Department of Computing Science and Engineering
Southeast University

Bayesian Description Logic Networks

Man Zhu

Department of Computing Science and Engineering
Southeast University
Nanjing, Jiangsu, China
Email mzhu@seu.edu.cn

Technical Report CSE-???

September, 2012

Abstract

Uncertainties have drawn attentions of various domains. Notable domains are data base and ~~the 'Web of data' - the~~ Semantic Web. In the Semantic Web, where complete data being unrealistic to get, numerous partial data exist. Besides, even saying the data already published, the correctness can not be guaranteed absolutely. In order to provide a mechanism for handling uncertainties, we propose *Bayesian dEscription Logic NETworks* (BelNET). BelNET tightly integrate Description Logics (DLs) with Bayesian networks, which provide ways to represent and reason with the uncertainties. In this paper, we introduce the representation of DL knowledge base in BelNET. Additionally, we provide tractable inference and learning algorithms for BelNET. BelNET is targeted with real-world applications. To be specific, ontology learning and query answering. In order to demonstrate the effectiveness of BelNET in practice, BelNET is evaluated by performing ontology learning on various sized data sources. We conclude from experimental results that BelNET is a feasible formalism for the Semantic Web, and the ontology learning algorithm is effective in terms of query answering compared with the state of the art ontology learning approaches.

1. What is the target research problem? What are the existing approaches? What limitations of existing approaches does the current paper address?

2. What is the main idea of the approach? Could you summarise the major steps? What are the strong points of the approach?

3. What are the weak points of the approach? What improvements would you like to see for the paper?

4. What are the claimed contributions of the paper? How well does the paper evaluate such contributions?

5. Details comments on typos and presentations.

Contents

Chapter 1

Introduction

1.1 Motivation

Uncertainties have drawn attentions of various domains. Notable domains are database [2] [13] and the domain of focus in this paper, the Semantic Web [11] [23]. Loosely speaking, there are two forms of uncertainties: partial and erroneous data. Although being targeted to construct a 'Web of Data', it is still unrealistic to demand complete data to be published on the Semantic Web. Under the open world assumption (OWA), which is the assumption made by Semantic Web languages, statements are unknown if they are not explicitly declared to be true and cannot be deduced by reasoners. These are called partialties in this paper. Secondly, due to the publishing mechanism of the Semantic Web, which allows data to be shared almost freely under open licenses, the Semantic Web contains data that might be erroneous in practice. Formalisms are needed to handle these uncertainties.

1.2 Why SRL?

In the early days, ML community have been focused on learning deterministic logical concepts. However, those methods failed to fit perfectly for uncertainties and large-scale circumstances, which leads to statistical methods that ignored relational aspects of the data, such as neural networks, generalized linear models. On the other hand, inductive logic programming (ILP) is designed to learn first-order rules directly which are much more expressive [19]. It is argued in [4] that inductive learning methods, could be fruitfully exploited for learning from the Semantic Web Data. For the last few years, the ILP community and the statistical machine learning community have been incorporating aspects of the complementary technology (machine learning, probability theory, and logic), which leads to the emerging area of statistical relational learning (SRL). It attempts to represent, reason, and learn in domains with complex relational and rich probabilistic structure [7]. Using SRL, two characters of the Semantic Web Data, which distinguish it from other data sets, can be easily handled: 1) Semantic Web Data are highly structured due to the relations between entities and the underlying ontology. 2) Semantic Web Data contains uncertainties as described above.

1.3 Contribution

In this paper, we propose a statistical relation models - Bayesian Description Logic Networks (BelNET). BelNET tightly integrate Bayesian Networks with Description Logics. Bayesian Network is a probabilistic graphical model, that is represented as DAG, and is able to represent a richer set of dependencies than undirected graphs [25].

Chapter 2

Related Work

2.1 SRL Models in General

Recent years have seen a great deal of work on a variety of statistical models that target relational learning tasks [7].

By and large, there are two approaches to the task of modeling relationships under uncertainties: Bayesian networks and Markov networks.

Directed acyclic probabilistic entity-relationship (DAPER) model is an probabilistic extension of the entity-relationship model. DAPER is a language combining plate models and probabilistic relation model, which is expressive, however, the algorithms for it have been a shortage.

Relational Markov networks (RMNs) [27] is proposed mainly for tasks of collective classification and link prediction. Motivated by the disadvantages of using models such as *probabilistic relational models* (PRMs), which is a relational extension to Bayesian networks, in collective classification, namely, the difficulty in achieving a coherent probabilistic model caused by cycles, and the indirect target function in generative training, Markov networks is used. In RMNs, relational clique template, represented by relational query language (RQL), is used to capture the correlations between features, such as document labels. In RMNs, the tasks, such as collective classification and link prediction, can be solved by MAP inference. For considerations of simplicity, efficiency, and accuracy, this is done by belief propagation, a local message passing algorithm introduced in [24].

Conventional statistical models always assume instance independence, which is contradicated by most relational data sets. *Relational Dependency Networks* (RDNs) [22] is capable of expressing and reasoning with such dependencies in a relational setting. The main difference between RDNs and other probabilistic relational models is that RDN is an approximate model where coherent probability distributions are not guaranteed to be specified. RDNs upgrade dependency networks to a first-order logic (FOL) representation with an entity-relationship model. In RDNs, conditional relational learners, to be specific, relational Bayesian classifiers (RBCs) and relational probability trees (RPTs), are used for parameter estimation and structure learning.

Markov logic networks [26] is a combination of first-order logic and markov networks. In a Markov logic network (MLN), each formula has an associated weight that reflects how strong a constraint the formula puts on possible worlds. The syntax of the formulas in an MLN is the standard syntax of first-order logic.

2.2 SRL Models for the Semantic Web

However, in the domain of the Semantic Web, we face a different situation. Firstly, the knowledge presented in the Semantic Web are based on DLs. DLs are decidable fragments of first-order logic closely related to propositional modal and dynamic logics [8]. As a result, the probabilistic structures being represented are different. Secondly, they make different possible world assumptions. Traditional machine learning approaches always make closed world assumption (CWA), where statements are false if they are not specified. Semantic Web languages make open world assumption (OWA), where statements can be true, false, or unknown. The unknown data pose new challenges to traditional approaches. In the following, we will introduce the SRL models for the Semantic Web.

P-CLASSIC [17] is a probabilistic version for DL language $CLASSIC$. P-CLASSIC is mainly motivated to be able to express the degree of overlap between concepts, thus in P-CLASSIC classes are updated to be probabilistic, and are represented by Bayesian networks. P-CLASSIC is able to deduce answers to probabilistic subsumption queries with inference algorithm in Bayesian networks. If the linear time inference is supported

Table 2.1: Comparison of SRL models

model	syntax	inference		learning		ref
		answering CPD	MAP	parameter	structure	
P-CLASSIC	DL	ComputeProbability	-	-	-	[17]
DAPER	ER	-	-	-	-	[10]
RMN	RQL	-	BP	MAP estimation, conjugate gradient, L-BFGS	-	[27]
RDN	ER	-	-	discriminative learning with PLL	conditional relational learner (RBCs, RPTs)	[22]
MLN	FOL	Gibbs sampling	MaxWalkSAT, MC-SAT, CPI, LAZY-(MC-)SAT	generative learning with PLL, discriminative learning with voted perceptron	TDSL, BUSL	[26]
$\mathcal{EL}^{++}$-LL	DL	MC-ILP	MAX-SAT	-	-	[23]

in the original network, P-CLASSIC also supports linear time inference. As shown in Table 2.1, the supporting algorithms are limited.

In [23], the authors propose log-linear DLs, which is a combination of log-linear models with DLs.

```
generally speaking, there are 4 types of data imperfections:
noise, insufficiently covered example space, inexactness, missing values
in the training examples.
```

Chapter 3

Bayesian Description Logic Networks

As discussed above, there are two kinds of uncertainties in the Semantic Web by and large. Namely, partialities and errors. BelNET is motivated by the need to handle these uncertainties. Recent years, uncertainties have been attracted attention of various domains, such as database where probabilistic databases are studied [2] [13]. In the interdisciplinary of machine learning, logic, and probability theory, statistical relational learning is booming, which attempt to represent, reason, and learn in domains with complex relational and rich probabilistic structure [7]. The idea of using probabilitic graphical model to represent the uncertainties and meanwhile the relational structures are maintained provides a promising candidate for handling the uncertainties in the Semantic Web. In this paper Bayesian networks are used to represent and model the uncertainties. They altogether are integrated in a SRL model named *Bayesian Description Logic networks* (Belnet), where DLs are modeled.

Definition 1 (Bayesian Description Logic Networks) *A Bayesian Description Logic network L is a directed graph, where*

- *nodes represent Description Logic concepts.*
- *links signify the existence of direct influences between the linked nodes. To be specific, two nodes are linked, if they represent exactly the two concepts in two sides of an inclusion axiom;*
- *the strength of influences are quantified by factors.*

Generally speaking, a BelNET models a Description Logic knowledge base. In the following, we first briefly review the variant of the DLs and generally describe the DL knowledge base. Secondly, we describe how a DL knowledge base is modeled in BelNET.

3.1 Syntax

The syntax of BelNET is equivalent to the syntax of the underlying Description Logic. For various Description Logics, we use suffixes, such as BelNET-$\mathcal{EL}^{++}$, where the underlying logic is DL $\mathcal{EL}^{++}$.

Generally speaking, a BelNET models a DL knowledge base. A DL knowledge base comprises TBox (*terminology*, i.e., the vocabulary of an application domain) and ABox (*assertions*). TBox consists of concepts denoting sets of individuals (we denote the set of concept names by N_C), and roles denoting binary relationships between individuals (we denote the set of role names by N_R). ABox contains assertions about named individuals (we denote the set of individual names by N_I) in terms of the TBox. We further categorize the ABox into two sets. One is the set of *individual type assertions* such as Holiday(Mid-Autumn_Festival), and the other is the set of *relationship assertions* between individuals such as country(Mid-Autumn_Festival, China). The assertions in the ABox are also called *facts*.

Example 1 *In this DL knowledge base example,* $N_C = \{$*newspaper, tabloid, broadsheet, whitevanman*$\}$, $N_R = \{$*reads*$\}$, $N_I = \{$*Mick, The Guardian, The Times, Daily Mirror, The Sun*$\}$.

The TBox contains:

$$\begin{aligned} \textit{broadsheet} \sqcup \textit{tabloid} &\sqsubseteq \textit{newspaper} \\ \textit{whitevanman} &\sqsubseteq \forall \textit{reads.tabloid} \end{aligned}$$

Table 3.1: Syntax and Semantics of the $\mathcal{S}$ family of Description Logics [12].

construct name	syntax	semantics	
atomic concept	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$	$\mathcal{S}$
atomic role	R	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$	
transitive role	$R \in \mathbf{R}_+$	$R^{\mathcal{I}} = (R^{\mathcal{I}})^+$	
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$	
disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$	
negation	$\neg C$	$\Delta^{\mathcal{I}} / C^{\mathcal{I}}$	
exists restriction	$\exists R.C$	$\{x \mid \exists y. < x, y > \in R^{\mathcal{I}}$ and $y \in C^{\mathcal{I}}\}$	
value restriction	$\forall R.C$	$\{x \mid \forall y. < x, y > \in R^{\mathcal{I}}$ implies $y \in C^{\mathcal{I}}\}$	
role hierarchy	$R \sqsubseteq S$	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$	$\mathcal{H}$
role chains	$R_1 \circ \ldots \circ R_n \sqsubseteq S$	$R_1^{\mathcal{I}} \circ \ldots \circ R_n^{\mathcal{I}} \subseteq S^{\mathcal{I}}$	$\mathcal{R}$
nominal	$\{o\}$	$\{o^{\mathcal{I}}\}$	$\mathcal{O}$
inverse role	R^-	$\{< x, y > \mid < y, x > \in R^{\mathcal{I}}\}$	$\mathcal{I}$
number	$\geqslant n\ P$	$\{x \mid \sharp\{y. < x, y > \in P^{\mathcal{I}}\} \geqslant n\}$	$\mathcal{N}$
restriction	$\leqslant n\ P$	$\{x \mid \sharp\{y. < x, y > \in P^{\mathcal{I}}\} \leqslant n\}$	
qualified number	$\geqslant n\ P.C$	$\{x \mid \sharp\{y. < x, y > \in P^{\mathcal{I}}$ and $y \in C^{\mathcal{I}}\} \geqslant n\}$	$\mathcal{Q}$
restriction	$\leqslant n\ P.C$	$\{x \mid \sharp\{y. < x, y > \in P^{\mathcal{I}}$ and $y \in C^{\mathcal{I}}\} \leqslant n\}$	

The ABox contains:

whitevanman(Mick)	*reads(Mick, Daily Mirror)*	*tabloid(Daily Mirror)*
broadsheet(The Guardian)	*broadsheet(The Times)*	*tabloid(The Sun)*

3.2 Semantics

In BelNET, Description Logic concepts are treated as random variables. The DL knowledge base is represented in a Bayesian network. The Bayesian network defines a joint probability distribution as $P(\mathbf{X} = \mathbf{x}) = \prod_{i=1}^{K} P(X_i = x_i|\mathbf{pa}_i)$ [1], where $\mathbf{pa}_i$ is the assignment of values to the variables in the parents of node X_i (denoted as $\mathbf{Pa}_i$).

Example 2 *Following Example 1, suppose we have the following probabilities, which are part of the full CPT:*

P(broadsheet)=.5 *P(tabloid)=.5* *P(whitevanman)=.3*

P(newspaper|broadsheet, ¬tabloid)=.5 *P(∀reads.tabloid|whitevanman)=1*

then the probability of the truth assignment (broadsheet, ¬tabloid, whitevanman, newspaper, ∀reads.tabloid) is computed by P(broadsheet) × P(¬tabloid) × P(whitevanman) × P(newspaper|broadsheet, ¬tabloid) × P(∀reads.tabloid|whitevanman) = .5 × .5 × .5 × .3 × 1 = .0375

[1]Boldface capital letters such as $\mathbf{X}$ are set of variables, and boldface lowercase letters such as $\mathbf{x}$ are assignments of values to the variables therein.

Chapter 4

Representation

Firstly, we observe that the axioms handled in this paper, namely GCIs and RIs, can be represented with statistical relational models. Dependencies exist in inclusion axioms. For example, a concept inclusion axiom $tabloid \sqsubseteq newspaper$ implies that knowing an individual being an instance of type $tabloid$ helps to infer it being an instance of type $newspaper$ (Fig. 4.1(a)). In fact, without uncertainties, the conditional probability that newspaper is true given the truthness of tabloid is 1, namely $P(newspaper^1|tabloid^1) = 1$. On the contrary, the dependency between two concepts also implies a subsumption relationship. The same case happens for role inclusions.

Bayesian networks are suitable for representing these directionality influences, which are what conveys in inclusions. This paper uses Bayesian networks to represent the probabilities. Description Logics allow describing complex concepts. Possible forms of DL concepts can be nominals ($\{o\}$), atomic concepts (A), negation ($\neg C$) conjunctions ($C \sqcap D$), disjunctions ($C \sqcup D$), (full) existential restrictions ($\exists r.C$), (full) universal restrictions ($\forall r.C$), qualified number restriction ($\leqslant n\ P.C$) and ($\geqslant n\ P.C$). In the following, the representation of a probabilistic DL concept is described respectively according to the syntax.

4.1 Atomic Concept, Negation and Nominal

Atomic concepts and nominals can be considered as random variables, and they are represented as nodes in BelNET. We only consider negations that occur immediately above atomic concepts. Negation is represented as a deterministic node, as shown in Figure 4.1 (b). The probability distribution for a negation is as follows:

$$P(\neg C|C) = \begin{cases} \mathbf{1} & \neg C = 1 - C \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

4.2 Conjunction

An individual belongs to the conjunction $C_1 \sqcap \ldots \sqcap C_k$, if it is an instance of all concepts $C_1 \ldots C_k$. Fig. 4.2(a) depicts the case for an intersection of two concepts C_1 and C_2. The concepts $C_1, \ldots, C_k$ that are operands of

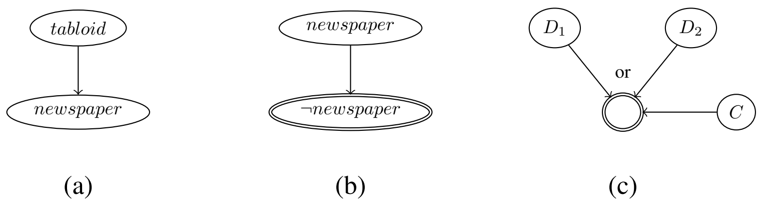

Figure 4.1: Dependencies in $\mathcal{EL}^{++}$ axioms. (a) represent the influence between two concepts in a inclusion axiom. (b) $\neg newspaper$ is a deterministic node representing the negation of atomic concept $newspaper$. (c) C represents a hybrid deterministic node.

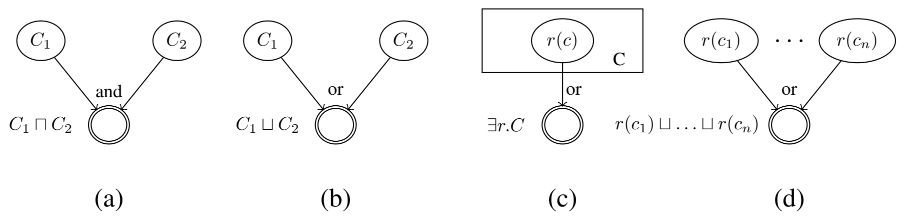

Figure 4.2: Deterministic nodes. (a) depicts a conjunction node. (b) depicts a disjunction node.

$C_1 \sqcap \ldots \sqcap C_k$ are treated as parent node to the deterministic node representing the conjunction.

$$P(C = 1|C_1, \ldots, C_k)_{and} = \begin{cases} \mathbf{1} & \forall i \in \{1, \ldots, k\}, C_i = 1 \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

In Fig. 4.2(a), the double circle node is a deterministic 'and' node. Deterministic means the value of the node is determined when observing the value of the parent nodes. The probability of the deterministic 'and' node representing $C_1 \sqcap \ldots \sqcap C_k$ is defined above.

4.3 Disjunction

An individual belongs to the disjunction $C_1 \sqcup \ldots \sqcup C_k$, if it is an instance of at least one of concepts $C_1 \ldots C_k$. Fig. 4.2(b) depicts the case for an disjunction of two concepts C_1 and C_2. The concepts $C_1, \ldots, C_k$ that are operands of $C_1 \sqcup \ldots \sqcup C_k$ are treated as parent node to the deterministic node representing the disjunction.

$$P(C = 1|C_1, \ldots, C_k)_{and} = \begin{cases} \mathbf{1} & \exists i \in \{1, \ldots, k\}, C_i = 1 \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

In Fig. 4.2(b), the double circle node is a deterministic 'or' node. The probability of the deterministic 'or' node representing $C_1 \sqcup \ldots \sqcup C_k$ is defined above.

4.4 Full Existential Restriction

Full existential restrictions are represented as deterministic 'or' nodes, where the conditional probability of $\exists r.C$ is defined. Fig. 4.2(c) depicts the nodes relative to a fullly existential restricted concept, where rectangle denotes a place notation for repeated variables. In Fig. 4.2(c), the binary role r is represented with one arity under the reason that the first arity is taken as variables. Fig. 4.2(d) is a expanded representation for Fig. 4.2(b). Role r is *partially instantiated* to be $r(C, B)$, where b_i is possible instance of type B.

$$P(\exists r.B|r(C, b_1), \ldots r(C, b_k)) = \begin{cases} \mathbf{1} & \exists i \in \{1, \ldots, k\}, r(C, b_k) = 1 \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

In order to treat the case of representing inclusions of the form such as $C \sqsubseteq D_1 \sqcup D_2$, we introduce a hybrid deterministic node, as depicted in Fig. 4.1(c), with the following CPD:

$$P(C|A, C_1 \ldots C_k) = \begin{cases} \mathbf{1} & \text{either } A = 1, \text{or } \forall i \in \{1, \ldots, k\}, C_i = 1 \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

4.5 Full Universal Restriction

Full universal restrictions are represented as a deterministic 'and' node, where the conditional probability of $\forall r.C$ is defined. Fig. 4.3(a) depicts the nodes relative to a fullly existential restricted concept. Fig. 4.3(b) is a expanded representation for Fig. 4.3(a). Role r is *partially instantiated* to be $r(C, b_i)$, where b_i is possible instances of type B.

$$P(\forall r.B|r(C, b_1), \ldots r(C, b_k)) = \begin{cases} \mathbf{1} & \forall i \in \{1, \ldots, k\}, r(C, b_k) = 1 \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

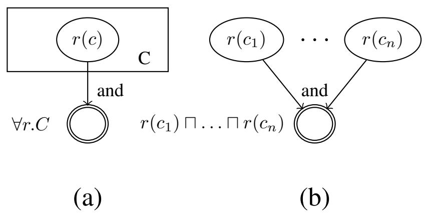

Figure 4.3: Representing universal restrictions and qualified number restrictions. (a) depicts a node for universal restriction. (b) depicts the expansion for (a).

4.6 Qualified Number Restriction

4.7 Network Construction

After initialize a concept (for example concept $\top$), and the Bayesian network G, the following procedure $ConstructBayesNetFromKB$ can be called in order to get a Bayesian network for the DL knowledge base. In this section, the factors attached to the nodes in the Bayesian network are not calculated.

Algorithm 1: ConstructBayesNetFromKB

```
input : Normalized DL $KB$, DL $concept$, $G$
1 begin
2     $children \leftarrow getSubClass(concept, KB)$;
3     $node \leftarrow getNode(concept, G)$;
4     foreach $child$ in $children$ do
5         $childnode \leftarrow ConstructBayesNetFromKB(KB, child, G)$;
6         $linkto(childnode, node)$;
7     return $node$;
8 end
```

Chapter 5

Inference

An important aspect of any probabilistic representation is the support for making inferences; having made some observations, how do we condition on these observations and update our probabilistic model? Our focus is on the most common query type: the *conditional probability query*, $P(\mathbf{Y}|\mathbf{E} = \mathbf{e})$.

$$P(\mathbf{Y}|\mathbf{E} = \mathbf{e}) = \frac{P(\mathbf{Y}, \mathbf{E} = \mathbf{e})}{\sum_y P(\mathbf{y}, \mathbf{e})}$$

It is well known that such inference problems are hard in the worst case, but are tractable for models with bounded *treewidth*. Examples of conditional probability queries are [16]:

prediction queries where we predict the "downstream" effects of various factors;

explanation queries where we reason from effects to causes;

intercausal reasoning also called *explaining away*, where different causes of the same effect can interact.

5.1 Exact Inference

The most common exact inference methods include *variable elimination* and *clique tree propagation*. Variable elimination algorithm sums out variables one at a time, multiplying the factors necessary for that operation. In another algorithm, a preconstructed clique tree is used as a data structure for exact inference. In the clique tree, messages are passed between the cliques in the clique tree, with the end result that the cliques are calibrated. The advantage of variables elimination is that it is flexible in online optimization without the need to preconstruct the clique tree data structure. However, the clique tree propagation could provides answers to multiple cliques using a single computation using dynamic programming.

5.2 Approximate Inference

The computational and space complexity of the clique tree is exponential in the treewidth of the network. In real-world applications of BelNET, we encounter networks with large treewidth. So approximate inference algorithms are needed.

There are two methodologies for solving inference in Bayesian networks approximately: *approximate the target distribution* and *particle-base methods*. The algorithms in the first category are very similar in flavor to the factor manipulation methods underlying exact inference. Examples of the first category are *loopy belief propagation*, *expectation propagation* , and *variational methods*. In the second category, the joint distribution is approximated as a set of instantiations to all or some of the variables in the network. The most common approximate inference algorithms in this category include *importance sampling*, *MCMC*.

Chapter 6

Learning

6.1 Parameter Estimation

There are two main approaches to dealing with parameter estimation task: one based on maximum likelihood estimation, and the other using Bayesian approaches.

The problem setting of the following sections are as follows: Suppose we want to learn the parameters for a Bayesian network with structure G and parameters θ. We are also given a data set D consisting of samples $\xi[1], \ldots, \xi[M]$. Each element in D is an instance in the knowledge base. Each column in D denotes a DL concept. $\xi[i,j]$ represents whether data element i is of type j. In the following, the data elements are assumed to be independent with each other.

6.1.1 Maximum Likelihood Estimation

The likelihood function is

$$\begin{aligned} L(\theta : D) &= \prod_m P_G(\xi[m] : \theta) \\ &= \prod_m \prod_i P(x_i[m] | pa_{Xi}[m] : \theta) \\ &= \prod_i \left[\prod_m P(x_i[m] | pa_{Xi}[m] : \theta) \right] \end{aligned}$$

The equations above show the *global decomposition* property of the likelihood function, which means the likelihood function decomposes as a product of independent terms, one for each CPD in the network. The maximum likelihood estimation (MLE) solution is the one that combines maximizations of each local likelihood function.

Suppose the parameters are organized in a table-CPD, then we will have a parameter $\theta_{x|u}$ for each combination of $x \in Val(X)$ and $u \in Val(U)$. The local likelihood function can be rewritten as:

$$\begin{aligned} L_X(\theta_{X|U} : D) &= \prod_m \theta_{x[m]|um} \\ &= \prod_{u \in Val(U)} \left[\prod_{x \in Val(X)} \theta_{x|u}^{M[u,x]} \right] \end{aligned}$$

The MLE parameters are

$$\hat{\theta}_{x|u} = \frac{M[u,x]}{M[u]} \tag{6.1}$$

However, when we have a very small number of data instances from which we estimate a parameter, the estimates we get can be very noisy, leading to *overfitting*.

6.1.2 Bayesian Estimation

The predictive model in Bayesian estimation has that

$$P(X_i|[M+1] = x_i|U[M+1] = u, D) = \frac{\alpha_{x_i|u} + M[x_i, u]}{\sum_i \alpha_{x_i|u} + M[x_i, u]} \tag{6.2}$$

As for the priors, we will use K2 prior, and BDe prior. Although it might be difficult to populate the 'complete' facts on the Semantic Web, the instance level knowledge in the Semantic Web itself provide ideal source for both inductive and deductive learning, which can be used for itself to grow. There are two main approaches dealing with parameter estimation for Bayesian networks: one is based on maximum likelihood estimation, and the other using Bayesian approaches [16]. Using the instances, firstly, we estimate the factors in the Belnet. In this paper, the parameters are estimated by maximum likelihood estimation, where the task is maximizing $L(\Theta) = \log p(X_i|\Theta)$.

6.1.3 Parameter Estimation with Partial Data

6.2 Structure Learning

6.2.1 Related Work

There are two approaches for finding structures in BNs. The first approach poses learning as a *constraint satisfaction* problem. To be specific, by estimating conditional independence properties between attributes in the data, a network that exhibits the observed dependencies and independencies are built. Usually this is done using a statistical hypothesis test, such as $\mathcal{X}^2$-test. The second approach poses learning as an *optimization* problem. In this approach, the task is to find a structure that maximizes some statistically motivated score. Commonly used scores include BIC/MDL score, BDe score. In general, this problem is NP-hard, so the standard solution is heuristic search. The commonly used baseline method is greedy hill-climbing search over the space of network structures, modified with a tabu list and random restarts. Although the first approach is efficient, it is sensitive to failures in independence tests. Thus, the optimization approach is a better tool for learning structure from data. In this section, we will report the important findings and works in this area.

The costly operation in the optimization approach is collecting various sufficient statistics about the data. When a large number of attributes exist in the domain, the number of possible candidates grow quickly. A large body of researches in learning structures from BNs work on reducing the candidate set.

[6] proposed to restrict possible parents to k promising candidates for a variable by using measure of dependence, such as the mutual information or correlation. Once the candidate parents are selected, the search procedure will be restricted to networks in which only these candidates can be the parents of the variable. In order to avoid finding inferior scoring network, this step will be iterated after a network is found until convergence. All considered measures are mutual information, KL-divergence, conditional mutual information, and score measure. It is reported in experiments that using the score selection method with candidate set of sizes 10 or 15, networks can be learned reasonably close to the one found by greedy hill-climbing in about half the running time and half the number of sufficient statistics. When there are 200 attributes, the speedup is larger than 3.

It has been proved that given an ordering $\prec$ on the variables in the network, finding the highest-scoring network consistent with $\prec$ is not NP-hard [3]. Based on this observation, [28] proposed an algorithm that conducts the search over the space of orderings $\prec$. In this approach, the search space is $2^{O(n \log n)}$, which is significantly smaller than $2^{\Omega(n^2)}$. From the evaluation, we observe that when there are fewer than 50 variables and more than $1,000$ records, the algorithm can at least find the network as good as the baseline method. When the number of variables get larger, the algorithm is better compared to the baseline method in terms of time of convergence and BDe score. It is worth noted that in a real world data set *stress*, which contains 173 instances and $6,100$ variables, the proposed approach converge in 300 seconds with a significantly better result than the baseline method. And in another dataset *diabetes*, which contains $10,000$ instances and 413 variables, it converges in $2,000$ seconds, also with a better result than the baseline method.

By and large, the above works are learning for a single high-scoring model from data with model selection. In small domains with a substantial amount of data, it has been shown that the highest scoring model is orders of magnitude more likely than any other [9]. In such cases, the model selection approach can provide a good approximation. However, under situations such as gene expression, where the amount of data is small relative to the size of the model, there might be many models that explain the data reasonablly well. Model selection can be viewed as simply arbitrarily selecting one from them. Several works [5] [15] [14] consider the edges as structural features based on the observation that an edge appear in this single structure does not necessarily appear in other likely structures, and they propose Bayesian learning. They estimate the extent to which a feature, e.g., the presence of an edge, is likely given the data by estimating it probability $P(f|D) = \sum_G P(G|D)f(G)$, where $f(G)$ is 1 is the feature holds in G. Because these methods average between models, they suffer from complexity problems. In [14], the complexity of calculating the posterior probabilities for all potential edges has been reduced to $O(n2^n)$, which saves the computation by a factor of n^2. The evaluations, which are conducted from a simulation that covers data sets with 20 to $10,000$ records over 5 to 25 discrete attributes.

6.2.2 Algorithms

Algorithm 2: Structure Learning With Partly Known Ordering

```
    input : Normalized DL $KB$, DL $concept$, $G$, a partly-known ordering $\prec$
1   begin
2       Randomly choose one ordering $\bar{\prec} = X_1 \prec \ldots \prec X_n$ that is consistent with $\prec$;
3       for $i \in \{1, \ldots, n\}$ do
4           $Pa \leftarrow$ all possible parents of $X_i$ consistent with $\prec$;
5           $\theta^{t'} \leftarrow$Expectation-Maximization($\mathcal{G}^t, \theta^t, \mathcal{D}$);
6           $\mathcal{G}^{t+1} \leftarrow$Structure-Learn($\mathcal{D}^*_{\mathcal{G}^t,\theta^{t'}}, \mathcal{G}^t$);
7           $\theta^{t+1} \leftarrow$Estimate-Parameters($\mathcal{D}^*_{\mathcal{G}^t,\theta^{t'}}, \mathcal{G}^{t+1}$);
8       return $\mathcal{G}^t, \theta^t$;
9   end
```

Algorithm 3: Structure EM algorithm

```
    input : Normalized DL $KB$, DL $concept$, $G$
1   begin
2       foreach $t = 0, 1, \ldots,$ until convergence do
3           $\theta^{t'} \leftarrow$Expectation-Maximization($\mathcal{G}^t, \theta^t, \mathcal{D}$);
4           $\mathcal{G}^{t+1} \leftarrow$Structure-Learn($\mathcal{D}^*_{\mathcal{G}^t,\theta^{t'}}, \mathcal{G}^t$);
5           $\theta^{t+1} \leftarrow$Estimate-Parameters($\mathcal{D}^*_{\mathcal{G}^t,\theta^{t'}}, \mathcal{G}^{t+1}$);
6       return $\mathcal{G}^t, \theta^t$;
7   end
```

Chapter 7

Evaluation

In this section, we provide empirical evidence that BelNET is a candidate formalism for modeling the uncertainties in the Semantic Web, and the structure learning algorithm as ontology learning performs better than the state of the art methods in terms of query answering.

7.1 Experimental Methodology

We examine various sized data sets (c.f. Table 7.1). The baseline method our structure learning algorithm compare to is the greedy hill-climbing search modified with random start and tabu list. We will use the same ADtree structure for computing the sufficient statistics used in the algorithms, and we will try hard to make it perform best.

The instances will be partitioned in to 10 subsamples. For each data set we will run the structure learning algorithm 10 times with one subsample left out for testing.

The measures used for evaluation are (1) scores of the model learned, such as BDe score and log-likelihood score, (2) running time, (3) the ability to answering the ABox queries (classify a statement as true or false).

Experiment 1. In this experiment, we will show that our structure learning algorithm is at least comparable to the greedy hill-climbing search in terms of BDe score, LL, and running time.

Experiment 2. In this experiment, we will compare our algorithm as an ontology learning approach to other state of the art methods [29] [18]. The comparison is based on measuring (1) the ability to answer the ABox queries of the training data (2) the ability to answer the ABox queries of the testing data (3) the ability to answer the ABox queries of the unknown data. Our algorithm will be carried out on 3 DL languages. Firstly, we will only generate subClassOf relationship between atomic concepts and the negation of atomic concepts. Secondly, we will generate $\mathcal{EL}^{++}$ axioms. Lastly, we will include negated atomic concepts in the DL $\mathcal{EL}^{++}$.

The learning results will be portrayed by ROC curves. In addition, we will show the BDe score of our model of the each run of our algorithm with the goal of illustrating how good the model learned summarizes the data.

Table 7.1: Statistics of the data sets for evaluation. The statistics include counts of concept, and object properties that have more than 0 instances, counts of sub class of axiom, and counts of instances.

ontology	# concept	# object property	# sub class	# instance	# attributes		
					$\mathcal{C}$	$\mathcal{EL}^{++}$	$\mathcal{EL}^{++}+\mathcal{C}$
madcows	13	0	29	13	26	≤91	≤104
people+pets	34	5	33	21	68	≤600	≤634
portal	21	2	256	75	42	≤233	≤254
mindswapper	26	12	45	126	52	≤363	≤389
univ_bench	22	15	36	>1,000	44	≤268	≤290
Adhesome	22	14	108	3032	44	≤267	≤289
financial	49	14	55	17941	98	≤1239	≤1281
DBpedia	394	855	610		788	≤7,8310	≤7,8704

7.2 Experimental Results

Chapter 8

Discussion

8.1 Conclusion

8.2 Future Work

References

[1] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A. Verkamo, et al. Fast discovery of association rules. *Advances in knowledge discovery and data mining*, 12:307–328, 1996.

[2] M. Cheema, X. Lin, W. Wang, W. Zhang, and J. Pei. Probabilistic reverse nearest neighbor queries on uncertain data. *Knowledge and Data Engineering, IEEE Transactions on*, 22(4):550–564, 2010.

[3] G. Cooper and E. Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine learning*, 9(4):309–347, 1992.

[4] C. d'Amato, N. Fanizzi, and F. Esposito. Inductive learning for the semantic web: What does it buy? *Semantic Web*, 1(1-2):53–59, 2010.

[5] N. Friedman and D. Koller. Being bayesian about network structure. a bayesian approach to structure discovery in bayesian networks. *Machine learning*, 50(1):95–125, 2003.

[6] N. Friedman, I. Nachman, and D. Peér. Learning bayesian network structure from massive datasets: the "sparse candidate" algorithm. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 206–215. Morgan Kaufmann Publishers Inc., 1999.

[7] L. Getoor and B. Taskar. *Introduction to statistical relational learning*. The MIT Press, 2007.

[8] B. Grosof, I. Horrocks, R. Volz, and S. Decker. Description logic programs: Combining logic programs with description logic. In *Proceedings of the 12th international conference on World Wide Web*, pages 48–57. ACM, 2003.

[9] D. Heckerman, C. Meek, and G. Cooper. A bayesian approach to causal discovery. Technical report, Technical Report MSR-TR-97-05, Microsoft Research, 1997.

[10] D. Heckerman, C. Meek, and D. Koller. Probabilistic entityrelationship models, prms, and plate models. *Introduction to statistical relational learning*, page 201, 2007.

[11] A. Hogan, A. Harth, A. Passant, S. Decker, and A. Polleres. Weaving the pedantic web. In *3rd International Workshop on Linked Data on the Web (LDOW2010), in conjunction with 19th International World Wide Web Conference, CEUR*, 2010.

[12] I. Horrocks and P. F. Patel-Schneider. KR and reasoning on the semantic web: OWL. In J. Domingue, D. Fensel, and J. A. Hendler, editors, *Handbook of Semantic Web Technologies*, chapter 9, pages 365–398. Springer, 2011.

[13] B. Jiang, J. Pei, Y. Tao, and X. Lin. Clustering uncertain data based on probability distribution similarity. *Knowledge and Data Engineering, IEEE Transactions on*, (99):1–1, 2011.

[14] M. Koivisto. Advances in exact bayesian structure discovery in bayesian networks. In *Proc. of the 22nd Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 241–248, 2006.

[15] M. Koivisto and K. Sood. Exact bayesian structure discovery in bayesian networks. *The Journal of Machine Learning Research*, 5:549–573, 2004.

[16] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. The MIT Press, 2009.

[17] D. Koller, A. Levy, and A. Pfeffer. P-classic: A tractable probablistic description logic. In *Proceedings of the National Conference on Artificial Intelligence*, pages 390–397. Citeseer, 1997.

[18] J. Lehmann, S. Auer, L. Bühmann, and S. Tramp. Class expression learning for ontology engineering. *Journal of Web Semantics*, 9:71–81, 2011.

[19] T. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[20] A. Moore and M. Lee. Cached sufficient statistics for efficient machine learning with large datasets. *Arxiv preprint cs/9803102*, 1998.

[21] A. Moore, J. Schneider, and K. Deng. Efficient locally weighted polynomial regression predictions. In *Proceedings of the 1997 International Machine Learning Conference. Morgan Kaufmann*. Citeseer, 1997.

[22] J. Neville and D. Jensen. Collective classification with relational dependency networks. In *Proceedings of the Second International Workshop on Multi-Relational Data Mining*, pages 77–91. Washington, DC: ACM Press, 2003.

[23] M. Niepert, J. Noessner, and H. Stuckenschmidt. Log-linear description logics. In *IJCAI*, pages 2153–2158, 2011.

[24] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[25] J. Pearl and T. Verma. *The logic of representing dependencies by directed graphs*. Computer Science Department, University of California, 1987.

[26] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1):107–136, 2006.

[27] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 485–492. Morgan Kaufmann Publishers Inc., 2002.

[28] M. Teyssier and D. Koller. Ordering-based search: A simple and effective algorithm for learning bayesian networks. In *Proceedings of the Twenty-first Conference on Uncertainty in AI (UAI)*, pages 584–590, 2005.

[29] J. Völker and M. Niepert. Statistical schema induction. *The Semantic Web: Research and Applications*, pages 124–138, 2011.

Appendix A

Caching Sufficient Statistics

In the basic tasks of probabilistic graphical models, namely inference, parameter learning and structure learning, the key operation is calculating the sufficient statistics. Generally speaking, there are 3 data structures that can facilitate this task, namely the frequent set finder [1], KDtree [21] and ADtree [20]. Among these recent techniques, ADtree is seen to be used in recent structure learning researches such as [28].