First-Order Bayes-Ball for CP-Logic

Nima Taghipour NIMA.TAGHIPOUR@CS.KULEUVEN.BE
Wannes Meert WANNES.MEERT@CS.KULEUVEN.BE
Jan Struyf JAN.STRUYF@CS.KULEUVEN.BE
Hendrik Blockeel HENDRIK.BLOCKEEL@CS.KULEUVEN.BE
Dept. of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3001 Leuven, Belgium

Abstract

Efficient probabilistic inference is key to the success of statistical relational learning. One issue that affects inference cost is the presence of irrelevant random variables. The Bayes-ball algorithm can identify such irrelevant variables in a propositional Bayesian network. This paper presents a lifted version of Bayes-ball, which works directly on the first-order level, and shows how this algorithm applies to CP-logic inference.

1. Introduction

Achieving efficient inference is a major goal in statistical relational learning (SRL). One issue that affects the computational cost of probabilistic inference is the presence of irrelevant random variables. In order to answer a given probabilistic query, one may need only a subset of the variables. Most SRL algorithms that are based on Bayesian networks (BNs) therefore need to compute the *minimal relevant network* (MRN) of the query, which only includes variables for the relevant ground atoms, and then run a Bayesian inference algorithm on the MRN to compute the answer.

A straightforward way to compute the MRN for a given query atom and probabilistic logic theory (e.g., a CP-logic theory, see Section 2) is to compute all proofs of the query atom (e.g., using SLD resolution) and to collect all ground clauses that are used therein. The MRN can be trivially constructed from these. This method works well if there is no evidence. When truth values for certain evidence atoms are given, all proofs of these atoms must be computed as well to construct the part of the BN that connects the evidence atoms to the query atom. This network may contain irrelevant atoms: some atoms encountered in certain proofs of an evidence atom may be D-separated (Shachter, 1998) from the query. Such irrelevant atoms can be removed by running Bayes-ball (Shachter, 1998), which has been designed to compute the relevant part of a BN given a query and evidence. The BN obtained after running Bayes-ball is the MRN.

This straightforward method for computing the MRN has the disadvantage that it initially computes a BN that may be larger than the MRN. Predoiu (2003) describes a first-order version of Bayes-ball called Magic Bayes-ball (MBB) that addresses this issue. MBB first performs a pass of so-called magic backward-chaining on the query atom followed by a number of iterations of magic forward-chaining on the evidence atoms. A disadvantage is that MBB may compute proofs for irrelevant evidence atoms.

We propose a different first-order version of the Bayes-ball algorithm called first-order Bayes-ball (FOBB), which resembles the original algorithm more closely. FOBB's main advantage over MBB is that it avoids proving irrelevant evidence atoms. While FOBB is a general method that applies to several probabilistic logics, we illustrate it for the particular case of causal probabilistic logic (CP-logic).

2. CP-Logic

CP-logic is a probabilistic logic modeling language that has been designed to model causal processes (Vennekens et al., 2006). The model takes the form of a CP-logic theory (CP-theory), which is a set of events in which each event is represented as a rule of the following form:

$$(h_1 : \alpha_1) \vee \ldots \vee (h_n : \alpha_n) \leftarrow b_1, \ldots, b_m.$$

with h_i atoms, b_i literals, and α_i *causal* probabilities; $0 < \alpha_i \leq 1$, $\sum \alpha_i \leq 1$. We call the set of all $(h_i : \alpha_i)$

Presented at *ILP-MLG-SRL*, Leuven, Belgium, 2009.

the *head* of the event, and the conjunction of literals b_i the *body*. If the body of a CP-event evaluates to true, then the event will happen and cause *at most one* of the head atoms to become true; the probability that the event causes h_i is given by α_i. Note that if $\sum \alpha_i < 1$, it is possible that no head atom is caused.

Meert et al. (2008) defined a transformation that can transform any non-recursive ground CP-theory into an equivalent Bayesian network (EBN). The EBN contains one node for each CP-theory atom and one so-called *choice node* for each CP-event (Fig. 1). The event's body atoms become the parents of the choice node in the EBN and the head atoms become the node's children. While all atom nodes in the EBN are binary variables, the choice nodes are multi-valued; they model the non-determinism and mutual-exclusiveness in the outcome of the events. All atom nodes are functional and all choice nodes are probabilistic. Meert et al. (2008) give details about the transformation.

3. Bayes-Ball

Bayes-ball (Shachter, 1998) identifies the MRN of a BN for a given set of query and evidence nodes. It is based on the analogy of a bouncing ball that travels through the BN (Fig. 1). The ball starts at the query nodes. Upon reaching each node, it may *pass through*, *bounce back* and/or be *blocked*, depending on the direction from which it came and on whether the node is probabilistically or functionally dependent on its parents (based on D-separation):

(a) an unobserved probabilistic node passes balls through, and in addition bounces balls back from children (e.g., $s(1,3) \rightarrow \mathbf{C_4} \rightarrow s(1,3)$)
(b) an observed node bounces balls back from parents (e.g., $C_2(1,1) \rightarrow \mathbf{r(1,1)} \rightarrow C_2(1,1)$), but blocks balls from children (e.g., $C_1(1,2) \rightarrow s(1,2) \not\rightarrow C_3$)
(c) a functional unobserved node always passes balls through (e.g., $C_2(1,2) \rightarrow \mathbf{t(1)} \rightarrow C_5$)

Nodes are marked at each visit of the ball, depending on the type of action performed on the ball. In the end, these marks indicate the relevance of each node.

4. First-Order Bayes-Ball

FOBB is based on the same principles as Bayes-ball, building upon the transformability of a CP-theory to an EBN. Its main advantage is the possibility to perform some steps at the first-order level. Therefore, several nodes can be represented by one first-order atom, and be visited in one single step. We explain the algorithm at a high level; a more detailed description will follow in a longer version of this paper.

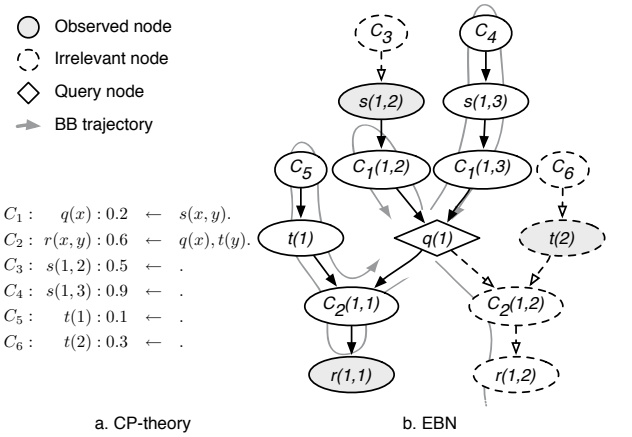

Figure 1. (a) CP-theory and (b) EBN. Parts irrelevant to the query $P(q(1)|r(1,1), s(1,2), t(2))$ are indicated with dashed lines; the remaining part is the MRN. The gray lines indicate a trace of Bayes-ball.

FOBB assumes that the query atoms Q are ground, the evidence atoms E are ground and consistent w.r.t. the CP-theory T, and that T has a finite grounding. The algorithm (Alg. 1) closely resembles the original Bayes-ball (see Shachter (1998)), the main differences are that it works with first-order atoms instead of BN nodes, and that it uses the first-order input CP-theory to compute the parents and children of a first-order atom, which may also be first-order.

Based on the example (Fig. 1), we explain some of the differences with the original Bayes-ball.

Instead of scheduling nodes, FOBB schedules triplets that contain an atom, a direction, and constraints on the logical variables of the atom. E.g., FOBB starts by scheduling the query: $\langle q(1), \mathit{fromChild}, \emptyset \rangle$. Next, FOBB retrieves this triplet from the schedule and computes its parents by matching $q(1)$ to the heads of T. In this case, only event C_1 matches, so FOBB schedules $\langle C_1(x,y), \mathit{fromChild}, \{x{=}1\} \rangle$. FOBB represents event choices as first-order atoms, adhering to the EBN transformation, which represents choices as BN nodes. Note that the just scheduled triplet actually represents multiple nodes in the EBN (due to its free variable y).

The previous paragraph illustrated how FOBB computes parents. FOBB computes children as follows. If the selected atom represents a choice then it schedules the head atoms of the event; if the atom is a regular atom then it matches the atom to the body atoms of T and schedules the choices of the matching events.

Bayes-ball marks the nodes it visits. FOBB also keeps

Algorithm 1 FOBB(T, Q, E)

$S = \emptyset$, $E_R = \emptyset$
for each $q \in Q$ **do**
  $S \leftarrow S \cup \langle q, \mathit{fromChild}, \emptyset \rangle$
while $S \neq \emptyset$ **do**
  pick and remove a $\langle p, \mathit{from}, C \rangle$ from S
  $E_{p,C} \leftarrow$ GetUnifiable(p,C,E)
  if $E_{p,C} \neq \emptyset$ **then**
    Choose one $e \in E_{p,C}$ such that $\exists\theta : p\theta = e$
    $E_R \leftarrow E_R \cup e$
    $S \leftarrow S \cup \langle p, \mathit{from}, C \cup \text{Invert}(\theta) \rangle$
  if $\mathit{from} = \mathit{fromChild} \wedge E_{p,C} = \emptyset$ **then**
    if $\neg$HasMark(p,C,top) **then**
      AddMark(p,C,top)
      for each $pa \in$ GetParents(p,C,T) **do**
        $S \leftarrow S \cup \langle pa, \mathit{fromChild}, \text{Project}(C,pa) \rangle$
    if $\neg$Det(p)$\wedge\neg$HasMark(p,C,bottom) **then**
      AddMark(p,C,bottom)
      for each $ch \in$ GetChildren(p,C,T) **do**
        $S \leftarrow S \cup \langle ch, \mathit{fromParent}, \text{Project}(C,ch) \rangle$
  if $\mathit{from} = \mathit{fromParent}$ **then**
    if $E_{p,C} \neq \emptyset \wedge \neg$HasMark($e$,$\emptyset$,$\mathit{top}$) **then**
      AddMark(e,$\emptyset$,top)
      for each $pa \in$ GetParents(e,$\emptyset$,T) **do**
        $S \leftarrow S \cup \langle pa, \mathit{fromChild}, \text{Project}(\theta,pa) \rangle$
    if $E_{p,C} = \emptyset \wedge \neg$HasMark($p$,$C$,$\mathit{bottom}$) **then**
      AddMark(p,C,bottom)
      for each $ch \in$ GetChildren(p,C,T) **do**
        $S \leftarrow S \cup \langle ch, \mathit{fromParent}, \text{Project}(C,ch) \rangle$
$R \leftarrow \{a |$ HasMark(a,$\emptyset$,top)$\wedge$ground(a)$\}$
return (R, E_R)

track of marks, but here these are marks for first-order atoms. It stores the marks as triplets in a mark table; each triplet consists of an atom, a set of constraints, and a set of marks. E.g., the initial marks for the query atom and for $C_1(x, y)$ are stored as $\langle q(1), \emptyset, \{top\} \rangle$ and $\langle C_1(x, y), \{x{=}1\}, \{top\} \rangle$.

During the execution of the algorithm, it is possible that we have to specialize an atom in the marks table. This occurs for example if a subset of the nodes represented by the atom interact differently with the rest of the BN. E.g., $\langle C_1(x, y), \mathit{fromChild}, \{x{=}1\} \rangle$ has as parent $s(x, y)$ with $x{=}1$. This parent represents a set of nodes that do not all interact in the same way with the rest of the BN because of the presence of evidence. More specifically, $s(1, y)$ for $y{=}2$ is part of the evidence, and $s(1, y)$ for $y{\neq}2$ is not. Therefore, they have to be marked differently and to distinguish these cases, FOBB uses the constraints in the triplets.

When FOBB terminates, all ground atoms marked at the top together with the visited evidence atoms constitute the MRN.

FOBB generalizes SLD-resolution since it also computes proofs for the query atom. The schedule generalizes SLD-resolution's current goal and the marks can be considered a form of tabling. A difference is that FOBB does not detect failing derivations; in a longer version of this paper, we will show how to address this.

5. Conclusions

We presented a first-order version of Bayes-ball called FOBB, which finds the MRN for a given set of query and evidence atoms.

Further work will research a version of FOBB for lifted inference, which will produce a non-ground MRN. This should be possible because FOBB already works at the lifted level; the main change is to detect ground atoms that can be considered identical with regard to inference. This resembles Singla and Domingos (2008)'s approach; major differences are that FOBB is meant for directed graphs instead of undirected graphs, and that it is dependent on the query and not a compilation technique.

Acknowledgments: GOA/08/008 'Probabilistic Logic Learning' to NT. Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen) to WM. Research Fund K.U.Leuven to JS.

References

Meert, W., Struyf, J., & Blockeel, H. (2008). Learning ground CP-logic theories by leveraging Bayesian network learning techniques. *Fundamenta Informaticae*, *89*, 131–160.

Predoiu, L. (2003). Magic Bayes-Ball: Ein Bayes-Ball-Algorithmus für Bayes'sche Datalog-Programme. Master's thesis, University of Freiburg, Germany.

Shachter, R. (1998). Bayes-Ball: The rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams). *Proc. of the 14th Conf. on Uncertainty in Artificial Intelligence* (pp. 480–487).

Singla, P., & Domingos, P. (2008). Lifted first-order belief propagation. *Proc. of the 23rd AAAI Conf. on Artificial Intelligence* (pp. 1094–1099).

Vennekens, J., Denecker, M., & Bruynooghe, M. (2006). Representing causal information about a probabilistic process. *Lecture Notes in Computer Science*, *4160*, 452–464.