Granular Jointree Probability Propagation

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

P. Lingras
Department of Math and Computing Science
Saint Mary's University
Halifax, Nova Scotia, Canada B3H 3C3
Pawan.Lingras@stmarys.ca

Abstract— **Jointree computation continues to be central to the theory and practice of probabilistic expert systems. Recent research has incorporated granular structures to facilitate propagation in the jointree. In this paper, we propose a method for granular jointree probability propagation. Our method extends the previous works by allowing the granular levels to communicate with each other. It is explicitly demonstrated that our granular approach increases the amount of parallelism during probability propagation.**

I. Introduction

Bayesian networks [7] are a formal framework for uncertainty management. However, the probabilistic inference process is actually conducted on a *jointree* [1], [8] constructed from a Bayesian network. The *Hugin* [3] architecture is regarded as the premier algorithm for probability propagation in jointrees. One limitation of this algorithm, however, is that a jointree node cannot send a message to a neighbour until it has received messages from all its other neighbours [8].

More recently, several researchers have proposed representing Bayesian networks with multiple jointrees, called *hierarchical Markov networks* [10] and *nested jointrees* [5] (see also the *maximal prime decomposition* [6] of Bayesian networks). Although these representations sound different, they are similar in the sense that the representation is a hierarchy of jointrees. That is, each node in the root jointree may have a nested jointree. The nested jointrees may in turn possess other nested jointrees themselves in a recursive fashion. Nevertheless, the important point is that these granular structures are only utilized to facilitate the traditional Hugin algorithm. In other words, the nested jointrees only help compute the messages for the root jointree in a more efficient manner. However, no study has ever investigated a method for propagating probabilties in a truly granular fashion.

In this paper, we propose a method for *granular* jointree probability propagation. Given a Bayesian network, we represent it as an hierarchical Markov network (a hierarchy of jointrees). More importantly, given two neighbour nodes in the root jointree, we allow these two nodes to pass messages to each other. Unlike other methods, however, we also allow the nested jointrees to pass messages to each other. That is, our method allows messages to be propagated at the "root" level as well as the "nested" level. Our granular approach to probability propagation is quite significant as we explicitly demonstrate that this method can increase the amount of parallel computation. That is, we show that one root node can send a message to a neighbour, while still waiting for messages from its other neighbours.

This paper is organized as follows. Section 2 reviews traditional jointree probability propagation. Advanced jointree probability propagation is briefly discussed in Section 3. In Section 4, we introduce the notion of granular jointree probability propagation. The conclusion is presented in Section 5.

II. Traditional Jointree Probability Propagation

Let U be a finite set of discrete variables, each with a finite domain. Let V be the Cartesian product of the variable domains. A *joint probability distribution* [8] is a function p on V such that $0 \leq p(v) \leq 1$ for each configuration $v \in V$ and $\sum_{v \in V} p(v) = 1.0$. Henceforth, we may say p is on U with the frame V understood. The *marginal distribution* $p(X)$ for $X \subseteq U$ is defined as $\sum_{U-X} p(U)$. If $p(X) > 0$, then the *conditional probability distribution* $p(Y|X)$ is defined as $p(YX)/p(X)$. A *potential* is a function ϕ on V such that $\phi(v)$ is a nonnegative real number and $\sum_{v \in V} \phi(v)$ is positive, i.e., at least one $\phi(v) > 0$.

Definition 1: [7] A *Bayesian network* is a directed acyclic graph (DAG) D together with a conditional probability distribution $p(a_i|P_i)$ for each variable a_i in D, where P_i denotes the parent set of variable a_i in D.

Example 1: One DAG D for a Bayesian network on $U = \{a, b, c, d, e, f, g, h, i, j, k, l\}$ is shown in Figure 1. Given DAG D, the corresponding conditional probability distributions are $p(a)$, $p(b|a)$, $p(c|a)$, $p(d|b, c)$, $p(e|a)$, $p(f)$, $p(g|d, f)$, $p(h|e)$, $p(i|e, g, h)$, $p(j)$, $p(k|h, j)$, and $p(l|k)$.

The DAG graphically encodes *conditional independencies* [9] regarding the variables in U. More specifically, the independencies encoded in a DAG indicate that the product of the conditionals is a unique joint probability distribution $p(U)$ on U. In our example, the independencies encoded in the DAG of Figure 1 indicates that the product of the corresponding conditionals is a unique joint distribution $p(U)$, namely,

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

The important point is that Bayesian networks provide a semantic modeling tool which greatly facilitate the acquisition of probabilistic knowledge. Assuming binary variables,

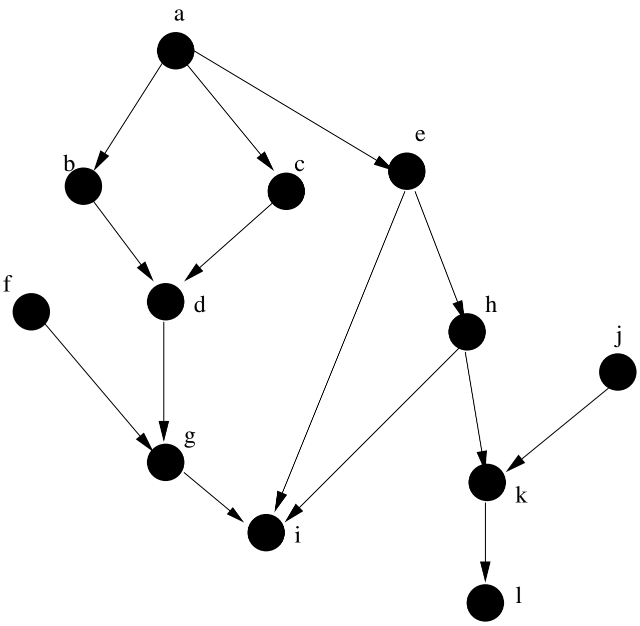

Fig. 1. A Bayesian network.

specifying $p(U)$ directly would require 4095 prior probabilities ($2^{12}-1$ for twelve variables), while the Bayesian network conditionals can be specified using only 33 conditional probabilities. We now turn our attention to probabilistic inference.

Probabilistic inference means computing the probability values of a particular set of variables given that other variables take on certain values, i.e., computing $p(X|Y=y)$.

Example 2: For simplicity, suppose the Bayesian network in Figure 1 is only defined on the set $\{a,b,c,d\}$ of variables. Thus, the joint distribution is defined as:

$$p(a,b,c,d) \quad = \quad p(a)\cdot p(b|a)\cdot p(c|a)\cdot p(d|b,c).$$

Consider what is involved to compute the marginal $p(d)$. By definition,

$$\begin{aligned}
p(d) &= \sum_{a,b,c} p(a)\cdot p(b|a)\cdot p(c|a)\cdot p(d|b,c)\\
&= \sum_{b,c} p(d|b,c)\cdot \sum_{a} p(a)\cdot p(b|a)\cdot p(c|a)\\
&= \sum_{b,c} p(d|b,c)\cdot \sum_{a} \phi_1(a,b,c)\\
&= \sum_{b,c} p(d|b,c)\cdot \phi_1(b,c)\\
&= \sum_{c}\sum_{b} p(d|b,c)\cdot \phi_1(b,c)\\
&= \sum_{c}\sum_{b} \phi_2(b,c,d)\\
&= \sum_{c} \phi_2(c,d)\\
&= \phi_2(d),
\end{aligned}$$

where $\phi_1(a,b,c) = p(a)\cdot p(b|a)\cdot p(c|a)$, and $\phi_2(b,c,d) = p(d|b,c)\cdot \phi_1(b,c)$.

The above example demonstrates the *necessity* of jointrees. Consider the context of the potentials that were created in the last example, namely, $\{a,b,c\}$, $\{b,c,d\}$, $\{c,d\}$, and $\{d\}$. These four subsets of attributes can be organized as a jointree. In other words, $\{\{a,b,c\}$, $\{b,c,d\}$, $\{c,d\}$,$\{d\}\}$ is an *acyclic hypergraph* [1]. Every jointree (acyclic hypergraph) is equivalent to a chordal undirected graph.

The DAG is converted into a chordal undirected graph by applying the moralization and triangulation procedures. The *moralization* [7] D^m of a DAG D on set U of variables is the unique undirected graph defined as:

$$D^m = \{(a,b)|a,b \in F_i \text{ for at least one variable } a_i \in U\},$$

where F_i is the family set $\{a_i\}\cup P_i$ of a_i.

Example 3: The *moralization* D^m of the DAG D in Figure 1 is depicted in Figure 2.

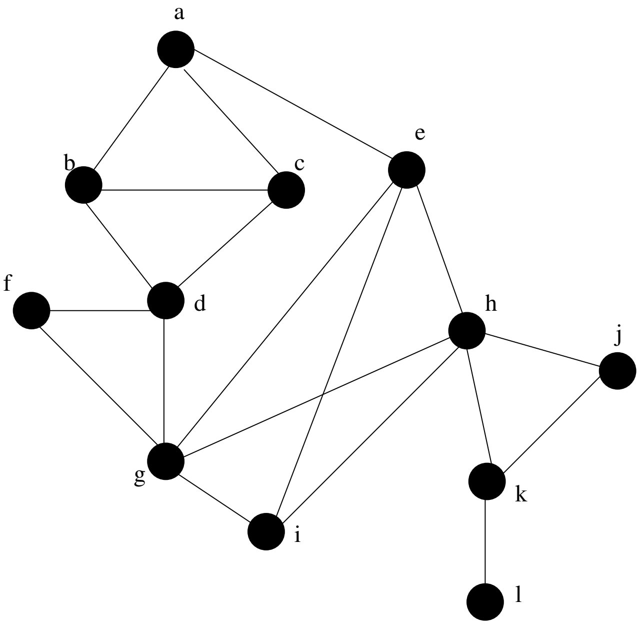

Fig. 2. The *moralization* of the Bayesian network in Figure 1.

If necessary, undirected edges are added to the moralization D^m to create a chordal graph D^t [4]. An undirected graph is *chordal* (or *triangulated*) if each cycle of length four or more possesses an edge (a,b) between two nonadjacent nodes a and b in the cycle.

Example 4: The undirected graph D^m in Figure 2 is not triangulated. For instance, the cycle $(a,c),(c,d),(d,g),(g,e),(e,a)$ does not possess a chord. One possible *triangulation* D^t of $\mathcal{D}^m$ is obtained by adding an edge between every pair of variables in $\{a,b,c,d,e,g\}$.

The maximal cliques of this triangulated graph are $\{d,f,g\}$, $\{a,b,c,d,e,g\}$, $\{e,g,h,i\}$, $\{h,j,k\}$, and $\{k,l\}$. Each maximal clique is represented by a node in a jointree, defined as follows.

Definition 2: [8] A *jointree* is a tree with the property that any variable in two nodes is also in every *separating set* on the path between the two.

Example 5: One possible jointree for the Bayesian network in Figure 1 is depicted in Figure 3. We label the nodes of this jointree as dfg, $abcdeg$, $eghi$, hjk and kl. The separating sets are $\{d, g\}$, $\{e, g\}$, $\{h\}$, and $\{k\}$.

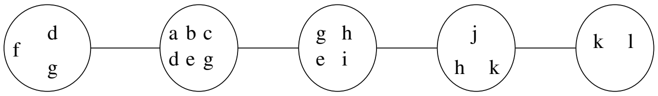

Fig. 3. One possible *jointree* for the Bayesian network in Figure 1.

We now focus on propagating probabilities in a jointree. We will refer to the method in [3] as the *Hugin* jointree propagation method. The Hugin method is regarded as the best jointree propagation algorithm [2], [8].

The objective of jointree propagation is to compute a marginal distribution $p(X)$ for each jointree node X. The first step is to construct an initial potential $\phi(X)$ for each node X in the jointree. For each configuration x of X, set $\phi(x) = 1.0$. Next, assign each conditional $p(a_i|P_i)$ in the Bayesian network to precisely one node X containing $\{a_i\} \cup P_i$ and set $\phi(X) = \phi(X) \cdot p(a_i|P_i)$.

Example 6: The conditionals for the Bayesian network in Figure 1 can be assigned to the constructed jointree in Figure 3 as follows. First, we set $\phi_1(d, f, g) = 1.0$, $\phi_2(a, b, c, d, e, g) = 1.0$, $\phi_3(e, g, h, i) = 1.0$, $\phi_4(h, j, k) = 1.0$ and $\phi_5(k, l) = 1.0$. Second, we can multiply the conditionals to the jointree potentials as follows:

$$\begin{aligned}
\phi_1(d, f, g) &= \phi_1(d, f, g) \cdot p(f) \cdot p(g|d, f), \\
\phi_2(a, b, c, d, e, g) &= \phi_2(a, b, c, d, e, g) \cdot p(a) \cdot p(b|a) \cdot \\
&\quad\ p(c|a) \cdot p(d|b, c) \cdot p(e|a), \\
\phi_3(e, g, h, i) &= \phi_3(e, g, h, i) \cdot p(h|e) \cdot p(i|e, g, h), \\
\phi_4(h, j, k) &= \phi_4(h, j, k) \cdot p(j) \cdot p(k|h, j), \\
\phi_5(k, l) &= \phi_5(k, l) \cdot p(l|k).
\end{aligned}$$

The Hugin jointree propagation method works as follows [8]. One node is chosen as the root node. Each separator S also has a potential $\phi(S)$ initialized to all ones.

Rule 1. Each nonroot node waits to send its message to a given neighbour until it has received messages from all its other neighbours.

Rule 2. The root node waits to send messages to its neighbours until it has received messages from all of them.

Rule 3. When a node is ready to send its message to a particular neighbour, it computes the message by marginalizing its current table to its intersection with this neighbour, and then sends the message to the separator between it and the neighbour.

Rule 4. When a separator receives a message ψ from one of its two nodes, it divides the message ψ by its current table ϕ, sends the quotient ψ/ϕ on to the other node, and then replaces ϕ with ψ.

Rule 5. When a node receives a message, it replaces its current table with the product of that table and the message.

Rules 1 and 2 force the propagation to move in to the root and then back out to the leaves. At the end of the inward pass, the table at the root r is $p(r)$. At the end of the outward pass, the tables on all of the nodes are marginals. Moreover, all messages in the outward pass are marginal distributions.

For example, suppose we pick node dfg in Figure 3 as the root. During the inward pass, messages are propagated from node kl to node dfg. During the outward pass, messages are propgated from node dfg to node kl. Thus, eight messages are propagated *sequentially*.

III. Granular Representations of Bayesian Networks

Although several researchers [5], [6] have proposed various granular representations of Bayesian networks, we will focus our discussion on *hierarchical Markov networks* (HMNs) [10].

Example 7: The Bayesian network in Figure 1 can be faithfully represented by the HMN in Figure 4.

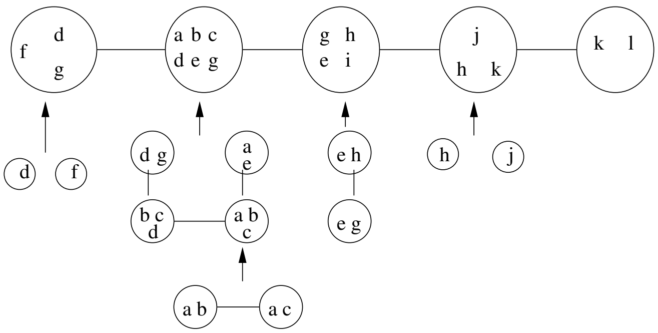

Fig. 4. The unique HMN for the Bayesian network in Figure 1.

Granular representations are desirable as they can encode more independency information than a traditional jointree.

Example 8: In the Bayesian network of Figure 1, variables g and h are conditionally independent given variable e. This independency is encoded in the HMN of Figure 4 but *not* in the conventional jointree of Figure 3.

The important point is that the query processing algorithms in [5], [6], [10] only allow the nested jointree to help its parent node send messages to its neighbours. For example, the nested jointree for node $eghi$ in Figure 4 only helps $eghi$ to send its messages to its neighbour nodes $abcdeg$ and hjk. In the next section, we extend this approach by allowing the nested jointrees to talk to each other.

IV. Granular Probability Propagation

We begin by showing some limitations of only using the nested jointrees to facilitate the Hugin propagation algorithm.

The CPTs of the Bayesian network are assigned to nodes in the HMN in the same fashion as before. In particular, it can be verified that the CPT $p(f)$ must be assigned under the root node dfg, the CPTS $p(a)$, $p(b|a)$, $p(c|a)$ and $p(e|a)$ must be assigned under root node $abcdeg$, and finally $p(h|e)$ must be assigned under root node $gehi$.

Given the above initial assignment of CPTS, the Hugin propagation algorithm has the following restrictions during the outward pass:

(i) Node $abcdeg$ must wait for its neighbour dfg to send message $p(dg)$ before $abcdeg$ is able to compute $p(abcdeg)$.

(ii) Only after computing $p(abcdeg)$ is $abcdeg$ allowed to send message $p(ge)$ to its neighbour $gehi$.

(iii) Node $gehi$ must wait until it receives $p(ge)$ before it is able to compute $p(gehi)$.

(iv) Only after computing $p(gehi)$ is $gehi$ allowed to send message $p(h)$ to its neighbour hjk.

We can, however, facilitate this propagation by better utilizing the nested jointrees. By the above remarks regarding the assignment of CPTs to the HMN, node $abcdeg$ is able to compute $p(e)$ as follows:

$$\sum_{a,b,c} p(a)p(b|a)p(c|a)p(e|a) \quad = \quad \sum_{a,b,c} p(abce) \quad = \quad p(e).$$

Even though the Hugin algorithm wants node $abcdeg$ to send $p(ge)$ to node $gehi$, let us allow node $abcdeg$ to send $p(e)$ to $gehi$, as illustrated in Figure 5. Again, by the above remarks regarding CPT assignment, node $gehi$ is now able to compute its message $p(h)$:

$$\sum_{e} p(e)p(h|e) \quad = \quad \sum_{e} p(eh) \quad = \quad p(h).$$

Node $gehi$ can now send $p(h)$ to its neighbour node hjk, as depicted in Figure 5.

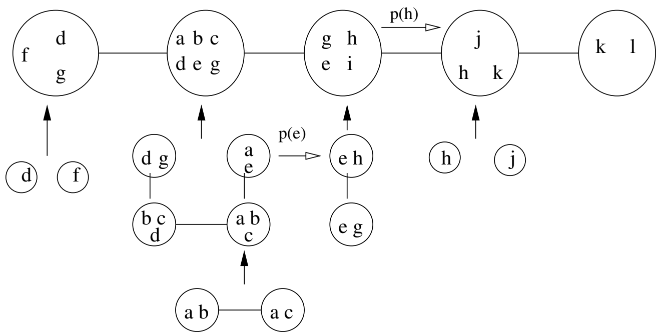

Fig. 5. By allowing the internal structure of node $abcdeg$ to send the marginal $p(e)$ to the internal structure of node $gehi$, $gehi$ is able to send its message $p(h)$ to its neighbour hjk even before node dfg sends $p(dg)$ to node $abcdeg$.

The important point is that the Hugin method prohibits node $gehi$ from sending its message $p(h)$ to node hjk until its receives message $p(ge)$ from node $abcdeg$. Moreover, $abcdeg$ is unable to send $p(ge)$ to $gehi$ until it has received message $p(dg)$ from node dfg. Our approach allows messages to be sent in a granular fashion. Whereas $abcdeg$ is unable to send $p(ge)$ to $gehi$, it is indeed able to send the smaller message $p(e)$. In turns out, however, that receiving $p(e)$ gives node $gehi$ enough information to send its message $p(h)$ to its neighbour hjk. Our approach increases the amount of parallel computation; while node dfg is sending $p(dg)$ to $abcdeg$, node $gehi$ is sending $p(h)$ to hjk.

V. Conclusion

To the best of our knowledge, this is the first work to propose *granular* probability propagation in Bayesian networks. Although previous works [5], [6], [10] have suggested representing Bayesian networks in a granular fashion, the nested levels are only utilized to help the *Hugin* [3] algorithm send messages in the root jointree. Our work extends [5], [6], [10] by also allowing the nested levels to communicate with each other. This granular approach is very promising as we explicitly demonstrated in Section 4 that our method can increase the amount of parallel computation during probability propagation. Future work will formalize and implement the granular approach put forth in this manuscript.

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*, vol. 30, 479–513, 1983.

[2] C. Huang and A. Darwiche. Inference in belief networks: a procedural guide. *International Journal of Approximate Reasoning*, 11(1): 393–405, 1994.

[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] U. Kjaerulff. "Triangulation of graphs - Algorithms giving small total state space," Technical report R 90-09, University of Aalborg, Denmark, 1990.

[5] U. Kjaerulff. Nested junction trees. *Proc. 13th Conf. on Uncertainty in Artificial Intelligence*, 302–313, 1997.

[6] K.G. Olesen and A.L. Madsen. Maximal prime subgraph decomposition of Bayesian networks. *IEEE Transactions on Systems, Man, and Cybernetics*, B, 32(1):21-31, 2002.

[7] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 1988.

[8] G. Shafer. *Probabilistic Expert Systems*, Society for Industrial and Applied Mathematics, 1996.

[9] S.K.M. Wong, C.J. Butz and D. Wu. On the implication problem for probabilistic conditional independency. *IEEE Trans. Sys. Man. Cyb.*, A, 30(6):785–805, 2000.

[10] 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*, 52-59, 2001.