Approximate Belief Updating in Max-2-Connected Bayes Networks is NP-Hard

Erez Karpas

Faculty of Industrial Engineering
Technion, Haifa, Israel
karpase@technion.ac.il

Solomon Eyal Shimony and Amos Beimel

Dept. of Computer Science
Ben-Gurion University
P.O. Box 645, Beer-Sheva, Israel
{shimony, beimel}@cs.bgu.ac.il

Abstract

A max-2-connected Bayes network is one where there are at most 2 distinct directed paths between any two nodes. We show that even for this restricted topology, null-evidence belief updating is hard to approximate.

Key words: Bayes network, Complexity, Max-k-connected

1 Introduction

Bayes networks are a compact representation of the joint probability distribution over a set of random variables. Reasoning about them is of major interest in both theoretical and applied AI [5]. A Bayes network $\mathcal{B} = (N, P)$ represents a probability distribution as a directed acyclic graph N where its set of nodes V stands for random variables (in this paper, each random variables $X \in V$ takes values from a finite domain $\mathrm{Dom}(X)$), and P, a set of tables of conditional probabilities (CPTs) – one table for each node $X \in V$. For each

[1] This work was carried out while the first author was at Ben-Gurion University

possible value $x \in \mathrm{Dom}(X)$, the respective table lists the probability of the event $X = x$ given each possible value assignment to (all of) its parents. The joint probability of a complete state (assignment of values to all variables) is given by the product of $|V|$ terms taken from the respective tables [5] (where $|V|$ is the cardinality of V, i.e., the number of nodes). That is, with $\mathrm{Parents}(X)$ denoting the parents of X in G, we have:

$$\Pr(V) = \prod_{X \in V} \Pr(X|\,\mathrm{Parents}(X)).$$

Probabilistic reasoning (inference) has several forms [5,7], but only *belief updating* (defined below) is discussed here. Additionally, a distinction is made between a problem with *evidence*, which is a partial assignment $\mathcal{E}$ to some of the variables (presumably *observed* values for some of the variables), and a reasoning problem with no evidence. The belief updating problem is: compute marginal distributions for all other (non-evidence) variables given the evidence, i.e., compute $\Pr(X = x|\mathcal{E})$ for all $X \in V$ and for each possible value x of X. If $\mathcal{E} = \emptyset$, then the problem is called null-evidence belief updating.

As inference over Bayes networks is hard in the general case [1,2,6], complexity analysis of sub-classes of Bayes networks is of extreme importance: knowledge of the exact frontier of tractability impacts heavily on the type of Bayes networks one may wish to acquire from experts or learn from data [3].

A max-k-connected Bayes network is one with at most k distinct directed paths between any two nodes. In [7] it was shown that belief updating in max-k-connected Bayes networks was NP-hard to approximate for $k \geq 3$, even with no evidence, and can be done efficiently where $k = 1$ (note that this latter class is a strict superclass of poly-trees [5,7]). It was also shown that belief updating is hard for $k = 2$. However, the question whether this restricted version of the problem is easy to approximate was left open. In this paper we show that null-evidence belief updating for $k = 2$ is also hard to approximate.

2 Main Result

Definition A (relative) approximation problem [2] in max-2-connected Bayes networks consists of:

Input: A max-2-connected Bayes network $\mathcal{B}$, a node X in $\mathcal{B}$, a value $x \in \mathrm{Dom}(X)$, and an approximation error threshold ϵ.
Output: an approximation of $\Pr(X = x)$, p, such that $\Pr(X = x)(1 + \epsilon) \geq p \geq \Pr(X = x)/(1 + \epsilon)$.

Theorem 1 *Approximate belief updating in max-2-connected Bayes networks is NP-Hard.*

The proof of Theorem 1 is by reduction from the bounded degree directed Hamilton cycle problem (see [4]). The Hamilton cycle decision problem is: Given an undirected graph $G = (V, E)$, is there a cycle that passes through every vertex of G exactly once. This problem is NP-Complete even if the degree of each vertex in the graph is at most 4 (see [4]). The problem remains hard for directed graphs where the in-degree and out-degree of each vertex is at most 4 (because every undirected edge can be viewed as an incoming edge and an outgoing edge), and so it is NP-Complete for directed graphs with a total degree (incoming+outgoing) of 8.

Proof of Theorem 1. Given a directed graph G with a maximum total degree of 8, we show how to construct a max-2-connected Bayes network, where by approximating the distribution of some node s, we can decide if there is a Hamilton cycle in G.

Let $G = (V, E)$, where $|V| = n$ and $|E| = m$, be a directed graph with total degree at most 8. We construct a max-2-Connected Bayes network as follows:

(1) For each directed edge $e_i \in E$ create a multi-valued node e_i in the Bayes network. The possible values for e_i are $\{\perp, 0, 1, 2, \ldots, n-1\}$ with uniform priors (that is, the probability of each assignment a to the edge nodes is $\Pr(a) = 1/(n+1)^m$). We can interpret the values of these nodes as encoding a Hamiltonian cycle in the following way:
- Assigning $\perp$ to e_i means that e_i is not in the Hamilton cycle.
- Assigning $e_i = k \in \{0, 1, \ldots, n-1\}$ means that e_i is the $k + 1^{th}$ edge in the cycle.

(2) For each $v_i \in V$ create a binary-valued node v_i in the Bayes network. The parents of v_i are all e_k such that v_i is an end-point of e_k (that is, $e_k = (v_i, v_{i'})$ or $e_k = (v_{i'}, v_i)$ for some $v_{i'}$). Because the maximum degree of the graph is 8, the size of each of these CPTs is $(n+1)^8$.

The CPTs are such that $\Pr(v_i = T|\,\text{Parents}(v_i)) = 1$ iff v_i has exactly one incoming edge with value other than $\perp$, exactly one outgoing edge with value other than $\perp$, the value of the incoming edge is j, and the value of the outgoing edge is $(j+1) \mod n$, for some $0 \leq j \leq n-1$. This can be done easily for any assignment to $\text{Parents}(v_i)$. If there are not exactly two parents with a value other than $\perp$, $\Pr(v_i = T|\,\text{Parents}(v_i)) = 0$. Otherwise, there are exactly two parents (edges) with value other than $\perp$. If one of these edges is incoming with value j and the other is outgoing with value $(j+1) \bmod n$, then $\Pr(v_i = T|\,\text{Parents}(v_i)) = 1$. Otherwise $\Pr(v_i = T|\,\text{Parents}(v_i)) = 0$.

(3) Create a binary-valued "and" node s, with all v_i nodes as parents. That is, define the CPT of s so that $\Pr(s = T|\,\text{Parents}(s)) = 1$ iff all v_i have

value T. We avoid an exponential-size CPT by using the standard trick of actually implementing s by using a tree of 2-input "and" nodes to increase the fan-in (see [1]).

As an example of the reduction, the graph in Fig. 1, results in the Bayes network of Fig. 2. For example, u is the end-point of four edges (w, u), (x, u), (y, u), and (u, v) in G, thus, these edge nodes are the parents of the vertex node u. The CPT for node u can be seen in Table 1.

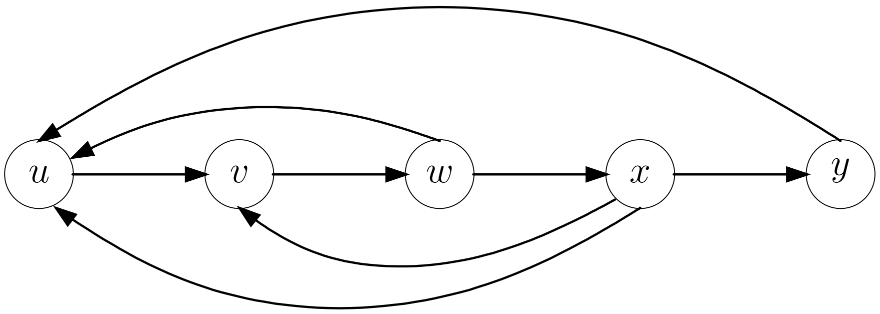

Fig. 1. The original graph G.

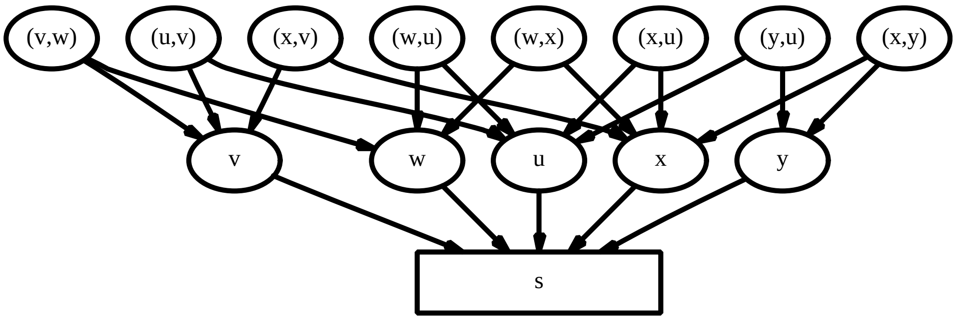

Fig. 2. 2-connected Bayes network representing G.

We now have to prove the following:

(1) The resulting Bayes network is max-2-connected.
(2) $\Pr(s = T) > 0$ iff G has a Hamilton cycle.

To see that the resulting Bayes network is max-2-connected, note that the path from any e_i node to any v_i node is of length 1, and therefore there cannot be two distinct paths from any e_i node to any v_i node (as there are no parallel edges in the Bayes network we constructed). The number of distinct paths from any e_i node to s is exactly 2. If $e_i = (v_{i_1}, v_{i_2})$, then the two paths from e_i to s are $e_i \to v_{i_1} \to s$ and $e_i \to v_{i_2} \to s$. Clearly, there are no other paths in this Bayes network, and therefore it is max-2-connected.

Next we prove that $\Pr(s = T) > 0$ iff G has a Hamilton cycle. If there is a Hamilton cycle in G, then we can choose any edge in the cycle, and assign value 0 to the corresponding node in the Bayes network. We can continue

(u,v)	(w,u)	(x,u)	(y,u)	$\Pr(u = T \mid (u,v),(w,u),(x,u),(y,u))$
1	$\perp$	$\perp$	0	1
2	$\perp$	$\perp$	1	1
3	$\perp$	$\perp$	2	1
4	$\perp$	$\perp$	3	1
0	$\perp$	$\perp$	4	1
1	$\perp$	0	$\perp$	1
2	$\perp$	1	$\perp$	1
3	$\perp$	2	$\perp$	1
4	$\perp$	3	$\perp$	1
0	$\perp$	4	$\perp$	1
1	0	$\perp$	$\perp$	1
2	1	$\perp$	$\perp$	1
3	2	$\perp$	$\perp$	1
4	3	$\perp$	$\perp$	1
0	4	$\perp$	$\perp$	1

Table 1
The CPT for node u (only the lines where $\Pr(u = T|(u,v),(w,u),(x,u),(y,u)) > 0$ are shown. $\Pr(u = T|(u,v),(w,u),(x,u),(y,u)) = 0$ for all other values of $(u,v),(w,u),(x,u),(y,u)$.

along the Hamilton cycle and assign values $1,\dots,n-1$ to the nodes in the Bayes network corresponding to the following edges in the cycle. Assign $\perp$ to all other edge nodes. Mark this assignment as a. Then $\Pr(s = T|a) = 1$, because the value of each vertex node is T with probability 1, and therefore the value of s is T with probability 1. $\Pr(s = T|a) = 1$ and $\Pr(a) = \frac{1}{(n+1)^m}$ (because priors are uniform) and, therefore, $\Pr(s = T) > 0$.

Conversely, if $\Pr(s = T) > 0$, then there exists some complete assignment a, such that $\Pr(s = T|a) > 0$. According to the CPT of node s, for every assignment χ, $\Pr(s = T|\chi)$ is either 0 or 1. Because $\Pr(s = T) > 0$, $\Pr(s = T|a)$ cannot be 0, and therefore $\Pr(s = T|a) = 1$ and $\Pr(a) > 0$. An assignment to the edge nodes gives us a complete assignment to all nodes with probability 1, so $\Pr(a) = \frac{1}{(n+1)^m}$.

Denote by $a(e_i)$ the value that a assigns to node e_i. There must exist some $e_i = (v_{i_1}, v_{i_2})$ such that $a(e_i) \neq \perp$, because otherwise, all edge nodes are assigned $\perp$, and therefore all vertex nodes will have value F with probability 1, and the value of s is F with probability 1. Let $a(e_i) = k \in \{0, 1, \dots, n-$

1}. Because $a(s) = T$, we know that all vertex nodes are assigned T, and specifically, $a(v_{i_2}) = T$. This can only happen if v_{i_2} has exactly one incoming edge with value other than $\perp$ ($a(e_i) = k$), and exactly one outgoing edge with value other than $\perp$, because otherwise we would have $a(v_{i_2}) = F$. Assume, without loss of generality, that the outgoing edge is e_j. It must have value $(k + 1) \bmod n$, because otherwise the value of v_{i_2} would have been F with probability 1. Assume $e_j = (v_{i_2}, v_{i_3})$. By the same reasoning as above we know that $a(v_{i_3}) = T$, so there exists some outgoing edge from v_{i_3} with value $(k + 2) \bmod n$. By repeating this process, we follow a cycle C in the graph.

Observe that C is simple: it could not reach a previously visited vertex, because that vertex would have 2 different incoming edges, and would be assigned value F. Additionally, the cycle must visit all vertices, because a vertex that is not visited would have 0 incoming edges, and would be assigned value F. Thus C is Hamiltonian. Therefore, it is NP-hard to decide whether $\Pr(s = T) > 0$, and thus belief updating in max-2-connected Bayes networks is hard to approximate within any bounded factor. $\square$

References

[1] G. F. Cooper, The computational complexity of probabilistic inference using Bayesian belief networks, Artificial Intelligence 42 (2-3) (1990) 393–405.

[2] P. Dagum, M. Luby, Approximating probabilistic inference in Bayesian belief networks is NP-hard, Artificial Intelligence 60 (1) (1993) 141–153.

[3] S. Dasgupta, Learning polytrees, in: Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99), San Francisco, CA, 1999, pp. 134–14.

[4] M. R. Garey, D. S. Johnson, Computers and Intractability, A Guide to the Theory of NP-completeness, W. H. Freeman and Co., 1979.

[5] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, San Mateo, CA, 1988.

[6] S. E. Shimony, Finding MAPs for belief networks is NP-hard, Artificial Intelligence 68 (2) (1994) 399–410.

[7] S. E. Shimony, C. Domshlak, Complexity of probabilistic reasoning in directed-path singly-connected Bayes networks, Artificial Intelligence 151 (1-2) (2003) 213–225.