A Query Processing Algorithm for Hierarchical Markov Networks

C.J. Butz
Department of Computer Science
University of Regina
Regina, Saskatchewan, Canada S4S 0A2
E-mail: butz@cs.uregina.ca

J. Liu
Department of Computer Science
University of Regina
Regina, Saskatchewan, Canada S4S 0A2
E-mail: liuji114@cs.uregina.ca

Abstract

Hierarchical Markov networks (HMNs) were recently proposed as a faithful representation of Bayesian networks. In this paper, we propose a query processing algorithm for HMNs. This method takes one query processing algorithm for a traditional Markov network and extends it to a hierarchy of Markov networks. Experimental results explicitly demonstrate the effectiveness of our approach. The work here will then be useful to any problem utilizing Bayesian networks, such as traditional information retrieval, web search, user profiling, multi-agents and e-commerce.

1 Introduction

Probability theory provides a formal foundation for uncertain reasoning [12]. In particular, *Bayesian networks* (BNs) [6] use the notion of *probabilistic conditional independence* [13] to facilitate the acquisition of a joint probability distribution. Several researchers have naturally suggested that Bayesian networks be applied in traditional information retrieval [2, 7, 10, 16], web search [9], user profiling [11], multi-agents [5, 12, 17] and e-commerce [4].

To facilitate the inference process, a BN is typically transformed into a *Markov network* (MN) [12, 13]. A MN is an acyclic hypergraph together with a marginal distribution for each hyperedge in the hypergraph. The local propagation algorithm in [3] has long been regarded in the BN community as the best method for query processing. This method involves fixing a particular *jointree* [1] for an acyclic hypergraph. More recently, experimental results suggested that fixing a particular jointree leads to increased computation when compared to using an acyclic hypergraph [14].

On the other hand, *hierarchical Markov networks* (HMNs) [15] were proposed as an alternative representation of BNs. As the name suggests, an HMN is a hierarchy of MNs. One advantage of the HMN representation over the MN representation is that the HMN representation is guaranteed to represent precisely those independencies encoded in a BN [15]. No study, however, has put forth a method for probabilistic inference in HMNs.

In this paper, we develop a query processing algorithm for HMNs. Our method extends the one in [14] from a single MN to multiple MNs. Optimizing probabilistic inference means taking advantage of independencies to reduce computation during query processing. One salient feature of our approach is that we can utilize every independence holding in a BN. It is then not surprising that our experimental results indicate query processing in HMNs can be more efficient than in traditional MNs.

This paper is organized as follows. In Section 2, we review three kinds of probabilistic networks. Our query processing algorithm for HMNs is described in Section 3. In Section 4, experimental results are reported. The conclusion is given in Section 5.

2 Probabilistic Networks

2.1 Bayesian networks

Let U be a finite set of discrete random variables, each with a finite set of mutually exclusive states. Obviously, it may be impractical to define a joint distribution on U directly: for example, one would have to specify 2^n entries for a distribution over n binary variables. BNs utilize *conditional independencies* [13] to facilitate the acquisition of probabilistic knowledge.

Let X, Y and Z be disjoint subsets of variables in R. Let x, y, and z denote arbitrary values of X, Y and Z, respectively. We say Y and Z are *conditionally independent* given X under the joint probability distribution p, denoted $I(Y, X, Z)$, if $p(y|x, z) = p(y|x)$, whenever $p(x, z) > 0$. $I(Y, X, Z)$ can be equivalently written as

$$p(y, x, z) \quad = \quad \frac{p(y, x) \cdot p(x, z)}{p(x)}. \qquad (1)$$

A *Bayesian network* (BN) [6] is a pair $\mathcal{B} = (D, C)$. In this pair, D is a *directed acyclic graph* (DAG) on a set U of variables, and $C = \{p(a_i|P_i) \mid a_i \in D\}$ is the corresponding set of *conditional probability tables* (CPTs), where P_i denotes the *parent set* of variable a_i in the DAG D. We will use the terms BN and DAG interchangeably if no confusion arises. The *d-separation* method [6] can be used to read independencies from a DAG. For instance, $I(d, b, e)$, $I(c, \emptyset, f)$, $I(h, g, i)$ and $I(defh, b, g)$ all hold by d-separation in the DAG D in Fig. 1.

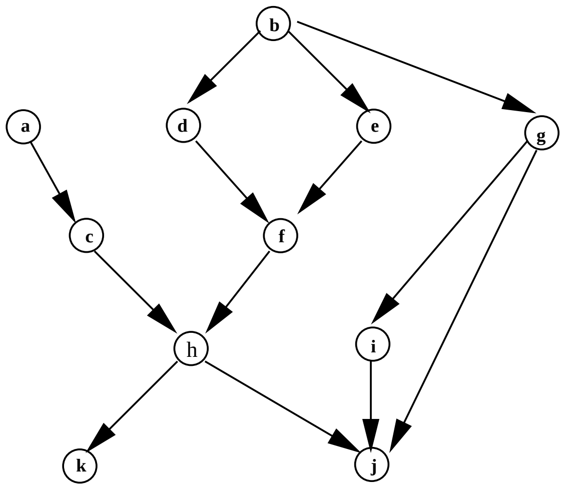

Figure 1. A Bayesian network on variables $U = \{a, b, c, d, e, f, g, h, i, j, k\}$.

Example 1 *Consider the BN $\mathcal{B} = (D, C)$, where D is the DAG in Fig. 1 on $U = \{a, b, c, d, e, f, g, h, i, j, k\}$ $= abcdefghijk$, and C is the corresponding set of CPTs. The conditional independencies encoded in the DAG D indicate that the product of the CPTs in C define a unique joint probability distribution $p(U)$:*

$$\begin{aligned} p(U) &= p(a)p(b)p(c|a)p(d|b)p(e|b)p(f|d,e)p(g|b) \\ & \quad p(h|c,f)p(i|g)p(j|g,h,i)p(k|h). \end{aligned} \quad (2)$$

2.2 Markov networks

In order to facilitate the processing of queries, a BN is usually transformed into a MN. The *moralization* and *triangulation* procedures [6] are applied to transform the DAG in a triangulated (chordal) undirected graph. A triangulated graph can be conveniently represented as an *acyclic hypergraph* [1], where each hyperedge in the hypergraph represents a maximal clique in the triangulated graph.

Example 2 *One possible acyclic hypergraph, obtained by applying the moralization and triangulation procedures on the DAG in Fig. 1, is $\{ac, bdef, bfg, cfh, fgh, ghij, hk\}$. This MN expresses the joint distribution in Ex. 1 as*

$$\begin{aligned} & p(U) \\ = & \frac{p(bdef)p(bfg)p(fgh)p(ghij)p(cfh)p(ac)p(hk)}{p(bf)p(fg)p(gh)p(fh)p(c)p(h)}. \end{aligned} \quad (3)$$

Traditionally, in the probabilistic reasoning literature, an acyclic hypergraph is fixed as a *jointree* [1]. One possible jointree for the above acyclic hypergraph is depicted in Fig. 2.

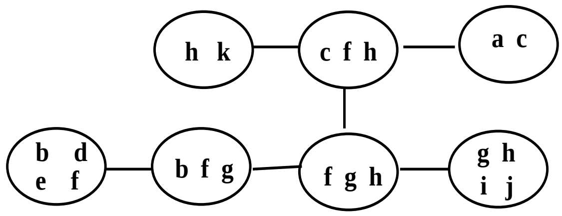

Figure 2. One possible jointree for the acyclic hypergraph in Example 2.

Note that both probabilistic inference methods [3, 14] will store marginals for every term in the right side of Eq. (3).

2.3 Hierarchical Markov networks

In [15], it was suggested that BNs be represented as *hierarchical Markov networks* (HMNs). Due to lack of space, we refer the reader to [15] for details on the HMN representation and its construction.

Example 3 *The BN in Fig. 1 can be represented by the unique HMN illustrated in Fig. 3. Here $\mathcal{H}_0 = \{ac, cfh, bdefgh, ghij, hk\}$, $\mathcal{H}_1 = \{bg, bde, def, fh\}$, $\mathcal{H}_2 = \{c, f\}$, $\mathcal{H}_3 = \{gh, gi\}$, $\mathcal{H}_4 = \{bd, be\}$, and*

$$\begin{aligned} child(bdefgh) &= \mathcal{H}_1, \\ child(cfh) &= \mathcal{H}_2, \\ child(ghij) &= \mathcal{H}_3, \\ child(bde) &= \mathcal{H}_4. \end{aligned}$$

We refer to $\mathcal{H}_0$ as the root MN in the HMN **H** $=$ $\{\mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4\}$

This HMN encodes the following independency information:

$$p(U) = \frac{p(ac)p(cfh)p(hk)p(bdefgh)p(ghij)}{p(c)p(h)p(fh)p(gh)},$$

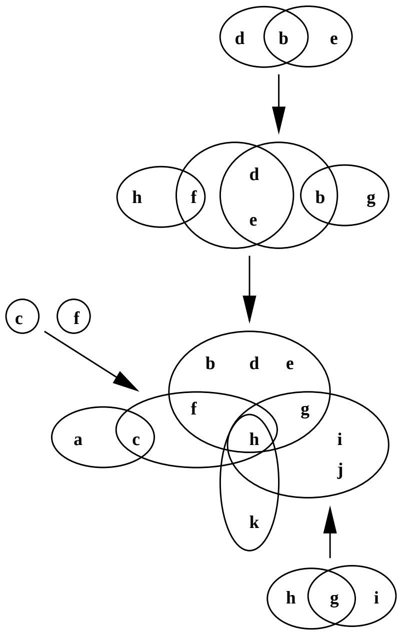

Figure 3. The HMN for the DAG D in Fig. 1.

$$
\begin{aligned}
p(bdefgh) &= \frac{p(fh)p(def)p(bde)p(bg)}{p(f)p(de)p(b)}, \\
p(cf) &= p(c)p(f), \\
p(bde) &= \frac{p(bd)p(be)}{p(b)}, \\
p(ghi) &= \frac{p(gh)p(gi)}{p(g)}.
\end{aligned}
$$

3 Processing Queries in HMNs

Before presenting our algorithm, we first make it clear which marginals are stored in the knowledge base.

3.1 The Knowledge Base

A hyperedge X is called a *leaf*, if X does not contain another MN. The *context* of MN $\mathcal{H}$, denoted $context(\mathcal{H})$, is the set X of variables on which $\mathcal{H}$ is defined.

Example 4 *Hyperedge hk is a leaf hyperedge. On the contrary, $ghij$ is not a leaf hyperedge as it contains the MN $\mathcal{H}_3$. Here $context(\mathcal{H}_3) = ghi$.*

We can choose to store three types of marginal distributions for the hyperedges in the HMN representation: (i) those for leaf hyperedges, (ii) those necessarily computed for type (i), and (iii) those hyperedges X, where $context(child(X)) \subset X$.

An example of a type (i) marginal is $p(def)$. To illustrate a type (ii) marginal, consider what is involved to compute the type (i) marginal $p(def)$. Since we are given the initial conditional $p(f|de)$, we must compute $p(def)$ as follows:

$$p(def) = p(de)p(f|de).$$

The task then turns to computing $p(de)$. It can be easily verified that $p(de)$ can only be computed after $p(bde)$ is obtained. Since bde is a hyperedge in the HMN, the computed marginal $p(bde)$ is not discarded. Instead, it is stored in the knowledge base together with the type (i) marginals. Finally, the type (iii) marginals in our running example are $p(cfh)$ and $p(ghij)$.

3.2 A Query Processing Algorithm

Our method for processing a query in a HMN can be viewed as recursively applying the method suggested in [14], which is described next.

Suppose we wish to compute $p(X)$ from a MN $\mathcal{H}$. The *selective reduction algorithm* [8] is applied for this purpose. First, mark the nodes in X. Next, repeatedly apply the following two operations until neither can be applied: (i) delete an unmarked node that occurs in only one hyperedge, and (ii) delete a hyperedge that is contained by another hyperedge.

Example 5 *The selective reduction of the set bj of variables in the MN $\mathcal{H}$ of Example 2 is $\{bgh, fgh, ghj\}$. The query $p(bj)$ can then be processed as:*

$$
\begin{aligned}
p(bj) &= \sum_{fgh} \frac{p(bfg)p(fgh)p(ghj)}{p(fg)p(gh)} \\
&= \sum_{gh} \frac{p(ghj)}{p(gh)} \sum_{f} \frac{p(bfg)p(fgh)}{p(fg)} \\
&= \sum_{gh} \frac{p(ghj)}{p(gh)} \sum_{f} p(bfgh) \\
&= \sum_{gh} \frac{p(bgh)p(ghj)}{p(gh)} \\
&= \sum_{gh} p(bghj).
\end{aligned}
$$

Since the selective reduction algorithm is an efficient method for processing queries in a single MN [14] and a HMN is a hierarchy of MNs, we can process queries in a HMN by recursively calling the selective reduction algorithm.

Suppose we wish to compute $p(X)$ from a HMN $\mathbf{H}$. Call the selective reduction algorithm of X on $\mathcal{H}_0$. While there exists a hyperedge h such that $p(h)$ cannot be obtained by marginalization from a stored marginal, call the selective reduction algorithm of h on $child(h)$.

Example 6 *Consider answering the query $p(bj)$ in the HMN $\mathbf{H} = \{\mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4\}$ of Example 3. As previously mentioned, the root MN of $\mathbf{H}$ is $\mathcal{H}_0$. The selective reduction of bj on $\mathcal{H}_0$ is $\{bgh, ghj\}$. This means that the desired marginal $p(bj)$ can be computed as*

$$p(bj) \quad = \quad \sum_{gh} \frac{p(bgh)p(ghj)}{p(gh)}. \tag{4}$$

Here $p(ghj)$ can be obtained from $p(ghij)$ stored in the knowledge base. On the contrary, $p(bgh)$ cannot be obtained by marginalizing any stored table. Hence, we need to call the selective reduction algorithm again. The child of hyperedge $bdefgh$ in $\mathcal{H}_0$ is the MN $\mathcal{H}_1$, i.e., $child(bdefgh) = \mathcal{H}_1$. The selective reduction of bgh on $\mathcal{H}_1$ is $\mathcal{H}_1$ itself. Thus, the missing marginal $p(bgh)$ can be computed by

$$\begin{aligned} p(bgh) &= \sum_{def} \frac{p(bg)p(bde)p(def)p(fh)}{p(b)p(de)p(f)} \\ &= \sum_{de} \frac{p(bg)p(bde)}{p(b)p(de)} \sum_{f} \frac{p(def)p(fh)}{p(f)} \\ &= \sum_{de} \frac{p(bg)p(bde)}{p(b)p(de)} \sum_{f} p(defh) \\ &= \sum_{de} \frac{p(bg)p(bde)p(deh)}{p(b)p(de)} \\ &= \frac{p(bg)}{p(b)} \sum_{de} \frac{p(bde)p(deh)}{p(de)} \\ &= \frac{p(bg)p(bh)}{p(b)}. \end{aligned}$$

After $p(bgh)$ is obtained, the original query $p(bj)$ can be answered using Eq. (4).

4 Experimental Results

In this section, empirical studies are conducted on three approaches to answering queries, namely, the jointree approach [3], the acyclic hypergraph approach [14], and the HMN approach presented in Section 3. Our comparison is based on the BN in Fig. 1, which is a complicated version of the well-known “Asia” BN commonly used in the probabilistic reasoning literature.

Five queries are considered: $p(ai)$, $p(de)$, $p(dg)$, $p(eh)$, $p(kj)$. Four of the five queries involve variables that are close together for the following reasoning. Xiang [17] states that probabilistic queries $p(X)$ tend to be localized in practice, namely, the variables in X are indeed likely to be found in close proximity. Nevertheless, we have also included $p(ai)$ where variables a and i are spread apart in the BN.

Given that the BN in Fig. 1 can be represented by the acyclic hypergraph in Example 2, the jointree in Fig. 2, or the HMN in Fig. 3, we count the number of multiplications (*), divisions $(/)$, and additions (+) needed to answer each of the five queries. The experimental results obtained are listed in Table 1.

Table 1. Required operations for five queries.

p(ai)	Jointree	Acyclic Hypergraph	HMN
$*$	48	16	16
$/$	20	8	8
$+$	16	24	24
Total	104	48	48
p(de)	**Jointree**	**Acyclic Hypergraph**	**HMN**
$*$	0	0	0
$/$	0	0	0
$+$	12	12	4
Total	12	12	4
p(dg)	**Jointree**	**Acyclic Hypergraph**	**HMN**
$*$	16	16	8
$/$	8	8	4
$+$	20	20	8
Total	44	44	20
p(eh)	**Jointree**	**Acyclic Hypergraph**	**HMN**
$*$	32	8	8
$/$	16	4	4
$+$	24	16	8
Total	72	28	20
p(kj)	**Jointree**	**Acyclic Hypergraph**	**HMN**
$*$	40	8	8
$/$	20	4	4
$+$	32	16	16
Total	92	28	28

The experimental results reported in Table 1 clearly demonstrate that our method for processing queries in a HMN is superior to the jointree method [3], which is regarded as the state-of-the-art algorithm in the BN community. In addition, our proposed method shows modest improvement over the acyclic hypergraph approach [14]. Our future experiments will be based on the large real-world BNs found in sample data sets.

There are two primary factors that contribute to the improvement of the well established jointree method. First of all, fixing a particular jointree will always benefit certain queries while hindering others (see [14] for a more thorough discussion). Secondly, query optimization means taking advantage of independencies while processing a query. Only the HMN encodes those and only those independencies encoded in a BN. For instance, $I(c, \emptyset, f)$, $I(h, g, i)$, and $I(b, de, f)$ are encoded in the BN and the HMN. On the contrary, these independencies are not encoded in the jointree and acyclic hypergraph representations. Failure to represent known independency information can lead to increased computation.

5 Conclusion

In this paper, we suggested a query processing algorithm for hierarchical Markov networks. Our method can be seen as recursively applying an efficient query processing algorithm developed for a single Markov network. The experimental results in Table 1 demonstrate the effectiveness of our approach. Since the hierarchical Markov network has many advantages over the Markov network representation [15], and given the explanation of the favorable experimental results at the end of Section 4, we feel that the hierarchical Markov network representation could eventually become the standard representation of Bayesian networks. Thus, the encouraging results reported in this paper are useful to any work applying BNs, including traditional information retrieval [2, 7, 10, 16], web search [9], user profiling [11], multi-agents [5, 12, 17] and e-commerce [4].

References

[1] C. Beeri, R. Fagin, D. Maier and M. Yannakakis. On the desirability of acyclic database schemes, *Journal of the Association for Computing Machinery*, **30**, 479-513, 1983

[2] R. Fung and B. Del Favero, Applying Bayesian networks to information retrieval. *Communication of ACM*, **38**(3), 42-48,57, 1995.

[3] F.V. Jensen. S.L. Lauritzen and K.G. Olesen, Bayesian updating in causal probabilistic networks by local computations, *Computational Statistics Quarterly*, **4**, 269-282, 1990

[4] J. Ji, L. Zheng and C. Liu, The intelligent electronic shopping system based on Bayesian customer modeling. *First Asia-Pacific Conference on Web Intelligence*, 574-578, 2001.

[5] S. Lee, C. Sung and S. Cho, An effective conversational agent with user modeling based on Bayesian network. *First Asia-Pacific Conference on Web Intelligence*, 428-432, 2001.

[6] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 1998

[7] B. Ribeiro-Neto, I. Silva and R. Muntz, “Bayesian network models for information retrieval”, Soft Computing in Information Retrieval: Techniques and Applications. F. Crestani and G. Pasi (Eds.), Springer Verlag, 259–291, 2000.

[8] R.E. Tarjan and M. Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM J. Comput.*, 13(3):566-579, 1984

[9] I. Silva, B. Ribeiro-Neto, P. Calado, E. Moura and N. Ziviani, Link-based content-based evidential information in a belief network model. *Twenty-third International Conference on Research and Development in Information Retrieval*, 96–103, 2000.

[10] H.R. Turtle and W.B. Croft, Inference networks for document retrieval. *Thirteenth International Conference on Research and Development in Information Retrieval*, 1-24, 1990.

[11] S.K.M. Wong and C.J. Butz, A Bayesian approach to user profiling in information retrieval. *Technology Letters*, **4**(1), 50-56, 2000.

[12] S.K.M. Wong and C.J. Butz, Constructing the dependency structure of a multi-agent probabilistic network, *IEEE Transactions on Knowledge and Data Engineering*, **13**, 3, 395-415, 2001.

[13] S.K.M. Wong, C.J. Butz and D. Wu. On the implication problem for probabilistic conditional independency, *IEEE Transactions on Systems, Man, and Cybernetics*, Part A: Systems and Humans, Vol. 30, No. 6, 785-805, 2000.

[14] S.K.M. Wong, D. Wu and C.J. Butz, Probabilistic reasoning in Bayesian networks: a relational database approach, *Proceedings of the Sixteenth Canadian Conference on Artificial Intelligence(AI)*, 583-590, 2003

[15] S.K.M. Wong, C.J. Butz and D. Wu, On undirected representations of Bayesian networks, *ACM SIGIR Workshop on Mathematical/Formal Models in Information Retrieval(MF/IR)*, 52-59, 2001

[16] S.K.M. Wong and Y.Y. Yao, On modeling information retrieval with probabilistic inference, *ACM Transactions on Information Systems*, **13**, 38-68, 1995.

[17] Y. Xiang, *Probabilistic Reasoning in Multi-Agent Systems: A Graphical Models Approach*, Cambridge Publishers, 2002.