Knowledge-Based Systems 75 (2015) 30–40

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

TBox learning from incomplete data by inference in BelNet$^+$

Man Zhu [a,b,*], Zhiqiang Gao [a,b], Jeff Z. Pan [c], Yuting Zhao [c], Ying Xu [a,b,d], Zhibin Quan [a,b]

[a] School of Computer Science & Engineering, Southeast University, PR China
[b] Key Laboratory of Computer Network and Information Integration, Southeast University, Ministry of Education, PR China
[c] Department of Computer Science, The University of Aberdeen, UK
[d] Faculty of Information Technology, Monash University, Australia

ARTICLE INFO

Article history:
Received 14 March 2014
Received in revised form 5 November 2014
Accepted 8 November 2014
Available online 18 November 2014

Keywords:
Ontology learning
TBox learning
Probabilistic description logics
Semantic web
Evaluation framework

ABSTRACT

In this work we deal with the problem of TBox learning from *incomplete* semantic web data. TBox, or conceptual schema, is the backbone of a Description Logic (DL) ontology, but is always difficult to obtain. Existing approaches either fail in getting correct results under incompleteness or learn results that are not enough to resolve the incompleteness. We propose to transform TBox learning in DL into inference in the extension of Bayesian Description Logic Network (abbreviated as BelNet$^+$), whereby the structure in the data is leveraged when evaluating the relationships between two concepts. BelNet$^+$, integrating the probabilistic inference capability of Bayesian Networks with the logical formalism of DL ontologies – Description Logics, supports promising inference. In this paper, we firstly explain the details of BelNet$^+$ and introduce a TBox learning approach based on BelNet$^+$. In order to overcome the drawbacks of current evaluation metrics, we then propose a novel evaluation framework conforming to the Open World Assumption (OWA) generally made in the semantic web. Finally the results from empirical studies on comparisons with the state-of-the-art TBox learners verify the effectiveness of our approach.

1. Introduction

Ontologies are basic building blocks of the semantic web [12,25]. The number of semantic web datasets has approximately doubled since 2011, and it has grown by 270% if the category social networking is taken into account [27]. However, the knowledge acquisition bottleneck has resulted in inexpressive schemata (also known as TBoxes, while the data part of ontologies are called ABoxes) on the semantic web [2,13].

One way of enriching TBoxes is to (semi-) automatically learn TBoxes, which has been seen in learning from unstructured documents [2] and semi-structured documents [22]. Given the fast development of semantic data, one way of exploiting them [30,7] is to learning TBoxes from semantic data (ABoxes) [32]. However, in an environment like the semantic web, data generally suffer from incompleteness issue [33], which consequently hinders the learners from getting correct results. In this paper, we focus on learning TBox from incomplete ABox data.

This problem has attracted attention from both machine learning and data mining domains. For example, a number of studies have applied Inductive Logic Programming (ILP) to learning Description Logic (DL) knowledge bases. Lehmann et al. [17] extensively studied the properties of $\mathcal{ALC}$ (a basic DL) and $\mathcal{EL}$ (a light-weight language) refinement operators, which were used in the ILP algorithm. Since the refinement operators are designed to traverse the possible candidates, the approach is effective over complete data. However, the candidate scores are based on both positive and negative examples by making closed world assumption (CWA) – assuming true of the specified and derivable statements, and false otherwise – which, under the incomplete semantic web data, leads to lots of noisy negative examples. Consequently, a candidate concept that best describes the other one but is over specialized will be selected at last. For example, one might learn an axiom $Grandson \sqsubseteq Male \sqcap \neg\ Person$ (*Grandson* is a *Male* who is not a *Person*) from a dataset without statements like "individual grandsons are person". In the data mining domain, Völker and Niepert [28] used association rule mining to learn TBox from semantic web knowledge base such as DBpedia. The measures used to select candidate TBox axioms are support and confidence, where negative examples are out of consideration but are undoubtedly useful in specializing the axioms and decreasing the redundancies in the results. Furthermore, TBox axioms are learned for respective and independent targets, which leads to either over or under specialized result sets. Lastly, the metrics precision, recall, and F1-measure commonly used by current approaches are sensitive to minor changes in the gold standard ontologies. For example, consider a set containing "$Father \sqsubseteq Male$".

* Corresponding author at: School of Computer Science & Engineering, Southeast University, PR China.

http://dx.doi.org/10.1016/j.knosys.2014.11.004
0950-7051/

Replacing "*Father* $\sqsubseteq$ *Male*" and "*Father* $\sqsubseteq \exists$*hasChild*.$\top$" with "*Father* $\sqsubseteq$ Male $\sqcap \exists$hasChild.$\top$" will decrease recall from 1/2 to 0. To summarize, the problem of learning TBox from incomplete semantic web data remains challenging because:

- Little attention is paid to approaches dealing with the incompleteness in the data.
- An evaluation framework to compare existing approaches is lacking.

In order to address those challenges we make the following four contributions in this paper.

- We generate the negative examples according to CWA in a manner similar to [17]. However, to solve the noisy issue brought by CWA and incompleteness, we adopt an approach that instead of considering the instances of concept pairs only, uses inference in a Bayesian network that leverages the structure in the data.
- In order to foster promising inference on subsumption and disjointness axioms, we extend BelNet [33] to BelNet$^+$. BelNet combines Bayesian networks with DLs by representing DL concepts as nodes and subsumptions with links. In BelNet$^+$, we extend the semantics of links in BelNet by using additional links for disjointness. Compared to BelNet, BelNet$^+$ is more effective in detecting disjoint concepts and answering queries.
- We consider the TBox learning as instance classification. In order to conform to the Open World Assumption (OWA) generally made in the semantic web, we extend the traditional confusion matrix by considering unknown results (neither true nor false), and propose the metrics using the new confusion matrix correspondingly. Our extension of traditional evaluation metrics reflects more objectively on the performance of the learners.
- In order to evaluate the state of the art TBox learners, we set up gold standard ontologies correspondingly. Meanwhile, in our evaluation framework, the quality of the gold standard ontologies is more easily guaranteed.

The rest of the paper is organized as follows. In Section 2, notions in DLs are introduced. In Section 3, we introduce the proposed model, BelNet$^+$, in detail. Then we describe the TBox learning approach in Section 4. In Section 5, we describe the proposed evaluation framework for TBox learners. The empirical performance evaluations are shown in Section 6. We briefly review the related work in Section 7. Section 8 presents conclusions drawn from the work and identifies areas for future research.

2. Ontology & description logic

An ontology comprises TBox (*terminology*, *i.e.*, the vocabulary of an specific domain) and ABox (*assertions*). TBox consists of concepts denoting sets of individuals, and roles denoting binary relationships between individuals. We denote the set of concept names by N_C, the set of all concept expressions by N_C^+, and 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. ABox contains two sets of assertions. One is the set of concept assertions such as *Grandson(Mathiew)*, and the other is the set of role assertions between individuals such as *hasChild(Paul, Mathiew)*. The assertions in the ABox are also called *facts*.

Description Logics (DLs) provide the logical formalism for ontologies in the semantic web. Table 1 shows the syntax and semantics of a specific DL language $\mathcal{ALC}$. In DLs, interpretations are used to assign meanings to syntactic constructs. An *interpretation* $\mathcal{I}$ consists of a non-empty set $\Delta^{\mathcal{I}}$. An *interpretation function* $\cdot^{\mathcal{I}}$ assigns to each object $a \in N_I$ an element of $\Delta^{\mathcal{I}}$, to each atomic concept $A \in N_C$ a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and to each atomic role $r \in N_R$ a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. An interpretation $\mathcal{I}$ *satisfies* a subsumption axiom $C \sqsubseteq D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, if it satisfies an equality $C \equiv D$ if $C^{\mathcal{I}} = D^{\mathcal{I}}$, and it satisfies a disjointness axiom $C \sqcap D \sqsubseteq \bot$ if $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$. If $\mathcal{T}$ is a set of axioms, then $\mathcal{I}$ satisfies $\mathcal{T}$ iff $\mathcal{I}$ satisfies each element of $\mathcal{T}$. If $\mathcal{I}$ satisfies an axiom (resp. a set of axioms), then we say that it is a *model* of this axiom (resp. the set of axioms). An ABox $\mathcal{A}$ is *consistent* with respect to a TBox $\mathcal{T}$, if there is an interpretation that is a model of both $\mathcal{A}$ and $\mathcal{T}$ [1]. A concept C is *satisfiable* with respect to $\mathcal{T}$ if there exists a model $\mathcal{I}$ of $\mathcal{T}$ such that $C^{\mathcal{I}}$ is nonempty. An ontology O is *incoherent* iff there exists an unsatisfiable named concept in O [6]. Please refer to [12] for further details of DLs.

Table 1
Syntax and semantics of DL $\mathcal{ALC}$.

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	$\forall r \cdot C$	$(\forall r \cdot C)^{\mathcal{I}} = \{a \mid \forall b \cdot (a,b) \in r^{\mathcal{I}} \text{ implies } b \in C^{\mathcal{I}}\}$
Disjunction	$C \sqcup D$	$(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$
Negation	$\neg C$	$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
Existential restriction	$\exists r \cdot C$	$(\exists r \cdot C)^{\mathcal{I}} = \{a \mid \exists b \cdot (a,b) \in r^{\mathcal{I}} \text{ and } b \in C^{\mathcal{I}}\}$

For an ontology O, its *ABox materialization* is $M(O) = \{C(a) \mid C \in N_C^+, a \in N_I, O \models C(a)\}$, where $O \models C(a)$ if every interpretation that satisfies O also satisfies $C(a)$. If the ABox materialization of an ontology O contains the same set of ABox assertions as in O then we say that O is *ABox materialized*. Moreover, the *parent set* (parents) of $C \in N_C^+$ is defined as $Pa(C) = \{C' \in N_C^+ \mid O \models C' \sqsubseteq C$, and there is no C'' such that $O \models C' \sqsubseteq C''$, and $O \models C'' \sqsubseteq C\}$, where $O \models C' \sqsubseteq C''$, if every model of O also satisfies $C' \sqsubseteq C''$.

3. BelNet$^+$

BelNet$^+$ integrates DLs with Bayesian networks. The syntax of BelNet$^+$ is defined by the subsumption and disjointness axioms in DLs. The semantics is defined by a probability distribution over a Bayesian network.

3.1. Syntax

Definition 1. A *Bayesian subsumption axiom* is in the form of $D|C_1, \ldots, C_n$, where $C_i \sqsubseteq D$, $C_i \not\equiv \bot$, $D \not\equiv \bot$, $i \in \{1, \ldots, n\}$ and $\nexists j,\ k \in \{1, \ldots, n\}$ such that $C_j \sqsubseteq C_k$. If $D' \equiv D$, then label D with an *alias* D'.

Definition 2. A *Bayesian disjoint axiom* is in the form of $D|\overline{C}$, where $C \sqcap D \sqsubseteq \bot$ and $C \not\equiv \bot$, $D \not\equiv \bot$.

Definition 3. A BelNet$^+$ contains a set of Bayesian subsumption axioms ($D|C_1, \ldots, C_n$) and a set of Bayesian disjoint axioms ($D|\overline{C}$), together with an ontology ABox. A BelNet$^+$ defines a Bayesian network $\mathcal{B}$ as follows:

- $\mathcal{B}$ contains one binary node associated with a conditional probability table (CPT) calculated from the ABox for each C_i and D appearing in either the Bayesian subsumption axioms or the Bayesian disjoint axioms.
- $\mathcal{B}$ links from node C_i to node D for each Bayesian subsumption axiom $D|C_1, \ldots, C_n,\ i \in \{1, \ldots, n\}$.

- There is a link between node C and node D in each Bayesian disjoint axiom, and the direction is found by Algorithm 1.

Links in a BelNet$^+$ can be *conditional*, which means that the assignments of one node fully determine that of the other one. Fig. 1 shows an example of conditional links. With the assignments for variables *Female* and *Male*, we know for sure the assignment of $Female \sqcup Male$ by the semantics of DLs.

For convenience, in this paper we use the same symbol for both the concept in DL ontology and the corresponding node in the Bayesian network.

Example 1. (Given an ontology containing TBox){$Male \sqsubseteq Person$, $Female \sqsubseteq Person$, $Male \sqcap Female \sqsubseteq \bot$}, the corresponding BelNet$^+$ contains the following Bayesian subsumption axioms and Bayesian disjoint axioms:

$Person|Male, Female$

$Male|\overline{Female}$

this BelNet$^+$ specifies a Bayesian network structure as shown in Fig. 2, where the CPTs are learned by parameter learning (Section 3.3).

In the following, we will prove that a BelNet$^+$ is guaranteed to define a Bayesian network as a directed acyclic graph (DAG).

We define the subgraph of $\mathcal{B}$ with links for subsumption axioms as $\mathcal{B}^{\rightarrow}$.

Proposition 1. *$\mathcal{B}^{\rightarrow}$ is a DAG.*

Proof. Suppose there is a directed cycle in $\mathcal{B}^{\rightarrow}$, say $C_1 \rightarrow \cdots \rightarrow C_n \rightarrow C_1$, which suggests that $C_1 \sqsubseteq C_2 \ldots C_{n-1} \sqsubseteq C_n, C_n \sqsubseteq C_1$, then $C_1 \equiv \cdots \equiv C_n$. This conflicts with the rule of Bayesian subsumption axioms that equivalent concepts are represented as alias. □

If we denote the node, say C, in $\mathcal{B}^{\rightarrow}$ whose indegree ($d^-_{\mathcal{B}}(C)$) or outdegree ($d^+_{\mathcal{B}}(C)$) equals to 0 as *terminal node*, we have the following propositions:

Proposition 2. *$\mathcal{B}^{\rightarrow}$ contains at least one terminal node.*

Proof. Suppose each node in $\mathcal{B}^{\rightarrow}$ has at least a pair of incoming and outcoming edges. Let $P := C_1 \rightarrow \cdots \rightarrow C_n$ be the longest path in $\mathcal{B}^{\rightarrow}$. Suppose the outcoming edge of C_n links to C. If C is not on P, the path $C_1 \rightarrow \ldots C_n \rightarrow C$ is longer than P, which is a contradiction. Therefore, $C = C_i$, for some $i, i \in \{1, \ldots, n\}$, and $C_i \rightarrow \cdots \rightarrow C_n \rightarrow C_i$ forms a cycle in $\mathcal{B}^{\rightarrow}$, which conflicts with Proposition 1. □

If we denote the subgraph of $\mathcal{B}^{\rightarrow}$ by deleting a node C in $\mathcal{B}^{\rightarrow}$ whose indegree or outdegree is 0, and also the links connected with C as $\mathcal{B}^{\rightarrow} \setminus C$, the following proposition holds:

Proposition 3. *$\mathcal{B}^{\rightarrow} \setminus C$ contains at least one terminal node.*

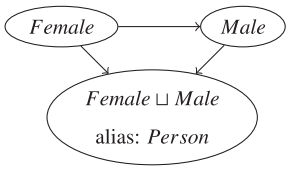
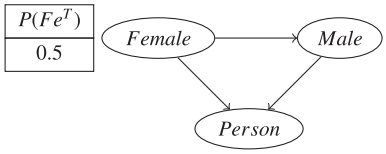


Fig. 1. A BelNet$^+$ example. The links from *Male* and *Female* to $Female \sqcup Male$ are conditional.



Fe	$P(Ma^T\|Fe)$
T	0.25
F	0.75

Fe	Ma	$P(Pe^T\|Fe, Ma)$
T	T	0.50
T	F	0.75
F	T	0.75
F	F	0.50

Fig. 2. A motivated BelNet$^+$ example. *Ma*, *Fe*, and *Pe* are short for *Male*, *Female*, and *Person*. *T* and *F* are short for value `TRUE` and `FALSE`.

Proposition 3 is easy to be followed after we prove Propositions 1 and 2. Because every subgraph of $\mathcal{B}^{\rightarrow}$ is also a DAG, then for every subgraph of $\mathcal{B}^{\rightarrow}$, Proposition 3 also holds.

Theorem 1. *At least one link direction assignments for Bayesian disjoint axioms exist, such that $\mathcal{B}$ is a DAG.*

Proof. We denote the subgraph of $\mathcal{B}$ with all nodes and the undirected edges for Bayesian disjoint axioms as $\mathcal{B}^{-\rightarrow}$. Algorithm 1 finds at least one edge assignments. Algorithm 1 will terminate, because each time in the first while loop, the total number of nodes will be decreased by 1 in $\mathcal{B}^{\rightarrow}$ and $\mathcal{B}^{-\rightarrow}$, and in the second, $\mathcal{B}^{-\rightarrow}$ will have 1 less node. Line 4, 6 and 10 guarantee C to be a terminal node, and it is impossible that a loop in $\mathcal{B}$ goes through C. At last $\mathcal{B}'$ will be guaranteed to be a DAG. Fig. 3 depicts an example of this procedure. At first, in (1), the inputs are $\mathcal{B}^{-\rightarrow}$ and $\mathcal{B}^{\rightarrow}$, and $\mathcal{B}$ is initialized with all nodes, and the links in $\mathcal{B}^{\rightarrow}$. After removing terminal nodes in $\mathcal{B}^{\rightarrow}$ and adding links to $\mathcal{B}$, we get the three graphs in (4) which corresponds to the end of the first while loop in Algorithm 1 on line 7. In the second while loop, all terminal nodes in $\mathcal{B}^{-\rightarrow}$ are removed, and the final $\mathcal{B}$ is shown in (6) (Fig. 3). □

3.2. Semantics

The semantics of a BelNet$^+$ is based on joint probability distributions over the Bayesian network generated:

$$P(\mathcal{B}) = P(C_1, \ldots, C_n) = \prod_{i=1}^{n} P(C_i|Pa(C_i)) \quad (1)$$

where C_is are nodes in $\mathcal{B}$, and $Pa(C_i)$ is the parent set of C_i.

Example 2. By calculating from the BelNet$^+$ shown in Fig. 2, the joint probability of the existence of an instance who is a Female, a Male, and a Person is $0.5 \times 0.25 \times 0.50 = 0.0625$. The probability of an instance who is a Female and also a Person is $0.5 \times 0.25 \times 0.75 + 0.5 \times 0.75 \times 0.75 = 0.375$. In this example ($P(Person^F|Female^T, Male^F) = 0.25$), the probabilities calculated still suggest that the first instance is less probable.

A BelNet$^+$ can be viewed as a template for generating ABoxes. Given different sets of conditional probability tables (or CPTs), or different set of Bayesian axioms, it can generate different ABoxes.

3.3. Parameter estimation

The parameters in a BelNet$^+$ refer to the CPT in the Bayesian network defined by it. In this part, we will discuss how the parameters can be learned from semantic web data.

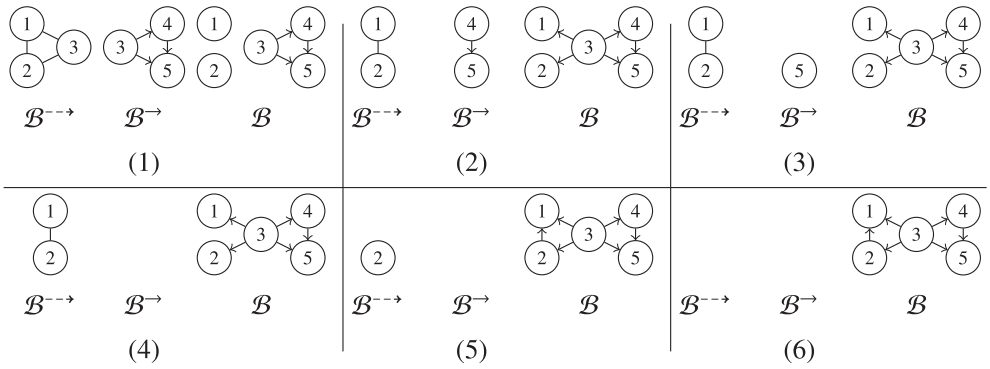


Fig. 3. An example of finding edge assignment to ensure the DAG property of the Bayesian network generated.

It is natural to use a finite ontology domain $\Delta^{\mathcal{I}}$ to restrict all individuals in the ABox of a BelNet$^+$. We assume that $\Delta^{\mathcal{I}}$ contains all individual names in the BelNet$^+$, and an individual name o is always interpreted to itself, *i.e.*, $o^{\mathcal{I}} = o$.

We call all interpretations related to individual o a *possible observation* $\mathbf{o}$. For example, $C^{\mathcal{I}} = \{a, b\}$, then there are two possible observations, where $C^{\mathbf{o_1}} = \{a\}$, $C^{\mathbf{o_2}} = \{b\}$. A possible observation is an interpretation which assigns at most one element to one concept. Actually, under a possible observation, C_i has two values: $\mathtt{T}$ and $\mathtt{F}$. For a specific observation $\mathbf{o}$, $\mathbf{o}$ supports $C_i^{\mathtt{T}}$ if $o \in C_i^{\mathcal{I}}$, and $\mathbf{o}$ supports $C_i^{\mathtt{F}}$ if $o \notin C_i^{\mathcal{I}}$. These cases can be abbreviated as $C_i^{\mathbf{o}}$. $\mathcal{B}^{\mathbf{o}}$ is short for $\{C_1^{\mathbf{o}}, \ldots, C_n^{\mathbf{o}}\}$, where C_i, $i \in \{1, \ldots, n\}$ is a node in $\mathcal{B}$.

Algorithm 1. Finding link directions for $\mathcal{B}$

input: $\mathcal{B}^{\dashrightarrow}$, subgraph of $\mathcal{B}$ with undirected edges for disjoint axioms
$\mathcal{B}^{\rightarrow}$, subgraph of $\mathcal{B}$ with links for subsumption axioms
output: $\mathcal{B}'$

```
Initialize B' with nodes in B^→ and B^⇢, and edges in B^→;
while there exists a terminal node C in B^→ do
    if d^-_B(C) is 0 then
        add links from C to all C's neighbors in B^⇢ to B';
    else
        add links from all C's neighbors to C in B^⇢ to B';
    delete C and the edges connected with C from B^→ and B^⇢;
while B^⇢ is not empty do
    C ← a node in B^⇢;
    add links from all neighbors to C in B^⇢ to B';
    delete C and the edges connected with C from B^⇢;
return B';
```

For a *marginal node* C, which has no parents in $\mathcal{B}$, the *marginal probability* is a table of $P(C^{\sharp})$, where $\sharp \in \{\mathtt{TRUE}, \mathtt{FALSE}\}$. Furthermore, $P(C^{\mathtt{TRUE}})$ is the probability that a possible observation supports C, *i.e.*, $P(o \in C^{\mathcal{I}})$. Similarly $P(C^{\mathtt{FALSE}})$ is the probability that a possible observation does not support C, *i.e.*, $P(o \notin C^{\mathcal{I}})$. Actually the parameters depends on the number of individuals satisfying concept C in the ontology, as we will see below. For convenience, in the following $\mathtt{TRUE}(/\mathtt{FALSE})$ is shortened to be $\mathtt{T}(/\mathtt{F})$.

The CPTs will be learned from the ontology ABox. We assume that all possible observations are independent. By Eq. (1). the likelihood of all possible observations $\{\mathbf{o}\}$ is

$$L(\theta : \{\mathbf{o}\}) = \prod_{\mathbf{o}} \prod_{i=1}^{n} \theta_{C_i^{\mathbf{o}}|Pa(C_i)^{\mathbf{o}}} = \prod_{i=1}^{n} \theta_{C_i^{\mathbf{o}}|Pa(C_i)^{\mathbf{o}}}^{N[C_i^{\mathbf{o}}|Pa(C_i)^{\mathbf{o}}]} \tag{2}$$

where θ denotes the set of CPT values, and $N[C_i^{\mathbf{o}}|Pa(C_i)^{\mathbf{o}}]$ is the number of possible observations satisfying $C_i^{\mathbf{o}}|Pa(C_i)^{\mathbf{o}}$. Maximizing this likelihood by setting the derivative of the log-likelihood of Eq. (2) with respect to its CPTs to 0 results in

$$\theta_{C_i^{\mathbf{o}}|Pa(C_i)^{\mathbf{o}}} = \frac{N[C_i^{\mathbf{o}}|Pa(C_i)^{\mathbf{o}}]}{N[Pa(C_i)^{\mathbf{o}}]} \tag{3}$$

In order to avoid the cases where $N[Pa(C_i)^{\mathbf{o}}] = 0$, we add one "imaginary" possible observation to it.

Example 3. Given the BelNet$^+$ in Example 1. In addition, we also have an ABox:

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

Then the estimation of $\theta_{Person^{\mathtt{T}}|Female^{\mathtt{T}},Male^{\mathtt{F}}}$ is $\frac{1+0.5}{1+1} = 0.75$.

The learned CPTs are shown in Fig. 2.

3.4. Inference

BelNet$^+$ can answer arbitrary probability query: "Given a BelNet$^+$, what is the probability of a Bayesian subsumption/disjoint axiom?" More formally, the conditional probability query is given by

$$P(D|C_1, \ldots, C_n) = \frac{P\left(\sqcup_{i=1}^{n} C_i^T, D^T\right)}{P\left(\sqcup_{i=1}^{n} C_i^T\right)} \tag{4}$$

and

$$P(D|\overline{C}) = 1 - P(D^T, C^T) \tag{5}$$

Eqs. (4) and (5) can be calculated by joint probabilities over the Bayesian networks. Joint probability queries can be answered in Bayesian networks. In our work, we pay less attention to networks with a large tree-width. The complexity of exact inference algorithm – junction tree algorithm – is exponential to the tree-width of the networks, which is acceptable. In our implementation, we use junction tree algorithm [9] to do the task.

3.5. Structure learning

Structure learning is a specific type of knowledge discovery that learns a dependency structure, being able to give promising answers to queries "what is the probability of a Bayesian subsumption/disjoint axiom?". So the task of *structure learning* in BelNet$^+$ is to find a BelNet$^+$ $\mathcal{B}$ that makes the data the most probable. This is similar to the task of structure learning in Bayesian networks except that the structure we learn needs to be a BelNet$^+$. In other words, the links in the structure need to be corresponded to subsumption or disjoint relationships. If we denote the candidate structures in a domain as $\mathcal{B}^+$, and that of the same domain in

Bayesian networks as $\mathcal{B}$, we have $\mathcal{B}^+ \subseteq \mathcal{B}$. Thus, we can share the structure scores from that in Bayesian networks structure learning.

Structure Score. Choices for score functions used in Bayesian network structure learning include maximum likelihood, Bayesian score that is based on a Bayesian perspective encoding uncertainties both over structure and over parameters, and extensions of Bayesian score. Likelihood measurement suffers from overfitting, and prefers more complex networks to simpler ones, which is not always the preference in practice. For handling over-fitting problems and more efficient numerical computation of the Bayesian score [15], we will adopt the decomposable Bayesian score with Dirichlet priors as our score function.

Algorithm 2. Structure learning in BelNet$^+$

input: Bayesian network $\mathcal{B} =< V, E_{conditional} >, \mathcal{M} =< C, Inst_C >, max\_iter$
output: $\mathcal{B}'$
1 Initialize *best_score* with the score of $\mathcal{B}$;
2 **for** *each pair of nodes* **do**
3 | *cache* the score for adding/deleting/reversing the link between them;
4 **while** *max_iter* not reached **do**
5 | **while** *best operation not found and cache not completely traversed* **do**
6 | | *op* ← the best operation from the cache;
7 | | **if** *op satisfies the* selection criteria **then**
8 | | | best operation found;
9 | **if** *best operation found and new_score* $\geqslant$ *best_score* **then**
10 | | do operation *op*, and label the network as $\mathcal{B}'$
11 | | *best_score* ← score of $\mathcal{B}'$;
12 | **else**
13 | | **return** $\mathcal{B}'$;

Structure Search. We knew from literature that "Given a dataset $\mathcal{D}$ and a decomposable score function, finding $\mathcal{G}^* = \arg\max_{\mathcal{G}\in\mathcal{G}_d} score(\mathcal{G} : \mathcal{D})$ is $\mathcal{NP}$-hard." [15]. The BelNet$^+$ structure would additionally have the property that the links correspond to subsumption or disjoint relationships. Thus, instead of aiming for an algorithm that will always find the highest-scoring network, we resort to heuristic algorithms that attempt to find the best network but are not guaranteed to do so. The algorithm adopted here is a modified version of the structure learning algorithm in Bayesian networks [23]. The Bayesian network structure learning algorithm can only recover the structure that is equivalent in terms of representing the independencies among the nodes to the real underlying structure [15]. In this paper, the preference is a single structure that is concise and can directly be used to extract axioms. To achieve this goal, we incorporate this preference in Algorithm 2.

Roughly speaking, the structure learning algorithm starts from an initial structure (with nodes, and the conditional links between nodes), and iteratively tries to find the best operation (in terms of adding/deleting/reversing) that can be carried out from the current structure, unlike in [33]. This process iterates until no better structure can be found, or the step reaches the maximum threshold (c.f. Algorithm 2). Two thresholds are involved in this procedure. One controls the maximum number of parent nodes a node can have, the other controls the maximum number of iterations for this procedure to exit.

Selection criteria. After an operation is selected by the score function, in order to meet the demand of BelNet$^+$, to be specific, preference is given to structures whose links signify the special dependencies "subsumption" and "disjointness", different from [33]. The operations not satisfying the requirements are filtered out by the selection criteria.

We denote the candidate operation as *op*, where op_{head} is the node to which the link points, and op_{tail} represents the node from which the link starts. Further, we denote the count of instances belonging to concepts op_{tail} and op_{head} as $M[op_{head}, op_{tail}]$, the count of instances belonging to concept op_{head} as $M[op_{head}]$, similar for $M[op_{tail}]$. Then, *operation op will be selected iff either* $M[op_{head}, op_{tail}] = M[op_{tail}]$ *and* $M[op_{tail}] > threshold_{parent}$ *or* $M[op_{head}, op_{tail}] = 0$ *and* $M[op_{head}] \neq 0$ and $M[op_{tail}] \neq 0$. In this paper, the $threshold_{parent}$ is 0.

It happens that some concepts in the ontology contain a large number of missing values. Those corresponding nodes are out of consideration in the post-processing step. The rest of nodes are called *informative nodes*.

Algorithm 3 shows how the post-processing step works. In Algorithm 3, besides Bayesian disjoint axioms, which are considered in [33] in a smaller scale, the candidate Bayesian subsumption axioms are also generated by inference over B, and the results of the inference can be considered as weights of the candidates. In practice, in order to select the axioms from the weighted results, we use thresholds. Since the Bayesian network constructed can behave differently for Bayesian subsumption axioms and Bayesian disjoint axioms, we use $\text{threshold}_{subsumption}$ and $\text{threshold}_{disjoint}$ respectively for the selection.

Although the results from Algorithm 3 look quite simple, such as relations between pair of concepts generated from pre-processing, these are however the basis for more complex axioms, as shown below.

- If there is more than one Bayesian subsumption axiom $\{D_1|C, D_2|C, \ldots, D_n|C\}$, generate $C \sqsubseteq \sqcap_{i\in\{1,\ldots,n\}} D_i$.
- If there is more than one Bayesian subsumption axiom $\{D|C_1, D|C_2, \ldots, D|C_n\}$, generate $\sqcup_{i\in\{1,\ldots,n\}} C_i \sqsubseteq D$.
- Bayesian disjoint axioms correspond to disjoint axioms in ontologies.

4. TBox learning as inference

After describing the details of BelNet$^+$, we will introduce how the TBox can be learned with BelNet$^+$. The learning approach includes three main steps:

(1) *Pre-processing.* In pre-processing, given an ontology O, for each $C \in N_C^+$ and $r \in N_R$, pre-processing creates nodes corresponding to C and $\exists r.\top$. Conditional links are added among the nodes. Furthermore, ABox materialization will be carried out on each node generated in this step. We denote the ABox materialized ontology as O^+. The result of this step is denoted by $\mathcal{B}^0$.
(2) *Learning Bayesian network.* Structure learning (c.f. Section 3.5) will be carried out on $\mathcal{B}^0$ over O^+. After that, parameter learning will fill the CPTs attached with the structure learned. We denote the result of this step as $\mathcal{B}$.
(3) *Post-processing.* Having a Bayesian network learned, TBox axioms are extracted through inference over $\mathcal{B}$. See below for details.

Algorithm 3. Post-processing in BelNet$^+$

input: $\mathcal{B}$, threshold$_{disjoint}$, threshold$_{subsumption}$
output: $\mathcal{O}'$
1 Initialize an empty list named *axiomlist* and an ontology $\mathcal{O}'$;
2 **for** *any two different informative nodes* C_i *and* C_j **do**
3 **if** $P(C_i|C_j) > threshold_{subsumption}$ **then**
4 add $(< C_i|C_j >, P(C_i \sqsubseteq C_j)) \rightarrow axiomlist$;
5 **if** $P(C_i|\overline{C_j}) > threshold_{disjoint}$ **then**
6 add $(< C_i|\overline{C_j} >, P(C_i|\overline{C_j})) \rightarrow axiomlist$;
7 sort *axiomlist* ASC according to the probabilities;
8 **for** *each axiom in axiomlist* **do**
9 **if** *adding axiom to* $\mathcal{O}'$ *not causing inconsistency* **then**
10 add *axiom* $\rightarrow \mathcal{O}'$;
11 **return** $\mathcal{O}'$;

5. A novel evaluation framework

The set of axioms learned by TBox learning systems can be viewed as an application of information retrieval on a knowledge base. From this perspective, the performance of a TBox learning system can either be judged by human experts or be evaluated by traditional IR measures. Using traditional IR measures, an axiom learned is *correct* if it can be entailed by the gold standard ontology. However, both methodologies suffer from disadvantages: human experts are subjective to some extent, and there are various representations for a domain, consequently, the evaluations by IR measures are sensitive to gold standard ontologies.

From another perspective, the TBox in an ontology assists classifying instances with DL reasoners. Although it is impossible to explicitly make all true statements of the interested domain, it is still workable to get as many facts as possible through reasoning. In this way, the TBox can be viewed as a set of classification "rules" to classify the instances. Based on this observation, we extend metrics used in classification.

Below we firstly introduce the notations we will use in the evaluation framework.

Notations. We denote the original ontology (the input of ontology learners) as $\mathcal{O}$, and the output as $\mathcal{O}'$. Furthermore, the gold standard ontology is denoted by $\mathcal{O}^S$.

Definition 4 (*Gold standard ontology*). An ontology $\mathcal{O}^S$ is called a gold standard ontology for $\mathcal{O}$, if $\mathcal{O}^S$ satisfies:

- $\mathcal{O}^S$ is both consistent and coherent.
- $\mathcal{O}^S$ entails all correct (with respect to the knowledge of domain experts) ABox statements with the vocabulary of its ontology counterpart $\mathcal{O}$.

Property 1. *The gold standard ontology* $\mathcal{O}^S$ *of an ontology* $\mathcal{O}$ *can be non-unique.*

Property 1 is straight forward. A gold standard ontology for $\mathcal{O}$ can be the one with ABox knowledge not explicitly stated but inferred. In the extreme case, another gold standard ontology for $\mathcal{O}$ may explicitly state all ABox statements.

If we view the TBox as a set of classification rules, the result of classifying an instance a towards a concept A with respect to an ontology $\mathcal{O}$ is

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

In order to incorporate the *unknown* values in the classification results, we extend the traditional confusion matrix used in the evaluation of binary classification [10] by considering "unknown" as a specific classification result (c.f. Table 2).

With this extension in hand, several classical metrics used by classification problems are (extended) as follows:

$$Accuracy(U) = \frac{TP + TN + w \cdot TU}{P_C + N_C + w \cdot U_C}$$
$$ErrorRate(U) = 1 - Accuracy(U)$$
$$Precision(U) = \frac{TP}{TP + FP(N) + w \cdot FP(U)}$$
$$Recall(U) = \frac{TP}{TP + FN(P) + w \cdot FU(P)}$$
$$F - Measure(U) = \frac{(1+\beta)^2 \cdot Recall(U) \cdot Precision(U)}{\beta^2 \cdot Recall(U) + Precision(U)}$$
$$TP\_rate = \frac{TP}{P_C}$$
$$FP(N)\_rate(U) = \frac{FP(N)}{N_C}$$
$$FP\_rate(U) = \frac{FP(U) + FP(N)}{N_C}$$

Traditional *Accuracy*, *ErrorRate*, *Precision*, *Recall* and *F-Measure* are calculated from the extended metrics when w is 0. Traditional ROC graph is formed by plotting *TP_rate* over $FP(N)\_rate(U)$.

We demonstrate the necessity of this confusion matrix extension by Example 4:

Example 4. Table 3 shows 3 ontologies. The first one is the gold standard ontology, $\mathcal{O}_1$ and $\mathcal{O}_2$ are two ontologies to be evaluated. If we calculate the accuracy for classifying concept Female, then using the traditional confusion matrix,

$$Accuracy(\mathcal{O}_1, \text{Female}, \mathcal{O}^S) = Accuracy(\mathcal{O}_2, \text{Female}, \mathcal{O}^S) = \frac{2+1}{4}$$

but apparently $\mathcal{O}_2$ contains one incorrect subsumption axiom. In the new framework, if $w = 1$, then

Table 2
The extended confusion matrix. *T* and *F* are short for *True* and *False*. *P*, *N* and *U* are short for *Positive*, *Negative* and *Unknown* respectively. *FP*(*N*) is short for *False Positives from Negatives* (set of positive results which should be labeled as negatives).

	P_C	N_C	U_C
P	*TP*	*FP*(*N*)	*FP*(*U*)
N	*FN*(*P*)	*TN*	*FN*(*U*)
U	*FU*(*P*)	*FU*(*N*)	*TU*

Table 3
An example of gold standard ontology and test ontologies. All ontologies have the same set of concept names: Female, Male, Mother, Daughter, and Child.

	$\mathcal{O}^S$	$\mathcal{O}_1$	$\mathcal{O}_2$
TBox	Female ⊓ Male ⊑ ⊥	Mother ⊑ Female Daughter ⊑ Female Female ⊓ Male ⊑ ⊥	Mother ⊑ Female Daughter ⊑ Female Female ⊓ Male ⊑ ⊥ Child ⊑ Daughter
ABox	Female(a), Female(b) Male(c), Child(d)	Mother(a), Daughter(b) Male(c), Child(d)	Mother(a), Daughter(b) Male(c), Child(d)

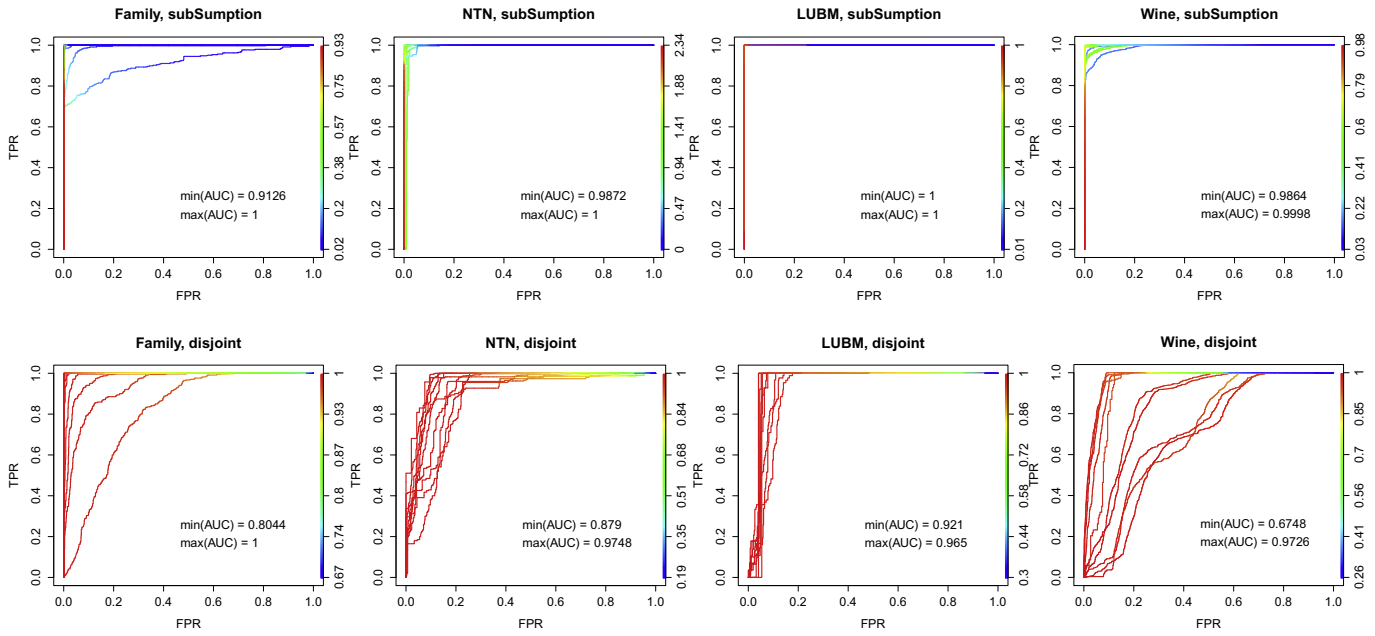


Fig. 4. ROC (Receiver Operating Characteristics) curves for each dataset, varying the size of the dataset (10–100%). AUC is short for "Area Under ROC Curve".

$$Accuracy(O_1, \text{Female}, \mathcal{O}^S) = \frac{2+1+1}{4}$$

$$Accuracy(O_2, \text{Female}, \mathcal{O}^S) = \frac{2+1}{4}$$

The above measures are used in binary classification evaluations. When evaluating multi-class problem, we simply use an average (weighted) of the above measures. Suppose the importance of the concepts are ranked with weights $w_1, \ldots, w_n$, then the average (weighted) value of a specific measure is

$$\frac{\sum_{i=1}^{n} w_i \cdot Measure(A_i)}{\sum_{i=1}^{n} w_i}$$

where *Measure* can be replaced by any of the (extended) metrics listed above. A_i denotes the i-th concept.

Property 2. *The (extended) metrics listed above are the same in all gold standard ontologies* $\mathcal{O}^S$.

Property 2 holds because the (extended) metrics are calculated by the extended confusion matrix (Table 2), and according to the definition of gold standard ontologies, the confusion matrix is the same in all gold standard ontologies.

Property 2 indicates the stability of a gold standard ontology. In other words, the variations in gold standard ontologies have no influence on the evaluation results.

6. Experiments

We have implemented a prototype of BelNet$^+$ with the TBox learning algorithm in Java and Scala. We designed and carried out the experiments to highlight the effect of incompleteness on learning methods. In this section, we evaluate the performance of the proposed learning method focusing on answering the following three questions: (1) How promising is the inference in BelNet$^+$? (2) How are the performance of the four approaches, namely DLLearner, Goldminer, BelNet, and BelNet$^+$, under the existence of incompleteness? (3) Will the amount of incompleteness be decreased with TBox learning?

6.1. Experimental setup

6.1.1. Dataset

The datasets used in the experiments include: Family,[1] Semantic Bible (NTN),[2] LUBM,[3] and Wine.[4] We manually constructed gold standard ontologies for the datasets.[5]

In order to quantify the degree of incompleteness of an ontology O, we denote incompleteness by the percentage of unknown answers to all possible queries in the form "Is individual a an instance of concept A?".

To be specific, the *incompleteness* of an ontology O is quantified by

$$\frac{\{f(a,A,O) | f(a,A,O) \text{ is unknown}\}}{|\{f(a,A,O)\}|}$$

where $a \in N_I$ and $A \in N_C$.

The relevant statistics of the datasets and the corresponding gold standard ontologies are shown in Table 5, in which we calculate the number of named concepts, object properties, number of subClassOf, equivalentClass, disjointWith axioms, number of individuals, the DL expressibility, and the incompleteness of the corresponding ontologies. As is shown in the table, semantic bible, LUBM and Wine contain more incompleteness than that in Family.

It is worth noticing that the DL expressibility of the ontologies chosen is not restricted to certain DL languages. In our approach, all concept expressions in the original ontology are treated as concepts.

To demonstrate the capability of TBox learners in handling incompleteness, we create subontologies of the original ontologies with different levels of incompleteness. We partition the ABox into 10 parts. Then we randomly select one of them, and add it to the TBox as the first subontology. By randomly selecting and adding

[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://kaon2.semanticweb.org.
[5] https://github.com/fresheye/belnet/blob/master/ontology/.

one part to the existing largest subontology each time, we finally get 10 subontologies. This procedure will be carried out 10 times, each of which with a different initial start subontology. In order to clearly demonstrate the performance, the result ontologies only contain axioms learned.

6.1.2. Default values and thresholds

Goldminer consists of 4 tunable parameters, namely support and confidence in learning subsumptions and disjointness separately. We tried parameters in the scope of $[0-1]$ for Goldminer, and finally we chose the support threshold to be 0, and confidence threshold to be 0.9 for learning subsumptions, and 0.1 (support), 0.8 (confidence) for learning disjointness, which is also the setting recommended in [5], in order to get a higher F-measure.

In BelNet$^+$, we tried different combinations of maximum number of parents and maximum number of iterations. We set 5 as the maximum number of parents and 100 as the maximum number of iterations, because the results are almost stable with these settings. In addition, we only learn axioms among the concepts containing at least 10% of individuals. The corresponding concepts for informative nodes contain at least one individual.

To set parameters $\text{threshold}_{disjoint}$ and $\text{threshold}_{subsumption}$, we draw the ROC (Receiver Operating Characteristic) curves for each dataset (c.f. Fig. 4). An axiom is true if it can be entailed by the gold standard ontology, and false if not. We selected the thresholds by setting FPR (False Positive Rate) <0.1 and TPR (True Positive Rate) >0.7. The thresholds selected are shown in Table 4.

6.2. Experimental results

6.2.1. Performance of inference

The first experiment is to demonstrate the effectiveness of the inference in BelNet$^+$. For each of the four datasets, we performed the experiments by conducting two kinds of inference, namely inference for probabilities of Bayesian subsumption axioms and Bayesian disjoint axioms, in the $\mathcal{B}$ learned.

Quality of Inference. We consider the inference results as the output of a binary classifier. By consulting the gold standard ontologies $\mathcal{O}^S$, the correctness of the corresponding axioms can be calculated.

Suppose the ontology learned is $\mathcal{O}$, *precision* and *recall* are calculated as follows:

$$Precision(\mathcal{O}^S, \mathcal{O}') = \frac{|\{\alpha|\alpha \in \mathcal{O}' \text{ and } \mathcal{O}^S \models \alpha\}|}{|\{\alpha|\alpha \in \mathcal{O}'\}|}$$

$$Recall(\mathcal{O}^S, \mathcal{O}') = \frac{|\{\alpha|\alpha \in \mathcal{O}^S \text{ and } \mathcal{O}' \models \alpha\}|}{|\{\alpha|\alpha \in \mathcal{O}^S\}|}$$

where α is a subsumption or disjointness axiom. *F-measure* is the harmonic mean of precision and recall. In Fig. 4, we report the quality of the inference by drawing the ROC curves on each partition of the four datasets. We find that:

- The inference of Bayesian subsumption axioms obtain better results than that of Bayesian disjoint axioms. As we can find from Section 3.7, the probability of a Bayesian subsumption axiom is a normalized measure. However, the probabilities of Bayesian disjoint axioms depend on probability queries like $P(C^T, D^T)$. On semantic web, the number of individuals belonging to a pair of concepts is not large enough, which deviates the results.
- The AUCs, a.k.a. the probability that inference as a classifier ranks higher for correct axioms than incorrect axioms, in the figure are quite high. Thus, the effectiveness of inference is confirmed.
- For both subsumption axioms and disjointness axioms, the performance of inference gets better with the size of datasets growing.
- After the thresholds are selected, we compare the precision, recall, and F-measure of the axioms learned by the four approaches. As shown in Table 6, BelNet$^+$ outperforms the other three approaches in terms of F-measure. Worth noticing is that

Table 4
The thresholds & AUC of each partition per dataset (t_d: $\text{threshold}_{disjoint}$, AUC_d: $AUC_{disjoint}$, t_s: $\text{threshold}_{subsumption}$, AUC_s: $AUC_{subsumption}$). If the thresholds under constraint FPR < 0.1 and TPR > 0.7 are not available, we relax the constraint of FPR by 0.1.

%	Family				NTN				LUBM				Wine			
	t_d	AUC_d	t_s	AUC_s	t_d	AUC_d	t_s	AUC_s	t_d	AUC_d	t_s	AUC_s	t_d	AUC_d	t_s	AUC_s
10	0.9900	0.8044	0.4082	0.9126	0.9982	0.9128	0.8670	0.9931	0.9833	0.9210	0.8657	1.0000	0.9811	0.9210	0.9349	1.0000
20	0.9900	0.9348	0.3729	0.9902	0.9976	0.8790	0.6053	0.9886	0.9861	0.9374	0.9176	1.0000	0.9902	0.9374	0.9562	1.0000
30	0.9935	0.9756	0.8869	0.9995	0.9982	0.8834	0.3630	0.9872	0.9903	0.9282	0.9198	1.0000	0.9919	0.9282	0.9643	1.0000
40	0.9945	0.9899	0.9263	0.9995	0.9980	0.9020	0.6444	1.0000	0.9805	0.9555	0.9353	1.0000	0.9936	0.9555	0.9848	1.0000
50	0.9963	0.9981	0.9486	1.0000	0.9982	0.9249	0.7477	0.9999	0.9866	0.9542	0.9277	1.0000	0.9902	0.9542	0.9873	1.0000
60	0.9966	0.9992	0.9581	1.0000	0.9991	0.9589	0.8311	0.9996	0.9728	0.9509	0.9435	1.0000	0.9880	0.9509	0.9796	1.0000
70	0.9961	0.9997	0.9595	1.0000	0.9997	0.9538	0.8154	0.9987	0.9672	0.9561	0.9582	1.0000	0.9879	0.9561	0.9814	1.0000
80	0.9961	0.9998	0.9589	1.0000	0.9994	0.9381	0.3417	0.9945	0.9821	0.9650	0.9671	1.0000	0.9904	0.9650	0.9875	1.0000
90	0.9956	1.0000	0.9549	1.0000	0.9987	0.9650	0.6390	0.9998	0.9551	0.9591	0.9738	1.0000	0.9935	0.9591	0.9876	1.0000
100	0.9913	1.0000	0.9569	1.0000	0.9972	0.9748	0.9550	1.0000	0.9985	0.9459	0.9987	1.0000	0.8945	0.9459	0.8765	1.0000

Table 5
Statistics of the datasets for evaluation. The dataset name end with 's is the gold standard dataset.

Ontology	# concepts	# object properties	# $\sqsubseteq/\equiv/\perp$	# individuals	DL expressibility	Incompleteness
Family	19	4	27/0/0	202	$\mathcal{AL}$	0.609
Family′	19	4	27/17/14	202	$\mathcal{ALC}$	0.267
Semantic Bible	49	29	51/0/5	724	$\mathcal{SHOIN}(\mathcal{D})$	0.887
Semantic Bible′	49	29	52/6/34	724	$\mathcal{SHOIN}(\mathcal{D})$	0.048
LUBM	43	25	36/6/0	1555	$\mathcal{ALEHI}(\mathcal{D})$	0.946
LUBM′	43	25	36/6/52	1555	$\mathcal{SHI}(\mathcal{D})$	0.097
Wine	142	13	126/61/1	162	$\mathcal{SHOIN}$	0.957
Wine′	142	13	186/61/21	162	$\mathcal{SHOIN}$	0.197

Table 6
Quality of inference for 50% and 100% of the datasets (P: Precision, R: Recall, F: F-measure, DLer: DLLearner, Gold: Goldminer, Bel: BelNet, Bel$^+$: BelNet$^+$). Bold values indicate the best results in the comparisons.

%		DLLearner	Goldminer	BelNet	BelNet$^+$	DLLearner	Goldminer	BelNet	BelNet$^+$
		Family				NTN			
50	Precision	0.0778	0.5455	1.0000	1.0000	0.1875	0.5193	0.6244	0.9683
	Recall	0.0074	0.9630	0.5935	0.7561	0.1512	0.7679	0.7791	0.5577
	F-measure	0.0044	0.6964	0.7448	**0.8611**	0.1674	0.6196	0.6932	**0.7077**
100	Precision	0.0556	0.5175	0.8148	0.9306	0.2500	0.6179	0.6127	1.0000
	Recall	0.2222	0.9259	0.4634	0.8293	0.1977	0.8571	0.7791	0.7791
	F-measure	0.0889	0.6640	0.5908	**0.8770**	0.2208	0.7181	0.6860	**0.8758**
		LUBM				Wine			
50	Precision	0.3023	0.3293	0.5347	0.8755	0.3333	0.3953	0.6700	0.4340
	Recall	0.2045	0.3611	0.4412	0.3529	0.0821	0.1498	0.1164	0.2816
	F-measure	0.2440	0.3445	0.4835	**0.5031**	0.1317	0.2172	0.1979	**0.3382**
100	Precision	0.2558	0.3474	0.5802	0.7147	0.0993	0.4765	0.6100	0.4998
	Recall	0.3409	0.3889	0.4118	0.5294	0.2657	0.3478	0.1594	0.3382
	F-measure	0.2923	0.3670	0.4817	**0.6083**	0.1446	0.4021	0.2528	**0.4034**

the precision of BelNet$^+$ is always the highest in all datasets, which also confirms our expectation that BelNet$^+$ gives promising results of queries.
- From the whole dataset (c.f. the rows in Table 6 for data partition 100%), which is the real world ontology, BelNet$^+$ also outperforms other learners.

6.2.2. Performance of instance classification

We now compare the performance of BelNet$^+$ with DLLearner, Goldminer and BelNet in our proposed evaluation framework.

Quality of Classification. In order to show the effect of incompleteness over learners, we partition the datasets with respect to ABox assertions. Fig. 5 illustrates the average accuracy of classifying the instances in each dataset. Because there is no preference as to the concepts to be classified, we set equal weights to each concepts. We demonstrate the average accuracies on both training sets and the whole datasets. The upper row figures show the average accuracy on the training sets, and figures in the row below are the average accuracy on the whole dataset. From these figures, it is not hard to find:

- Although the average accuracies on training sets of BelNet$^+$ are not guaranteed to be the highest, they are the highest on the whole datasets in all of the tests. It proves that BelNet$^+$ is effective in instance classification under the existence of incompleteness.
- The average accuracy of BelNet$^+$ on the whole datasets goes closer to that on the training datasets. This shows that the performance of BelNet$^+$ gets better with the size of datasets growing, which is the same conclusion with that in the previous sections.
- Among the four learners, BelNet behaves similarly with BelNet$^+$ in terms of trend. This is not surprising, because BelNet$^+$ is an extension of BelNet.
- The average accuracies of DLLearner and Goldminer are relatively low on Family dataset, which shows that the performance of the two learners is affected more by the incompleteness in the datasets.

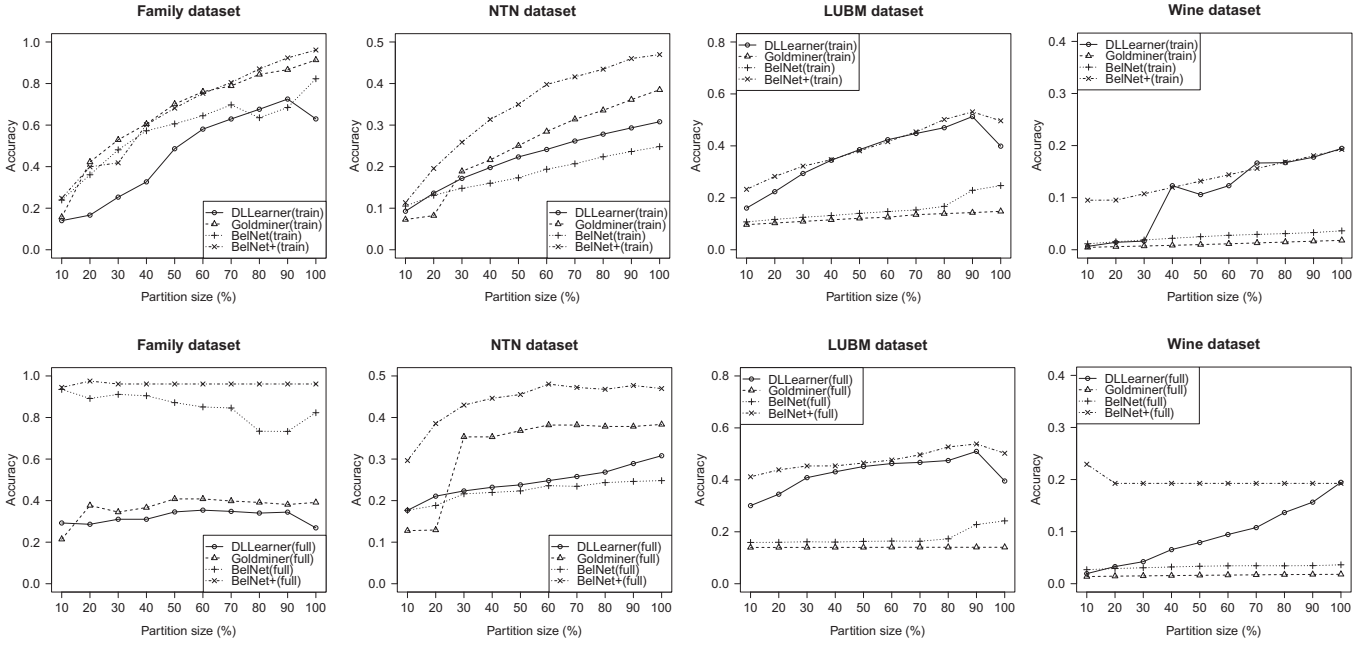


Fig. 5. Average accuracy of instance classification for each dataset, varying the size of the dataset. The figures in the upper row show the average accuracies on training sets, and figures in the row below show that on the whole datasets.

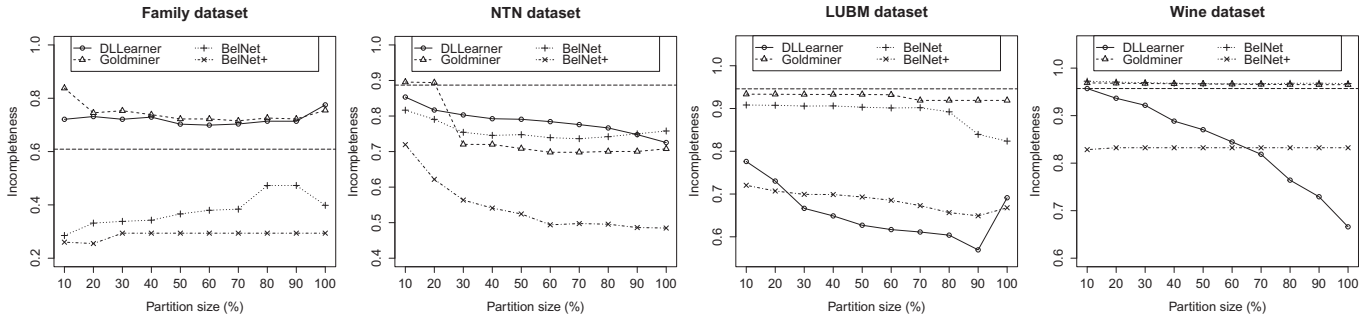


Fig. 6. Incompleteness in the ontologies learned by learners for each data partition. The straight lines in the figures indicate the incompleteness in the original datasets.

Effectiveness of Incompleteness Reduction. Having verified the performance of the learning approaches, in the sequel we will evaluate the effectiveness of the learners in reducing incompleteness. Fig. 6 represents the incompleteness of the ontologies learned by the learners, where the incompleteness in the original ontologies is shown by dashed line. We find that:

- Among the four learners, the performance of DLLearner is better when the dataset is larger.
- In all datasets, BelNet and BelNet$^+$ successfully improve the completeness in the original datasets.
- Compared with all other three learners, BelNet$^+$ decreases the most incompleteness except on dataset LUBM and Wine when the partition size is relatively large. LUBM is a large dataset, making CWA still causes DLLearner to get a large set of consistent expressive TBox axioms, which decreases the incompleteness in the dataset. On the expressive dataset Wine, DLLearner is able to generate specific concepts when learning concept definitions. As a result, the learned axioms are effective in decreasing incompleteness.

7. Related work

Since we are facing an era in which semantic web data grows very rapidly, learning TBox from ABox data has attracted lots of attention in the past 5 years. In this section, we notice a subset of works of ontology learning and statistical relational learning (SRL) that (1) focus on learning TBox axioms from ABox data, or (2) SRL models that handle DLs, or have applications in TBox learning.

Inductive Logic Programming. Inductive logic programming (ILP) marries machine learning and data mining, whose survey can be found in [4,3]. In particular, Jens Lehmann et al. developed *DLLearner* [18,19] to learn $\mathcal{ALC}$ *concept descriptions* from ontologies based on ILP techniques, where the candidate concept descriptions were generated by a downward refinement operator. After that, in [11], they particularly focused on handling larger datasets, such as DBpedia. TBox learning using ILP takes advantages of well defined refinement operators, which generatively or specifically search towards the target concept. These methods perform quite well when the data quality is relatively high. However, when the dataset suffers from incompleteness (or noise), these methods would drop into local optimum descriptions for concepts due to the incorrect "false" values generated by making CWA.

Association Rule Mining. As a classical data mining method for mining relationships, association rule mining (ARM) is applied in TBox learning problems. Johanna Völker et. al. learned $\mathcal{EL}$ axioms from ontologies based on association rule mining method [28], and in [5], this approach was further extended to learn disjointness axioms. The prototype Goldminer was also implemented. Realizing that learning from semantic web data suffered from a lack of negative examples when using OWA, Galárraga et al. [8] proposed a *rule* mining model supporting OWA scenario by introducing a new confidence measure in association rule mining. However, these methods mainly use support and confidence thresholds to export the final rules, which work unexpectedly when there is noise or data imbalance. Besides, these methods tend to learn a large number of irrelevant results, which put an extra burden on end-users of ontology learning applications. In addition, association rule mining is also applied to mine rules from dynamic ontologies for providing predictive reasoning [20,21].

Statistical Relational Learning. Koller et al. extended DL CLASSIC with nodes in a BN representing probabilistic information of the individuals in a specific class [16], and the model was called P-CLASSIC. It is closely related to the representation in BelNet$^+$. However, in BelNet$^+$, the edges correspond to the specific type of dependency – subsumption. BLP [14] 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) concepts are modeled with random variables; (3) schema level ontology learning is enabled. OntoBayes [31] extends OWL with annotating RDF triples with probabilities and dependencies. All these models have not been applied to TBox learning. In [24], $\mathcal{EL}^{++}$-*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. Besides these works, there are attempts that learn ABox using graphical models. For example, Rajput and Haider presented a semantic annotation framework that extracts ABox data using Bayesian networks [26]. In [29], Wang et al. proposed a data level information integration method with the aid of ontologies.

8. Conclusion and future work

In this paper, we deal with the following issues in the context of the semantic web: (1) in semantic web, making CWA results in noisy data; (2) learning one axiom a time leads to incorrect results in the existence of incompleteness.

To be specific, we propose an extension of Bayesian Description Logic Network, called BelNet$^+$ and correspondingly introduce a procedure to learn TBox axioms. In order to learn schemata from incomplete ABox, DL concept expressions correspond to probabilistic nodes, subsumption and disjointness relationships between DL concept expressions are represented as links. Learning schemata is transformed into structure learning and inference in BelNet$^+$, which, from the experiments, were shown to be effective for learning from incomplete semantic data in the proposed evaluation framework in terms of accuracy and the ability to reduce incompleteness.

Additionally, in order to overcome the drawbacks of current evaluation metrics used in TBox learning, a.k.a. subjectiveness of domain experts and sensitiveness in the gold standard ontologies, we propose a novel evaluation framework taking *unknowns* that widely spread the semantic web into consideration. Evaluations demonstrate the effectiveness of our approach.

In the future, we will explore the following aspects: (1) we use exact inference in BelNet$^+$, which is not efficient enough for networks with large tree-width. We will study this issue and use approximate methods in the future; (2) ABox materialization on all instances costs too much for a large dataset, such as DBpedia, we will find scalable solutions of BelNet$^+$ on very large datasets.

Acknowledgements

This work is partially funded by the National Science Foundation of China under Grant 61170165, the EU IAPP K-Drive project (286348) and the EPSRC WhatIf project (EP/J014354/1). The authors would like to thank Campbell Wilson for proof-reading the document.

References

[1] Franz Baader, Werner Nutt, The Description Logic Handbook, New York, NY, USA, 2003, pp. 43–95.
[2] Philipp Cimiano, Ontology Learning and Population from Text – Algorithms, Evaluation and Applications, Springer, 2006.
[3] Claudia d'Amato, Nicola Fanizzi, Floriana Esposito, Inductive learning for the semantic web: what does it buy?, Semantic Web 1 (1, 2) (2010) 53–59
[4] Nicola Fanizzi, Claudia D'Amato, Floriana Esposito, Machine Learning Methods for Ontology Mining, 2010, pp. 131–153.
[5] Daniel Fleischhacker, Johanna Vlker, Inductive learning of disjointness axioms, in: On the Move to Meaningful Internet Systems: OTM 2011, Lecture Notes in Computer Science, vol. 7045, 2011, pp. 680–697.
[6] Giorgos Flouris, Zhisheng Huang, Jeff Z. Pan, Dimitris Plexousakis, Holger Wache, Inconsistencies, negations and changes in ontologies, in: Proceedings of the 21st National Conference on Artificial Intelligence, vol. 2, 21, 2006, pp. 1295–1300.
[7] Achille Fokoue, Felipe Meneguzzi, Murat Sensoy, Jeff Z. Pan, Querying linked ontological data through distributed summarization, in: Proceedings of the 26th Conference on Artificial Intelligence, AAAI'12, 2012, pp. 31–37.
[8] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, Fabian Suchanek. AMIE: association rule mining under incomplete evidence in ontological knowledge bases, in: Proceedings of the 22nd International Conference on World Wide Web, WWW '13, 2013, pp. 413–422.
[9] Jayanta K. Ghosh, Probabilistic networks and expert systems: exact computational methods for bayesian networks, Int. Stat. Rev. 76 (2) (2008) 306–307.
[10] Haibo He, Edwardo A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.
[11] Sebastian Hellmann, Jens Lehmann, Sören Auer, Learning of OWL class descriptions on very large knowledge bases, Int. J. Semantic Web Inf. Syst. (IJSWIS) 5 (2) (2009) 25–48.
[12] Ian Horrocks, Peter F. Patel-Schneider, KR and reasoning on the semantic web: OWL, in: Handbook of Semantic Web Technologies, 2011, pp. 365–398.
[13] Qiu Ji, Zhiqiang Gao, Zhisheng Huang, Reasoning with noisy semantic data, in: Proceedings of the 8th Extended Semantic Web Conference on The Semanic Web: Research and Applications – ESWC'11, vol. Part II, 2011, pp. 497–502.
[14] Kristian Kersting, Luc De Raedt, Towards combining inductive logic programming with Bayesian networks, in: Inductive Logic Programming, Lecture Notes in Computer Science, vol. 2157, 2001, pp. 118–131.
[15] Daphne Koller, Nir Friedman, Probabilistic Graphical Models: Principles and Techniques – Adaptive Computation and Machine Learning, The MIT Press, 2009.
[16] Daphne Koller, Alon Levy, Avi Pfeffer, P-CLASSIC: a tractable probabilistic description logic, in: Proceedings of the 14th National Conference on Artificial Intelligence and 9th Conference on Innovative Applications of Artificial Intelligence, AAAI'97/IAAI'97, 1997, pp. 390–397.
[17] Jens Lehmann, Learning OWL Class Expressions, PhD Thesis, University of Leipzig, 2010.
[18] Jens Lehmann, Pascal Hitzler, A refinement operator based learning algorithm for the ALC description logic, in: Proceedings of the 17th International Conference on Inductive Logic Programming, ILP'07, 2008, pp 147–160.
[19] Jens Lehmann, Pascal Hitzler, Concept learning in description logics using refinement operators, Mach. Learn. 78 (1–2) (2010) 203–250.
[20] Freddy Lecue, Jeff Z. Pan, Predicting knowledge in an ontology stream, in: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI'13, 2013, pp. 2662–2669.
[21] Freddy Lecue, Jeff Z. Pan, Consistent knowledge discovery from evolving ontologies, in: Proceedings of the 29th Conference on Artificial Intelligence, AAAI'15, 2015, to appear.
[22] Tom M. Mitchell, Justin Betteridge, Andrew Carlson, Estevam Hruschka, Richard Wang, Populating the semantic web by macro-reading internet text, in: Proceedings of the 8th International Semantic Web Conference, ISWC '09, 2009, pp. 998–1002.
[23] Richard E. Neapolitan, Learning Bayesian Networks, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2003.
[24] Mathias Niepert, Jan Noessner, Heiner Stuckenschmidt, Log-linear description logics, in: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, IJCAI'11, vol. 3, 2011, pp. 2153–2158.
[25] Jeff Z. Pan, Edward Thomas, Yuan Ren, Stuart Taylor, Tractable Fuzzy and Crisp Reasoning in Ontology Applications, IEEE Comput. Intell. M. 7 (2) (2012) 45–53.
[26] Quratulain Rajput, Sajjad Haider, BNOSA: a bayesian network and ontology based semantic annotation framework, Web Semantics: Science, Services Agents World Wide Web 9 (2) (2011) 99–112.
[27] Max Schmachtenberg, Christian Bizer, Heiko Paulheim, Adoption of the linked data best practices in different topical domains, in: Proceedings of the 13th International Conference on The Semantic Web, ISWC'13, vol. Part I, 2014, pp. 245–260.
[28] Johanna Völker, Mathias Niepert, Statistical schema induction, in: Proceedings of the 8th Extended Semantic Web Conference on The Semantic Web: Research and Applications, ESWC'11, vol. Part I, 2011, pp. 124–138.
[29] Chao Wang, Jie Lu, Guangquan Zhang, Integration of ontology data through learning instance matching, in: IEEE/WIC/ACM International Conference on Web Intelligence, 2006, pp. 536–539.
[30] Honghan Wu, Boris Villazn-Terrazas, Jeff Z. Pan, Jos Manul Gmez-Prez, Exploiting semantic web datasets: a graph pattern based approach, in: Proceedings of the 8th Chinese Semantic Web and Web Science Conference, CSWS'14, 2014, pp. 167–173.
[31] Yi Yang, Jacques Calmet, OntoBayes: an ontology-driven uncertainty model, in: Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), CIMCA '05, vol. 1, 2005, pp. 457–463.
[32] Man Zhu, DC proposal: ontology learning from noisy linked data, in: Proceedings of the 10th International Conference on The Semantic Web, ISWC'11, vol. Part II, 2011, pp. 373–380.
[33] Man Zhu, Zhiqiang Gao, Jeff Z. Pan, Yuting Zhao, Ying Xu, Zhibin Quan, Ontology learning from incomplete semantic web data by BelNet, in: Proceedings of the 2013 IEEE 25th International Conference on Tools with Artificial Intelligence, ICTAI '13, 2013, pp. 761–768.