Learning Probabilistic Description Logic Concepts Under Alternative Assumptions on Incompleteness

Pasquale Minervini, Claudia d'Amato, Nicola Fanizzi, and Floriana Esposito

LACAM Laboratory – Dipartimento di Informatica
Università degli Studi di Bari Aldo Moro – via E. Orabona, 4 - 70125 Bari - Italia
firstName.lastName@uniba.it

Abstract. Real-world knowledge often involves various degrees of uncertainty. For such a reason, in the Semantic Web context, difficulties arise when modeling real-world domains using only purely logical formalisms. Alternative approaches almost always assume the availability of probabilistically-enriched knowledge, while this is hardly known in advance. In addition, purely deductive exact inference may be infeasible for Web-scale ontological knowledge bases, and does not exploit statistical regularities in data. Approximate deductive and inductive inferences were proposed to alleviate such problems. This article proposes casting the concept-membership prediction problem (predicting whether an individual in a Description Logic knowledge base is a member of a concept) as estimating a conditional probability distribution which models the posterior probability of the aforementioned individual's concept-membership given the knowledge that can be entailed from the knowledge base regarding the individual. Specifically, we model such posterior probability distribution as a generative, discriminatively structured, Bayesian network, using the individual's concept-membership w.r.t. a set of *feature* concepts standing for the available knowledge about such individual.

1 Introduction

Real-world knowledge often involves various degrees of uncertainty. For such a reason, in the context of Semantic Web (SW), difficulties arise when trying to model real-world domains using purely logical formalisms. For this purpose, the Uncertainty Reasoning for the World Wide Web Incubator Group [1] (URW3-XG) identified the requirements for representing and reasoning with uncertain knowledge in the SW context, and provided a number of use cases showing the clear need for explicitly representing and reasoning in presence of uncertainty [23]. As a consequence, several approaches, particularly focusing on enriching knowledge bases and inference procedures with probabilistic information has been proposed. Some approaches extend knowledge representation formalisms actually used in

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

the SW (such as [7]), while others rely on probabilistic enrichment of Description Logics [1] (DLs) or logic programming formalisms (such as [28]).

Uncertainty is pervasive in real-world knowledge, but it is often hard to elicit it on both the logical and the probabilistic side. Machine Learning (ML) methods have been proposed to overcome several potential limitations of purely deductive reasoning and ontology engineering [9, 19, 34]. These limitations are inherent to i) the difficulty of engineering knowledge bases in expressive SW formalisms, ii) taking regularities in data into account, iii) performing approximate reasoning on Web-scale SW knowledge bases, and iv) reasoning in presence of incomplete knowledge (because of the Open-World Assumption), noise and uncertainty.

Various ML techniques have been extended to tackle SW representations. These encode regularities emerging from data as statistical models that can later be exploited to perform efficiently a series of useful tasks, bypassing the limitations of deductive reasoning and being able to cope with potential cases of inconsistency.

One of these tasks is the prediction of assertions, which is at the heart of further often more complex tasks such as query answering, clustering, ranking and recommendation. Data-driven forms of assertion prediction could be useful for addressing the cases where, for various reasons related to cases of incompleteness and inconsistency, it is not possible to logically infer the truth value of some statements (i.e. assertions which are not explicitly stated in nor derivable from the knowledge base). An example of such cases is the following:

Example 1. Consider a knowledge base $\mathcal{K}$ modeling familial relationships, where persons (each represented by an individual in the ontology) are characterized by multiple classes (such as `Father`, `Uncle`) and relationships (such as `hasChild`, `hasSibling`). By relying on purely deductive reasoning, it might not be possible to assess whether a certain property holds for a given person. For example, it might not be possible to assess whether John is an uncle or not. Assuming the property is represented by the concept `Uncle` and the person by the individual `john`, this can be formally expressed as:

$$\mathcal{K} \not\models \texttt{Uncle}(\texttt{john}) \wedge \mathcal{K} \not\models \neg\texttt{Uncle}(\texttt{john}),$$

i.e. it is not possible to deductively infer from $\mathcal{K}$ whether the property "uncle" holds for the person John.

Semantic Web knowledge representation languages make the *Open World Assumption*: a failure on deductively infer the truth value of a given fact does not imply that such fact is false, but rather that its truth value cannot be deductively inferred from the KB. This differs from the *Negation As Failure*, commonly used in databases and logic programs. Other issues are related to the distributed nature of the data across the Web: multiple, mutually conflicting pieces of knowledge may lead to contradictory answers or flawed inferences.

Most approaches to circumvent the limitations of incompleteness and inconsistency rely on extensions of the representation languages or of the inference

services (e.g. *ontology repairing* [37] and *epistemic* reasoning [13] and *paraconsistent* reasoning [29]).

An alternative solution consists in relying on data-driven approaches to address the problem of missing knowledge. The prediction of the truth value of an assertion can be cast as a *classification* problem, to be solved through *statistical learning* [41]: domain entities described by an ontology can be regarded as *statistical units*, and their properties can be statistically inferred even when they cannot be deduced from the KB. Several methods have been proposed in the SW literature (see [34] for a recent survey). In particular, Statistical Relational Learning [14] (SRL) methods face the problem of learning in domains showing both a complex relational and a rich probabilistic structure. A major issue with the methods proposed so far is that the induced statistical models (as those produced by kernel methods, tensor factorization, etc.) are either difficult to interpret by experts and to integrate in logic-based SW infrastructures, or computationally impractical (see Sect. 5.1).

Contribution

A learning task can be either *generative* or *discriminative* [24], depending on the structure of the target distribution. Generative models describe the joint probability of all random variables in the model (e.g. a joint probability distribution of two sets of variables $\Pr(\mathbf{X}, \mathbf{Y})$). Discriminative models directly represent aspects of the distribution that are important for a specific task (e.g. a conditional probability distribution of a set of variables given another $\Pr(\mathbf{Y} \mid \mathbf{X})$). The main motivation behind the choice of a discriminative model is described by the *main principle* in [41]: "If you possess a restricted amount of information for solving some problem, try to solve the problem directly and never solve a more general problem as an intermediate step. It is possible that the available information is sufficient for a direct solution but is insufficient for solving a more general intermediate problem". Discriminative learning can also be useful for feature selection (e.g. in the context of a mining or ontology engineering task). In [5], authors show that many feature selection methods grounding on information theory ultimately try to optimize some approximation of a conditional likelihood (that is, a quantity proportional to the true posterior class probabilities in a set of instances).

In this article, we propose a method for predicting the concept-membership relation of an arbitrary individual with respect to a given target concept given a set of training individuals (members and non-members) within a DL knowledge base. The proposed method relies on a Bayesian network with generative parameters (which can be computed efficiently) and discriminative structure (which maximize the predictive accuracy of the model). The proposed model can be used also with knowledge bases expressed in expressive DLs used within the SW context, such as $\mathcal{SHOIN}(\mathcal{D})$ and $\mathcal{SROIQ}(\mathcal{D})$.

In particular, the proposed method relies on a committee of features (represented by possibly complex concepts) to define a set of random variables

$\{F_1, \ldots, F_n\}$. Such variables are then used to model a posterior probability distribution of the target concept-membership, conditioned the membership w.r.t. the aforementioned feature concepts $\Pr(C \mid F_1, \ldots, F_n)$: the value of each F_i depends on the concept-membership w.r.t. the i-th feature concept, and C is a Boolean random variable whose conditional probability distribution depends on the value of the F_i's. The proposed method relies on an inductive process: it incrementally builds a Bayesian classifier through a set of hill-climbing searches in the space of feature concepts using DL refinement operators [26].

This paper is organized as follows: Sect. 2 contains an introduction to the Bayesian network formalism of describing independence relations among a set of variables. In Sect. 3 we describe Terminological Bayesian Classifier models for class-membership prediction in DL knowledge bases, and how such models can be learned from data. In Sect. 4 we provide an empirical evaluation of the discussed model. Finally, in Sect. 6 we summarize the proposed approach and discuss possible research directions.

2 Bayesian Networks and Bayesian Classifiers

Graphical models [20] (GMs) are a popular framework that allows a compact description of 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, i.e. *directed* and *undirected* graphical models, based directed and undirected graphs respectively.

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 vertexes.

In a BN, 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 vertexes that are not descendants of X_i given a joint state of its parents. More formally: $\forall X_i : X_i \perp\!\!\!\perp S \,|\, parents(X_i)$, where the function $parents(X_i)$ returns the parent vertexes of X_i in the DAG representing the BN.

The conditional independence assumption allows representing 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).

In BNs, common inference tasks (such as calculating the most likely value for some variables, their marginal distribution or their conditional distribution given some evidence) are NP-hard. However, such inference tasks are less complex for particular classes of BNs such tasks, approximate inference algorithms exist to efficiently infer in restricted classes of networks. For example, the *variable elimination* algorithm has linear complexity in the number of vertexes if the BN is a singly connected network [20].

Approximate inference methods for BNs also exist in literature such as *Monte Carlo* algorithms, *belief propagation* or *variational methods* [20]. 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 BN structures is super-exponential in the number of vertexes, making it generally impractical to perform an exhaustive search through the space of its possible models.

Looking for a trade-off between efficiency and expressiveness, we focus on *Bayesian network classifiers*, where a Bayesian network is used to model the conditional probability distribution of a single variable, representing a concept-membership relation.

For its simplicity, accuracy and low time complexity in both inference and learning, we first focused on a particular subclass of Bayesian network classifiers. *Naïve Bayesian classifier* models 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 category of models is especially interesting since it proved to be effective also in contexts in which the underlying independence assumptions do not hold [12], outperforming more recent approaches [6].

However, such strong independence assumptions may not capture correlations between feature concepts properly. Therefore, we also consider employing generic Bayesian network structures and polytree structures among feature variables, while retaining the edges from the class variable to feature variables. We avoid performing an exhaustive search in the space of possible structures (that, in the case of Bayesian classifiers, may be too complex to perform) and take the path also used in [17] and [33] of performing an hill climbing search, making modifications at the network structures at each step until we get to an (possibly local) optimal solution.

3 Terminological Bayesian Classifiers for Concept-membership Prediction

We propose employing the Bayesian network classifier [20] formalism to represent the statistical relations among a set of concepts in a given knowledge base. In particular, we aim at using such BN to model the conditional probability distribution $\Pr(C \mid F_1, \ldots, F_n)$, representing the probability that a generic individual in a knowledge base is a member of a target concept C given its concept-membership relation w.r.t. a set of *feature* concepts $\{F_1, \ldots, F_n\}$ (the random variables in the network can be considered as indicator functions taking different values depending on the concept-membership relation between the individual and the corresponding concept).

An intuitive method for mapping the values of the random variable to the corresponding concept-membership relation is considering the variable as a Boolean indicator function, assuming value `True` iff the individual is an instance of the concept, `False` iff it is an instance of its complement, and otherwise considering the variable as *non-observable*: this allows to consistently handle the Open World Assumption (OWA) characterizing the semantics of standard DLs, where it is common to have *partial knowledge* about the concept-membership relations of an individual.

However, this setting implies that not knowing the concept-membership relation w.r.t. a feature concept is *uninformative* [36] when predicting the concept-membership relation w.r.t. a given target concept; this is a strong assumption that does not hold in general. We will refer to such kind of networks as *Terminological Bayesian Classifiers* (TBCs). More formally:

Definition 1. *(Terminological Bayesian Classifier) A Terminological Bayesian Classifier (TBC) $\mathcal{N}_\mathcal{K}$, with respect to a knowledge base $\mathcal{K}$, is defined as a pair $\langle \mathcal{G}, \Theta_\mathcal{G} \rangle$, representing respectively the structure and parameters of a BN, in which:*

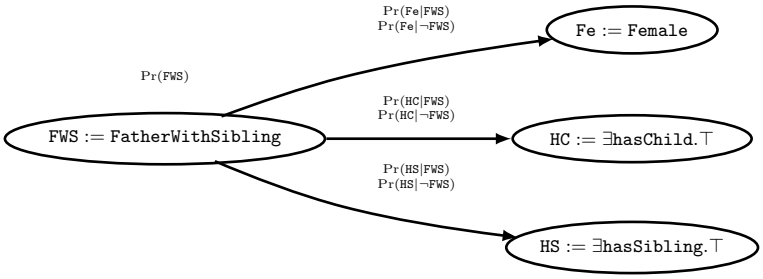
- *$\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ is an augmented directed acyclic graph, in which:*
  - *$\mathcal{V} = \{F_1, \ldots, F_n, C\}$ (vertexes) is a set of random variables, each linked to a concept defined over $\mathcal{K}$. Each F_i ($i = 1, \ldots, n$) is a Boolean random variable, whose value depends on the membership w.r.t. a feature concept, while C is a Boolean variable which indicates the membership relation to the target concept (we will use the names of variables in $\mathcal{V}$ to represent the corresponding 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.*

A very simple but effective structure is naïve Bayesian one (also described in section 2), which relies on the assumption the concept-membership w.r.t. each of the feature concepts are independent given the concept-membership relation w.r.t. the target concept; this results in the edge set $\mathcal{E} = \{\langle C, F_i \rangle \mid i \in \{1, \ldots, n\}\}$.

Example 2. (Example of Terminological Naïve Bayesian Classifier) Given the following set of feature concepts [2]:

$$\mathcal{F} = \{Fe := Female, HC := \exists hasChild.\top, HS := \exists hasSibling.\top\},$$

and a target concept $FWS := FatherWithSibling$, a terminological naïve Bayesian classifier expressing the target concept in terms of the feature concepts is the following:



We can also express correlations between feature concepts which may be useful for making the conditional probability distribution more accurate, by relaxing the constraints on the edge set $\mathcal{E}$; we consider allowing for generic (acyclic) graph structures among feature variables, and for polytree (or singly connected tree) graph structures, which allow for exact inference to be calculated in polynomial time [20].

Let $\mathcal{K}$ be a knowledge base and a a generic individual so that $\mathcal{K} \models HC(a)$, and the membership relation between a to the concepts Fe and HS is not known, i.e. $\mathcal{K} \not\models C(a)$ and $\mathcal{K} \not\models \neg C(a)$, where C is either Fe or HS. It is possible to infer, through the given network, the probability that the individual a is a member of the target concept FWS:

$$\Pr(FWS(a) \mid HC(a)) = \frac{\Pr(FWS)\Pr(HC \mid FWS)}{\Pr(HC)},$$

where $\Pr(HC) = \Pr(FWS)\Pr(HC \mid FWS) + \Pr(\neg FWS)\Pr(HC \mid \neg FWS)$. □

In the following, we define the problem of learning a TBC $\mathcal{N}_\mathcal{K}$, given a knowledge base $\mathcal{K}$ and a set of positive, negative and neutral training individuals.

The problem consists in finding a TBC $\mathcal{N}^*_\mathcal{K}$ maximizing an arbitrary scoring criterion, given a set of training individuals $Ind_C(\mathcal{K})$. Such individuals are organized in positive, negative and neutral examples, accordingly to their concept-membership relation w.r.t. the target concept C in $\mathcal{K}$.

More formally:

Definition 2. *(Terminological Bayesian Classifier Learning Problem)*
The TBC learning problem can be defined as follows:

[2] Here concepts have been aliased for brevity.

Given *:*

- *a* target concept C*;*
- *a set of training individuals* $Ind_C(\mathcal{K})$ *in a knowledge base* $\mathcal{K}$ *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 w.r.t. 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 for finding the optimal network $\mathcal{N}_\mathcal{K}^*$ may be too large to be exhaustively explored. For such a 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 concept 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* [21], where forward selection of attributes generally increases the classifier accuracy. The algorithm proposed here is organized in two nested loops: the inner loop is concerned with exploring the space of possible features (concepts), e.g. by means of DL refinement operators;e the outer loop implements the abstract greedy feature selection strategy (such as forward selection [18]). Both procedures are guided by a scoring function defined over the space of TBC models.

Algorithm 1 Scoring function-driven hill climbing search for a new concept to add to the committee of DL concepts used to construct the Terminological Bayesian Network.

function $Grow(\mathcal{F}, Ind_C(\mathcal{K}), Start)$

1: $C \leftarrow Start$;
2: {Iteratively refine the concept C until a stopping criterion is met}
3: **repeat**
4: {Let $\mathcal{C}$ be the set of (upward and downward) refinements of the concept C obtained by means of the ρ refinement operator:}
5: $\mathcal{C} \leftarrow \{C' \in \rho_\uparrow(C) \cup \rho_\downarrow(C) \mid |C'| \leq min(|C| + depth, maxLength)\}$;
6: {Select the concept in the set of refinements $\mathcal{C}$ providing the highest increase to the score (measured by the *Score* function) to the TBC obtained (using the $ConstructNetwork$ function) by adding the selected concept to the set $\mathcal{C}$}
7: $C \leftarrow \arg\max_{C' \in \mathcal{C}} Score(ConstructNetwork(\mathcal{F} \cup \{C'\}, Ind_C(\mathcal{K})), Ind_C(\mathcal{K}))$;
8: **until** Stopping criterion; {E.g. no further improvements in score}
9: **return** C

In the inner loop, outlined in Alg. 1, the search through the space of concept definitions is performed as a hill climbing search, using the $\rho^{cl}_{\downarrow}$ refinement operator [26] ($\rho^{cl}_{\downarrow}(C)$ 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 $\mathcal{F}'$ and finds an 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 2 Forward Selection Backward Elimination method for the incremental construction of terminological Bayesian classifiers.

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

1: $t \leftarrow 0, \mathcal{F}^t \leftarrow \emptyset$;
2: **repeat**
3: $t \leftarrow t + 1$;
4: {A new committee is selected among a set of possible candidates (represented by the set of committees $\hat{\mathcal{F}}$), obtained by either adding of removing a set of concepts to the structure, so as to maximize the score of the corresponding TBC (measured by means of the *Score* function)}
5: $\hat{\mathcal{F}} = \{Grow(\mathcal{F}^{t-1}, Ind_C(\mathcal{K}), \top), Shrink(\mathcal{F}^{t-1}, Ind_C(\mathcal{K}), max)\}$;
6: $\mathcal{F}^t \leftarrow \arg\max_{\mathcal{F} \in \hat{\mathcal{F}}} Score(ConstructNetwork(\mathcal{F}, Ind_C(\mathcal{K})), Ind_C(\mathcal{K}))$;
7: **until** Stopping criterion; {E.g. the maximum number of concepts in $\mathcal{F}$ was reached}
8: $\mathcal{N}^t_{\mathcal{K}} \leftarrow ConstructNetwork(\mathcal{F}^t, Ind_C(\mathcal{K}))$;
9: **return** $\mathcal{N}^t_{\mathcal{K}}$;

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

1: {Finds the best network that could be obtained by removing at most max feature concepts from the network structure, w.r.t. a given scoring criterion *Score*}
2: $\hat{\mathcal{F}} \leftarrow \{\mathcal{F}' \subseteq \mathcal{F} : |\mathcal{F}| - |\mathcal{F}'| \leq max\}$;
3: $\mathcal{F}^* \leftarrow \arg\max_{\mathcal{F} \in \hat{\mathcal{F}}} Score(ConstructNetwork(\mathcal{F}, Ind_C(\mathcal{K})), Ind_C(\mathcal{K}))$;
4: **return** $\mathcal{F}^*$;

In the outer loop, outlined in Alg. 2, it is possible to implement a variety of feature selection strategies [18]. In this specific case, we propose a *Forward Selection Backward Elimination* (FSBE) method, which at each iteration considers adding a new concept to the network (by means of the *Grow* function) or removing an existing one (by means of the *Shrink* function).

Different Assumptions on the Ignorance Model

During the learning process, it may happen that the concept membership between a training individual and some of the feature concepts is unknown. The reason of such missingness can be taken into account, when learning the parameters of the statistical model [20]. Formally, the missing data handling method

depends on the probability distribution underlying the missingness pattern [36], which in turn can be classified on the basis of its behavior with respect to the variable of interest:

- **Missing Completely At Random** (MCAR) – the variable of interest $\mathbf{X}$ is independent from its observability $O_{\mathbf{X}}$, 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 [36]:

$$P_{missing} \models (O_{\mathbf{X}} \perp\!\!\!\perp \mathbf{X});$$

- **Missing At Random** (MAR) – happens when the observability of the variable of interest $\mathbf{X}$ depends on the value of some other variable in the probabilistic model:

$$P_{missing} \models (O_{\mathbf{X}} \perp\!\!\!\perp \mathbf{x}^{\mathbf{y}}_{hidden} \,|\, \mathbf{x}^{\mathbf{y}}_{obs});$$

- **Not Missing At Random/Informatively Missing** (NMAR, IM) – here, the actual value of the variable of interest influences the probability of its observability:

$$P_{missing} \models (O_{\mathbf{X}} \not\perp\!\!\!\perp \mathbf{X}).$$

Example 3. (Different Ignorance Models in Terminological Bayesian Classifiers) Consider the network in Ex. 2: 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* [20] 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 [17] of setting network parameters to their maximum likelihood value, which is both accurate and efficient. This decision is further motivated by [33], which empirically motivated that generative discriminatively structured Bayesian networks retain both the accuracy of discriminative networks and the efficiency of parameter learning and ability to handle partial evidence typical of generative networks.

As scoring function, similarly to [17], we adopt the conditional log-likelihood on positive and negative training individuals, defined as [3]:

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

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

A problem with using simply *CLL* as scoring criterion is that it tends to favor complex structures [20] that overfit the training data. To avoid overfitting, we penalize the conditional log-likelihood through the *Bayesian Information Criterion* (BIC) [20], 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:

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

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).

For relaxing the independence assumptions in naïve Bayes structures, we follow the approach also discussed in [17, 33] to perform an hill-climbing search in the space of structures, by looking for the one maximizing the (penalized) CLL. The exploration in the space of possible structures is performed by making atomic modification to the structure between feature variables, and consist in atomic operation of either edge addition, removal or reversal.

When learning network parameters from **MAR** data, a variety of techniques is available, such as Expectation-Maximization (EM) or gradient ascent [20]. In this work, we employ the EM algorithm, as outlined in Alg. 3: it first initializes network parameters using estimates that ignore missing data; then, it considers individuals whose membership w.r.t. 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 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:

- **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.

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: $\{\mathcal{N}^0_{\mathcal{K}}$ is 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: $\quad \{vals(X_i, parents(X_i))$ represents the set of possible values for X_i and its parents$\}$
8: $\quad$ **for** $X_i \in \mathcal{V}, \langle x_i, \pi_{x_i} \rangle \in vals(X_i, parents(X_i))$ **do**
9: $\quad\quad \bar{n}(x_i, \pi_{x_i}) \leftarrow \bar{n}(x_i, \pi_{x_i}) + \Pr(x_i, \pi_{x_i} \mid \mathcal{N}_{\mathcal{K}});$
10: $\quad$ **end for**
11: **end for**
12: **return** $\{\bar{n}(x_i, \pi_{x_i})\};$

Table 1. Ontologies considered in the experiments.

Ontology	Expressivity	#Axioms	#Inds.	#Classes	#ObjProps.
BioPax (Proteomics)	$\mathcal{ALCHN}(\mathcal{D})$	773	49	55	47
Family-Tree	$\mathcal{SROIF}(\mathcal{D})$	2059	368	22	52
MDM0.73	$\mathcal{ALCHOF}(\mathcal{D})$	1098	112	196	22
NTNames	$\mathcal{SHOIN}(\mathcal{D})$	4434	724	49	29
Wine	$\mathcal{SHOIN}(\mathcal{D})$	1046	218	142	21

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 $O_i = \texttt{False}$ iff $\mathcal{K} \not\models F_i(a)$ and $\mathcal{K} \not\models \neg F_i(a)$, $O_i = \texttt{True}$ 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. However, to empirically evaluate the impact of doing so, we include the observability of a variable in the model by allowing its possible values to be a part of $\{\texttt{True}, \texttt{False}, \texttt{Unknown}\}$ (the best partition is chosen by the search process itself, considering each of the alternatives and choosing the one providing the major increase in the penalized CLL), and compare it with the result obtained allowing variables to vary in $\{\texttt{True}, \texttt{False}\}$ only.

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 Terminological Bayesian Classifiers from real world ontologies.

Starting from a set of real ontologies [4] (outlined in Table 1), we generated a set of 20 random query concepts (each corresponding to a DL complex concept) for each ontology [5], so that the number of individuals belonging to the target query concept C (resp. $\neg C$) was at least of 10 elements and the number of individuals in C and $\neg C$ was in the same order of magnitude. A DL reasoner [6]

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

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

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

Table 2. Statistics for cross-validated accuracy results on the generated data sets: for each of the ontologies, 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 *accuracy* (left) and the *area under the precision-recall curve* (right) metrics.

Biopax (Proteomics)	Generic		Polytree		Naïve Bayes	
{ T, F }, FS	0.95 ± 0.1	0.94 ± 0.15	0.96 ± 0.1	0.94 ± 0.15	0.95 ± 0.1	0.94 ± 0.15
Any in {T,U,F}, FS	0.95 ± 0.1	0.93 ± 0.16	0.95 ± 0.1	0.92 ± 0.17	0.95 ± 0.1	0.92 ± 0.17
{ T, F }, FSBE	0.95 ± 0.1	0.94 ± 0.15	0.95 ± 0.1	0.94 ± 0.15	0.95 ± 0.1	0.94 ± 0.15
Any in {T,U,F}, FSBE	0.95 ± 0.1	0.92 ± 0.17	0.95 ± 0.1	0.93 ± 0.17	0.95 ± 0.1	0.93 ± 0.17
Family-Tree	Generic		Polytree		Naïve Bayes	
{ T, F }, FS	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0
Any in {T,U,F}, FS	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0
{ T, F }, FSBE	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0
Any in {T,U,F}, FSBE	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0
MDM0.73	Generic		Polytree		Naïve Bayes	
{ T, F }, FS	0.95 ± 0.08	0.87 ± 0.26	0.95 ± 0.08	0.87 ± 0.26	0.95 ± 0.08	0.87 ± 0.26
Any in {T,U,F}, FS	0.97 ± 0.06	0.9 ± 0.23	0.97 ± 0.06	0.9 ± 0.23	0.97 ± 0.06	0.9 ± 0.23
{ T, F }, FSBE	0.95 ± 0.07	0.87 ± 0.26	0.95 ± 0.08	0.87 ± 0.26	0.95 ± 0.08	0.87 ± 0.26
Any in {T,U,F}, FSBE	0.97 ± 0.06	0.9 ± 0.23	0.97 ± 0.06	0.9 ± 0.23	0.97 ± 0.06	0.9 ± 0.23
NTNames	Generic		Polytree		Naïve Bayes	
{ T, F }, FS	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0
Any in {T,U,F}, FS	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0
{ T, F }, FSBE	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0
Any in {T,U,F}, FSBE	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0	1 ± 0
Wine	Generic		Polytree		Naïve Bayes	
{ T, F }, FS	0.92 ± 0.1	0.89 ± 0.18	0.92 ± 0.1	0.89 ± 0.18	0.92 ± 0.1	0.89 ± 0.18
Any in {T,U,F}, FS	0.92 ± 0.12	0.92 ± 0.14	0.92 ± 0.12	0.92 ± 0.14	0.92 ± 0.12	0.92 ± 0.14
{ T, F }, FSBE	0.92 ± 0.1	0.89 ± 0.18	0.92 ± 0.1	0.89 ± 0.18	0.92 ± 0.1	0.89 ± 0.18
Any in {T,U,F}, FSBE	0.92 ± 0.12	0.9 ± 0.16	0.92 ± 0.12	0.9 ± 0.16	0.93 ± 0.11	0.91 ± 0.16

was employed to decide deductively about the concept-membership of individuals to query concepts.

Experiments consisted in predicting the membership w.r.t. automatically generated concept queries in the form of Terminological Bayesian Classifiers, using different sets of constraints on possible structures (and then obtaining naïve Bayes structures, polytrees or generic Bayesian networks), and on the possible values taken by variables. For predicting the membership w.r.t. the generated query concepts, different constraints on the available values for the variable in networks were empirically evaluated, allowing them to be either {`True`, `False`} or to also take a `Unknown` value, which represents the case in which it is not possible to entail an individual's membership w.r.t. a concept nor to its complement.

During the learning process, we set the *depth* parameter to 3 and *maxLength* to 6 (3 in the case of Family-Tree, for efficiency reasons); for exploring the space of concepts we employed the ψ refinement operator [26], available in the DL-Learner [25] framework, for moving both upwards and downwards in the concept lattice starting from the concept $\top$.

Regarding the feature selection strategy (corresponding to the outer loop in Alg. 2), two different methods were empirically evaluated, namely Forward Selection (FS) and Forward Selection Backward Elimination (FSBE), where the

former only adds (at most) one concept and the latter also considers removing one concept from the committee at each iteration.

Results (expressed using the Accuracy and the Area Under the Precision-Recall curve, calculated as proposed in [10]) have been obtained through k-fold cross validation (with $k = 10$); we evaluated the proposed approach in the Concept-membership prediction task, which consisted in predicting the membership w.r.t. automatically generated query concepts, which was also used in [34] and whose results are summarized in table 2.

From empirical evaluations, it emerged that looking for more complex structures under the penalized CLL did not provide any significant gain over simple naïve Bayesian structures, confirming the simplicity and the accuracy of naïve Bayes network classifiers. There was no statistically significant difference observed adopting different feature selection methods.

On the other hand, it was shown that the missing value handling method impacted on the effective accuracy of the proposed approach: including the *observability* of a concept-membership relation, i.e. whether it can or cannot be proved true or false from the knowledge base, within the probabilistic model, positively impacted on the final accuracy (but making the induced model dependent on the particular ignorance mechanism).

5 Related Works

The problem of managing uncertain knowledge in the SW context has been focused particularly from the knowledge representation perspective. Several approaches, particularly focusing on enriching knowledge and inference procedures with probabilistic information has been proposed. Some of them extend knowledge representation formalisms actually used in the SW. For example: PR-OWL [7] extends the semantics of OWL through the first-order probabilistic logic formalism of Multi-Entity Bayesian Networks [22]. Other approaches rely on probabilistic enrichment of Description Logics [1] (DLs) or logic programming formalisms. Specifically, [15, 28] rely on probabilistic lexicographic entailment from probabilistic default reasoning.

Log-Linear DLs [31] and CR$\mathcal{ALC}$ [8] extend DLs by means of probabilistic graphical models [20]. Similarly, in [16] authors propose probabilistic extension of the DL-Lite language based on Bayesian Networks. In [2], authors propose using Binary Decision Diagrams for efficient reasoning over probabilistic ontologies based on distribution semantics.

To handle vagueness, also fuzzy extensions of Description Logics have been proposed in literature (see e.g. [3, 38, 39]).

5.1 Machine Learning Methods for Knowledge Base Completion

The idea of leveraging Machine Learning methods for handling incomplete and noisy knowledge bases is being explored in SW literature. A variety of methods have been proposed for predicting the truth value of assertions in Web ontologies:

those include generative probabilistic models (e.g. [11, 32, 35]), kernel methods (e.g. [27,42]), matrix and tensor factorization methods (e.g. [30,40]) and energy-based models (e.g. [4]).

An issue with existing methods is that they either rely on a possibly expensive search process, or induce statistical models that are not meaningful to human experts. For example, kernel methods induce models (such as separating hyperplanes) in a high-dimensional feature space implicitly defined by a kernel function. The underlying kernel function itself usually relies on purely syntactic features of the neighborhood graphs of two individual resources (such as their common subtrees [27] or isomorphic subgraphs [42]): in both cases, there is not necessarily an explicit meaning of such syntactic features in terms of domain knowledge.

The Latent variable method in [35], the matrix or tensor factorization methods in [30,40], and the energy-based models in [4], try to explain the observations (assertions) in terms of latent classes or attributes, which also may be not meaningful to the domain experts and knowledge engineers.

The approaches in [32] and [11] try to overcome this limitation by expressing the induced model using a probabilistic extension of the $\mathcal{ALC}$ Description Logic and Markov Logic, respectively. However, inference in these models is intractable in general: inference in [32] and [11] reduces to probabilistic inference to the corresponding ground graphical model.

6 Conclusions and Future Work

This article proposes a method based on discriminatively structured Bayesian networks to predict whether an individual is an instance of a given target concept, given the available knowledge about the individual (in the form of its concept-membership relation w.r.t. a set of feature concepts. Instead of modeling a fully fledged joint probability distribution among concepts in the knowledge base, we face the simpler problem directly model the conditional probability distribution of the aforementioned target concept-membership given other, informative and eventually inter-correlated, feature concept-memberships. We then propose a score-based approach to incrementally build the discriminatively structured Bayesian network, using Description Logic refinement operators [26].

References

[1] Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook. Cambridge University Press (2003)

[2] Bellodi, E., Lamma, E., Riguzzi, F., Albani, S.: A distribution semantics for probabilistic ontologies. In: Bobillo, F., Carvalho, R.N., da Costa, P.C.G., d'Amato, C., Fanizzi, N., Laskey, K.B., Lukasiewicz, T., Martin, T., Nickles, M. (eds.) URSW. CEUR Workshop Proceedings, vol. 778, pp. 75–86. CEUR-WS.org (2011)

[3] Bobillo, F., Straccia, U.: fuzzydl: An expressive fuzzy description logic reasoner. In: FUZZ-IEEE. pp. 923–930. IEEE (2008)

[4] Bordes, A., Glorot, X., Weston, J., Bengio, Y.: A semantic matching energy function for learning with multi-relational data - application to word-sense disambiguation. Machine Learning 94(2), 233–259 (2014)
[5] Brown, G., Pocock, A., Zhao, M.J., Luján, M.: Conditional likelihood maximisation: A unifying framework for information theoretic feature selection. J. Mach. Learn. Res. 13, 27–66 (Mar 2012)
[6] 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)
[7] Carvalho, R.N., Laskey, K.B., da Costa, P.C.G.: Pr-owl 2.0 - bridging the gap to owl semantics. In: Bobillo, F., Carvalho, R.N., da Costa, P.C.G., d'Amato, C., Fanizzi, N., Laskey, K.B., Laskey, K.J., Lukasiewicz, T., Martin, T., Nickles, M., Pool, M. (eds.) URSW. CEUR Workshop Proceedings, vol. 654, pp. 73–84. CEUR-WS.org (2010)
[8] Cozman, F.G., Polastro, R.B.: Complexity analysis and variational inference for interpretation-based probabilistic description logic. In: Bilmes, J., Ng, A.Y. (eds.) UAI. pp. 117–125. AUAI Press (2009)
[9] d'Amato, C., Fanizzi, N., Esposito, F.: Inductive learning for the semantic web: What does it buy? Semantic Web 1(1-2), 53–59 (2010)
[10] Davis, J., Goadrich, M.: The relationship between precision-recall and roc curves. In: ICML 2006. pp. 233–240. ACM, New York, NY, USA (2006)
[11] Domingos, P., Lowd, D., Kok, S., Poon, H., Richardson, M., Singla, P.: Uncertainty reasoning for the semantic web i. pp. 1–25. Springer (2008)
[12] 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)
[13] Donini, F.M., Lenzerini, M., Nardi, D., Nutt, W., Schaerf, A.: An epistemic operator for description logics. Artif. Intell. 100(1-2), 225–274 (1998)
[14] Getoor, L., Taskar, B.: Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning). The MIT Press (2007)
[15] Giugno, R., Lukasiewicz, T.: P-shoq(d): A probabilistic extension of shoq(d) for probabilistic ontologies in the semantic web. In: Proceedings of the European Conference on Logics in Artificial Intelligence. pp. 86–97. JELIA '02, Springer-Verlag, London, UK, UK (2002)
[16] Greco, S., Lukasiewicz, T. (eds.): Scalable Uncertainty Management, Second International Conference, SUM 2008, Naples, Italy, October 1-3, 2008. Proceedings, Lecture Notes in Computer Science, vol. 5291. Springer (2008)
[17] Grossman, D., Domingos, P.: Learning bayesian network classifiers by maximizing conditional likelihood. In: Brodley, C.E. (ed.) ICML. vol. 69 (2004)
[18] Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L. (eds.): Feature Extraction, Foundations and Applications. Springer (2006)
[19] Hitzler, P., van Harmelen, F.: A reasonable semantic web. Semantic Web 1(1-2), 39–44 (2010)
[20] Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press (2009)
[21] Langley, P., Sage, S.: Induction of selective bayesian classifiers. In: de Mántaras, R.L., Poole, D. (eds.) UAI. pp. 399–406. Morgan Kaufmann (1994)
[22] Laskey, K.B.: Mebn: A language for first-order bayesian knowledge bases. Artif. Intell. 172(2-3), 140–178 (2008)
[23] Laskey, K.J., Laskey, K.B.: Uncertainty reasoning for the world wide web: Report on the urw3-xg incubator group. In: URSW2008

[24] Lasserre, J., Bishop, C.M.: Generative or discriminative? getting the best of both worlds. BAYESIAN STATISTICS 8, 3–24 (2007)
[25] Lehmann, J.: Dl-learner: Learning concepts in description logics. Journal of Machine Learning Research 10, 2639–2642 (2009)
[26] Lehmann, J., et al.: Concept learning in description logics using refinement operators. Mach. Learn. 78, 203–250
[27] Lösch, U., Bloehdorn, S., Rettinger, A.: Graph kernels for RDF data. In: Simperl, E., et al. (eds.) Proceedings of ESWC'12. LNCS, vol. 7295, pp. 134–148. Springer (2012)
[28] Lukasiewicz, T.: Expressive probabilistic description logics. Artif. Intell. 172(6-7), 852–883 (2008)
[29] Maier, F., Ma, Y., Hitzler, P.: Paraconsistent OWL and related logics. Semantic Web 4(4), 395–427 (2013)
[30] Nickel, M., Tresp, V., Kriegel, H.P.: A Three-Way Model for Collective Learning on Multi-Relational Data. In: Getoor, L., et al. (eds.) Proceedings of ICML'11. pp. 809–816. Omnipress (2011)
[31] Niepert, M., Noessner, J., Stuckenschmidt, H.: Log-linear description logics. In: Walsh, T. (ed.) IJCAI. pp. 2153–2158. IJCAI/AAAI (2011)
[32] Ochoa-Luna, J.E., 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)
[33] Pernkopf, F., Bilmes, J.A.: Efficient heuristics for discriminative structure learning of bayesian network classifiers. J. Mach. Learn. Res. 11, 2323–2360 (Aug 2010)
[34] 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)
[35] 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)
[36] Rubin, D.B.: Inference and missing data. Biometrika 63(3), 581–592 (1976)
[37] Stoilos, G., Cuenca Grau, B., Motik, B., Horrocks, I.: Repairing ontologies for incomplete reasoners. In: Aroyo, L., et al. (eds.) Proceedings of the 10th International Semantic Web Conference, ISWC2011. LNCS, vol. 7031, pp. 681–696. Springer (2011)
[38] Straccia, U.: A fuzzy description logic. In: Mostow, J., Rich, C. (eds.) AAAI/IAAI. pp. 594–599. AAAI Press / The MIT Press (1998)
[39] Straccia, U.: Towards a fuzzy description logic for the semantic web (preliminary report). In: Gómez-Pérez, A., Euzenat, J. (eds.) ESWC. Lecture Notes in Computer Science, vol. 3532, pp. 167–181. Springer (2005)
[40] Tresp, V., Huang, Y., Bundschus, M., Rettinger, A.: Materializing and querying learned knowledge. In: Proceedings of the First ESWC Workshop on Inductive Reasoning and Machine Learning on the Semantic Web (IRMLeS 2009) (2009)
[41] Vapnik, V.N.: Statistical learning theory. Wiley, 1 edn. (Sep 1998)
[42] de Vries, G.K.D.: A Fast Approximation of the Weisfeiler-Lehman Graph Kernel for RDF Data. In: Blockeel, H., et al. (eds.) ECML/PKDD (1). LNCS, vol. 8188, pp. 606–621. Springer (2013)