Lossless Decomposition of Bayesian Networks

Dan Wu
School of Computer Science
University of Windsor
Windsor Ontario
Canada N9B 3P4

Abstract

In this paper, we study the problem of information preservation when decomposing a single Bayesian network into a set of smaller Bayesian networks. We present a method that losslessly decomposes a Bayesian network so that no conditional independency information is lost and no extraneous conditional independency information is introduced during the decomposition.

1. Introduction

Recent research on Bayesian networks has seen the trend of extending the Bayesian network model to handle large, complex, or dynamic domains. Typical such extensions include the *Object Oriented* Bayesian network model (OOBN) (Koller & Pfeffer 1997), the *Multiply Sectioned* Bayesian network model (MSBN) (Xiang 2002), and the *dynamic* Bayesian network model (DBN) (Murphy 2002). All these extensions aim to provide knowledge representations and probabilistic inference algorithms for large, complex, or dynamic domains.

Divide and conquer is an important problem-solving technique, especially for conceptually large and difficult problems. It divides the original problem into smaller subproblems and conquers the subproblems so as to solve the original large and difficult problem. To some extent, the OOBN model and the MSBN model can be considered as applications of divide and conquer technique. For example, modeling an aircraft using a single Bayesian network would be an enormous task, if not possible, for any human domain expert. A Bayesian network modeling an aircraft, if constructed, will have a huge number of variables representing different parts of the aircraft, and no single human expert can possibly possess all the knowledge to build such a network. However, one can divide the problem of modeling an aircraft into smaller subproblems, such as modeling the aircraft engine, the communication system, the safety system, etc., and then combine all the components together. In the MSBN model, this means building Bayesian networks for the engine, the communication system, and the safety system. By combining Bayesian networks constructed for different components of an aircraft, one would obtain the Bayesian network model of the whole aircraft. In the OOBN model, this means creating classes for different components of an aircraft such as aircraft engine class, safety system class, and etc. Each class encapsulates potentially complex internal structure represented as a Bayesian network. Objects created from these classes are organized as a traditional Bayesian network while each object itself encapsulates an embedded Bayesian network.

In this paper, we study an important topic that has seldom been discussed previously in these extensions. That is, the problem of information preservation. By information, we mean the *conditional independency* (CI) encoded in the graphical structure of a Bayesian network. The importance of information preservation is self-explanatory. For instance, different agents in the MSBN model may encode CIs in their respective subdomains. These CIs should also hold in the original problem domain. For otherwise, what the agent believes in its subdomain is not true in the original domain. Therefore, it would be highly desirable to preserve the CIs existing in the problem domain during the divide and conquer process such that no existing CIs are removed *and* no extraneous CIs are introduced during the process. In other words, one would expect that the CIs encoded in the problem domain is preserved, and thus equivalent to the CIs encoded in the subdomains after dividing the original domain. More specifically, we investigate the following problem:

> Assuming that a large and complex problem domain can be modelled as a single Bayesian network B (at least theoretically), how would one divide (decompose) this single Bayesian network into a set of smaller Bayesian networks SB_i, so that the CIs encoded in B is equivalent to the CIs encoded in all SB_i?

A very simple and intuitive decomposition method will be presented and the equivalence of CIs encoded naturally follows from the decomposition. One salient feature of the proposed decomposition is that it is not entirely a graphical operation, instead, the decomposition method makes full use of the algebraical properties of Bayesian networks that were recently revealed (Wu & Wong 2004).

The paper is organized as follows. In Section 2, relevant background knowledge is reviewed. In Section 3, the proposed lossless decomposition method is presented. A motivating example is first studied. A generalization of the example is then thoroughly investigated. Conclusion is in Section 4.

2. Background

A *Bayesian network* (BN) (Pearl 1988) is a probabilistic graphical model defined over a set V of random variables. A BN consists of a graphical and a numerical component. The graphical component is a *directed acyclic graph* (DAG). Each vertex in the DAG corresponds one-to-one to a random variable in V, and we thus use the term variable and node interchangeable. The numerical component is a set $\mathcal{C}$ of *conditional probability distributions* (CPDs). For each variable $v \in V$, there exists one-to-one a CPD $p(v|\pi_v)$ in $\mathcal{C}$, where π_v denotes parents of v in the DAG. The product of the CPDs in $\mathcal{C}$ yields a *joint probability distribution* (JPD) over V as

$$p(V) \quad = \quad \prod_{v \in V} p(v|\pi_v),$$

and we call this equation the *Bayesian factorization* (BF). Obviously, the BF is *unique* to a BN.

One of the most important notions in BNs is conditional independency. Let $X,\ Y,\ Z$ be three subsets of V, we say that X is *conditional independent* (CI) of Z given Y, denoted $I(X,\ Y,\ Z)$, if and only if

$$p(X,\ Y,\ Z) = \frac{p(X,\ Y) \cdot p(Y,\ Z)}{p(Y)}$$

when $p(Y) \neq 0$.

The DAG of a BN encodes CIs that hold among nodes in V. More precisely, given a *topological ordering* of all the variables in a DAG so that parents always precede their child in the ordering, every node is conditional independent of all its predecessors given its parents. That is, a topological ordering induces a set $\{I(a_i,\ \pi_{a_i},\ \{a_1,\ \ldots,\ a_{i-1}\} - \pi_{a_i}),\ a_i \in V\}$ of CIs which are called *causal input list* (CIL) (Verma & Pearl 1988). The DAG of a BN may have many different topological orderings, each of which induces a different CIL. However, all these CILs are *equivalent* since their respective closures under the *semi-graphoid* (SG) axioms (Pearl 1988) are the same. The closure of a CIL contains all the CIs encoded in a BN, which can not only be numerically verified by the definition of CI with respect to the BF, but also can be graphically identified by the *d-separation* criterion from the DAG of the BN (Pearl 1988).

A BN can be *moralized and triangulated* to produce a junction tree for inference purposes (Jensen 1996). Extensive research has been done on triangulating a BN to produce a junction tree that facilitates more effective inference (Kjaerulff 1990; Larranaga *et al.* 1997; Wong, Wu, & Butz 2002). The notion of junction tree plays an important role in the lossless decomposition in the following section. However, we are not concerned if the junction tree facilitates better inference in this paper.

We generalize the notion of topological ordering to a subset of variables in a DAG. Let V represent the set of all variables in a DAG, a subset $X \subseteq V$ of variables is said to be in a topological ordering with respect to the DAG, if for each variable $a_i \in X$, the variables in the intersection of a_i's ancestors and X precede a_i in the ordering.

Although a BN is traditionally defined as a probabilistic graph model (i.e., the DAG) augmented with a set of CPDs, alternatively and equivalently, a BN can also be defined in terms of the CPD factorization of a JPD as follows.

Definition 1 Let $V = \{a_1,\ \ldots,\ a_n\}$. Consider the CPD factorization of $p(V)$ below:

$$p(V) \quad = \prod_{a_i \in V,\ a_i \notin A_i,\ A_i \subseteq V} p(a_i|A_i). \tag{1}$$

For $i = 1,\ \ldots,\ n$, if (1) each $a_i \in V$ appears exactly once as the head [1] of one CPD in the above factorization, and (2) the graph obtained by depicting a directed edge from b to a_i for each $b \in A_i$ is a DAG, then the DAG drawn according to (2) and the CPDs $p(a_i|A_i)$ in Eq. (1) define a BN. In fact, the factorization in Eq. (1) is a Bayesian factorization.

By looking at the CPD factorization of a JPD, if it satisfies the conditions in Definition 1, then this CPD factorization defines a BN and it is in fact a BF. From this BF, one can produce the DAG of the BN according to condition (2) in Definition 1, from which one may further obtain any topological ordering and its associated CIL. Hence, it can be claimed that the BF of a BN encapsulates all necessary information to infer any CIL of a BN. On the other hand, from any CIL of a DAG, one can obtain the unique BF of the BN. Therefore, the notions of CIL and BF are synonymous. If one of them is known, the other is also known. Since the closure of a CIL contains all CIs encoded in a BN, one may also say that the BF of a BN and all CIs encoded in a BN are synonymous.

3. Lossless Decomposition

As the BN model becomes popular and establishes itself as a successful framework for uncertain reasoning, efforts have been made to extend BNs to model large and complex domains. Having realized that modeling a large and complex domain using a single BN is not feasible, if not possible, both the OOBN and MSBN model, which are notable extensions of the traditional BN model, have tried to model different portions of the large and complex domain as BNs under the terminology "agent" in MSBN and "class" in OOBN. By combining the BN representations of these smaller portions, if one desires, a single large BN modeling the whole problem domain can be recovered. However, during these endeavors of extending BNs, the problem of information preservation was not addressed. In other words, whether the CIs encoded in the single large BN is equivalent to the CIs encoded in the BNs modeling smaller portions of the original problem domain is questionable. There are two directions or two scenarios to address the information preservation problem. One scenario is to study how to divide (decompose) a single BN into many smaller BNs without losing any CI information. The other scenario is to assume

[1]Given a CPD $p(X|Y)$, X is called the *head* and Y is called the *tail* of this CPD. A marginal $p(X)$ can also be considered as a CPD with its tail empty (i.e., $p(X|\emptyset)$)

that there are small BNs constructed for different portions of a problem domain which can be presumably modelled by a single BN, and one needs to verify if the CIs encoded in all these small BNs is equivalent to the CIs in the single BN. In this paper, we study the first scenario and leave the second scenario to another paper.

Imagine a large and complex problem domain which can be modelled as a single large BN. If one divides (or decomposes) this BN into a set of smaller BNs each of which models a portion of the original domain, from the information preservation perspective, it is then reasonably expected and highly desirable that the CIs encoded in the single large BN is equivalent to all the CIs encoded in all the smaller BNs. Such a decomposition is called a *lossless decomposition*. In the following, a simple method for losslessly decomposing a BN into a set of smaller BNs, which will be called *sub*BNs in this paper, is presented.

An Example

We begin with a simple example to illustrate the idea of the proposed lossless decomposition.

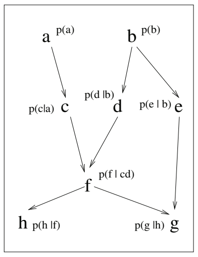


Figure 1: The Asia travel BN.

Consider the Asia travel BN defined over $V = \{a, \ldots, h\}$ from (Lauritzen & Spiegelhalter 1988). Its DAG and CPDs associated with each node are depicted in Figure 1. The JPD $p(V)$ is obtained as:

$$\begin{aligned} p(V) &= p(a)\cdot p(b)\cdot p(c|a)\cdot p(d|b)\cdot p(e|b) \\ &\quad \cdot p(f|cd)\cdot p(g|ef)\cdot p(h|f). \end{aligned} \tag{2}$$

The DAG in Figure 1 is moralized and triangulated (Huang & Darwiche 1996) so that a junction tree such as the one in Figure 2 (i) is constructed. This junction tree consists of 6 cliques depicted as round rectangles, denoted $c_1 = ac, c_2 = bde, c_3 = cdf, c_4 = def, c_5 = fh, c_6 = efg$, and 5 separators depicted as smaller rectangles attached to the edge connecting two incidental cliques, denoted $s_1 = c, s_2 = de, s_3 = df, s_4 = f, s_5 = ef$.

If one can construct a small BN (*sub*BN) defined over variables in each $c_i, i = 1, \ldots 6$, and show that CIs encoded in the BN in Figure 1 is equivalent to CIs encoded in the to-be-constructed *sub*BNs for each c_i, then the goal of lossless decomposition is achieved. And this is the basic idea of the proposed decomposition method.

Following this line of reasoning, consider the following equation in Eq. (3) which is called *Markov factorization* (MF) in this paper. It relates the JPD $p(V)$ to marginals $p(c_i)$ for each clique and marginals $p(s_j)$ for each separator in the junction tree:

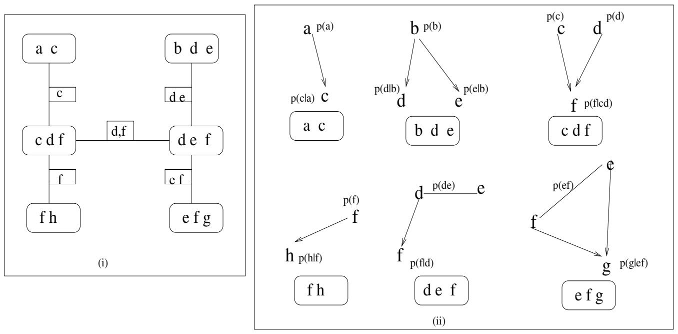


Figure 2: (i) A junction tree constructed from the DAG in Figure 1. (ii) The *sub*BNs obtained by decomposing the BN in Figure 1 with respect to the junction tree in (i).

$$p(V) = \frac{p(c_1)\cdot p(c_2)\cdot p(c_3)\cdot p(c_4)\cdot p(c_5)\cdot p(c_6)}{p(s_1)\cdot p(s_2)\cdot p(s_3)\cdot p(s_4)\cdot p(s_5)}. \tag{3}$$

The objective here is to obtain a BF for each numerator in Eq. (3). Comparing Eq. (2) with Eq. (3), one can see that both represent the same JPD $p(V)$, but in different algebraic forms. The apparent difference is that Eq. (2) is a factorization of CPDs and thus has no denominators, while Eq. (3) is a factorization of marginals and has denominators. If one wants to *algebraically* equate these two questions, one option is to manipulate Eq. (2) as follows by multiplying and dividing $\Pi_{j=1}^{5} p(s_j)$ at the same time to Eq. (2), which will result in[2]:

$$\begin{aligned} p(V) &= \overbrace{[a, c|a]}^{c_1}\cdot\overbrace{[b, d|e, e|b]}^{c_2}\cdot\overbrace{[f|cd]}^{c_3}\cdot\overbrace{[1]}^{c_4}\cdot\overbrace{[h|f]}^{c_5}\cdot\overbrace{[g|fe]}^{c_6}\cdot \\ &\quad \frac{c\cdot de\cdot df\cdot f\cdot ef}{c\cdot de\cdot df\cdot f\cdot ef} \end{aligned} \tag{4}$$

It is noted that all the CPDs in Eq.(2) still exist in Eq.(4), together with the newly multiplied separator marginals $\Pi_{j=1}^{5} p(s_j)$ (as numerators and denominators). All these CPDs are regrouped into different square bracket in Eq.(4), and each square bracket corresponds to a clique in the junction tree. A CPD is assigned to a bracket corresponding to a clique c_i as long as the union of the head and tail of this CPD is a subset of c_i. A CPD can only be assigned to one arbitrary clique if the union of its head and tail are subsets of two or more cliques. We use $\phi_{\mathbf{c_i}}$, called *potential*, to denote the result of the multiplication in each square bracket, and we have

$$\begin{aligned} \phi_{c_1}(ac) &= \quad p(a)\cdot p(c|a), & \phi_{c_4}(def) &= 1 \\ \phi_{c_2}(bde) &= p(b)\cdot p(d|e)\cdot p(e|b), & \phi_{c_5}(fh) &= p(h|f) \\ \phi_{c_3}(cdf) &= \quad p(f|cd), & \phi_{c_6}(efg) &= p(g|fe) \end{aligned} \tag{5}$$

Recall that the idea of the proposed lossless decomposition is to obtain a BF for each $p(c_i)$, we thus examine carefully each numerator and denominator in Eq. (4).

[2]Due to limited space, we write a for $p(a)$, $b|d$ for $p(b|d)$, etc.

It is obvious that: $\phi_{c_1}(ac) = p(a) \cdot p(c|a) = p(ac)$ and $\phi_{c_2}(bde) = p(b) \cdot p(d|b) \cdot p(e|b) = p(bde)$. In other words, the factorizations of $\phi_{c_1}(ac)$ and $\phi_{c_2}(bde)$ as shown in Eq. (4) are already in the form of BFs respectively by Definition 1, therefore, two *sub*BN can be created for clique c_1 and c_2 as shown in Figure 2 (ii).

Consider $\phi_{c_5}(fh) = p(h|f)$, evidently, this is not a BF. To make it a BF, we can multiply it with the separator marginal $p(f)$ that was multiplied as numerator in Eq. (4), and this results in $\phi_{c_5}(fh) = p(h|f) \cdot p(f) = p(fh)$, which is a BF now and thus defines a *sub*BN for clique c_5 shown in Figure 2 (ii).

For $\phi_{c_6}(efg) = p(g|ef)$, we can multiply it with the separator marginal $p(ef)$ which results in $\phi_{c_6}(efg) = p(g|ef) \cdot p(ef) = p(efg)$. Again this is a BF now and thus defines a *sub*BN for clique c_6 shown in Figure 2 (ii).

So far, we have successfully obtained BFs for $p(c_1)$, $p(c_2)$, $p(c_5)$, and $p(c_6)$, and we have consumed the separator marginals $p(f)$ and $p(ef)$ during this process. We still need to make the remaining $\phi_{c_3}(cdf) = p(f|cd)$ and $\phi_{c_4}(def) = 1$ BFs respectively by consuming the remaining separator marginals, i.e., $p(c)$, $p(de)$ and $p(df)$.

In order to make $\phi_{c_3}(cdf) = p(f|cd)$ a BF, we need to multiply it with $p(cd)$, however, we only have the separator marginals $p(c)$, $p(de)$ and $p(df)$ at our disposal. It is easy to verify that we cannot mingle $p(de)$ with $\phi_{c_3}(cdf) = p(f|cd)$ to obtain $p(cdf)$. Therefore, $p(de)$ has to be allocated to $\phi_{c_4}(def)$ such that $\phi_{c_4}(def) = 1 \cdot p(de)$. We now only have $p(df)$ at our disposal for making $\phi_{c_4}(def)$ a BF. Note that $p(df) = p(d) \cdot p(f|d)$, and this factorization helps make $\phi_{c_4}(def) = p(de)$ a BF by multiplying $p(f|d)$ with $\phi_{c_4}(def)$ to obtain $\phi_{c_4}(def) = p(de) \cdot p(f|d) = p(def)$. We are now left with the separator marginal $p(c)$ and $p(d)$ (from the factorization of the separator marginal $p(df)$) and $\phi_{c_3}(cdf)$, and $p(c)$ and $p(d)$ have to be multiplied with $\phi_{c_3}(cdf)$ to yield $\phi_{c_3}(cdf) = p(f|cd) \cdot p(c) \cdot p(d) = p(cdf)$. We have thus so far successfully and algebraically transformed each ϕ_{c_i} into a BF of $p(c_i)$ by incorporating the separator marginals (or its factorization) multiplied. Each of these BFs of $p(c_i)$ define a *sub*BN for its respective clique c_i shown in Figure 2 (ii).

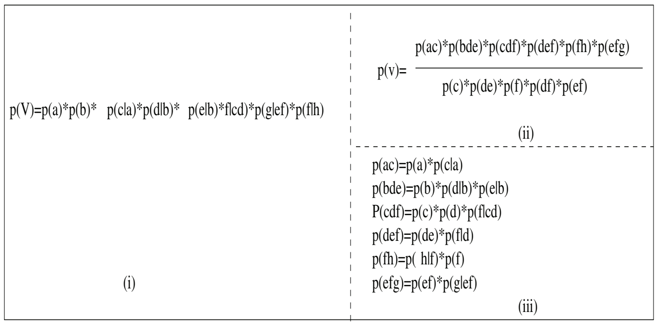
Table 1 briefly summarizes this process of "making" BFs of $p(c_i)$ for each clique c_i. It can be seen that this process is simply a process of properly allocating separator marginals multiplied in Eq. (4).

It is perhaps worth pointing out that a BN may produce different junction trees via moralization and triangulation. The algebraic manipulation just demonstrated depends on the form of the MF in Eq. (3), which is determined by the particular junction tree structure in Figure 2 (i). In other words, the structure of the junction tree somewhat determines the decomposition, and we thus call the junction tree in Figure 2 (i) the *skeleton* of the decomposition. Henceforth, when we refer to the decomposition of a BN, it not only refers to the *sub*BNs produced as those in Figure 2(ii), but also includes the skeleton of the decomposition such as the junction tree in Figure 2(i).

Interested readers can verify that a similar algebraic manipulation bearing the same idea of allocating separator marginals exists for each different junction tree structure that can be produced from the given BN.

	receives	result
ϕ_{c_1}	nothing	$p(a) \cdot p(c\|a) = p(ac)$
ϕ_{c_2}	nothing	$p(b) \cdot p(d\|b) \cdot p(e\|b) = p(bde)$
ϕ_{c_3}	$\underline{p(c)}, \underline{p(d)}$	$p(f\|cd) \cdot \underline{p(c)} \cdot \underline{p(d)} = p(cdf)$
ϕ_{c_4}	$\underline{p(de)}, \underline{p(f\|d)}$	$\underline{p(de)} \cdot \underline{p(f\|d)} = p(def)$
ϕ_{c_5}	$\underline{p(f)}$	$p(h\|f) \cdot \underline{p(f)} = p(fh)$
ϕ_{c_6}	$\underline{p(ef)}$	$p(g\|ef) \cdot \underline{p(ef)} = p(efg)$

Table 1: Allocating separator marginals, the underlined terms are either the separator marginals or from the factorization of a separator marginal.


p(V)=p(a)*p(b)* p(c|a)*p(d|b)* p(e|b)*f|cd)*p(g|ef)*p(f|h)

(i)

p(v)= p(ac)*p(bde)*p(cdf)*p(def)*p(fh)*p(efg) / p(c)*p(de)*p(f)*p(df)*p(ef)

(ii)

p(ac)=p(a)*p(c|a)
p(bde)=p(b)*p(d|b)*p(e|b)
P(cdf)=p(c)*p(d)*p(f|cd)
p(def)=p(de)*p(f|d)
p(fh)=p(h|f)*p(f)
p(efg)=p(ef)*p(g|ef)

(iii)

Figure 3: (i) The original BF. (ii) The MF with respect to the junction tree in Figure 2 (i). The BFs of the *sub*BNs produced in Figure 2 (ii).

It still remains to be shown if the CIs encoded in the BN in Figure 1 is equivalent to the CIs encoded in the decomposition in Figure 2. In fact, the equivalence follows naturally from the algebraical transformation used to produce the decomposition. Consider the three equations in Figure 3. We have just demonstrated how to algebraically transform the BF in Figure 3 (i) to the equations in Figure 3 (ii) (the MF) and (iii) (BFs for *sub*BNs). By substituting those BFs in Figure 3 (iii) for the numerators in Figure 3 (ii), one will obtain the BF in Figure 3 (i). In other words, one can algebraically transform back and forth between the equation in Figure 3 (i) and the equations in Figure 3 (ii) and (iii). Any CI that holds with respect to the equation in Figure 3 (i) will also hold with respect to the equations in Figure 3 (ii) and (iii), and vice versa. This means that both the original BN and its decomposition encodes the same CI information.

The Lossless Decomposition Method

One may perhaps attribute the successful lossless decomposition example to sheer luck. In the following, we will show that this is not a coincidence but an unavoidable elegant consequence.

Every clique c_i in the junction tree was initially associated with a clique potential ϕ_{c_i}. Every clique potential has to mingle with some appropriate separator marginal or its factorization if necessary to be transformed into a BF. This perfect arrangement of separator marginals is not a coincidence, in fact, it can always be achieved as we explain below.

Assigning either a separator marginal or the factor(s) in its factorization to a clique potential, as shown before, must satisfy one necessary condition, namely, condition (1) of Definition 1, in order for the product of the clique potential with the allocated separator marginal(or its factorization) to be a BF. That's to say, for each ϕ_{c_i}, we need a CPD with a_j as head for each $a_j \in c_i$. If a variable, say a_j, appears m times in m cliques in the junction tree, then each of these m cliques will need a CPD with a_j as head. However, the original BN only provides one CPD with a_j as head, and we are short of $m-1$ CPDs (with a_j as head). Fortunately, m cliques containing a_j implies the junction tree must have exactly $m-1$ separators containing the variable a_j (Huang & Darwiche 1996), therefore the $m-1$ needed CPDs with a_j as head will be supplied by the $m-1$ separator marginals (or their factorizations). This analysis leads to a simple procedure to allocate separator marginals.

Procedure: Allocate Separator Marginals (ASM)

Step 1. Suppose the CPD $p(a_i|\pi_{a_i})$ in the BF is assigned to a clique c_k to form ϕ_{c_k}. If the variable a_i appears in a separator s_{kj} between c_k and c_j, then draw a small arrow originating from a_i in the separator s_{kj} and pointing to the clique c_j. If variable a_i also appears in other separators in the junction tree, draw a small arrow on a_i in those separators and point to the neighboring clique away from clique c_k's direction. Repeat this for each CPD $p(a_i|\pi_{a_i})$ in the BF of a given BN.

Step 2. Examine each separator s_i in the junction tree, if the variables in s_i all pointing to one neighboring clique, then the separator marginal $p(s_i)$ will be allocated to that neighboring clique , otherwise, $p(s_i)$ has to be factorized so that the factors in the factorization can be assigned to appropriate clique indicated by the arrows in the separator.

The procedure ASM can be illustrated using Figure 4. If all variables in the same separator are pointing to the same neighboring clique, that means the separator marginal as a whole (without being factorized) will be allocated to the neighboring clique, for example, the separator marginals $p(c)$, $p(de)$, $p(f)$, and $p(ef)$ in the figure. If the variables in the separator are pointing to different neighboring cliques, that means the separator marginal has to be factorized before the factors in the factorization can be allocated according to the arrow. For example, the separator marginal $p(df)$ has to be factorized so that the factor $p(d)$ is allocated to $\phi_{c_3}(cdf)$ and $p(f|d)$ is allocated to $\phi_{c_4}(def)$. (The factorization of a separator marginal will be further discussed shortly.)

Although an appropriate allocation of the separator marginals can always be guaranteed to satisfy condition (1) of Definition 1, one still needs to show that such an allocation will not produce a directed cycle when verifying condition (2) of Definition 1. It is important to note that a directed cycle can be created in a directed graph if and only if one draws a directed edge from the descendant of a node to the node itself.

Consider a clique c_i in a junction tree and its neighboring

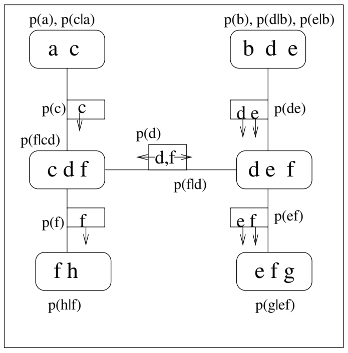


Figure 4: Allocating separate marginals by the procedure ASM.

cliques. Between c_i and each of its neighboring clique, say clique c_j, is a separator s_{ij} whose separator marginal $p(s_{ij})$ or some factors in its factorization can possible be allocated to the clique potential ϕ_{c_i}. As the example in the previous subsection shows, sometimes, the separator marginal ϕ_{c_i} as a whole will be allocated to ϕ_{c_i}; sometimes, some factors in the factorization of $p(s_{ij})$ will be allocated to ϕ_{c_i}. Suppose the separator marginal $p(s_{ij})$ is allocated to ϕ_{c_i}. If one follows the rule of condition (2) in Definition 1 to draw directed edges based on the original CPDs assigned to ϕ_{c_i} and the newly allocated separator marginal $p(s_{ij})$, no directed cycle will be created, because the original CPDs assigned to ϕ_{c_i} are from the given BN, which will not cause any cycle, and the variables in s_{ij} will be ancestors of all other variables in the clique, which will not create any cycle as well. Suppose the separator marginal $p(s_{ij})$ has to be factorized first as a product of CPDs, and only some of the CPDs in the factorization will be allocated to c_i (and the rest will be allocated to c_j). In this case, it is possible that the CPDs in the factorization allocated to c_i will cause a directed cycle if $p(s_{ij})$ is not factorized appropriately . For example, in the previous subsecton, we decomposed the separator marginal $p(df)$ as $p(df) = p(d) \cdot p(f|d)$. In fact, we could have decomposed it as $p(df) = p(f) \cdot p(d|f)$ and assigned the factor $p(d|f)$ to c_3, which would result in $\phi_{c_3}(cdf) = p(c) \cdot p(f|d) \cdot p(f|cd)$. It is easy to verify that ϕ_{c_3}, after incorporating the allocated CPD $p(d|f)$, satisfies the condition (1) but not (2) of Definition 1, which means that $\phi_{c_3}(cdf) = p(c) \cdot p(f|d) \cdot p(f|cd) \neq p(cdf)$ and it is not a BF. It is important to note that the incorrect factorization $p(df) = p(f) \cdot p(d|f)$ does not follow the topological ordering of the variables d and f (d should precede f in the ordering) with respect to the original DAG, in which f is a descendant of d. Drawing a directed edge from f to d, as dictated by the CPD $p(d|f)$, would mean a directed edge from the descendant of d, namely, the variable f to the variable d itself, and this is exactly the cause of creating a directed cycle. However, if we factorize $p(df)$ as we did previously, there will be no problem. This is because when we factorize $p(df)$ as $p(df) = p(d) \cdot p(f|d)$, we were following the topological ordering of the variables d and f with respect to the original DAG such that the heads of the CPDs in the factorization are not ancestors of their respective tails in the original DAG.

Therefore, if the procedure ASM indicates that a separator marginal $p(s_i)$ has to be factorized before it can be allocated to its neighboring clique, then $p(s_i)$ must be factorized based on a topological ordering of the variables in s_i with respect to the original DAG.

Obtaining the factorization of a separator marginal $p(s_i)$ with respect to the topological ordering of the variables in s_i is not as difficult as it seems. [3] Factorizing $p(s_i)$ is in fact no difference than creating a BF of $p(s_i)$ (or creating a BN over s_i). According to (Pearl 1988), for each variable x in s_i, this amounts to to find out a minimal subset of s_i, denoted $MS(x)$, which are predecessors of x with respect to the topological ordering of s_i, such that, given $MS(x)$, x is conditional independent of all its other predecessors (excluding $MS(x)$) in the original DAG. $p(s_i)$ can then be factorized as $p(s_i) = \prod_{x \in s_i} p(x|MS(x))$. Finding $MS(x)$ for each $x \in s_i$ can be done by tracing back from x through all possible paths in the original DAG to find out all its ancestors. For each such path, if it contains variables which are predecessors of a with respect to the topological ordering of s_i, then the predecessor closest to a is added to $MS(x)$.

To summarize, the proposed lossless decomposition method for a given BN is as follows.

Procedure: Decompose

```
Input:   a BN
Output:  the skeleton and the subBNs
Step 1.  Obtain a junction tree from the given BN. Assign all
         CPDs in the BF of the given BN to a proper clique of
         the junction tree.
Step 2.  Invoke ASM procedure to allocate separator marginals.
Step 3.  Factorizing separator marginals if dictated by the result of
         ASM.
Step 4.  Construct the subBNs for each clique in the junction tree
         obtained in step 1 using the outcomes of step 2 and 3.
Step 4.  Return the junction tree and the subBNs obtained in Step
         1 and 4 as output.
```

Theorem 1 The procedure *Decompose* produces a lossless decomposition of Bayesian networks.

4. Conclusion and Discussion

In this paper, we have investigated the problem of losslessly decomposing a BN into a set of smaller *sub*BNs. The proposed decomposition method produces a decomposition skeleton which is a junction tree and a set of *sub*BNs each of which corresponds to a clique in the junction tree. The CI information encoded in the original given BN is the same as the CI information encoded in the skeleton and the *sub*BNs in the decomposition. This is simply because the original BN and the decomposition are algebraically identical as explained by Figure 3. A lossless decomposition is desirable when dividing a large and complex BN into a set of smaller ones, because it guarantees that the CI information before the decomposition and after the decomposition is the same. For example, in the MSBN model, each agent is represented by a BN, the combined knowledge from all agents is the union of the CIs from each agent. It would be desirable that the combined knowledge also holds in the problem domain. If not, that means some agent may have some CIs only valid in its subdomain but not valid in the problem domain.

[3]Due to limited spaces, a complete implemented algorithm will not be presented here but its idea is illustrated briefly. The proof of Theorem 1 will also be omitted.

Acknowledgment

The authors wish to thank referees for constructive criticism and financial support from NSERC, Canada.

References

Huang, C., and Darwiche, A. 1996. Inference in belief networks: A procedural guide. *International Journal of Approximate Reasoning* 15(3):225–263.

Jensen, F. 1996. *An Introduction to Bayesian Networks*. UCL Press.

Kjaerulff, U. 1990. Triangulation of graphs—algorithms giving small total state space. Technical report, JUDEX, Aalborg, Denmark.

Koller, D., and Pfeffer, A. 1997. Object-oriented bayesian networks. In *Thirteenth Conference on Uncertainty in Artificial Intelligence*, 302–313. Morgan Kaufmann Publishers.

Larranaga, P.; Kuijpers, C.; Poza, M.; and Murga, R. 1997. Decomposing bayesian networks: triangulation of the moral graph with genetic algorithms. *Statistics and Computing* 7(1):19–34.

Lauritzen, S., and Spiegelhalter, D. 1988. Local computation with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society* 50:157–244.

Murphy, K. 2002. *Dynamic bayesian networks: representation, inference and learning*. Ph.D. Dissertation.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, California: Morgan Kaufmann Publishers.

Verma, T., and Pearl, J. 1988. Causal networks: Semantics and expressiveness. In *Fourth Conference on Uncertainty in Artificial Intelligence*, 352–359.

Wong, S.; Wu, D.; and Butz, C. 2002. Triangulation of bayesian networks: a relational database perspective. In *The Third International Conference on Rough Sets and Current Trends in Computing (RSCTC 2002), LNAI 2475*, 389–397.

Wu, D., and Wong, S. 2004. The marginal factorization of bayesian networks and its application. *Int. J. Intell. Syst.* 19(8):769–786.

Xiang, Y. 2002. *Probabilistic Reasoning in Multuagent Systems: A Graphical Models Approach*. Cambridge.