Ontology Learning from Incomplete Data by BelNet

Man Zhu[1], Zhiqiang Gao[1], Jeff Z. Pan[2], Yuting Zhao[2], Ying Xu[1]

[1]School of Computer Science & Engineering, Southeast University, P.R. China
[2]Department of Computer Science, The University of Aberdeen, UK

Abstract. Recent years have seen a dramatic growth of semantic web data. Schemas Learning from semantic web data becomes an increasingly pressing issue. In this paper, we propose Bayesian Description Logic Networks (BelNet), to deal with the problem of learning general concept inclusions and disjointness over incomplete semantic data. We implemented the BelNet approach and compared our prototype implementation with a state-of-the-art ontology learning systems.

1 Introduction

The knowledge acquisition bottleneck has resulted in inexpressive schemas on the semantic web (SW) [1], which gives rise to the research of ontology learning - the process of knowledge extraction from diverse data sources [10]. Among the sources explored, the instance level of SW data have been considered to be promising for plentiful reasons. SW data is growing rapidly; e.g., from May 2009 to March 2010, the number of RDF triples has grown from 4.7 billion to 16 billion. Data mining and machine learning techniques, such as association rule mining [15], inductive logic programming (ILP) [11], can be applied to SW data straight forward owing to the similarity of SW data to database in terms of structure.

However, the problem of learning schemas from instance-level data is non-trivial. By making open-world assumption (OWA), the SW generally concerns known true statements. The truth values of unspecified and underivable statements should be assumed as *unknown* [14]. The attempt of making closed-world assumption - assuming true of the specified and derivable statements, and false otherwise - is risky in learning from SW data, where incompleteness is generally acknowledged as inevitable [4]. On the other hand, conforming to the assumption made in SW results in a dataset filled with value *true* without sufficient *false*, which is considered to be important in most learning algorithms, such as ILP. The approach proposed in this paper adopts a probabilistic point of view to deal with the aggressive 'false' under 'risky' CWA.

To address the incompleteness (as a kind of uncertainty) issue, we propose the Bayesian description logic Network (cf. Sec 3), or simply BelNet, a description logic based Bayesian Network [13] for learning schema (TBox) axioms from data axioms (in ABox). BelNet is designed to deal with the issues of (i) learning one single axiom a time and (ii) learning crisp ontological axioms. In the presence of incompleteness, aiming at one best axiom can lead to serious over-fitting problem. For example, one might learn the axiom that 'Father' is a Person with a 'Daughter' child ($Father \sqsubseteq \exists hasChild.Daughter$), if all fathers in the data set happen to have daughters. To address this issue, a global target function is introduced in BelNet for leading the learner out of the local optimum. Moreover, learning crisp axioms may reject the 'probable' correct answers. Take the family dataset (cf. Sec 6.2) as an example, where all fathers accidentally have at least a job; also, due to incompleteness, there is a father who has no known children. A crisp ontology learner might learn the axiom $Father \sqsubseteq \exists hasJob.\top$ and ignore the possible axiom $Father \sqsubseteq \exists hasChild.\top$. To address this issue, a weighted approach is used in BelNet to keep both axioms, with a lower weight for the latter axiom.

We have intensively studied the properties of BelNet, theoretically and practically. In BelNet, the links normally signify the subsumption relationship. Given the ABox data in the ontology, BelNet firstly learns the structure that best encodes the subsumption dependencies supported by ABox data. From the structure, general concept inclusions (GCIs) are extracted directly. In addition, we propose an approach to generating candidate weighted GCIs, which are consequently transformed into linear time inferencing in BelNet (cf. Sec 4). We compare the performance of the ontology learning approach using BelNet with the state-of-the art system DLLearner (cf. Sec 5). Our experiments show: 1) the proposed approach is able to learn TBox axioms even when the TBox knowledge in the ontology is quite rare or even vacant; 2) the results of the proposed approach decrease the incompleteness of the input ontology largely (cf. Sec 6).

The rest of the paper is organized as follows. In Section 2, we recall the basic notions of Description Logics, that are helpful for understanding this paper. Bayesian description logic Networks (BelNet) will be introduced in Section 3, following with Section 4, an algorithm of applying BelNet in ontology learning task. Section 5 proposes a comparison framework and describes the concrete implementation of different possible world assumptions. In Section 6, we describe the experimental results and present the conclusions in Section 7.

2 Preliminary

2.1 Bayesian Networks

Bayesian networks (BNs), also known as belief networks (or Bayes nets for short), belonged to the family of probabilistic graphical models. These graphical structures are used to represent knowledge about an uncertain domain. In particular, each node in the graph represents a random variable, while the edges between the nodes represent probabilistic dependencies among the corresponding random variables. BNs correspond to the graphical model structure known as a *directed acyclic graph* (DAG). BNs are both mathematically rigorous and intuitively understandable. They enable an effective representation and computation of the *joint probability distribution* (JPD) over a set of random variables [13].

A more formal definition of a BN can be given [3]. A Bayesian network B is an annotated acyclic graph that represents a JPD over a set of random variables V. The network is defined by a pair $B = \langle \mathcal{G}, \Theta \rangle$, where $\mathcal{G}$ is the DAG whose nodes $V_1, V_2, ..., V_n$ represent random variables, and whose edges represent the direct dependencies between these variables. The graph $\mathcal{G}$ encodes independence assumptions, by which each variable V_i is independent of its nondescendents give its parents in $\mathcal{G}$. For example, a simple $\mathcal{G}$ $Grandfather \rightarrow Father \rightarrow Male \rightarrow Person$ encodes that 'given *Grandfather*, *Father* is independent of its nondescendent node *Person*'. The second component Θ denotes the set of parameters of the network. This set contains the parameter $\theta_{V_i|Pa_{V_i}} = P_B(v_i|Pa_{v_i})$ for each realization v_i of V_i conditioned on Pa_{v_i}, the set of parents of V_i in $\mathcal{G}$. Accordingly, B defines a unique JPD over V, namely:

$$P_B(V_1, V_2, ..., V_n) = \prod_{i=1}^{n} P_B(V_i|Pa_{V_i}) = \prod_{i=1}^{n} \theta_{V_i|Pa_{V_i}} \tag{1}$$

Belief propagation, also known as *sum-product message passing* is a widely used message passing algorithm for performing inference on graphical models, and will be used in BelNet as well for inference. There are two main approaches to dealing with the parameter estimation task: one based on *maximum likelihood estimation*, and the other using Bayesian approaches. In BelNet, the Bayesian approach will be adopted for a parameter estimation less probable of overfitting.

2.2 Description Logic $\mathcal{ALC}$

Description Logics (DLs) provide the logical formalism for ontologies and the Semantic Web. 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 concept assertions such as Holiday(Mid-Autumn_Festival), and the other is the set of role assertions between individuals such as country(Mid-Autumn_Festival, China). The assertions in the ABox are also called *facts*.

We briefly introduce DL $\mathcal{ALC}$, which is the DL language for representation in BelNet. Please refer to [6] for further details of DLs. Interpretations are used to assign a meaning to syntactic constructs. An *interpretation* $\mathcal{I}$ consists of a non-empty set $\Delta^{\mathcal{I}}$. An *interpretation function* $\cdot^{\mathcal{I}}$ assigns to every object $a \in N_{\mathcal{I}}$ an element of $\Delta^{\mathcal{I}}$, to every atomic concept $A \in N_C$ a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and to every atomic role $r \in N_R$ a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.

Table 1. $\mathcal{ALC}$ syntax and semantics

	construct	syntax	semantics
	atomic concept	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
	atomic role	r	$r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
	top concept	$\top$	$\Delta^{\mathcal{I}}$
	bottom concept	$\bot$	$\emptyset$
	conjunction	$C \sqcap D$	$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
	universal restriction	$\forall r.C$	$(\forall r.C)^{\mathcal{I}} = \{a \mid \forall b.(a,b) \in r^{\mathcal{I}} \text{ implies } b \in C^{\mathcal{I}}\}$
$\mathcal{U}$	*disjunction*	$C \sqcup D$	$(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$
$\mathcal{C}$	*negation*	$\neg C$	$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
$\mathcal{E}$	*existential restriction*	$\exists r.C$	$(\exists r.C)^{\mathcal{I}} = \{a \mid \exists b.(a,b) \in r^{\mathcal{I}} \text{ and } b \in C^{\mathcal{I}}\}$

3 Bayesian Description Logic Network

In connection with a DL ontology, the corresponding Bayesian Description Logic Network (BelNET) is a graph-based knowledge representation showing relationships between concepts. In general a BelNET contains two components:

The structure of an BelNet is a directed acyclic graph (DAG), where
- vertexes represent Description Logic concepts (expressions).
- links signify the existence of direct influences between the linked vertexes. To be specific, two nodes are linked, if they represent exactly the two concepts in two sides of an inclusion axiom; links can be *conditional*, which means the vertex on one side of the link is completely determined by the other node. (c.f. Figure 3)

The numerical information relies on statistics approach against the facts in the ontology ABox, and shows how and in which way the ontology ABox is supporting the relations (links) between two concept nodes.

The reason for not choosing arbitrary links between any pair of nodes is because firstly, current representation supports efficient inferencing in the network. Secondly, the network itself reveals the subsumption relationship of interest. Thirdly, the network structure already encodes the independency information in the underlying data.

In the following, we will firstly introduce how to build the graph structure of BelNET from an ontology, and then we will illustrate how to use the information in the ontology ABox to calculate the Joint Probability Distribution (JPD) for the BelNET.

3.1 Building DAG for BelNET of an ontology

We start from a simple situation. Given an ontology $\mathcal{O}$, let N_C^+ be all concept expressions appearing in $\mathcal{O}$. For any $C \in N_C^+$, We define its *nearest parents* $Pa(C) = \{C' \in N_C^+ \mid \mathcal{O} \models C' \sqsubseteq C$, and there is no C'' such that $\mathcal{O} \models C' \sqsubseteq C''$, and $\mathcal{O} \models C'' \sqsubseteq C\}$.

In this paper, we are particularly interested in ontologies with rich ABox.

Definition 1 ***(ABox Materialisation)*** *For an ontology O, its* ABox materialisation $M_A(O) = \{a : A | A \in N_C^+, a \in N_I, O \models a : A\}$. *If $O = O \cup M_A(O)$, then we say O is* ABox materialised.

Given a consistent ontology $\mathcal{O} =< T, A >$, its corresponding BelNet graph, denoted as $Bel(\mathcal{O})$, is generated with the following steps:

1. For each $C \in N_C^+$, there is a C vertex in $Bel(\mathcal{O})$;
2. If $C' \in Pa(C)$, then there is a link from vertex C' to C;
3. If $C' \equiv C$ and $C \in V$, then label C with an alias C';

For convenience in this paper we use the same symbol for both the concept in DL ontology and the corresponding vertex in the graph.

Example 1 *Fig. 1 (a) shows the graphical representation of the $Bel(\mathcal{O})$, for which the TBox of ontology $\mathcal{O}$ contains:*

$$\begin{aligned} Father &\sqsubseteq Parent & Mother &\sqsubseteq Female \\ Mother &\sqsubseteq Parent & Daughter &\sqsubseteq Female \\ Daughter &\sqsubseteq Child & Parent &\sqsubseteq \exists married.\top \\ Son &\sqsubseteq Child & Child &\sqsubseteq \exists hasParent.\top \end{aligned}$$

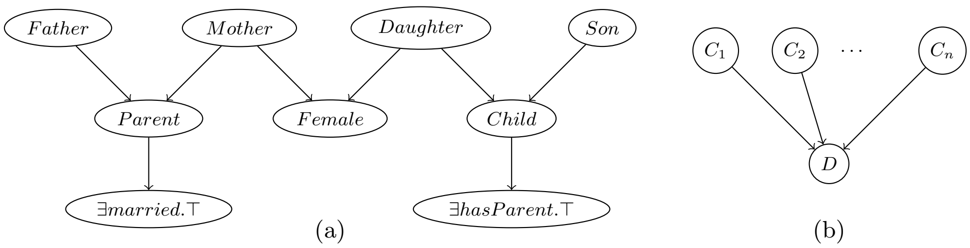


Fig. 1. BelNet graphical representation.

Proposition 1 *Given an ontology $\mathcal{O}$, $Bel(\mathcal{O})$ is a DAG.*

Proof. Assume there is a directed circle $C_1 \to C_2 \to \ldots \to C_n \to C_1$ in $Bel(\mathcal{O})$, then $\mathcal{O} \models C_i \sqsubseteq C_{i+1}, i \in \{1, \ldots, n-1\}$, and $\mathcal{O} \models C_n \sqsubseteq C_1$ then $\mathcal{O} \models C_1 \equiv C_2 \equiv \ldots \equiv C_n$. In $Bel(\mathcal{O})$, $C_2, \ldots, C_n$ are alias of vertex C_1, which conflicts with the assumption. Thus no directed cycle in $Bel(\mathcal{O})$ exists, and $Bel(\mathcal{O})$ is a DAG.

3.2 Generating Joint Probability Distribution for BelNet

Along with ontology TBox constructs the links in $Bel(\mathcal{O})$, ontology ABox contributes to the parameters on the links, which reflects the supportiveness from the evidences (ABox assertions) to the BelNET graph. Now we introduce how to generate the *Conditional Probability Tables* (CPT) for each vertex with her parents, in the BelNet graph $Bel(\mathcal{O})$.

It is natural to use a finite ontology domain $\mathbf{\Delta}^{\mathcal{I}}$ to restrict all elements in the possible world in the BelNET. For convenience, we assume $\mathbf{\Delta}^{\mathcal{I}}$ contains all individual names in the ontology, and an individual name o is always interpreted to itself, *i.e.*, $o^I = o$.

We call each element o in $\Delta^{\mathcal{I}}$ a *possible observation*. A possible observation is an interpretation which assigns at most one element to one concept. We assume that all possible observations are independent.

Marginal nodes are those having no parent in $Bel(\mathcal{O})$. The *marginal probability* of a marginal node C is a table of $P(C^\sharp)$, where $\sharp \in \{\texttt{TRUE}, \texttt{FALSE}\}$. Furthermore, $P(C^{\texttt{TRUE}})$ is the probability that a possible observation supports C, *i.e.*, $o \in C^{\mathcal{I}}$. Similarly $P(C^{\texttt{FALSE}})$ is the probability that a possible observation does not support C, *i.e.*, $o \notin C^{\mathcal{I}}$. Actually the values are related to the number of individuals satisfiying concept C in the ontology. For convenience in the following $P(C^{\texttt{TRUE}})(/P(C^{\texttt{FALSE}}))$ is shortened to $P(C^T)(/P(C^F))$.

Definition 2 ***(Bayesian subsumption axiom)*** *A Bayesian subsumption axiom is in the form of* $D|C_1, \ldots, C_n$, *where* $C_i \sqsubseteq D, i \in \{1, \ldots, n\}$.

Fig. 1 (b) shows the graph of a Bayesian subsumption axiom. The vertexes in $Bel(\mathcal{O})$ are treated as random variables, so the *Conditional Probability Tables* (CPT) of a Bayesian subsumption axiom is calculated based on the *Bayesian subsumption function*

$$P(D|C_1, \ldots, C_n) = \frac{P(D, C_1, \ldots, C_n)}{P(C_1, \ldots, C_n)} \tag{2}$$

where $P(C_1, \ldots, C_n)$ is a discrete probability distribution, and $P(C_1^{\sharp_1}, \ldots, C_n^{\sharp_n})$, $\sharp_i \in \{T, F\}$, is the probability that a possible observation o satisfies $o \in C_i^{\mathcal{I}}$ if $\sharp_i = T$, or $o \notin C_i^{\mathcal{I}}$ if $\sharp_i = F$. This can be abbreviated as $\mathbf{C}_i^o$.

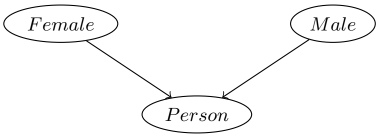
Example 2 *Given an ontology* $\mathcal{O} =< T, A >$, *where* T *includes* $\{Male \sqsubseteq Person, Female \sqsubseteq Person\}$. *We also have ABox as:*

$$Person(a), Person(b), Male(a), Female(b)$$

Fig. 2 shows the marginal probabilities for Female *and* Male*, and the CPT for* Person*.*

Actually the CPT reflects how much degree the ABox supports the subsumption axioms. Obviously we have following proposition.

$Female$	$P(Female)$
T	0.5
F	0.5



$Male$	$P(Male)$
T	0.5
F	0.5

$Female$	$Male$	$P(Person^T \mid Female, Male)$
T	F	1
F	T	1

Fig. 2. Motivated BelNet Example

Proposition 2 *In the BelNET of an ABox materialised ontology O, we have $P(D^T|C^T) = 1$, if $C \in Pa(D)$.*

Proof. Follows directly from the steps of transforming ontology into BeLNet, nodes C in $Pa(D)$ satisfy $C \sqsubseteq D$. $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, which means the probability of a possible observation o satisfies $o \in C^{\mathcal{I}}$ and $o \in D^{\mathcal{I}}$ is the same as the probability of $o \in C^{\mathcal{I}}$, a.k.a. $P(D^T, C^T) = P(C^T)$. From equation 2, $P(D^T|C^T) = \frac{P(D^T,C^T)}{P(C^T)} = 1$.

Lemma 1 *Given a consistent and ABox materialised ontology $\mathcal{O}$, and the corresponding $Bel(\mathcal{O})$, $P(D^T|Pa(D)) = 1$, if there is a $C_i \in Pa(D)$, whose value is true, and there exists a possible observation o satisfies $\mathcal{O}$ and $o \in D^{\mathcal{I}}, o \in C_i^{\mathcal{I}}$.*

Now we can measure the global supportiveness from evidences in ontology ABox to a BelNET graph $Bel(\mathcal{O})$.

Definition 3 ***(BelNET Joint Probability Distribution(JPD))*** *Given an ontology O and $Bel(\mathcal{O})$, the joint probability distribution is defined as:*

$$P(V_1, \ldots, V_n) = \prod_{i=1}^{n} P(V_i|Pa(V_i)) \tag{3}$$

where $V_1, \ldots, V_n$ are all vertexes in $Bel(\mathcal{O})$.

Proposition 3 *Given a consistent and ABox materialised ontology $\mathcal{O}$ and $Bel(\mathcal{O})$, we have $P(o) = \prod_{Pa(V_i)=\emptyset} P(Vi^o)$, if o satisfies $\mathcal{O}$.*

Proposition 3 follows from Lemma 1 and Equation (3).

As a direct conclusion from the joint probability distribution function, we have that

Theorem 1 *For an interpretation $\mathcal{I}$ satisfies a consistent and ABox materialised ontology $\mathcal{O}$ and $Bel(\mathcal{O})$ $\mathcal{G}$, we have $P(\mathcal{G}^{\mathcal{I}}) = \prod_{Pa(V_i)=\emptyset} P(V_i^v)^k$, k is the total number of observation o in $\mathcal{I}$ such that $o \in V_i^{\mathcal{I}}$ if $v = T$ and $o \notin V_i^{\mathcal{I}}$ if $v = F$.*

Proof.

$$
\begin{aligned}
P(\mathcal{G}^{\mathcal{I}}) &= \prod_{o \in \Delta^{\mathcal{I}}} P(\mathcal{G}^{o}) \\
&= \prod_{o \in \Delta^{\mathcal{I}}} \prod_{Pa(V_i)=\emptyset} P(V_i) \\
&= \prod_{Pa(V_i)=\emptyset} P(V_i^{v})^{k}
\end{aligned}
$$

Continuing to Example 2 in Fig. 2, we have $P(\mathcal{I}|\mathcal{G}) = 0.5^2 \times 0.5^2 \times 1^2$.

By now we have introduced the BelNET model for a ontology, and intensively studied the features of BelNET for an ABox materialised ontology which having rich ABox assertions. In the next section we will introduce how to learn the BelNET structure from the evidences in ontology ABox.

4 Learning with BelNet

The learning approach includes 3 main steps: *Pre-processing.* Given an ontology $\mathcal{O}$, for each $A \in N_C$ and $r \in N_R$, pre-processing creates nodes corresponding to A and $\exists r.\top$ in $Bel(\mathcal{O})$. Because all individuals belonged to concept $\forall r.A$, we generate $\exists r.A \sqcap \forall r.A$ instead as the approximation for $\forall r.A$.

4.1 Structure Learning

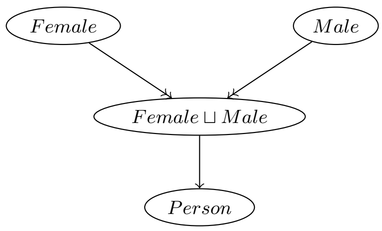


Fig. 3. BelNet Example

Definition 4 (TBox targeted structure learning in BelNet) *Given an ABox-enriched ontology $\mathcal{O} =< T, A >$, find a BelNet $\mathcal{G} =< V, E >$, such that $P(\mathcal{G}, A)$ is maximized, under the constraint that each link in $\mathcal{G}$ corresponds to a subsumption dependency relation.*

Roughly speaking, the BNs structure learning algorithm starts from an initial structure (in our case, the structure is initialized with all the nodes from pre-processing, and no edges), and try to find the best operation (in terms of adding / deleting /reversing) that can be carried out from the current structure. This process iterates until no better structure in terms of specific score function can be found, or the step reaches the maximum step (c.f. Algorithm 1).

Algorithm 1: structure learning in BelNet

input : BelNet $\mathcal{G} =< V, E >$, $E = \emptyset$, $\mathcal{M} =< C, Inst_C >$, max_iter
output: $\mathcal{G}'$

begin
  Initialize $best\_score$ for $\mathcal{G}$;
  for *each pair of nodes* **do**
    cache the score for adding/deleting/reversing the link between them;
  while max_iter *not reached* **do**
    while *best operation not found and cache not fully visited* **do**
      $o \leftarrow$ the best operation from the cache;
      if o *satisfies the selection criteria* **then**
        if o*'s inverse already in* $\mathcal{G}$ **then**
          Label o's inverse with alias o;
          merge o and o';
        else
          best operation found;
    if *best operation found and* $new\_score \geq best\_score$ **then** do operation o, and label the network as $\mathcal{G}'$;
    add o into tabulist;
    update cache;
    $best\_score \leftarrow new\_score$;
    else **return** $\mathcal{G}'$;
end

Score Function Thus, in the Bayesian network structure learning algorithms, the vital part is evaluating an operation, a.k.a. adding / deleting a link, and reversing the direction of a link. This is done by score functions. The score functions used in Bayesian network structure learning include maximum likelihood measure, Bayesian score, and extensions of Bayesian score. Likelihood measure suffers from over-fitting problems, and always prefers complexer network to a simpler one, which is not always the real preference in practice. Due to the better performance in handling over-fitting problems of Bayesian score [8], we will adopt Bayesian score as our score function.

Property 1 *Given two candidate nodes N_1 and N_2 for node N, Bayesian score prefers to link N_1 to N, if $P(N^T|N_1^T) > P(N^T|N_2^T)$, and $P(N^T|N_1^F) < P(N^T|N_2^F)$.*

The local graph of node N $loc(N, \mathcal{G})$ is a subgraph of $\mathcal{G}$. $loc(N, \mathcal{G})$ is composed of node N, $Pa(N)$, and the links among them.

Property 2 *Only a local graph is needed to be considered when comparing $\mathcal{G}$ and $\mathcal{G}'$, if $\mathcal{G}'$ can be achieved from $\mathcal{G}$ by adding / deleting a link or reversing a link.*

Algorithm 2: Post-processing in BelNet

---

input : BelNet $\mathcal{G} =< V, E >$, with JPD, $threshold_{disjoint}$, $\mathcal{O}$
output: $\mathcal{O}'$

begin
  Initialize an empty *axiomlist*;
  for *each node who has more than one parent* **do**
    for *any combination of two parent nodes V_i, V_j* **do**
      if $P(V_i^T, V_j^T) < threshold_{disjoint}$ **then**
        add $(< V_i,$ disjointWith $V_j >, P(V_i^T, V_j^T)) \rightarrow axiomlist$;
  sort *axiomlist* ASC according to the probability;
  for *each element in axiomlist* **do**
    if *adding axiom to $\mathcal{O}$ not causing inconsistency* **then**
      add $axiom \rightarrow \mathcal{O}$;
end

Structure Selection After an operation is selected by the score function, in order to meet the demand of BelNet, to be specific, the preference over structures whose links signifying the special dependency called 'subsumption'.

We denote the candidate operation as O, where O_{head} is the node to which the link points, and O_{tail} represents the node from which the link starts. Further, we denote the count of instances that belongs to both concepts corresponding to O_{tail} and O_{head} as $M[O_{head}, O_{tail}]$, the count of instances belonging to concept O_{head} as $M[O_{head}]$, similar for $M[O_{tail}]$. Then, operation O will be selected iff $M[O_{head}, O_{tail}] = M[O_{tail}]$ and $M[O_{tail}] > threshold_{parent}$. Properly selected thresholds will help when there are errors in the dataset. However, the focus of the paper is dealing with incompleteness. In this case, $threshold_{joint} = threshold_{parent} = 0$.

4.2 Post-processing

After the structure of BelNet is learnt, we can extract various kinds of axioms from BelNet by inferencing in it. (refer to Table 2 for the details of this translation). The results of CPD query are the weights of the corresponding DL axioms. After the BelNet has been learned, post-processing extracts GCIs and disjointness from the BelNet by the following procedure:

Table 2. Extract DL axioms by inferencing BelNet

	DL	CPD query
conjunction	$\sqcap_{i=1}^{n} C_i$	$P(C_1^T, \ldots, C_n^T)$
disjunction	$\sqcup_{i=1}^{n} C_i$	$1 - P(C_1^F, \ldots, C_n^F)$
disjointness	$\sqcap_{i=1}^{n} C_i \sqsubseteq \bot$	$P(C_1^T, \ldots, C_n^T) < \text{threshold}_{disjoint}$
subsumption	$C \sqsubseteq D$	$P(D^T \mid C^T) > \text{threshold}_{subsume}$

(1) For each alias in the BelNet, generate an equivalent axiom. For example, if node C has the alias of D, generate $C \equiv D$.
(2) For each non-conditional link $C \to D$ in the BelNet, generate an axiom $C \sqsubseteq D$.
(3) Generate disjointness axioms by Algorithm 2.

5 Evaluation Metrics

Ontology learners can serve various purposes, which qualifies the ontology learners in various levels:

- **Ontology construction.** (Semi-)automatically constructing the knowledge base by mining from the ABox data in the ontology, the main focus of learner in this dimension is the correctness of the axioms learned.
- **Resolving incompleteness.** Helping users to construct a (near-)complete ontology knowledge base. The end result of this type of learner provides correct and non-vague answers towards the queries submitted.
- **Ontology understanding.** Helping users to understand how one concept can be defined in terms of a certain vocabulary. This type of learners try to provide detailed description for a set of individuals.

Among the three dimensions, ontology understanding is the highest level. The focus of this paper is on both ontology construction and resolving incompleteness.
Notations. We denote the original ontology (the input of ontology learners) as $\mathcal{O}$, and the output as $\mathcal{O}'$. In order to evaluate whether the ontology learner is able to work when there are only ABox data, we denote $\mathcal{O}^{\mathcal{T}-}$ as the ontology by removing TBox axioms from $\mathcal{O}$, and correspondingly denote $\mathcal{O}^{\mathcal{T}-\prime}$ as the output of the ontology learner with input $\mathcal{O}^{\mathcal{T}-}$. With these notations, *precision* and *recall* can be calculated as follows:

$$Precision(\mathcal{O}, \mathcal{O}') = \frac{|\{T | T \in \mathcal{O}' \text{ and } \mathcal{O} \models T\}|}{|\{T | T \in \mathcal{O}'\}|}$$

$$Recall(\mathcal{O}, \mathcal{O}') = \frac{|\{T | T \in \mathcal{O} \text{ and } \mathcal{O}' \models T\}|}{|\{T | T \in \mathcal{O}\}|}$$

F1-measure is the harmonic mean of precision and recall.

In order to evaluate the incompleteness of a dataset, we adopt a measure called *uncertainty ratio*, which is the percentage of unknown answers to all possible queries of the form "Is the individual a belonged to the concept A?". Uncertainty ratio is calculated as follows:

$$uncertainty(\mathcal{O}) = \frac{|\{f(a, A, \mathcal{O}) = \text{unknown} | a \in N_I \text{ and } A \in N_C\}|}{|N_I| \times |N_C|}$$

where

$$f(a, A, \mathcal{O}) = \begin{cases} \text{true} & \mathcal{O} \models A(a) \\ \text{false} & \mathcal{O} \models \neg A(a) \\ \text{unknown} & \text{otherwise} \end{cases}$$

Consequently, uncertainty ratio can be defined as follows:

$$uncertainty\_ratio(\mathcal{O}, \mathcal{O}') = uncertainty(\mathcal{O}')/uncertainty(\mathcal{O})$$

On the other hand, besides evaluating the uncertainty has been reducted, we can additionally evaluate the correctness of this uncertainty reduction result. This is done by first constructing a complete standard ontology by adding as more correct disjointness axiom as possible, denoted as $\mathcal{O}^S$, and

$$correctness(\mathcal{O}, \mathcal{O}^S) = \frac{|\{f(a, A, \mathcal{O}^S) \neq f(a, A, \mathcal{O}')\}|}{|\{f(a, A, \mathcal{O}^S\}|}$$

In all these measures, $\mathcal{O}'$ can be replaced by $\mathcal{O}^{\mathcal{T}-\prime}$, which qualifies the ability of the learner to learn only with ABox, and without any TBox.

6 Experiments

In the experiments, we are going to evaluate the proposed ontology learning approach by BelNet from the following aspects: 1) We evaluate the proposed approach by checking the correctness of the axioms learnt by BelNet, and whether the axioms in the input ontology can be learned. 2) We analyze to which extent the proposed method improves the incompleteness of the input dataset, and how reasonable the improvements are.

6.1 Experiment Setup

Datasets The experiments are carried out on 4 ontologies: family [1], semantic bible [2], LUBM [3], and financial [4] (c.f. Table 3, where we calculate the number of concepts (c for short), object properties (op for short), number of Subclassof,

[1] `https://github.com/fresheye/belnet/blob/master/ontology/family-benchmark_rich_background.owl`
[2] `http://www.semanticbible.com`
[3] `http://swat.cse.lehigh.edu/projects/lubm/`
[4] `http://www.cs.put.poznan.pl/alawrynowicz/financial.owl`

Equivalentclasses, DisjointClasses axioms, number of individuals, DL expressivity, and the uncertainty of the corresponding dataset.). The DL expressivity of the ontologies chosen are not restricted to $\mathcal{ALC}$. In the proposed approach, all concept expressions in the original ontology are treated as a concept, and if they exceed the expressivity of $\mathcal{ALC}$, they will be treated the same way as a named concept.

We do experiment on a computer with 4 core 2.27GHz CPU, and 4G RAM. We evaluate the performance of our approach under the existence of incompleteness by randomly partitioning the original dataset into 10 parts. Each time of the experiment, we randomly select one part from the partitions, and to which we add another randomly selected one at the second time. At last, we get the whole dataset, which is the completest one. This procedure is carried out 10 times in order to demonstrate the objectiveness of the evaluation.

Table 3. Statistics of the data sets for evaluation.

ontology	# c	# op	# $\sqsubseteq/\equiv/\perp$	# ind	DL expressivity	uncertainty
Family	19	4	27 / 0 / 0	202	$\mathcal{AL}$	0.609
Semantic Bible	49	29	51 / 0 / 5	724	$\mathcal{SHOIN}(\mathcal{D})$	0.887
LUBM	43	25	36 / 6 / 0	1555	$\mathcal{ALEHI}(\mathcal{D})$	0.946
Financial	60	16	55 / 0 / 113	17941	$\mathcal{ALCOIF}$	0.067

6.2 Results

We select the $threshold_{disjoint}$ by running experiment and check both uncertainty ratio and correctness of BelNet, and fix this parameter. For reason of space limitation, we only show the threshold versus the uncertainty ratio & correctness on benchmark dataset family (c.f. 4).

From the experiments we conclude that 1) The correctness of BelNet goes higher when the threshold is larger. 2) When the dataset size goes high, the correctness is better. 3) The uncertainty ratio is lower if the threshold is higher, and the uncertainty ratio is higher if the dataset is large. This is because when the dataset is larger, BelNet starts to add disjointness axioms carefully, which results in a higher uncertainty ratio, and when less disjointness are learnt, the correctness grow. In the following experiments, we will choose 0.1 as our threshold.

Fig. 5 represents the performance of structure learning in terms of precision, recall, and F1-measure. We tried different thresholds for GoldMiner, and finally we chose the support threshold to be 0.8 to get a higher precision. Because otherwise GoldMiner get neither a high precision nor a high recall. From the figure, we can see that 1) for most of the dataset, our method is better than DLLearner and GoldMiner in terms of precision, recall and F1-measure. 2) However, the recall is not high compared with precision. This is understandable because that as we get more data, structure learning is getting more axioms that can not be

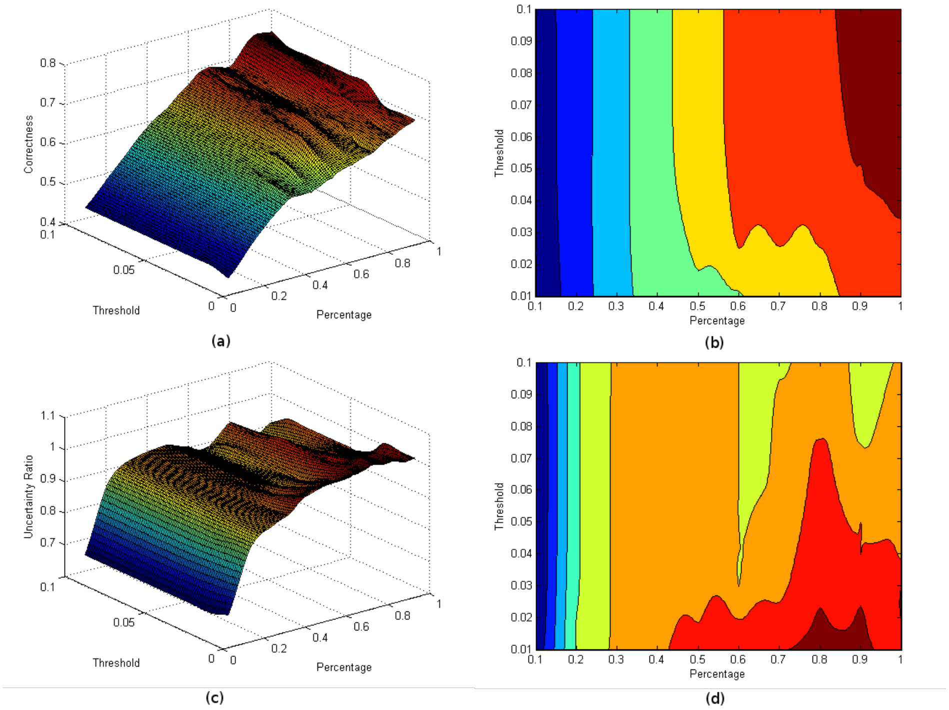


Fig. 4. The uncertainty ratio & correctness versus $threshold_{disjoint}$ and partition size. (a) represents the uncertainty in terms of $threshold_{disjoint}$ and the size of the dataset. (b) represents the contour of subgraph (a). (c) demonstrate the corresponding correctness in terms of $threshold_{disjoint}$ and the size of the dataset. (d) represents the contour of subgraph (c).

entailed by the original ontology, but these axioms can be true in practice. For example, from the family dataset, we get more axioms like $grandson \sqsubseteq son$, and these concepts belonged to the same level in the original ontology. Table 4 also shows a comparison of a snippet of the results of BelNet and DLLearner when the size of the dataset changes.

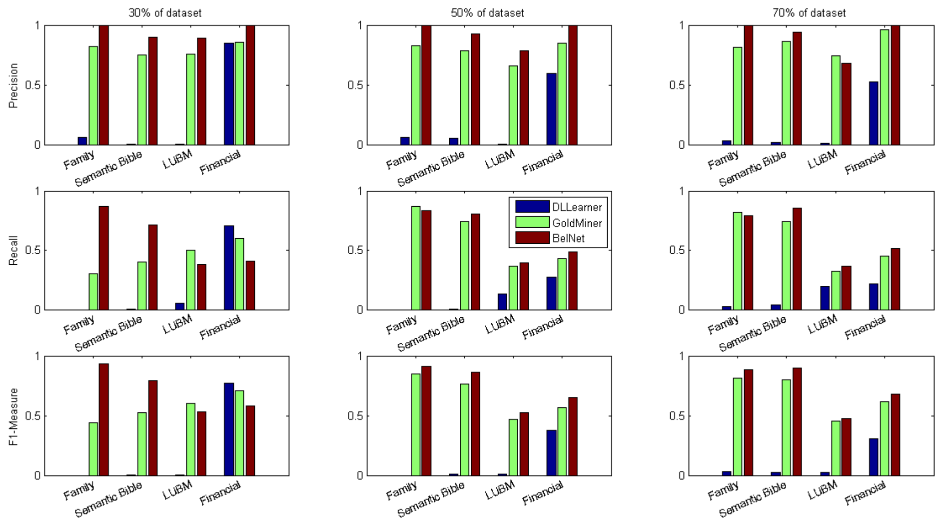


Fig. 5. The Precision, Recall, and F1-Measure of BelNet and state-of-the-art learners, in terms of the size of the data.

Table 4. Axioms learned for concept *Grandson*

%	BelNet ($Grandson \sqsubseteq$)	DLLearner ($Grandson \equiv$)
10	$Male$, $Grandchild$, $\exists hasParent.\top$, $Child$	$Male \sqcap \exists hasParent.\neg Person$
20	$Male$, $Grandchild$	$(Male \sqcap \neg Parent) \sqcup \neg Person$
30	$Male$, $Grandchild$	$(\neg Female \sqcap \neg Parent) \sqcap \forall hasChild.Mother$
40	$Male$, $Grandchild$	$\neg Female \sqcap \neg Grandparent \sqcap \forall hasSibling.Child$
50	$Male$, $Grandchild$	$\neg Female \sqcap \forall hasChild.(Child \sqcap \neg Parent)$
60	$Male$, $Grandchild$	$\neg Female \sqcap \forall married.\forall married.Son$
70	$Male$, $Grandchild$	$Person \sqcap \neg Female \sqcap \forall married.\forall hasParent.Sister$
80	$Grandchild$	$\neg Female \sqcap \forall married.\forall hasParent.Brother$
90	$Male$, $Grandchild$	$\neg Female \sqcap \exists hasParent. \leq 1 hasChild.GrandParent$
100	$Male$, $Grandchild$, Son	$Son \sqcap \exists hasParent.Child$

7 Related Work

Learning schemas from instance-level data has attracted attention since the fast development of semantic web. d'Amato et al. did a thorough survey [2] of the domain of ontology learning. In this section, we only notice a subset of work that focus on learning a broader sense of axioms from ABox data here. Due to the relationship between BelNet and statistical relational learning, important and closely related works on SRL models are also briefly reviewed in this section.

In [11], the authors developed *DLLearner* to learn $\mathcal{ALC}$ *cocnept descriptions* from ontologies based on ILP techniques, where the candidate concept descriptions are generated by a downward refinement operator.In addition, in [5], they particularly focused on larger datasets, such as DBpedia. DLLearner generates concept descriptions quite well when the data quality is relatively high. However, under the existence of incompleteness, which is the main focus of this paper, DLLearner would drop into local optimum description for concepts due to the incorrect 'false' values generated by making CWA. *Gold-Miner* [15] tries to learn $\mathcal{EL}$ axioms from ontologies based on association rule mining method. The target of *Gold-Miner* is not solving the incompleteness, which is the focus of this paper. In addition, Galárraga et al. [4] proposed a *rule* mining model supporting OWA scenario by introducing a new confidence measure in association rule mining.

We briefly review the SRL methods related to BelNet in terms that 1) they try to solve the task of ontology learning from semantic web data; 2) they are proposed in the context of semantic web; 3) they adopt Bayesian networks for handling uncertainties. Koller et al. extended DL CLASSIC with nodes in a Bayesian represent probabilistic information of the individuals in a specific class [9], which is closely related to the representation in BelNet. However, in BelNet, the edges correspond to the specific type of dependency (subsumption), but not the broader sense of dependency of any type. *Bayesian logic programs* (BLP) [7] unifies definite logic programs with Bayesian networks. In BLP, ground atoms are mapped to random variables. BelNet differs from BLP in that 1) the representation languages are different; 2) BelNet models concepts with random variables; 3) In addition, BelNet is suitable for schema level ontology learning. OntoBayes [16] extends OWL with annotating RDF triples with probabilities and dependencies. In [12], $\mathcal{EL}^{++}$-LL was proposed to extend crisp ontological axioms with weights. Using $\mathcal{EL}^{++}$-LL, a subset of coherent axioms can be learned from a set of *weighted* $\mathcal{EL}^{++}$ axioms.

8 Conclusion and Future Work

In this paper, we proposed Bayesian Description Logic Network (BelNet), for learning TBox axioms from incomplete ABox axioms. In BelNet, DL concept expressions correspond to probabilistic nodes, and subsumption relationships between DL concept expressions are represented as links. Probabilistic subsumption axioms can be extracted from the BelNet. The problem of learning DL axioms is transformed into structure learning in BelNet, which, from the experiment, was shown to be effective for learning from incomplete semantic data.

A technical report with full proofs and more details of the experiments can be found at: `https://github.com/fresheye/belnet/blob/master/TR/TR.pdf`.

In the future, we will further develop the inconsistency handling and reasoning techniques in BelNet. Consider an example where $\{Female \sqcap Male \sqsubseteq \bot, Female \sqsubseteq Person, Male \sqsubseteq Person\}$ are stated in the ontology. In BelNet, node Female and Male both link to node Person. Parameter estimation from a consistent ontology will lead to $P(Person^T|Female^T, Male^T)$ and $P(Person^F| Female^T, Male^T)$ are both 0, which conflicts $\sum_i P(C^i|D)$. This kind of situation can be represented by introducing three-value BelNet node, with an addition value of 'impossible', for detecting inconsistencies in the data.

References

1. P. Cimiano. *Ontology learning and population from text: algorithms, evaluation and applications*. Springer Verlag, 2006.
2. 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.
3. N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine learning*, 29(2-3):131–163, 1997.
4. L. Galárraga, C. Teflioudi, K. Hose, and F. M. Suchanek. Amie: Association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22nd international conference on World Wide Web*. ACM, 2013.
5. S. Hellmann, J. Lehmann, and S. Auer. Learning of owl class descriptions on very large knowledge bases. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(2):25–48, 2009.
6. 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.
7. K. Kersting and L. De Raedt. Towards combining inductive logic programming with bayesian networks. In *Inductive Logic Programming*, pages 118–131. Springer, 2001.
8. D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. The MIT Press, 2009.
9. 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.
10. J. Lehmann. Dl-learner: Learning concepts in description logics. *The Journal of Machine Learning Research*, 10:2639–2642, 2009.
11. J. Lehmann and P. Hitzler. Concept learning in description logics using refinement operators. *Machine Learning*, 78:203–250, 2010.
12. M. Niepert, J. Noessner, and H. Stuckenschmidt. Log-linear description logics. In *IJCAI*, pages 2153–2158, 2011.
13. J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.
14. A. Rettinger, U. Lösch, V. Tresp, C. d'Amato, and N. Fanizzi. Mining the semantic web. *Data Mining and Knowledge Discovery*, pages 1–50, 2012.
15. J. Völker and M. Niepert. Statistical schema induction. *The Semantic Web: Research and Applications*, pages 124–138, 2011.

16. Y. Yang and J. Calmet. Ontobayes: An ontology-driven uncertainty model. In *Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on*, volume 1, pages 457–463. IEEE, 2005.