysis* [11] can be used, where maximum likelihood network parameters are estimated using only available knowledge (i.e. ignoring missing data); we are adopting the heuristic used in [8] of setting network's parameters to their maximum likelihood value, which is both accurate and efficient. As scoring function, similarly to [8], we adopt the conditional log-likelihood on positive and negative training individuals, defined as [3]:

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

A problem with using simply $\mathcal{CLL}$ as scoring criterion is that it tends to favour complex structures [11] that overfit the training data. To avoid overfitting, we penalize the conditional log-likelihood through the *Bayesian Information Criterion* (BIC) [11], where the penalty is proportional to the number of independent parameters in a network (according to the minimum description length principle) and is defined as follows:

$$BIC(\mathcal{N}_{\mathcal{K}} \mid Ind_C(\mathcal{K})) = \mathcal{CLL}(\mathcal{N}_{\mathcal{K}} \mid Ind_C(\mathcal{K})) - \frac{\log N}{2} |\Theta_{\mathcal{G}}|; \tag{1}$$

where N is the number of data points and $|\Theta_{\mathcal{G}}|$ is the number of independent parameters in the network. 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, which is the target concept node). Without constraining the space of possible network structures, finding a structure which is optimal under some criterion may require an exhaustive search in the space of possible structures. However, in the case of TAN networks, if the scoring function is decomposable, there is an efficient method of finding a globally optimal network structure [6]. In this work, we create a complete weighted digraph among feature variables, each directed edge weighted with the BIC score (defined using the model's log-likelihood) gain that adding that edge would provide to the network, and then find the maximum weighted spanning tree structure using Chu-Liu-Edmonds algorithm (which has a $O(V^2)$ time complexity on dense digraphs, where V is the number of nodes). When learning network parameters from **MAR** data, a variety of techniques is available, such as Expectation-Maximization (EM), MCMC sampling or gradient ascent [11]. In this work, EM is used as outlined in Alg. 3: it first initialises network parameters using estimates that ignore missing data; Then, it considers 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 concept 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

[3] When used to score networks, conditional log-likelihoods are calculated ignoring available knowledge about the membership between training individuals and the target concept.

until convergence (e.g. when the improvement in log-likelihood is lower than a specific threshold). At each iteration, the EM algorithm applies the following two steps:

Algorithm 3 Outline for our implementation of the EM algorithm for parameter learning from MAR data in a terminological Bayesian classifier.

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

1: $\{\mathcal{N}^0_{\mathcal{K}}$ was initialized with arbitrary heuristic parameters $\Theta^0_{\mathcal{G}}\}$
2: $\mathcal{N}^0_{\mathcal{K}} = \langle \mathcal{G}, \Theta^0_{\mathcal{G}} \rangle, \mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle; t \leftarrow 0;$
3: **repeat**
4: $\quad \{\bar{n}(x_i, \pi_{x_i})\} \leftarrow ExpCounts(\mathcal{N}_{\mathcal{K}}, Ind_C(\mathcal{K}));$
5: $\quad \{$Network parameters $\Theta^{t+1}_{\mathcal{G}}$ are updated according to the inferred expected counts$\}$
6: $\quad$ **for** $X_i \in \mathcal{V}, \langle x_i, \pi_{x_i} \rangle \in vals(X_i, parents(X_i))$ **do**
7: $\quad\quad \theta^{t+1}_{\mathcal{G}}(x_i, \pi_{x_i}) \leftarrow \frac{\bar{n}(x_i, \pi_{x_i})}{\sum_{x'_i \in vals(X_i)} \bar{n}(x'_i, \pi_{x_i})};$
8: $\quad$ **end for**
9: $\quad t \leftarrow t + 1;$
10: $\quad \mathcal{N}^t_{\mathcal{K}} = \langle \mathcal{G}, \Theta^t_{\mathcal{G}} \rangle;$
11: $\quad \{$The iterative process stops when improvements in log-likelihood are $\leq$ a threshold$\}$
12: **until** $\mathcal{L}(\mathcal{N}^t_{\mathcal{K}} \mid Ind_C(\mathcal{K})) - \mathcal{L}(\mathcal{N}^{t-1}_{\mathcal{K}} \mid Ind_C(\mathcal{K})) \leq \tau;$
13: **return** $\mathcal{N}^t_{\mathcal{K}};$

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

1: $\mathcal{N}_{\mathcal{K}} = \langle \mathcal{G}, \Theta_{\mathcal{G}} \rangle, \mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle;$
2: **for** $X_i \in \mathcal{V}, \langle x_i, \pi_{x_i} \rangle \in vals(X_i, parents(X_i))$ **do**
3: $\quad \bar{n}(x_i, \pi_{x_i}) \leftarrow 0;$
4: **end for**
5: $\{\bar{n}(x_i, \pi_{x_i})$ will contain the expected counts for $(X_i = x_i, parents(X_i) = \pi_{x_i})\}$
6: **for** $a \in Ind_C(\mathcal{K})$ **do**
7: $\quad$ **for** $X_i \in \mathcal{V}, \langle x_i, \pi_{x_i} \rangle \in vals(X_i, parents(X_i))$ **do**
8: $\quad\quad \bar{n}(x_i, \pi_{x_i}) \leftarrow \bar{n}(x_i, \pi_{x_i}) + \Pr(x_i, \pi_{x_i} \mid \mathcal{N}_{\mathcal{K}});$
9: $\quad$ **end for**
10: **end for**
11: **return** $\{\bar{n}(x_i, \pi_{x_i})\};$

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

For structure learning in TAN Bayesian networks from MAR data, this works used the Structural EM (SEM) algorithm [11]. In SEM, outlined in Alg. 4, 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 (maximizing the expected score of the network) 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 wrt a scoring function, then the SEM algorithm will monotonically improve

Algorithm 4 Outline for our implementation of the Structural EM algorithm for structure and parameter learning from MAR data in a terminological Bayesian classifier.

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

1: $t \leftarrow 0$;
2: $\mathcal{N}_{\mathcal{K}}^0 \leftarrow EM(\mathcal{N}_{\mathcal{K}}^0, Ind_C(\mathcal{K}))$;
3: **repeat**
4: $t \leftarrow t + 1$;
5: {The $FindStructure$ function returns a new network maximizing the *expected* network score (wrt the actual network) with its parameter initialised to an initial heuristic value}
6: $\mathcal{N}_{\mathcal{K}}^t \leftarrow FindStructure(Ind_C(\mathcal{K}) \mid \mathcal{N}_{\mathcal{K}}^{t-1})$
7: $\mathcal{N}_{\mathcal{K}}^t \leftarrow EM(\mathcal{N}_{\mathcal{K}}^t, Ind_C(\mathcal{K}))$;
8: {The iterative process stops when $\mathcal{N}_{\mathcal{K}}^t$ does not show improvements in score over $\mathcal{N}_{\mathcal{K}}^{t-1}$}
9: **until** $Score(\mathcal{N}_{\mathcal{K}}^t, Ind_C(\mathcal{K})) \leq Score(\mathcal{N}_{\mathcal{K}}^{t-1}, Ind_C(\mathcal{K}))$;
10: **return** $\mathcal{N}_{\mathcal{K}}^{t-1}$;

such score. At each iteration of the SEM algorithm, we find the very same approach we used with MCAR data, except that we employ the *expected* value of the BIC score [11] on training individuals.

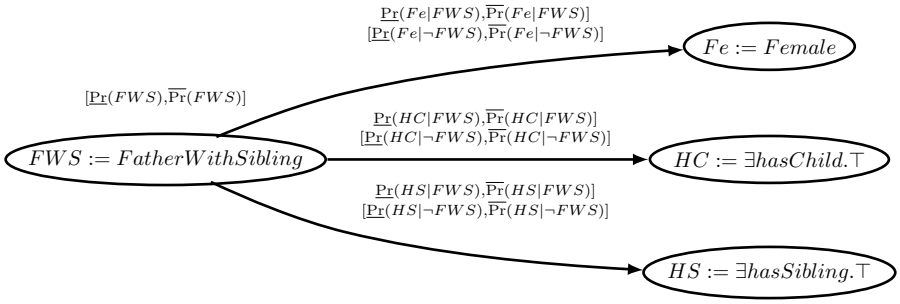
When data is **NMAR/IM** it may be harder to model, since we cannot assume that observed and missing values follow the same distributions. However, it is generally possible to extend the probabilistic model to produce one where the MAR assumption holds; if the value of a variable associated to the feature concept F_i is informatively missing, we can consider its observability as a indicator Boolean variable O_i (such that $O_i = False$ iff $\mathcal{K} \not\models F_i(a)$ and $\mathcal{K} \not\models \neg F_i(a)$, $O_i = True$ otherwise) and include it in our probabilistic model, so that F_i's ignorance model satisfies the MAR assumption (since the probability of F_i to be observable depends on the always observable indicator variable O_i). Doing this may however raise some problems, since the induced probabilistic model will be dependent on the specific ignorance model in the training set, and changes in such missingness pattern may impact on the model's effectiveness.

An alternate solution proposed in literature is *Robust Bayesian Estimation* [16] (RBE), which allows to learn interval-valued conditional probability distributions which explicitly represent the uncertainty about network parameters. 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. Such interval-valued and posterior intervals [4] can be calculated in closed form, as described in [16]. To score each induced network, we empirically choose to calculate posterior intervals, get their central point and then use them as probability values to calculate e.g. the BIC score as in Eq. 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 3. (Example of Terminological Naïve Bayesian Classifier using RBE) Consider again the terminological naïve Bayesian classifier in Example 1: when learning in

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

presence of NMAR data, 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.



Interval-valued network parameters can be calculated efficiently [16]. E.g. the parameters associated to the feature concept HC can be calculated as follows:

$$\overline{n}(HC|FWS)=n(?|FWS)+n(HC|?)+n(?|?); \quad \underline{n}(HC|FWS)=n(?|FWS)+n(\neg HC|?)+n(?|?);$$
$$\overline{\Pr}(HC|FWS)=\frac{n(HC|Fa)+\overline{n}(HC|FWS)}{n(Fa)+\overline{n}(HC|FWS)}; \quad \underline{\Pr}(HC|FWS)=\frac{n(HC|FWS)}{n(FWS)+\underline{n}(HC|FWS)};$$

where $n(? \mid FWS) = |\{a \in Ind^{+}_{FWS}(\mathcal{K}) \mid \mathcal{K} \not\models HC(a) \text{ and } \mathcal{K} \not\models \neg HC(a)\}|$, $n(HC\ |?) = |\{a \in Ind^{0}_{FWS}(\mathcal{K}) \mid \mathcal{K} \models HC(a)\}|$ and $n(?\ |?) = |\{a \in Ind^{0}_{FWS}(\mathcal{K}) \mid \mathcal{K} \not\models HC(a) \wedge \mathcal{K} \not\models \neg HC(a)\}|$. 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 FWS belongs to the posterior probability interval $[\underline{\Pr}(FWS \mid HC), \overline{\Pr}(FWS \mid HC)]$, where:

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

4 Experiments

In this section we empirically evaluate the impact of adopting different missing knowledge handling methods and search strategies, during the process of learning (naïve and TAN) TBCs from real world ontologies. Starting from a set of real ontologies [5] (outlined in Table 1), we generated a set of 20 random query concepts for each ontology [6], 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

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

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

Ontology	DL Expressivity	#Axioms	#Individuals	#Classes	#ObjectProperties
MDM0.73	$\mathcal{ALCHOF}(\mathcal{D})$	1098	112	196	22
LEO	$\mathcal{ALCHIF}(\mathcal{D})$	430	61	32	26
FAMILY-TREE	$\mathcal{SROIF}(\mathcal{D})$	2059	368	22	52
WINE	$\mathcal{SHOIN}(\mathcal{D})$	1046	218	142	21
BIOPAX (PROTEOMICS)	$\mathcal{ALCHN}(\mathcal{D})$	773	49	55	47

Table 1. Ontologies considered in the experiments.

same order of magnitude. A DL reasoner [7] was employed to decide on the theoretical concept-membership of individuals wrt the query concepts. In experiments, we relearned such concept queries as (naïve and TAN) TBCs, using individuals retrieved by each query (resp. its complement) as positive (resp. negative) examples. The evaluated missing knowledge handling methods were Robust Bayesian Estimation (ROBUST) and, for naïve and TAN networks respectively: Available Case Analysis (ACA and TACA), the (structural) EM algorithm (EM and SEM), and two additional approaches aiming at including a features' observability in the resulting model (IM^3 and IM^2 for naïve and TIM^3 and TIM^2 for TAN structures). The last two approaches build networks which are dependant on the ignorance model: IM^3 and TIM^3(where IM here stands for *Informatively Missing*) makes use of three-valued feature variables taking a value in $\{True, False, Unknown\}$ when the membership to the associated feature concept is respectively true, false or not known; while IM^2 and TIM^2 employ two-valued feature variables, taking a value in $\{True, Other\}$, when the membership to the associated feature concept is respectively true or either false or not known. During experiments,

FS	ACA	EM	IM^3	IM^2	TACA	SEM	TIM^3	TIM^2	ROBUST
LEO	$.96 \pm .14$	$.96 \pm .14$	$.91 \pm .2$	$.96 \pm .14$	$.93 \pm .18$	$.93 \pm .18$	$.89 \pm .21$	$.93 \pm .18$	$.9 \pm .2$
MDM	$.88 \pm .25$	$.88 \pm .25$	$.85 \pm .27$	$.92 \pm .2$	$.85 \pm .26$	$.85 \pm .26$	$.73 \pm .32$	$.86 \pm .26$	$.62 \pm .39$
WINE	$.89 \pm .21$	$.89 \pm .21$	$.87 \pm .23$	$.92 \pm .18$	$.89 \pm .22$	$.89 \pm .22$	$.83 \pm .25$	$.86 \pm .24$	$.78 \pm .3$
F-T	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0
PROTEOMICS	$.95 \pm .13$	$.95 \pm .13$	$.86 \pm .22$	$.96 \pm .11$	$.81 \pm .25$	$.81 \pm .25$	$.77 \pm .26$	$.8 \pm .26$	$.75 \pm .26$
FFSBE	ACA	EM	IM^3	IM^2	TACA	SEM	TIM^3	TIM^2	ROBUST
LEO	$.96 \pm .14$	$.96 \pm .14$	$.91 \pm .2$	$.96 \pm .14$	$.93 \pm .18$	$.93 \pm .18$	$.89 \pm .21$	$.93 \pm .18$	$.9 \pm .2$
MDM	$.88 \pm .25$	$.88 \pm .25$	$.86 \pm .27$	$.92 \pm .21$	$.85 \pm .26$	$.85 \pm .26$	$.73 \pm .32$	$.86 \pm .26$	$.63 \pm .39$
WINE	$.9 \pm .21$	$.9 \pm .21$	$.87 \pm .23$	$.92 \pm .19$	$.89 \pm .22$	$.89 \pm .22$	$.83 \pm .25$	$.86 \pm .24$	$.78 \pm .3$
F-T	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0
PROTEOMICS	$.95 \pm .13$	$.95 \pm .13$	$.86 \pm .22$	$.96 \pm .11$	$.81 \pm .25$	$.81 \pm .25$	$.77 \pm .26$	$.8 \pm .26$	$.76 \pm .26$

Table 2. Statistics for cross-validated AUC-PR 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 then used to evaluate the methods using k-fold cross validation (with $k = 10$) through the AUC-PR metric.

refinements were only allowed to contain conjunctions/disjunctions of concepts, com-

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

plements and existential restrictions, and refinements started from concept $\top$. To avoid overfitting, the greedy network construction was driven by the BIC score in Eq. 1. In experiments, each of the 20 generated query concepts, 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 such pairs of positive/negative examples, k-fold cross validation (with $k = 10$) was used to estimate k Area Under the Precision-Recall Curve [3] (AUC-PR) values (for ROBUST we used the midpoint of each posterior interval was used to associate a probability to concept-memberships), using inferred concept-membership probability to rank testing individuals. Results are summarised in Table 2. Parameters $depth$ and $maxLength$ (indicating resp. 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 almost every case, forcing the existence of a maximum (penalized) likelihood tree structure to exist between feature concepts did not benefit the ranking capability: for example, in the IM^3/TIM^3 and IM^2/TIM^2 cases, naïve Bayesian networks had significantly greater AUC-PR values than TAN counterparts (with $p < 0.01$ under a Student's paired t-test); a reasons for this is that BIC-driven search, because of the higher cost of adding feature concepts (depending on the higher number of network parameters), prevents the introduction of discriminative feature concepts in the network to keep its structure simple. This is also the reason that caused, in all feature selection strategies, IM^2/TIM^2 to have higher AUC-PR scores than their IM^3/TIM^3 counterparts (with $p < 0.01$). Comparing the methods to learn the parameters of naïve Bayesian networks on different assumptions on the missingness pattern, it emerged that IM^2 had greater AUC-PR results with all the experimented feature selection approaches (with $p < 0.01$), suggesting that the missingness of concept-membership information was informative (except in the case of the LEO ontology, where other approaches to dealing with missing informations had similar results). Also, using the midpoint of Robust Bayesian Estimators' posterior intervals led to worse results when ranking target concept-membership probabilities. However, it can still be used to explicitly represent the uncertainty on parameters caused by missing knowledge within the Semantic Web.

5 Conclusions and Future Work

This paper proposes a method, terminological Bayesian classifiers, to efficiently estimate the probability that a generic individual belongs to a specific target concept, given its concept-membership relation to a set of DL feature concepts; this work focused on network structures which allow for efficient inference and learning, and empirically evaluated different methods to handle missing data resulting from the adoption of the OWA. In the future, we aim at exploring other network structure which allow for efficient inference and learning, at extending this framework towards role-membership prediction and to evaluate it more extensively on real world ontologies.

References

[1] Caruana, R., Niculescu-Mizil, A.: An empirical comparison of supervised learning algorithms. In: Proceedings of the 23rd international conference on Machine learning. pp. 161–

168. ICML '06, ACM, New York, NY, USA (2006)
[2] Corani, G., Zaffalon, M.: Learning reliable classifiers from small or incomplete data sets: The naive credal classifier 2. Journal of Machine Learning Research 9, 581–621 (2008)
[3] Davis, J., Goadrich, M.: The relationship between precision-recall and roc curves. In: ICML 2006. pp. 233–240. ACM, New York, NY, USA (2006)
[4] Domingos, P., Lowd, D., Kok, S., Poon, H., Richardson, M., Singla, P.: Uncertainty reasoning for the semantic web i. pp. 1–25. Springer (2008)
[5] 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)
[6] Friedman, N., Geiger, D., Goldszmidt, M., Provan, G., Langley, P., Smyth, P.: Bayesian network classifiers. In: Machine Learning. pp. 131–163 (1997)
[7] Getoor, L., Taskar, B.: Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning). The MIT Press (2007)
[8] Grossman, D., Domingos, P.: Learning bayesian network classifiers by maximizing conditional likelihood. In: Brodley, C.E. (ed.) ICML. ACM International Conference Proceeding Series, vol. 69. ACM (2004)
[9] Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L. (eds.): Feature Extraction, Foundations and Applications. Springer (2006)
[10] Hitzler, P., van Harmelen, F.: A reasonable semantic web. Semantic Web 1(1-2), 39–44 (2010)
[11] Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press (2009)
[12] Langley, P., Sage, S.: Induction of selective bayesian classifiers. In: de Mántaras, R.L., Poole, D. (eds.) UAI. pp. 399–406. Morgan Kaufmann (1994)
[13] Laskey, K.J., Laskey, K.B.: Uncertainty reasoning for the world wide web: Report on the urw3-xg incubator group. In: URSW2008
[14] Lehmann, J., et al.: Concept learning in description logics using refinement operators. Mach. Learn. 78, 203–250
[15] Luna, J.E.O., Cozman, F.G.: An algorithm for learning with probabilistic description logics. In: Bobillo, F., da Costa, P.C.G., d'Amato, C., Fanizzi, N., Laskey, K.B., Laskey, K.J., Lukasiewicz, T., Martin, T., Nickles, M., Pool, M., Smrz, P. (eds.) URSW. pp. 63–74 (2009)
[16] Ramoni, M., Sebastiani, P.: Robust learning with missing data. Mach. Learn. 45, 147–170 (October 2001)
[17] 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)
[18] Rettinger, A., Nickles, M., Tresp, V.: Statistical relational learning with formal ontologies. In: Buntine, W.L., Grobelnik, M., Mladenic, D., Shawe-Taylor, J. (eds.) ECML/PKDD (2). LNCS, vol. 5782, pp. 286–301. Springer (2009)
[19] Richardson, M., Domingos, P.: Markov logic networks. Mach. Learn. 62, 107–136 (February 2006)
[20] Rodrigues De Morais, S., Aussem, A.: Exploiting data missingness in bayesian network modeling. In: Proceedings of the 8th International Symposium on Intelligent Data Analysis: Advances in Intelligent Data Analysis VIII. pp. 35–46. IDA '09, Springer (2009)
[21] Rubin, D.B.: Inference and missing data. Biometrika 63(3), 581–592 (1976)
[22] Xu, Z., Tresp, V., Yu, K., Kriegel, H.P.: Infinite hidden relational models. In: Proceedings of the 22nd International Conference on Uncertainity in Artificial Intelligence (UAI 2006)
[23] Zaffalon, M., Corani, G., Mauá, D.: Utility-based accuracy measures to empirically evaluate credal classifiers. In: ISIPTA 2011. pp. 401–410. Innsbruck