Computing Marginals with Hierarchical Acyclic Hypergraphs

S. K. M. Wong and T. Lin
Department of Computer Science
University of Regina
Regina, Saskatchewan
Canada S4S 0A2
{wong, taolin}@cs.uregina.ca

Abstract

How to compute marginals efficiently is one of major concerned problems in probabilistic reasoning systems. Traditional graphical models do not preserve all conditional independencies while computing the marginals. That is, the Bayesian DAGs have to be transformed into a secondary computational structure, normally, acyclic hypergraphs, in order to compute marginals. It is well-known that some conditional independencies will be lost in such a transformation. In this paper, we suggest a new graphical model which not only equivalents to a Bayesian DAG, but also takes advantages of all conditional independencies to compute marginals. The input to our model is a set of conditional probability tables as in the traditional approach.

Introduction

In probabilistic reasoning systems (Neapolitan, 1989; Pearl, 1988), the domain knowledge is represented in terms of a *joint probability distribution* (JPD). The JPD is factorized in terms of *conditional probability tables* (CPTs) according to the *conditional independencies* (CIs) encoded by the graphical model using *Bayesian directed acyclic graphs* (DAGs) or *acyclic hypergraphs* (AHs).

One of the problems in probabilistic reasoning systems is how to compute marginals from an input set of CPTs. This problem has been studied intensively. One method is to transform the DAG into a AH and apply local propagation techniques (Jensen, 1996; Shafer et al., 1990) to compute the marginal for every *hyperedge* of the AH. However, AHs cannot represent some *embedded CIs* (Pearl, 1988). This means that some CIs encoded in the DAG cannot be preserved by such a transformation.

Many graphical models have been suggested for taking advantage of all CIs in the computation of marginals. Geiger (Geiger, 1988) and Shachter (Shachter, 1990) proposed *multiple undirected graphs* (MUGs) to faithfully represent a DAG. More recently, Kjaerulff (Kjaerulff, 1997) has demonstrated that multiple AHs (*nested junction trees*) can be used to compute marginals in a more efficient manner than one single AH. However, it is not known if these proposed models are not equivalent to the Bayesian DAGs.

In this paper, we use the split-free *hierarchical acyclic hypergraphs* (HAHs) model, which is equivalent to a Bayesian DAG (Wong et al., 2003), to compute the marginals without losing CI information. We show that a set of CPTs can be specified according to the graphical structure, there exists a computation sequence enable us to compute the marginals, and the JPD factorization is represented in terms of the product of such a set of CPTs. It is worth mentioning that the complexity of our method for computing marginals is NP-Complete (Cooper,1990) as the local propagation technique. The important point is that our model preserves all CIs in computing the marginals.

This paper is organized as follows. We include a brief review of basic concepts about probabilistic networks in Section 2. Section 3 introduces the notion of HAH. In Section 4, we introduce a special type of HAHs called split-free HAHs. Section 5 discusses how to compute the marginals with respect to every hyperedge of an AH. Section 6 suggests an approach to specify the input set of CPTs for the split-free HAHs. In Section 7, we show that the JPD can be factorized as a product of the input CPTs. The conclusion is presented in Section 8.

Background Knowledge

Here we briefly review some pertinent notions of probabilistic networks including Bayesian networks and acyclic hypergraphs.

Let U be a set of domain variables. We say Y and Z are conditionally independent given X with respect to a JPD $P(U)$, if $P(Y|XZ) = P(Y|X)$, where X, Y, Z are disjoint subsets of U. This *conditional independence statement* (CI) can be conveniently represented by a triplet: $I(Y, X, Z)$.

A *Bayesian network* (Pearl, 1988) is a *directed acyclic graph* (DAG) together with a set of CPTs corresponding to each node A_i in the DAG. A *Bayesian JPD* is defined by the product of those CPTs, namely:

$$P(U) = \prod_{i=1}^{n} P(A_i|Pa(A_i)),$$

where $Pa(A_i)$ are parents of the node A_i.

A *hypergraph* $\langle \mathbf{H}, U \rangle$ denotes a family of subsets of U, namely:

$$\langle\, \mathbf{H}, U\, \rangle \;=\; \{h_1, h_2, \ldots, h_n\},$$

where $h_i \subseteq U$ is called a *hyperedge* of $\langle \mathbf{H}, U \rangle$, and U is the union of all hyperedges, written $U = h_1 h_2 \cdots h_n$. We use $\mathbf{H}$ to denote the hypergraph $\langle \mathbf{H}, U \rangle$ if no confusion arises. We say U is the *context* of $\mathbf{H}$, written $\mathcal{CT}(\mathbf{H}) = U$.

A hypergraph $\mathbf{H}$ is *acyclic* if there exists a *hypertree construction ordering* $(h_1, h_2, \ldots, h_n)$ such that

$$s_i = h_i \cap (h_1 \cdots h_{i-1}) \subseteq h_j, \;\; 1 \leq j \leq i-1,$$

for $2 \leq i \leq n$, where s_i is called the *separator* of h_i. It is worth mentioning that given an acyclic hypergraph, there exists many hypertree construction orderings. In fact, for each hyperedge of an acyclic hypergraph, there is a construction ordering beginning with that hyperedge (Shafer et al., 1990). In the following discussion, we always assume the subscripts of the hyperedges specify a hypertree construction ordering for the given AH.

The *factorization* of the JPD according to the AH is the product of the CPT specified to each hyperedges. Thus,

$$P(U) \;=\; \prod P(r_i|s_i), \tag{1}$$

where $r_i = h_i - s_i$, $1 \leq i \leq n$. Let $S = \{s_2, s_3, \ldots, s_n\} = \{s_i | 2 \leq i \leq n\}$, which is referred to as the *set of separators* of $\mathbf{H}$. It should be noted that all hypertree construction orderings have the same set of separators.

In an AH, We refer to $P(h_i)$ as the marginal with respect to the hyperedge h_i. We say the hyperedge h_i of an AH *computable* if the marginal $P(h_i)$ can be computed. The AH is *computable* if marginals with respect to all hyperedges are computable.

Hierarchical Acyclic Hypergraphs

In this section, we define the special graphical structure called the *hierarchical set of acyclic hypergraph* (HAH). It consists of a set of acyclic hypergraphs whose contexts form a tree hierarchy.

We first define a *tree hierarchy* of hypergraphs. Let $\mathcal{H}$ denotes a family of hypergraphs, written

$$\mathcal{H} \;=\; \{\, \mathbf{H}_0,\, \mathbf{H}_1,\, \ldots,\, \mathbf{H}_n \,\}. \tag{2}$$

We call $\mathbf{H}_j$ a *descendant* of $\mathbf{H}_i$, if $\mathcal{CT}(\mathbf{H}_j) \subseteq \mathcal{CT}(\mathbf{H}_i)$, $0 \leq i, j \leq n$. A hypergraph $\mathbf{H}_j$ is a *child* of $\mathbf{H}_i$, if $\mathbf{H}_j$ is a descendant of $\mathbf{H}_i$ and there is no hypergraph $\mathbf{H}_k$ in $\mathcal{H}$ such that $\mathcal{CT}(\mathbf{H}_j) \subseteq \mathcal{CT}(\mathbf{H}_k) \subseteq \mathcal{CT}(\mathbf{H}_i)$. If $\mathbf{H}_j$ is a *child* of $\mathbf{H}_i$, we say $\mathbf{H}_i$ is a *parent* of $\mathbf{H}_j$. If the AHs have no parents, we say they are roots. We refer to the AHs without children as the *leafs*.

Definition 1 *A family of hypergraph $\mathcal{H}$ is called tree hierarchy, if:*

(1) There exists a root $\mathbf{H}_0$ such that for every hypergraph $\mathbf{H}_j$ in $\mathcal{H}$, $\mathcal{CT}(\mathbf{H}_j) \subset \mathcal{CT}(\mathbf{H}_0)$, $1 \leq j \leq n$.

(2) Assume $\mathbf{H}_j$ is the child of $\mathbf{H}_i$, Then there exists a hyperedge h of $\mathbf{H}_i$ such that $\mathcal{CT}(\mathbf{H}_j) \subseteq h$.

(3) For two distinct children, written $\mathbf{H}_1$ and $\mathbf{H}_2$, of $\mathbf{H}_i$ such that $\mathcal{CT}(\mathbf{H}_1) \subseteq h_j$, $\mathcal{CT}(\mathbf{H}_2) \subseteq h_k$, where h_j and h_k are two hyperedges of $\mathbf{H}_i$, j is not equal to k, i.e., $j \neq k$.

Let $\mathbf{H}_p$ be the parent of $\mathbf{H}_c$ in the tree hierarchy $\mathcal{H}$. The hyperedge h in $\mathbf{H}_p$ is *refinable*, if $\mathcal{CT}(\mathbf{H}_c) \subseteq h$. We say $\mathbf{H}_c$ is the child of h. A hyperedge without any child is called *non-refinable*.

Now we use the concept of tree hierarchy to define the *hierarchical set of acyclic hypergraphs* (HAH). We call $\mathcal{H}$ the hierarchical set of AHs if all hypergraphs in the tree hierarchy $\mathcal{H}$ are acyclic.

Example 1 *Consider a given HAH $\mathcal{H}$, written*

$$\mathcal{H} \;=\; \{\mathbf{H}_1,\, \mathbf{H}_2,\, \mathbf{H}_3,\, \mathbf{H}_4\},$$

where

$$\begin{aligned} \mathbf{H}_1 &= \{BCDEFGH, AC, HK, GHI\}, \\ \mathbf{H}_2 &= \{BG, BDE, DEF, CFH\}, \\ \mathbf{H}_3 &= \{C, F\}, \\ \mathbf{H}_4 &= \{BD, BE\}. \end{aligned}$$

The acyclic hypergraph $\mathbf{H}_1$ is the root of $\mathcal{H}$, which has one child $\mathbf{H}_2$. The acyclic hypergraphs $\mathbf{H}_3$ and $\mathbf{H}_4$ are the children of $\mathbf{H}_2$ and the decedents of $\mathbf{H}_1$.

The split-free HAHs

In this section, we introduce the definition of split-free HAHs. We use a special hyperedges ordering, denoted by the hierarchy construction ordering (HCO), to test if the given HAHs$\mathcal{H}$ is split-free. That is, if there is a HCO in $\mathcal{H}$, the we say $\mathcal{H}$ is a *split-free* HAHs. In the following discussion, we use the notation $h_i \prec h_j$ to denote h_i appearing previous to h_j in an hyperedges ordering.

Definition 2 *A HAH $\mathcal{H}$ is called the* split-free *HAH if there exists a special hyperedges ordering for all hyperedges of $\mathcal{H}$, written $h_1 \prec h_2 \prec, \ldots, \prec h_n$, such that,*

(1) In the ordering, the appearing sequence of the hyperedges that comes from the same AH $\mathbf{H}$ is the hypertree construction ordering of $\mathbf{H}$;

(2) If h_i is the refinable hyperedge such that $\mathcal{CT}(\mathbf{H}_c) \subseteq h_i$, then $h_k \prec h_i$, for every hyperedge $h_k \in \mathbf{H}_c$;

(3) If h_i is the refinable hyperedge such that $\mathcal{CT}(\mathbf{H}_c) \subseteq h_i$. then for the separator s_i of h_i, $s_i \subseteq h_j$, where h_j is the first hyperedges of $\mathbf{H}_c$ appearing in the ordering.

We refer to such an ordering as the hierarchy construction ordering (HCO) of $\mathcal{H}$.

Example 2 *Consider the HAH $\mathcal{H}$ shown in Figure 2(a). There exists a HCO such that,*

$$AD \prec AB \prec AC \prec \underline{ABC} \prec BCE,$$

where we use underlines to highlights refinable hyperedges of $\mathcal{H}$.

The hypertree construction ordering of the AHs $\mathbf{H}_1$ and $\mathbf{H}_2$ in the sequence are: $AD \prec \underline{ABC} \prec BCE$ and $AB \prec AC$. It can be verified that they are the hypertree construction orderings of $\mathbf{H}_1$ and $\mathbf{H}_2$, respectively. For the refinable hyperedge ABC such that $\mathcal{CT}(\mathbf{H}_2) \subseteq ABC$,

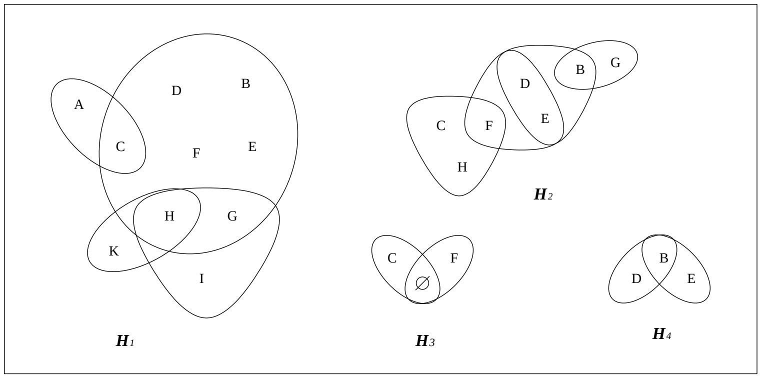

Figure 1: The hierarchical acyclic hypergraph given in Example 1.

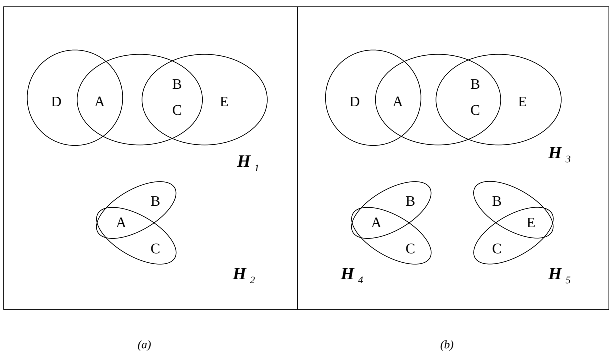

Figure 2: Two types of HAHs. The HAH shown at Part (a) is a split-free HAH. Part (b) is not a split-free HAH.

the condition $AB \prec ABC$ and $AC \prec ABC$ hold in the ordering. Consider the separator A of refinable hyperedge ABC. It satisfies $A \subseteq AB$, where AB is the first hyperedge of $\mathbf{H}_2$ appearing in the ordering.

Note that a split-free HAH may have many HCOs. For instance, another HCO for the HAH shown at Figure 2(a) is: $AD \prec AC \prec AB \prec \underline{ABC} \prec BCE$.

Note that not all HAHs have the HCOs. There exists no HCO for the HAH shown at Figure 2(b).

Consider a HCO of the split-free HAH. The following proposition states that the intersection of a hyperedge h_i with the union of the previous hyperedges in the ordering is contained by one of hyperedge h_j such that $h_j \prec h_i$.

Proposition 1 *Consider a HCO for the split-free HAH, written $h_1 \prec h_2 \prec \cdots \prec h_n$. Then $h_i \cap (h_1 \cdots h_{i-1}) \subseteq h_j$, where $h_j \prec h_i$ in the HCO.*

Proof. By induction. Consider the hyperedges of the root, by definition of HCO, it is obvious for the hyperedges of the root.

Assume h_i is a refinable hyperedge of the root and $\mathbf{H}_c$ is the child such that $\mathcal{CT}(\mathbf{H}_c) \subseteq h_i$. By definition of HCO, all hyperedges of $\mathbf{H}_c$ appears previous to h_i. It means that the union of them is still h_i. Hence, the intersection of $\mathcal{CT}(\mathbf{H}_c)$ and all previous hyperedges is still the separator of h_i. By definition of HCO, the separator is contained by the first hyperedge h_j of $\mathbf{H}_c$ and $h_j \prec h_i$. The argument above can be applied recursively for all hyperedges in the given split-free HAH.

Note that Proposition 1 enables us to search a HCO for the given HAH. Consider two hyperedges h_i and h_j of an acyclic hypergraph $\mathbf{H}$ in the HAH $\mathcal{H}$, where h_i is a refinable hyperedge, and $h_i \cap h_j = s$. If s is split by the decedents of h_i, by definition of HCO, then the condition $h_i \prec h_j$ should hold in any HCO of $\mathcal{H}$. Since h_i and h_j are in the same hypergraph $\mathbf{H}$, it follows that an available hypertree construction orderings of $\mathbf{H}$ for a HCO of $\mathcal{H}$ has to satisfy $h_i \prec h_j$. This restricts our choice to select the hyperedges ordering of $\mathbf{H}$: when we choose a hypertree construction ordering of $\mathbf{H}$, we have to check if $h_i \prec h_j$. If not, we have to select another hypertree construction ordering which satisfies the condition. If we cannot find any construction ordering satisfying the condition, then there is no HCO for $\mathcal{H}$.

Example 3 *Consider two HAHs shown in Figure 2 (a) and (b). Now we show how to determine if they have HCOs respectively.*

Consider hyperedges ABC and BCE in AH $\mathbf{H}_1$. In the child $\mathbf{H}_2$ of refinable hyperedge ABC, since AB and BC split $BC = ABC \cap BCE$, by definition of HCO, it means that BC cannot be the separator of BCE. Therefore, $ABC \prec BCE$ should hold in any HCO. At the same time, there are four hypertree construction orderings of $\mathbf{H}_1$, namely,

$$AD \prec ABC \prec BCE,$$
$$BCE \prec ABC \prec AD,$$
$$ABC \prec BCE \prec AD,$$
$$ABC \prec AD \prec BCE.$$

It can be found that only the first and the last hypertree construction orderings satisfy the condition $ABC \prec BCE$. After we add hyperedges of $\mathbf{H}_2$ in the ordering, we obtain the HCO for the HAH in Figure 2 (a). Therefore, it is a split-free HAH.

Applying the same method to the HAH in Figure 2 (b), we have two conditions to satisfy, namely, $ABC \prec BCE$ and $BCE \prec ABC$. Since none of the ordering would satisfy these two conditions, there is no HCO for the HAH. Therefore, it is not a split-free HAH.

The computability of AH

In this section, we show how to compute marginals with respect to every hyperedge in an AH $\mathbf{H}$ by explicitly assigning CPTs to the hyperedges of $\mathbf{H}$.

The following proposition states that an AH $\mathbf{H}$ is computable if we can provide the marginal with respect to the first hyperedge in a hypertree construction ordering of $\mathbf{H}$.

Proposition 2 *Consider an AH $\mathbf{H} = \{h_1, h_2, \ldots, h_n\}$, where subscripts specify the hypertree construction ordering of $\mathbf{H}$. If $P(h_1)$ can be computed, then for any hyperedge h_i, $2 \leq i \leq n$, $P(h_i)$ can be computed by specifying CPT $P(r_i|s_i)$ to the hyperedge h_i.*

Proof. By induction. Suppose $P(h_1)$ is computed, by specifying the CPT $P(r_2|s_2)$ to the hyperedge h_2, then we can compute $P(h_2)$ as:

$$P(h_2) \;=\; P(s_1)P(r_2|s_2) \;=\; \Big(\sum_{r_1} P(h_1)\Big)P(r_2|s_2).$$

In general, when we specify the CPT $P(r_i|s_i)$ to every hyperedge h_i, $i \leq 2$, the marginal with respect to the hyperedge h_i can be computed as:

$$P(h_i) \;=\; P(s_i)P(r_i|s_i) \;=\; \Big(\sum_{r_j} P(h_j)\Big)P(r_i|s_i),$$

where $s_i \subseteq h_j$, $1 \leq j \leq i-1$. It means that $P(h_i)$ can be computed if $P(h_j)$ is computed, where $j < i$. Therefore, following a given hypertree construction ordering of the acyclic hypergraph $\mathbf{H}$, we always can compute the marginal with respect to every hyperedge h_i in $\mathbf{H}$, by specifying CPTs $P(r_i|s_i)$ to the hyperedge h_i.

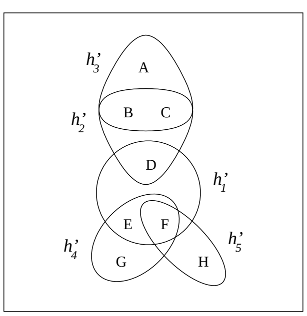

Figure 3: The hypergraph given in the Example 4.

Example 4 *Consider the hypergraph as shown in Figure 3, namely:*

$$\mathbf{H} = \{ABC, BCD, DEF, EFG, FH\}.$$

A hypertree construction ordering of $\mathbf{H}$ is:

$$ABC \prec BCD \prec DEF \prec EFG \prec FH.$$

If $P(h_1) = P(ABC)$ is computed, then we can specify $P(D|BC)$, $P(EF|D)$, $P(G|EF)$ and $P(H|F)$ to the hyperedge BCD, DEF, EFG and FH separately, so as to compute marginals $P(BCD)$, $P(DEF)$, $P(EFG)$ and $P(FH)$ consecutively.

It can be verified that the factorization of JPD $P(U)$ according to $\mathbf{H}$ is the product of specified CPTs, namely:

$$P(U) = P(ABC)P(D|BC)P(EF|D)P(G|EF)P(H|F).$$

Since an acyclic hypergraph may have a hypertree construction odering beginning with any hyperedge (Shafer et al., 1990), by Proposition 2, it follows that an AH is computable if we can compute the marginal with respect one of hyperedges of $\mathbf{H}$.

Marginals computation in split-free HAHs

In this section, we show that a set of CPTs can be specified according to the HCO of the given split-free HAH $\mathcal{H}$ in order to compute marginals with respect to hyperedges of $\mathcal{H}$. We refer to a HAH as *computable* if marginals with respect to all hyperedges are computable.

Consider a HAH $\mathcal{H}$. If one of the refinable hyperedges h equals to the context of its own child $\mathbf{H}_c$, namely, $h = \mathcal{CT}(\mathbf{H}_c)$, as soon as its child is computable, this refinable hyperedge h is computable as well. Then we only need to specify $P(h) = 1$ to this refinable hyperedge. Sometimes the refinable hyperedge properly contains the context of its child, thus, $(\mathbf{H}_c) \subset h$, then we always can specify the CPT $P(r|\mathcal{CT}(\mathbf{H}_c))$ to the refinable hyperedge, where $r = h - \mathcal{CT}(H_c)$. For instance, in Example 1, the refinable hyperedge CFH of $\mathbf{H}_2$ properly contains the context of its child $\mathbf{H}_3 = \{C, F\}$. We only need to specify CPT $P(H|CF)$ to this refinable hyperedge, if the marginal with respect to the child, namely $P(CF)$, is computable. In the following discussion, we will only consider how to specify CPTs to the non-refinable hyperedges.

The following theorem enables us to specify a set of CPTs for the given split-free HAH $\mathcal{H}$ such that the marginals with respect to every hyperedges of $\mathcal{H}$ can be computed.

Theorem 1 *A HAH $\mathcal{H}$ is computable if and only if it is split-free.*

Proof. Assume there exists a HCO for $\mathcal{H}$, namely, $h_1 \prec h_2 \prec, \ldots, \prec h_n$. By Proposition 1, Let $s_i = h_i \cap (h_1 \cdots h_{i-1}) \subseteq h_j$, where $h_i \prec h_j$ in the HCO. If the first hyperedge h_1 is computable, by specifying the CPTs $p(r_i|s_i)$ to the remaining hyperedges of the HCO, where $r_i = h_i - s_i$, then all hyperedges of the given HAH can be computed.

Suppose there exists no HCO for the HAH $\mathcal{H}$. It means that there exists an AH $\mathbf{H}$ in the HAH $\mathcal{H}$ such that, no matter how we specify the hypertree construction ordering of $\mathbf{H}$, there always exists at least one refinable hyperedge of $\mathbf{H}$, namely $\mathcal{CT}(\mathbf{H}_c) \subseteq h_i$, such that the separator s_i of h_i is located in two different hyperedges of $\mathbf{H}_c$, namely

h'_j and h'_k. By Proposition 1, we only can get $P(s_i)$ from the previous marginal computations, and there is no way to compute $P(h'_j)$ and $P(h'_k)$ consistently. It means that $\mathbf{H}_c$ cannot be computed.

By Theorem 1, given a HCO with respect to the split-free HAH, written $h_1 \prec h_2 \prec \ldots \prec h_n$, we can specify the CPT $P(r_i|s_i)$ to every non-refinable hyperedge h_i, where $s_i = h_i \cap (h_1 \cdots h_{i-1})$, $r_i = h_i - s_i$. If $P(h_j)$ is computable, then we may compute the marginal with respect to the hyperedge h_i, such that, $P(h_i) = P(r_i|s_i)P(s_i)$. It means that if h_1 in the HCO is computable, then all other hyperedges will be computed subsequently.

Example 5 *Consider a HAH shown at Figure 2(a), written*

$$\mathcal{H}_1 = \{\mathbf{H}_1, \mathbf{H}_2\},$$

where $\mathbf{H}_1 = \{AD, ABC, BCE\}$,
$\mathbf{H}_2 = \{AB, AC\}$.

There is a HCO of $\mathcal{H}_1$, written

$$AB \prec AC \prec \underline{ABC} \prec AD \prec BCE.$$

According to the given HCO, after providing the marginal $P(AB)$ to the hyperedge AB, the marginal with respect to the hyperedge AC can be computed by specifying the CPT $P(C|A)$ to the hyperedge AC. It means that the refinable hyperedge ABC is computed. By specifying $P(D|A)$ and $P(E|BC)$ to the hyperedges AD and BCE of $\mathbf{H}_1$, respectively, these two hyperedges can be computed as well. By definition, it means that $\mathcal{H}_1$ is a computable HAH.

Consider a HAH shown at Figure 2(b), namely:

$$\mathcal{H}_2 = \{\mathbf{H}_3, \mathbf{H}_4, \mathbf{H}_5\},$$

where $\mathbf{H}_3 = \{AD, ABC, BCE\}$,
$\mathbf{H}_4 = \{AB, AC\}$,
$\mathbf{H}_5 = \{BE, CE\}$.

Although we may provide any marginals to the hyperedges AD, AB or AC, respectively, we can not compute the marginals $P(BE)$ and $P(BC)$ at the same time. On the other hand, suppose we have the marginals $P(BE)$ and $P(BC)$ computed, the hyperedges AB and AC can not be computed simultaneously. Therefore, the HAH $\mathcal{H}_2$ is not computable.

The JPD factorization according to the split-free HAH

In previous section, we show that the HCO in the split-free HAH $\mathcal{H}$ enables us to assign a set of CPTs in order to compute marginals with respect to hyperedges of $\mathcal{H}$. In this section, moreover, we will show that the JPD can be factorized in terms of the product of such a set of CPTs.

Proposition 3 *Consider a HCO of the split-free HAH $\mathcal{H}$, written $h_1 \prec h_2 \prec \cdots \prec h_n$. The factorization of JPD $P(U)$ according to $\mathcal{H}$ can be represented as follows:*

$$P(U) = \prod_{i=1}^{n} P(r_i|s_i). \tag{3}$$

where $P(r_i|s_i)$, $1 \leq i \leq n$, is the CPT specified to each hyperedges h_i of $\mathcal{H}$.

Proof. Consider the root of the HAH $\mathcal{H}$, written $\mathbf{H}_{root} = \{h_1, h_2, \ldots, h_n\}$. The subscripts specify the hypertree construction ordering, which is the same as showing in the given HCO of $\mathcal{H}$. The JPD $P(U)$ can be factorized according to the root as follows:

$$P(U) = \prod_{i=1}^{n} P(r_i|s_i), \tag{4}$$

where s_i is the separator of h_i, $r_i = h_i - s_i$, $1 \leq i \leq n$.

Assume h_k is a refinable hyperedge such that $\mathcal{CT}(\mathbf{H}_k) \subseteq h_k$, where $\mathbf{H}_k = \{h'_1, h'_2, \ldots, h'_m\}$. It follows that:

$$P(h_k) = P(r_k|\mathcal{CT}(\mathbf{H}_k)) \prod_{j=1}^{m} P(r'_j|s'_j),$$

where $r_k = h_k - \mathcal{CT}(\mathbf{H}_k)$, $r_j = h'_j - s'_j$, $1 \leq j \leq m$. Hence,

$$P(r_k|s_k) = \frac{P(h_k)}{P(s_k)} = \frac{P(r_k|\mathcal{CT}(\mathbf{H}_k)) \prod_{j=1}^{m} P(r'_j|s'_j)}{P(s_k)}.$$

By definition of HCO, $s_k \subseteq h'_1$. It follows that:

$$P(r_k|s_k) = P(r_k|\mathcal{CT}(\mathbf{H}_k)) P(r'_1|s_k) \prod_{j=2}^{m} P(r'_j|s'_j), \tag{5}$$

where $r'_1 = h'_1 - s_k$.

Substitute Equation (5) into Equation (4), it follows that

$$P(U) = \Big(\prod_{i \neq k}^{n} P(r_i|s_i)\Big) P(r_k|\mathcal{CT}(\mathbf{H}_k)) \\ P(r'_1|s_k) \prod_{j=2}^{m} P(r'_j|s'_j). \tag{6}$$

The argument above thereafter can be applied recursively until all hypergraphs are considered for the HAH $\mathcal{H}$, which results in the factorization of JPD $P(U)$, as shown in Equation (3).

Example 6 *Consider the split-free HAH $\mathcal{H}$ given in Example 1. According to the HCO given in Equation (**??**), the CPTs specified to the corresponding hyperedges are listed as follows:*

$$\left\{ \begin{array}{l} P(BG), P(D|B), P(E|B), 1, \\ P(F|DE), P(C), 1, P(H|CF), 1, \\ P(A|C), P(K|H), P(I|GH) \end{array} \right\}. \tag{7}$$

In the HCO, the hypertree construction ordering for the root $\mathbf{H}_1$ is

$$\underline{BCDEFGH} \prec AC \prec HK \prec GHI.$$

The JPD $P(U)$ can be factorized as follows:

$$P(U) = P(BCDEFGH)P(A|C)P(K|H)P(I|GH). \tag{8}$$

Consider the refinable hyperedge $BDEFGH$, where $\mathcal{CT}(\mathbf{H}_2) \subseteq BDEFGH$. The hypertree construction ordering of $\mathbf{H}_2$ in the HCO is:

$$BG \prec \underline{BDE} \prec DEF \prec CFH.$$

The marginal $P(BCDEFGH)$ can be factorized as follows:

$$\begin{aligned} P(BCDEFGH) &= P(BG)P(DE|B) \\ & P(F|DE)P(CH|F). \end{aligned} \quad (9)$$

Consider the refinable hyperedge BDE of $\mathbf{H}_2$, where $\mathcal{CT}(\mathbf{H}_3) \subseteq BDE$. Since the first hyperedge of $\mathbf{H}_3$ appearing in the HCO is BD, which contains the separator B of BDE. It follows that the CPT specified to BDE, namely, $P(DE|B)$, can be rewritten

$$\begin{aligned} P(DE|B) &= \frac{P(DEB)}{P(B)} = \frac{P(BD)P(E|B)}{P(B)} \\ &= P(D|B)P(E|B). \end{aligned} \quad (10)$$

Substituting Equation (10) into Equation (9), it yields:

$$\begin{aligned} P(BCDEFGH) &= P(BG)P(D|B)P(E|B) \\ & P(F|DE)P(CH|F). \end{aligned} \quad (11)$$

Substituting Equation (11) into Equation (8), it yields:

$$\begin{aligned} P(U) = P(BG)P(D|B)P(E|B)P(F|DE)P(CH|F) \\ P(I|GH)P(A|C)P(K|H). \end{aligned}$$

The approach can be applied recursively until all AHs are considered. In the end, the CPTs in Equation (7) are specified to all hyperedges of $\mathcal{H}$. The factorization of JPD according to the HAH $\mathcal{H}$ thereafter can be represented by:

$$\begin{aligned} P(U) &= P(BG)P(D|B)P(E|B)P(F|DE)P(C) \\ & P(H|CF)P(A|C)P(K|H)P(I|GH). \end{aligned}$$

Conclusion

In this paper, we suggest a graphical model called the split-free HAH, which not only is equivalent to a Bayesian DAG, but also enables us to compute the marginals without losing any CI information. Similar to the conventional probabilistic network, we can specify an input set of CPTs according to the graphical structure. The JPD can be factorized as a product of such a set of CPTs.

Acknowledgement

The authors would like to thank anonymous referees for their useful suggestions and critical comments.

References

Beeri, C., Fagin, R., Maier, D., and Yannakakis, M.: On the desirability of acyclic database schemes. J. ACM vol. 30, No. 3, July 1983, pp479-513.

Gregory F. Cooper: The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks. Artificial Intelligence, Vol. 42, No.2-3, 1990, pp393-405.

D. Geiger: Towards the formalization of informational dependencies. Technical Report CSD-880053, University of California at Los Angeles, 1988.

Heckerman, D., Geiger, D. and Chickering, D.M.: Learning Bayesian networks:the combination of knowledge and statistical data. *Machine Learning*, 20:197-243, 1995.

Jensen, F.V.: An introduction to Bayesian networks. UCL Press, 1996.

U. Kjaerulff: Nested junction trees. In *Thirteenth Conference on Uncertainty in Artificial Intelligence*, pp 302-313. Providence, Rhode Island, 1997.

R. E. Neapolitan: Probabilistic Reasoning in Expert Systems. John Wiley & Sons, Inc. 1989

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

R.D. Shachter: A graph-based inference method for conditional independence. In *Seventh Conference on Uncertainty in Artificial Intelligence*, pp 353-360, Morgan Kaufmann Publishers, 1990.

G. R. Shafer, P. P. Shenoy: Probability Propagation, Annals of Mathematics and Artificial Intelligence, Vol.2, pp. 327-352, 1990

Wilhelm R$\ddot{o}$dder, Carl-Heinz Meyer: Coherent Knowledge Processing at Maximum Entropy by SPIRIT, In *Twelfth Conference on Uncertainty in Artificial Intelligence*, pp 470-476, Morgan Kaufmann Publishers, 1996.

S.K.M. Wong, T. Lin: An alternative characterization of a Bayesian Network, International Journal of Approximate Reasoning, Vol.33, No.3, pp221-234, 2003