The Bayesian Ontology Reasoner is BORN!

İsmail İlkan Ceylan[1⋆], Julian Mendez[1⋆], and Rafael Peñaloza[2]

[1] Theoretical Computer Science, TU Dresden, Germany
{ceylan,mendez}@tcs.inf.tu-dresden.de
[2] KRDB Research Centre, Free University of Bozen-Bolzano, Italy
rafael.penaloza@unibz.it

Abstract. Bayesian ontology languages are a family of probabilistic ontology languages that allow to encode probabilistic information over the axioms of an ontology with the help of a Bayesian network. The Bayesian ontology language $\mathcal{BEL}$ is an extension of the lightweight Description Logic (DL) $\mathcal{EL}$ within the above-mentioned framework. We present the system BORN that implements the probabilistic subsumption problem for $\mathcal{BEL}$.

1 Introduction

Bayesian ontology languages are a recently proposed family of knowledge representation formalisms capable of handling uncertainty [5]. They extend description logics (DLs) [3] with annotations that express the probabilistic dependencies among the axioms with the help of a Bayesian network (BN).

The best studied member of this family is $\mathcal{BEL}$, which extends the lightweight DL $\mathcal{EL}$. In recent work it has been shown that the well-known completion-based algorithm for deciding subsumption in $\mathcal{EL}$ [2] can be adapted to decide probabilistic subsumptions in $\mathcal{BEL}$ optimally w.r.t. computational complexity [7]. Unfortunately this approach destroys all the relevant properties required for efficient probabilistic entailments in a BN; thus, it is unlikely to produce a practical implementation before advanced optimizations are developed.

To obtain an efficient $\mathcal{BEL}$ reasoner, we follow a different strategy and exploit the highly optimized methods that have been developed for probabilistic logic programming. Specifically, we use ProbLog [11] as an engine for handling the probabilistic knowledge. We refer the interested reader to [13] for a description of the ProbLog 2 system.

Our approach takes advantage of the simple logical structure of $\mathcal{EL}$ and implements the completion rules in ProbLog. Using `jcel` [12], we normalize the ontology and extract a module [14] containing all the axioms that are relevant for the desired probabilistic entailment. The BN can also be encoded in ProbLog using a well-known reduction. ProbLog then takes over the task of deciding the probabilistic subsumption relation. Notice that the computation of a module is fundamental for a practical algorithm as it has been argued that well-structured

[⋆] Supported by DFG within the Research Training Group "RoSI" (GRK 1907).

ontologies usually have rather small modules, even if the total size of the ontology might be large.

In this paper we describe the implementation of BORN,[3] a Bayesian ontology reasoner based on the ideas sketched above. Our first experiments show that reasoning with this system is feasible.

2 The Bayesian Ontology Language $\mathcal{BEL}$

The Bayesian ontology language $\mathcal{BEL}$ is a probabilistic extension of the light-weight DL $\mathcal{EL}$ [4], where probabilities are encoded using a Bayesian network [10]. Formally, a *Bayesian network* (BN) is a pair $\mathcal{B} = (G, \Phi)$, where $G = (V, E)$ is a finite directed acyclic graph (DAG) whose nodes represent Boolean random variables, and Φ contains, for every node $x \in V$, a conditional probability distribution $P_\mathcal{B}(x \mid \pi(x))$ of x given its parents $\pi(x)$. Every BN $\mathcal{B}$ defines a unique joint probability distribution (JPD) over V given by

$$P_\mathcal{B}(V) = \prod_{x \in V} P_\mathcal{B}(x \mid \pi(x)).$$

As with classical DLs, the main building blocks in $\mathcal{BEL}$ are *concepts*, which are syntactically built as $\mathcal{EL}$ concepts. In $\mathcal{BEL}$, the domain knowledge is encoded via *probabilistic GCIs* of the form $\langle C \sqsubseteq D : x \rangle$ where $(C \sqsubseteq D)$ is an $\mathcal{EL}$ GCI and x is either a literal[4] over a fixed set of variables V or the empty set. A $\mathcal{BEL}$ TBox is a finite set of probabilistic GCIs over V. A $\mathcal{BEL}$ KB is a pair $(\mathcal{T}, \mathcal{B})$ where $\mathcal{T}$ is a $\mathcal{BEL}$ TBox and $\mathcal{B}$ is a BN, both defined over V.

A *contextual interpretation* is a pair $(\mathcal{I}, \mathcal{V}^\mathcal{I})$ where $\mathcal{I}$ is an $\mathcal{EL}$ interpretation and $\mathcal{V}^\mathcal{I}$ is a propositional interpretation. The contextual interpretation $(\mathcal{I}, \mathcal{V}^\mathcal{I})$ satisfies $\langle C \sqsubseteq D : x \rangle$ iff (i) $\mathcal{V}^\mathcal{I} \not\models x$, or (ii) $\mathcal{I} \models C \sqsubseteq D$. It is a *model* of the TBox $\mathcal{T}$ iff it is a model of all the probabilistic GCIs in $\mathcal{T}$.

Intuitively, axioms of the form $\langle C \sqsubseteq D : \emptyset \rangle$ represent crisp (or classical) axioms. In this view, the DL $\mathcal{EL}$ is an instance of $\mathcal{BEL}$ where all axioms are of the form $\langle C \sqsubseteq D : \emptyset \rangle$. For brevity, we often write $\langle C \sqsubseteq D \rangle$ to denote such axioms.

The probabilistic information provided by the BN is handled via the so-called *multiple-world semantics*. Briefly, a contextual interpretation describes a possible world; by assigning a probabilistic distribution over these interpretations, we describe the required probabilities, which should be consistent with the BN.

Definition 1. *A* probabilistic interpretation *is a pair $\mathcal{P} = (\mathfrak{I}, P_\mathfrak{I})$, where $\mathfrak{I}$ is a set of contextual interpretations and $P_\mathfrak{I}$ is a probability distribution over $\mathfrak{I}$ such that $P_\mathfrak{I}(\mathcal{I}) > 0$ only for finitely many interpretations $\mathcal{I} \in \mathfrak{I}$.*

$\mathcal{P}$ is a model *of the TBox $\mathcal{T}$ if every $\mathcal{I} \in \mathfrak{I}$ is a model of $\mathcal{T}$. $\mathcal{P}$ is* consistent *with the BN $\mathcal{B}$ if for every possible valuation $\mathcal{W}$ over V it holds that*

$$\sum_{(\mathcal{I}, \mathcal{W}) \in \mathfrak{I}} P_\mathfrak{I}(\mathcal{I}, \mathcal{W}) = P_\mathcal{B}(\mathcal{W}).$$

[3] Available under http://lat.inf.tu-dresden.de/systems/born

[4] We point out that in the general case $\mathcal{BEL}$ allows for a set of literals.

$\mathcal{P}$ is a model *of the KB $(\mathcal{B}, \mathcal{T})$ iff it is a model of $\mathcal{T}$ and consistent with $\mathcal{B}$.*

The main reasoning problem in $\mathcal{BEL}$ is answering probabilistic subsumption queries. In contrast to classical subsumption, in this case we are interested in computing the probability with which a subsumption relation holds, as given by the BN.

Definition 2 (probabilistic subsumption). *Let $\mathcal{K}$ be a $\mathcal{BEL}$ KB, and A, B $\mathcal{BEL}$ concept names. The* probability of the subsumption $A \sqsubseteq B$ *w.r.t. the probabilistic interpretation $\mathcal{P} = (\mathfrak{I}, P_{\mathfrak{I}})$ is*

$$P_{\mathcal{P}}(A \sqsubseteq B) := \sum_{(\mathcal{I},\mathcal{W}) \in \mathfrak{I}, \mathcal{I} \models A \sqsubseteq B} P_{\mathfrak{I}}(\mathcal{I}, \mathcal{W}).$$

The probability of $A \sqsubseteq B$ w.r.t. $\mathcal{K}$ *is* $P_{\mathcal{K}}(A \sqsubseteq B) := \inf_{\mathcal{P} \models \mathcal{K}} P_{\mathcal{P}}(A \sqsubseteq B)$.

We consider special restrictions over a $\mathcal{BEL}$ TBox $\mathcal{T}$ w.r.t. a valuation $\mathcal{V}^{\mathcal{I}} = \mathcal{W}$:

$$\mathcal{T}_{\mathcal{W}} := \{\langle A \sqsubseteq B : \emptyset \rangle \mid \langle A \sqsubseteq B : x \rangle \in \mathcal{T}, \mathcal{W} \models x\}.$$

Intuitively, every valuation defines an $\mathcal{EL}$ TBox and to decide the probability of a subsumption $A \sqsubseteq B$ it is enough to sum up the probabilities of the valuations $\mathcal{W}$ for which the classical entailment relation $\mathcal{T}_{\mathcal{W}} \models A \sqsubseteq B$ holds:

$$P_{\mathcal{K}}(A \sqsubseteq B) = \sum_{\mathcal{T}_{\mathcal{W}} \models A \sqsubseteq B} P_{\mathcal{B}}(\mathcal{W}).$$

Computing the probability of a subsumption query in $\mathcal{BEL}$ is shown to be PP-complete [7] by a reduction to inference in BNs. We refer to [6, 7] for the detailed definitions and proofs.

3 BORN: System Overview

BORN accepts a $\mathcal{BEL}$ KB $(\mathcal{T}, \mathcal{B})$ and a subsumption query as input and transforms the input into a probabilistic logic program. BORN is implemented on top of `jcel` [12]. It first normalizes the given $\mathcal{BEL}$ TBox $\mathcal{T}$ and then computes a module of $\mathcal{T}$ w.r.t. the given query by making use of existing features of `jcel`. Briefly, let $\mathcal{T}' \subseteq \mathcal{T}$ be two $\mathcal{BEL}$ TBoxes and let S be a signature. We say that $\mathcal{T}'$ is a module of $\mathcal{T}$ w.r.t. the signature S if for every subsumption query $\mathtt{q}$ with $\mathsf{sig}(\mathtt{q}) \subseteq S$ it holds that $\mathcal{T} \models \mathtt{q}$ iff $\mathcal{T}' \models \mathtt{q}$. Intuitively a module $\mathcal{T}'$ is a minimal subset of $\mathcal{T}$ containing all and only the relevant information w.r.t. a signature.

BORN requires a $\mathcal{BEL}$ KB $(\mathcal{T}, \mathcal{B})$ and a subsumption query as input. As an answer, BORN returns the probability for the query to hold.

Example 3. Consider the $\mathcal{BEL}$ KB $\mathcal{K} = (\mathtt{ABC}, \mathcal{B}_{\mathtt{ABC}})$ where

$$\begin{aligned}\mathtt{ABC} := \{\, & \langle \mathtt{A} \sqsubseteq \exists \mathtt{r.B} : x_0 \rangle, \langle \mathtt{A} \sqsubseteq \mathtt{C} : x_4 \rangle, \langle \mathtt{B} \sqsubseteq \exists \mathtt{s.C} : \neg x_3 \rangle, \\ & \langle \mathtt{C} \sqcap \mathtt{D} \sqsubseteq \mathtt{E} : x_5 \rangle, \langle \exists \mathtt{r.B} \sqsubseteq D : \neg x_2 \rangle, \langle \mathtt{C} \sqsubseteq \mathtt{E} : x_3 \rangle\},\end{aligned}$$

$\mathcal{B}_{\mathtt{ABC}}$ is the BN given in Figure 1 and the subsumption query $\mathtt{A} \sqsubseteq \mathtt{E}$. The relevant computations yield $P_{\mathcal{K}}(\mathtt{A} \sqsubseteq \mathtt{E}) = 0.1796$.

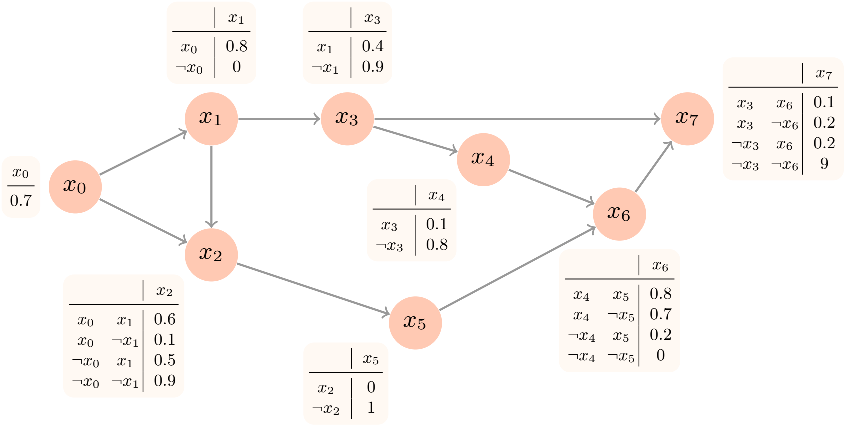

Fig. 1: The BN $\mathcal{B}_{\mathtt{ABC}}$ over the variables $V = \{x_i \mid 0 \leq i \leq 7\}$

We explain how each component of the ontology and the query is precisely represented. The $\mathcal{BEL}$ TBox ABC is a standard OWL 2 EL file with annotations representing literals:

```
SubClassOf(Annotation(:prob "x0"^^xsd:string) lat:a ObjectSomeValuesFrom(lat:r lat:b))
SubClassOf(Annotation(:prob "x4"^^xsd:string) lat:a lat:c)
SubClassOf(Annotation(:prob "\\+x3"^^xsd:string) lat:b ObjectSomeValuesFrom(lat:s lat:c))
SubClassOf(Annotation(:prob "x5"^^xsd:string) ObjectIntersectionOf(lat:c lat:d) lat:e)
SubClassOf(Annotation(:prob "\\+x2"^^xsd:string) ObjectSomeValuesFrom(lat:r lat:b) lat:d)
SubClassOf(Annotation(:prob "x3"^^xsd:string) lat:c lat:e))
```

We use the functional syntax throughout the paper for the sake of legibility, but all known OWL syntax paradigms are accepted by BORN. After performing module extraction, all axioms are converted into a set of clauses. For instance, the first axiom is rewritten as:

```
con('a'). con('b'). role('r').
subs('a', exists('r', 'b')) :- x0.
```

Clearly, `con`, `role` and `subs` are reserved predicate names used to denote concepts, roles and explicit subsumption relations. We assume that the BN and the query use the ProbLog syntax, that is roughly the Prolog syntax enriched with probabilistic annotations. A portion of the BN $\mathcal{B}$ from the Example 3 is then given by

```
0.7::x0.
0.8::x1:-x0.
0::x1:-\+x0.
0.6::x2:-x0,x1.
0.1::x2:-x0,\+x1.
0.5::x2:-\+x0,x1.
0.9::x2:-\+x0,\+x1.
```

Table 1: Ontologies and their sizes

Ontology	Size of the terminology	Size of the BN
$(\texttt{ABC}, \mathcal{B}_{\texttt{ABC}})$	6	8
$(\texttt{ABC}, \mathcal{B}'_{\texttt{ABC}})$		
$(\texttt{DBPEDIA}, \mathcal{B}_{\texttt{DBPEDIA}})$	266	17
$(\texttt{DBPEDIA}, \mathcal{B}'_{\texttt{DBPEDIA}})$		
$(\texttt{GO}, \mathcal{B}_{\texttt{GO}})$	23507	200
$(\texttt{GO}, \mathcal{B}'_{\texttt{GO}})$		

and the query from the Example 3 is given as `query(sub('a', 'e'))`.

We use an additional reserved symbol `sub` to represent all subsumption relations and not only the explicit ones. This distinction is forced for optimization reasons. Finally, we implement the deduction rules [4] for DL $\mathcal{EL}$ into the probabilistic logic program ProbLog as follows:

```
sub(X, B) :- subs(X, B).
sub(X, B) :- subs(A, B), sub(X, A), con(X), con(A), con(B).
sub(X, B) :- subs(and(A1, A2), B), sub(X, A1), sub(X, A2),
             con(X), con(A1), con(A2), con(B).
sub(X, exists(R, B)) :- subs(A, exists(R, B)), sub(X, A),
                        con(X), con(A), con(B), role(R).
sub(X, B) :- subs(exists(R, A), B), sub(X, exists(R, Y)), sub(Y, A),
             con(X), con(Y), con(A), con(B), role(R).
```

As an answer, BORN returns the subsumption relation with a probability value attached to it: `sub('a','e') : 0.1796`

The lack of Bayesian ontologies has led additional difficulties in evaluating BORN. To show some initial results on BORN, we have created the artificial ontologies listed in Table 1. `DBPEDIA` is adopted from `http://trill.lamping.unife.it/swish/` and `GO` is the Gene Ontology [1] annotated with literals. Each terminology is paired with two randomly generated BNs of the same size. The difference is in their independence assumptions, i.e. $\mathcal{B}_{\mathcal{O}}$ consists of independent nodes, whereas $\mathcal{B}'_{\mathcal{O}}$ is well-connected. Table 1 represents the relative sizes of terminologies and BNs, where the former is measured in the number of axioms and the latter in the number of nodes.

We ran our experiments on a PC equipped with a 2,3 GHz Intel Core i7 with 8 GB of main memory and obtained feasible execution times for the limited number of queries we have posed. Before moving to a detailed analysis, we stress the fact that all of the results we present here are preliminary. To speak of a general evaluation, there is still the need for further experiments with different types of ontologies and a large number of queries.

Performing module extraction prior to probabilistic inference turns out to be a simple but effective approach. The idea is that a module is usually much smaller than the original TBox. This is, of course, not always the case as evidenced by the module sizes of `ABC` w.r.t. some queries given in Figure 2. However, for some

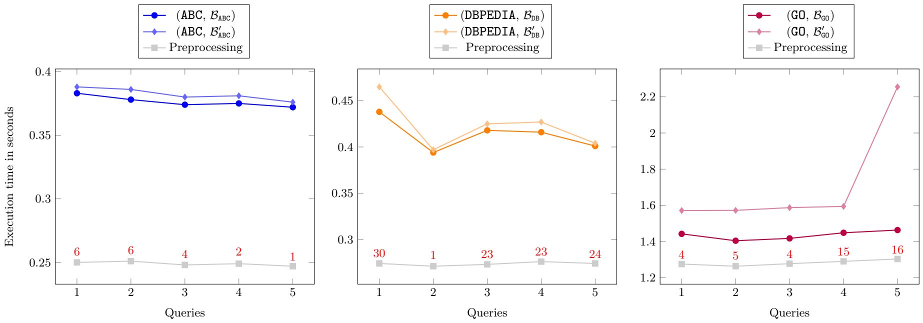

Fig. 2: Experiments run on BORN with the relevant module sizes

real terminologies (e.g. GO) the size of a module turns out to be extremely small compared to the original ontology. Notice that GO is very flat from its nature and leads to really small module sizes.

We report the module sizes (in the number of axioms), the total execution times and the execution times for preprocessing in Figure 2. The delay in preprocessing is mainly due to modularization, but preprocessings times remain rather stable over different queries compared to the total execution time. This time is below 1.4s even for GO, that consists of 23507 normalized axioms originally.

As expected, the size and shape of the network influences the performance greatly. We expect the number of the literals and the conditional dependencies to be much smaller compared to the sizes of terminologies. Given these assumptions, our initial experiments show that BORN is a promising tool for probabilistic reasoning over DL ontologies.

4 Conclusions

We have introduced the Bayesian Ontology reasoner BORN. To the best of our knowledge BORN is the first Bayesian reasoner over DL ontologies based on the multiple world semantics. Another probabilistic formalism is PR-OWL [8], which is based on a very different semantics. Closer to our semantics are the lightweight Bayesian ontology languages [9] for which no implementation has been provided yet. The closest existing system to BORN is the probabilistic DL reasoner TRILL [15]. However, TRILL forces independence of axioms.

We plan to create realistic test data and evaluate BORN and generalize the results to extensions of $\mathcal{EL}$ as well as to other DL-based ontology languages. Additionally, we think that further optimizations are possible for BORN. Ultimately, we plan to improve BORN to achieve a scalable probabilistic reasoner over arbitrary Bayesian DL ontologies.

References

1. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nature genetics 25(1), 25–29 (2000)
2. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: Proc. of IJCAI'05. AAAI Press (2005)
3. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, 2nd edn. (2007)
4. Brandt, S.: Polynomial Time Reasoning in a Description Logic with Existential Restrictions, GCI Axioms, and—What Else? In: Proc. of ECAI'04. pp. 298–302. IOS Press (2004)
5. Ceylan, İ.İ., Peñaloza, R.: Bayesian Description Logics. In: Proc. of DL'14. CEUR Workshop Proceedings, vol. 1193. CEUR-WS (2014)
6. Ceylan, İ.İ., Peñaloza, R.: The Bayesian Description Logic $\mathcal{BEL}$. In: Proc. of IJCAR'14. LNCS, vol. 8562, pp. 480–494. Springer Verlag (2014)
7. Ceylan, İ.İ., Peñaloza, R.: Tight Complexity Bounds for Reasoning in the Description Logic $\mathcal{BEL}$. In: Proc. of JELIA'14. vol. 8761, pp. 77–91. Springer Verlag (2014)
8. da Costa, P.C.G., Laskey, K.B., Laskey, K.J.: PR-OWL: A Bayesian Ontology Language for the Semantic Web. In: Uncertainty Reasoning for the Semantic Web I. LNCS, vol. 5327, pp. 88–107. Springer Verlag (2008)
9. D'Amato, C., Fanizzi, N., Lukasiewicz, T.: Tractable Reasoning with Bayesian Description Logics. In: Proc. of SUM'08. LNCS, vol. 5291, pp. 146–159. Springer Verlag (2008)
10. Darwiche, A.: Modeling and Reasoning with Bayesian Networks. Cambridge University Press (2009)
11. De Raedt, L., Kimmig, A., Toivonen, H.: ProbLog: A probabilistic prolog and its application in link discovery. In: Proc. of IJCAI'07. AAAI Press (2007)
12. Mendez, J.: jcel: A modular rule-based reasoner. In: Proc. of ORE'12. CEUR Workshop Proceedings, vol. 858. CEUR-WS (2012)
13. Renkens, J., Shterionov, D., Van den Broeck, G., Vlasselaer, J., Fierens, D., Meert, W., Janssens, G., De Raedt, L.: ProbLog2: From probabilistic programming to statistical relational learning. Proc. of NIPS'12 pp. 1–5 (2012)
14. Suntisrivaraporn, B.: Polynomial-Time Reasoning Support for Design and Maintenance of Large-Scale Biomedical Ontologies. Phd thesis, TU Dresden, Germany (2009)
15. Zese, R., Bellodi, E., Lamma, E., B, F.R., Aguiari, F.: Semantics and inference for probabilistic description logics. In: Uncertainty Reasoning for the Semantic Web III. LNCS, vol. 8816, pp. 79–99. Springer (2014)