On Undirected Representations of Bayesian Networks

S.K.M. Wong, C.J. Butz, D. Wu
Department of Computer Science
University of Regina
Regina, Saskatchewan, Canada, S4S 0A2
{wong,butz,danwu}@cs.uregina.ca

Abstract

Empirical studies clearly demonstrate the effectiveness of the *nested jointree* (NJT) representation in probabilistic inference. A NJT is a traditional *Markov network* (MN) together with a possible local MN nested in each clique. These nested MNs can themselves contain other nested MNs in a recursive manner. However, the NJT representation is not necessarily a *faithful* representation of a given *Bayesian network* (BN). This means that the effectiveness of a NJT has been demonstrated while only exploiting *some* of the independency information available in the given BN.

In this paper, we introduce a new kind of probabilistic network, called a *hierarchical Markov network* (HMN). A HMN is a hierarchy of MNs. We give an algorithm to transform a BN into a canonical HMN. The main result of this paper is that the constructed HMN is *unique* and *equivalent* to the input BN. Since HMNs are a *faithful* representation of BNs, a query may be optimized using independencies in a HMN that otherwise would have gone unrepresented in a NJT approach.

1 INTRODUCTION

Bayesian networks (BNs) [8] are a powerful tool for facilitating the acquisition of a probabilistic model. However, the *moralization* and *triangulation* procedures [8] are usually applied to transform a BN into a (decomposable) *Markov network* (MN) [3], since the latter can take advantage of the many efficient local propagation techniques [4, 6, 10] developed for computing marginal distributions. Thereby, BNs and MNs form a favorable pair for the acquisition and inference of probabilistic knowledge. However, the transformation of a BN into a MN is somewhat undesirable since the MN might not represent *all* of the independencies in the BN.

It is well-known that the main weakness of MNs is their inability to represent *embedded* independencies. However, it follows naturally that embedded independencies in a single BN can be represented using *multiple* MNs, which Geiger [2] and Shachter [9] call *multiple undirected graphs* (MUGs). Since MUGs are definitely not a compact representation, Geiger [2] stated that more research is needed in order to construct a *canonical basis.* More recently, Kjaerulff [5] took an initial step in this direction by introducing *nested jointrees* (NJTs). A NJT is a traditional MN together with a possible local MN nested in each clique. These nested MNs can themselves contain other nested MNs in a recursive manner. The important point of NJTs is that Kjaerulff [5] clearly demonstrated using ten large real-world BNs how multiple MNs can be used to process queries in a more efficient manner than in a single MN alone. However, it should be noted that NJTs are not a *faithful* representation of BNs. By definition, the root MN in a NJT is a traditional MN. Some *conditional independencies* (CIs), sacrificed in the transformation from a BN into a root MN, may not be recoverable by simply looking at the internal structure of the cliques in the root MN.

In this paper, we introduce a new kind of probabilistic network, called a *hierarchical Markov network* (HMN). A HMN is a rooted hierarchy of MNs. We give an algorithm to transform a BN into a canonical HMN. The main result of this paper is that the constructed HMN is *unique* and *equivalent* to the input BN. Since HMNs are a *faithful* representation of BNs, a query may be optimized using independencies in a HMN that otherwise would have gone unrepresented in a NJT approach. The above remarks seem to suggest that HMNs are a very natural representation of probabilistic knowledge with several desirable properties. It should be made clear, however, that BNs and HMNs are not adversarial representations. Instead BNs and HMNs form an enviable pair for the acquisition and inference of probabilistic knowledge.

This paper is organized as follows. Section 2 contains a review of Bayesian and Markov networks. In Section 3, we examine MUGs and NJTs. We introduce *hierarchical Markov networks* (HMNs) in Section 4. In Section 5, we present an algorithm to transform a BN into a unique and equivalent HMN. Advantages of our probabilistic network are given in Section 6. The conclusion is presented in Section 7.

2 BAYESIAN AND MARKOV NETWORKS

We begin our discussion by reviewing Bayesian and Markov networks. *Bayesian networks* (BNs) [8] greatly facilitate the acquisition of a *joint probability distribution* (jpd). However, a BN is usually transformed into a *Markov network* (MN) [3], since the latter can take advantage of the many efficient propagation techniques [4, 6, 10] developed for computing marginal distributions. Although BNs and MNs form a favorable pair for the acquisition and inference of probabilistic knowledge, the transformation of a BN into a MN is somewhat undesirable since the MN might not *faithfully* represent the BN.

A *Bayesian network* (BN) [8] is a *directed acyclic graph* (DAG) together with a corresponding set of *conditional probability tables* (CPTs). The terms BN and DAG will be used interchangeably if no confusion arises. For example, the BN $\mathcal{D}$ in Figure 1 (i) on the set of variables $R = \{A, B, C, D, E, F\}$, written as $R = ABCDEF$, indicates that the jpd $p(ABCDEF)$ can be factorized as:

$$p(R) = \quad (1)$$
$$p(A) \cdot p(B|A) \cdot p(C|A) \cdot p(D|B) \cdot p(E|C) \cdot p(F|DE).$$

The BN in Eq. (1) can be transformed into a MN introduced below.

A *Markov network* (MN) [3], called a *decomposable* MN by Pearl [8], is an acyclic hypergraph together with a corresponding set of marginal distributions. The moralization $\mathcal{U}_m$ of DAG $\mathcal{D}$ is shown in Figure 1 (ii). One possible chordal undirected graph $\mathcal{U}_t$, obtained by applying the triangulation procedure to $\mathcal{U}_m$, is shown in Figure 1 (iii). The *acyclic hypergraph* $\mathcal{H}$, defined by the maximal cliques of the chordal undirected graph $\mathcal{U}_t$, is $\mathcal{H} = \{ABC, BCD, CDE, DEF\}$. We will use the terms MN and acyclic hypergraph interchangeably if no confusion arises. The jpd is then written as a product of *potentials* $\phi(h)$ [3], $h \in \mathcal{H}$:

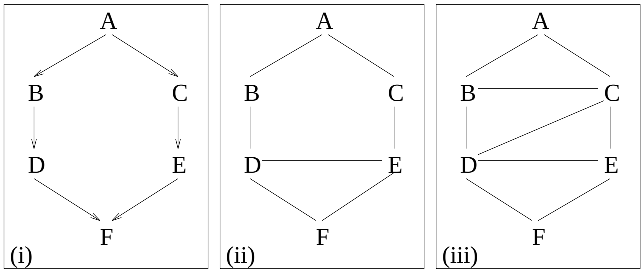

Figure 1: (i) A BN $\mathcal{D}$. (ii) The moralization $\mathcal{U}_m$ of $\mathcal{D}$. (iii) One possible triangulation $\mathcal{U}_t$ of $\mathcal{U}_m$. The MN defined by the maximal cliques of $\mathcal{U}_t$ is $\mathcal{H} = \{ABC, BCD, CDE, DEF\}$.

$$p(R) = \phi(ABC) \cdot \phi(BCD) \cdot \phi(CDE) \cdot \phi(DEF),$$

where the potentials are defined using the CPTs in Eq. 1:

$$\begin{aligned}\phi(ABC) &= p(A) \cdot p(B|A) \cdot p(C|A),\\ \phi(BCD) &= p(D|B),\\ \phi(CDE) &= p(E|C),\\ \phi(DEF) &= p(F|DE).\end{aligned}$$

In general, potentials are not uniquely defined. The fact that potentials $\phi(BCD)$ and $\phi(CDE)$ do not involve all variables in BCD and CDE, respectively, is entirely the consequence of the triangulation procedure [3]. The local propagation techniques can be applied to rewrite the factorization of $p(R)$ in terms of *marginal* distributions:

$$p(R) = \frac{p(ABC) \cdot p(BCD) \cdot p(CDE) \cdot p(DEF)}{p(BC) \cdot p(CD) \cdot p(DE)}. \quad (2)$$

The set of J-keys (or separating sets [3]) of $\mathcal{H}$ is $\{BC, CD, DE\}$. More generally, $\mathcal{H}$ is an *acyclic hypergraph* (a *hypertree* [10]) if and only if $\mathcal{H}$ has the *running intersection property* [3]. Given a *hypertree construction ordering* $h_1, h_2, \ldots, h_n$ for an acyclic hypergraph $\mathcal{H}$, and a *branching function* $b(i)$ for this ordering, the set $\mathcal{J}$ of *J-keys* for $\mathcal{H}$ is defined as:

$$\mathcal{J} = \{h_2 \cap h_{b(2)}, h_3 \cap h_{b(3)}, \ldots, h_n \cap h_{b(n)}\}. \quad (3)$$

An acyclic hypergraph $\mathcal{H}$ has a unique set of J-keys.

Although MNs can take advantage of the local propagation techniques for computing marginal distributions, transforming a BN into a MN may require that some conditional independency information be *sacrificed.* In the above example, both the *embedded* CI $I(B, A, C)$ and the *full* CI $I(AC, BE, DF)$ are represented in $\mathcal{D}$ but not in $\mathcal{H}$. (In the special case where a CI $I(Y, X, Z)$ involves all variables, i.e., $R = XYZ$, then $I(Y, X, Z)$ is called *full* or *nonembedded*; otherwise, it is called *embedded*.) In the remainder of the paper, we seek a new representation of probabilistic knowledge which saves all of the conditional independencies in a given Bayesian network, while at the same time maintains the desirable properties of a MN.

3 RELATED RESEARCH

The purpose here is to motivate the development of hierarchical Markov networks. Pearl explicitly states that in the strictest sense BNs are hypergraphs (see pg. 125 in [8]). In other words, a BN can be faithfully represented by *multiple* MNs [2, 9]. More recently, empirical studies explicitly demonstrate the effectiveness of multiple MNs in probabilistic *inference* [5].

Pearl [8] states that the main weakness of MNs is their inability to represent *embedded* CIs. However, a BN on n variables can be faithfully represented by n MNs, called *multiple undirected graphs* (MUGs) by Geiger [2] and Shachter [9], defined with respect to a causal input list. A *causal input list* (CIL) [8] for a BN $\mathcal{D}$ on n variables is a set of n CIs, denoted $CIL_{\mathcal{D}}$, following a fixed topological ordering of $\mathcal{D}$.

Example 1 One CIL $CIL_{\mathcal{D}}$ for the BN $\mathcal{D}$ in Figure 2 (i) following the topological ordering A, B, C, D is:

$$CIL_{\mathcal{D}} = \{I(A, \emptyset, \emptyset),\ I(B, A, \emptyset),\ I(C, A, B),\ I(D, BC, A)\}.$$

As shown in Figure 2 (ii), this CIL $CIL_{\mathcal{D}}$ defines the four acyclic hypergraphs $\mathbf{H} = \{\mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3\}$:

$$\begin{aligned}\mathcal{H}_0 &= \{h_1 = DBC,\ h_2 = BCA\},\\ \mathcal{H}_1 &= \{h_{11} = CA,\ h_{12} = AB\},\\ \mathcal{H}_2 &= \{h_{21} = BA,\ h_{22} = A\emptyset\},\\ \mathcal{H}_3 &= \{h_{31} = A\emptyset,\ h_{32} = \emptyset\emptyset\}.\end{aligned}$$

(Since no confusion should arise, we have sometimes written and depicted a set X as $X\emptyset$.)

Let $\mathcal{H}$ be a hypergraph. By $\mathcal{H} - X$, we denote the hypergraph obtained from $\mathcal{H}$ by deleting the set X of variables. More formally,

$$\mathcal{H} - X = \{\ h - X \mid h \in \mathcal{H}\} - \{\emptyset\}.$$

We say X *separates* Y (from the rest of the variables), if Y is the union of some disconnected components of the hypergraph $\mathcal{H} - X$.

Given a hypergraph $\mathcal{H}$, the set CI($\mathcal{H}$) of full CIs generated by $\mathcal{H}$ is defined as:

$$\text{CI}(\mathcal{H}) = \{X \Rightarrow\Rightarrow Y \mid X \text{ separates } Y \text{ in } \mathcal{H}\}. \quad (4)$$

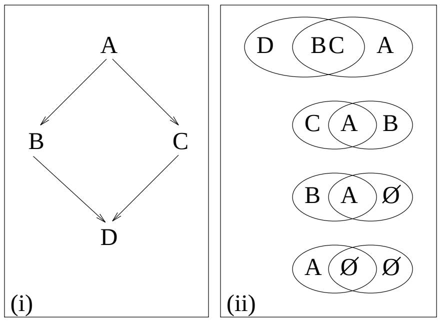

Figure 2: The single BN $\mathcal{D}$ in (i) can be equivalently represented by the four MNs in (ii).

More generally, the CI information in a set of MNs $\mathbf{H} = \{\mathcal{H}_0, \mathcal{H}_1, \ldots, \mathcal{H}_l\}$, denoted $CI(\mathbf{H})$, is defined as:

$$CI(\mathbf{H}) = CI(\mathcal{H}_0) \cup CI(\mathcal{H}_1) \cup \ldots \cup CI(\mathcal{H}_l). \quad (5)$$

By definition, $CIL_{\mathcal{D}} = CI(\mathbf{H})$ in Example 1.

Multiple MNs can be used to process queries in a more efficient manner. In [5], Kjaerulff introduced *nested jointrees* (NJTs). A NJT is a traditional MN $\mathcal{H}$ coupled with additional MNs contained by the hyperedges of $\mathcal{H}$. Each of these nested MNs can themselves contain other nested MNs in a recursive manner. Recall that many efficient local propagation techniques [4, 6, 10] have been developed for computing marginal distributions in MNs. These inference algorithms do not take advantage of any embedded CIs, since a MN can only represent full CIs. Thus, a query may be processed more efficiently in a NJT representation compared to a MN representation.

Example 2 [5] Consider a MN $\mathcal{H} = \{ABCD, ABE, BCF, ADG\}$, where hyperedge $h = ABCD$ is originally assigned the potential $\phi(CD)$. Suppose hyperedge $ABCD$ is going to send the message $\phi(AD)$ to its neighbor ADG. The information needed is $\phi(CD)$ together with the messages from h's neighbors ABE and BCF. In the local propagation techniques developed for MNs, the required message $\phi(AD)$ is computed as:

$$\phi(AD) = \sum_{BC} \phi(AB) \cdot \phi(BC) \cdot \phi(CD). \quad (6)$$

By using a NJT representation instead of a MN, $\phi(AD)$ can be computed more efficiently as:

$$\phi(AD) = \sum_{B} \phi(AB) \sum_{C} \phi(BC) \cdot \phi(CD). \quad (7)$$

Assuming binary variables and defining time cost as the number of arithmetic operations, Eqs. (6) and (7) imply a space cost of 16 and 8 and a time cost of 64 and 48, respectively.

It should be noted, however, that NJTs are not a *faithful* representation of BNs. By definition, the root jointree is a *traditional* MN. As demonstrated in Section 2, transforming a BN into a MN may necessitate that some CIs be sacrificed. This means that NJTs are not necessarily taking advantage of all CIs available in the given BN. More generally, MUGs are not a *basis*. In fact, Geiger [2] pointed out that further research is needed in order to determine how multiple DAGs and MUGs should be incorporated to create more compact graphical representations. Our main result is that we can construct a *unique* HMN, which is *equivalent* to the input BN. Thus, we believe that HMNs are the compact representation sought. Since the constructed HMN is a *faithful* representation of the input BN, it follows that a query may be processed more efficiently in a HMN representation compared to a NJT representation (see Section 6).

4 HIERARCHICAL MARKOV NETWORKS

Hierarchical Markov networks (HMNs) are introduced as a new kind of probabilistic network. We begin with a few pertinent notions.

Let $\mathcal{H}$ be a MN on $R = A_1 A_2 \cdots A_m$. The *context* of $\mathcal{H}$, denoted $context(\mathcal{H})$, is the set of nodes on which $\mathcal{H}$ is defined, i.e., $context(\mathcal{H}) = R$. Let $\mathcal{H}_0, \mathcal{H}_1, \ldots, \mathcal{H}_l$ be a set of MNs. We call $\mathcal{H}_j$ a *descendant* of $\mathcal{H}_i$, if $context(\mathcal{H}_j) \subseteq context(\mathcal{H}_i)$. If $\mathcal{H}_j$ is a descendant of $\mathcal{H}_i$, then we call $\mathcal{H}_i$ an *ancestor* of $\mathcal{H}_j$. We call $\mathcal{H}_j$ a *child* of $\mathcal{H}_i$, if $\mathcal{H}_j$ is a descendant of $\mathcal{H}_i$, and there does not exist a $\mathcal{H}_k$ such that $context(\mathcal{H}_j) \subseteq context(\mathcal{H}_k) \subseteq context(\mathcal{H}_i)$. If $\mathcal{H}_j$ is a child of $\mathcal{H}_i$, then we call $\mathcal{H}_i$ a *parent* of $\mathcal{H}_j$.

A *hierarchical Markov network* (HMN), denoted $\mathbf{H} = \{\mathcal{H}_0, \mathcal{H}_1, \ldots, \mathcal{H}_l\}$, is a hierarchy of MNs satisfying the following three conditions:

(i) there is only one *root* MN in $\mathbf{H}$,

(ii) if $\mathcal{H}_j$ is a child of $\mathcal{H}_i$, then there exists a hyperedge $h_i \in \mathcal{H}_i$ such that $context(\mathcal{H}_j) \subseteq h_i$, and

(iii) if $\mathcal{H}_j$ and $\mathcal{H}_k$ are two distinct children of $\mathcal{H}_i$, then $\mathcal{H}_j$ and $\mathcal{H}_k$ are contained by *distinct* hyperedges of $\mathcal{H}_i$.

Example 3: Let us verify that $\mathbf{H} = \{\mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2\}$, depicted in Figure 3, is a HMN, where

$$\begin{aligned} \mathcal{H}_0 &= \{h_1 = ABC,\ h_2 = BCGEF,\ h_3 = EFD\}, \\ \mathcal{H}_1 &= \{h_{11} = AB,\ h_{12} = AC\}, \\ \mathcal{H}_2 &= \{h_{21} = DE,\ h_{22} = DF\}. \end{aligned}$$

MN $\mathcal{H}_0$ fulfills condition (i) as the root of $\mathbf{H}$. Condition (ii) is also satisfied. $\mathcal{H}_1$ is a child of $\mathcal{H}_0$ and $context(\mathcal{H}_1) = ABC$ is a subset of $h_1 \in \mathcal{H}_0$. $\mathcal{H}_2$ is another child of $\mathcal{H}_0$ and $context(\mathcal{H}_2) = DEF$ is contained in $h_3 \in \mathcal{H}_0$. Condition (iii) is satisfied since $\mathcal{H}_1$ can be assigned to hyperedge h_1 in $\mathcal{H}_0$, while $\mathcal{H}_2$ can be assigned to h_3 in $\mathcal{H}_0$.

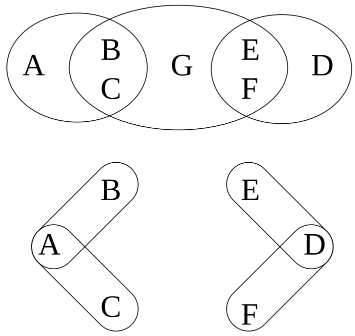

Figure 3: A HMN $\mathbf{H} = \{\mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2\}$.

Not every HMN can be faithfully represented by a single BN. The HMN $\mathbf{H} = \{\mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2\}$ in Ex. 3 cannot be faithfully represented by a BN. More formally,

$$\begin{aligned} CI(\mathbf{H}) &= CI(\mathcal{H}_0) \cup CI(\mathcal{H}_1) \cup CI(\mathcal{H}_2) \\ &= \{I(A, BC, GEFD),\ I(D, EF, ABCG), \\ &\quad\ I(B, A, C),\ I(E, D, F)\} \end{aligned}$$

cannot be faithfully represented by a single DAG.

On the contrary, a BN can always be faithfully represented by a HMN. Recall the BN $\mathcal{D}$ and $\mathbf{H} = \{\mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3\}$ in Ex. 1. Condition (i) in the definition of HMN is satisfied by $\mathcal{H}_0$. $\mathcal{H}_1$ is the only child of $\mathcal{H}_0$ and is contained by $h_2 \in \mathcal{H}_0$. Similarly, $\mathcal{H}_2$ is the only child of $\mathcal{H}_1$ and is contained by $h_{12} \in \mathcal{H}_1$, while $\mathcal{H}_3$ is the only child of $\mathcal{H}_2$ and is contained by $h_{22} \in \mathcal{H}_2$. Thus, condition (ii) is also satisfied. Condition (iii) is trivially satisfied. By definition, $\mathbf{H} = \{\mathcal{H}_0, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3\}$ in Ex. 1 is a HMN.

Transforming a BN on n variables into a HMN consisting of n MNs is not useful in practice. In the next section, we show how a BN can be equivalently transformed into a *canonical* HMN.

5 TRANSFORMING BNs TO HMNs

We give a procedure to transform a BN into a canonical HMN. It is then shown that the constructed HMN is *unique* and *faithful* (equivalent) to the input BN.

The construction of a HMN from a BN can be understood as recursively building a MN using *non-redundant* CIs in the BN. That is, the root MN in a HMN, unlike the one in a NJT, is *not* obtained by applying the standard moralization and triangulation procedures on the input BN. As in [12], we instead take full advantage of algorithms extensively used to analyze the properties of a relational database schema [1]. Thus, we may use database notation and write the CI $I(Y,X,Z)$ as $X \Rightarrow\Rightarrow Y|Z$. If the context XYZ is *fixed*, we may write $X \Rightarrow\Rightarrow Y|Z$ simply as $X \Rightarrow\Rightarrow Y$ with Z understood.

For our purposes we are primarily interested in a subset of CI($\mathcal{H}$). We define CF($\mathcal{H}$) as those CIs satisfying the following three conditions:

(i) $X \Rightarrow\Rightarrow Y$ is in CI($\mathcal{H}$).

(ii) $X \subseteq h$ for some $h \in \mathcal{H}$, and

(iii) $X' \Rightarrow\Rightarrow YX''$ is not in CI($\mathcal{H}$), where $X' \subset X$ and $X'' = X - X'$.

By definition, CF($\mathcal{H}$) is *unique* for a given hypergraph $\mathcal{H}$. It can also be shown that CF($\mathcal{H}$) is a *conflict-free* [12] set of full CIs.

We developed an algorithm in [12], called MAKEACYHYP here, that constructs a *unique* acyclic hypergraph $\mathcal{H}$ for a conflict-free set $\mathbf{C}$ of CIs. The main result was that $\mathbf{C} \models I(Y,X,Z)$ iff $I(Y,X,Z) \in CI(\mathcal{H})$.

Example 4 Consider the BN $\mathcal{D}$ in Figure 1 (i). As shown in Figure 4 (i), the hypergraph $\mathcal{H}$ corresponding to $\mathcal{D}$ is:

$$\mathcal{H} \quad = \quad \{A_iP_i \mid A_i \in \mathcal{D}\},$$

where P_i is the *parent set* of variable A_i in $\mathcal{D}$. The set CF($\mathcal{H}$) in this case consists of a single CI:

$$\text{CF}(\mathcal{H}) \quad = \quad \{\ DE \Rightarrow\Rightarrow F\ \}.$$

The unique acyclic hypergraph generated by MAKEACYHYP using CF($\mathcal{H}$) is:

$$\text{MAKEACYHYP(CF}(\mathcal{H})) \quad = \quad \{ABCDE, DEF\},$$

as shown in Figure 4 (ii).

Once the root MN $\mathcal{H}_0 = \{h_1, h_2, \ldots, h_n\}$ has been constructed, the embedded CIs for each hyperedge, denoted $\mathcal{D}_h$, are determined using the same procedure given by Kjaerulff [5]. The following algorithm BN2HMN constructs a specific HMN from an input BN.

Algorithm BN2HMN
Input: a DAG $\mathcal{D}$
Output: a HMN $\mathbf{H} = \{\mathcal{H}_0, \mathcal{H}_1, \ldots, \mathcal{H}_l\}$
BN2HMN($\mathcal{D}$)
begin

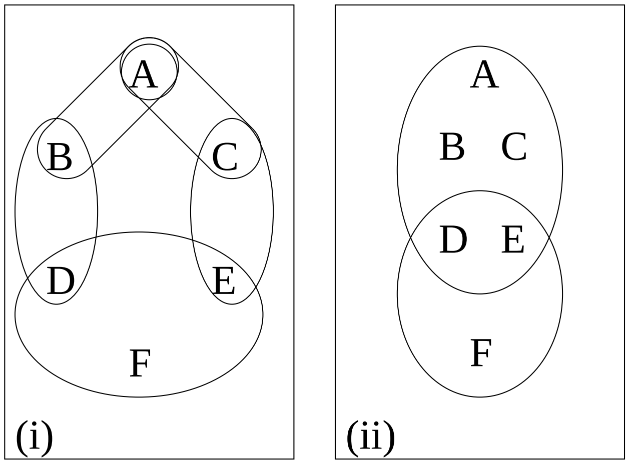

Figure 4: (i) The hypergraph $\mathcal{H}$ corresponding to the BN $\mathcal{D}$ in Figure 1 (i). (ii) MAKEACYHYP(CF($\mathcal{H}$)).

Step 1. Construct the root $\mathcal{H}_0$ of $\mathbf{H}$
$\mathcal{H} := \{A_iP_i \mid A_i \in \mathcal{D}\}$;
$\mathcal{H}_0 :=$ MAKEACYHYP(CF($\mathcal{H}$));
$\mathbf{H} := \{\mathcal{H}_0\}$;
Step 2. Recursively read the embedded CIs in $\mathcal{D}$
for each hyperedge h in any $\mathcal{H} \in \mathbf{H}$
  if $\mathcal{D}_h$ encodes a nontrivial CI
    $\mathcal{H}_h := \{A_iP_i \mid A_i \in \mathcal{D}_h\}$;
    $\mathbf{H} := \mathbf{H} \cup \{$MAKEACYHYP(CF($\mathcal{H}_h$))$\}$;
  end if
end for
return($\mathbf{H}$);
end

Example 5 Consider the BN $\mathcal{D}$ in Figure 5 (i). The hypergraph corresponding to $\mathcal{D}$ is $\mathcal{H} = \{\ A,\ AB,\ BC,\ BD,\ CDE,\ BEF,\ FG,\ FH,\ FGHI,\ J,\ K,\ IJKL\}$. By definition,

$$\text{CF}(\mathcal{H}) \quad = \quad \{B \Rightarrow\Rightarrow A,\ BE \Rightarrow\Rightarrow CD,\ F \Rightarrow\Rightarrow ABCDE,\ I \Rightarrow\Rightarrow JKL\}.$$

As can be seen in Figure 5 (ii), MAKEACYHYP(CF($\mathcal{H}$)) $= \{AB,\ BCDE,\ BEF,\ FGHI,\ IJKL\}$.

Once the root $\mathcal{H}_0$ is constructed, the same procedure is applied recursively on the hyperedges of $\mathcal{H}_0$. Consider the hyperedge $h = BCDE$. The sub-DAG defined by h is

$$\mathcal{D}_h \quad = \quad \{(B,C),(B,D),(C,E),(D,E)\}.$$

The hypergraph corresponding to $\mathcal{D}_h$ is $\mathcal{H}_h = \{BC,\ BD,\ CDE\}$. Then CF($\mathcal{H}_h$) $= \{CD \Rightarrow\Rightarrow B\}$, giving MAKEACYHYP(CF($\mathcal{H}_h$)) $= \{BC, BD, CDE\}$. The hyperedge BCD can be further decomposed, as can hyperedges $FGHI$ and $IJKL$ in $\mathcal{H}_0$. The constructed HMN $\mathbf{H}$ = BN2HMN($\mathcal{D}$) is shown in Figure 5 (ii).

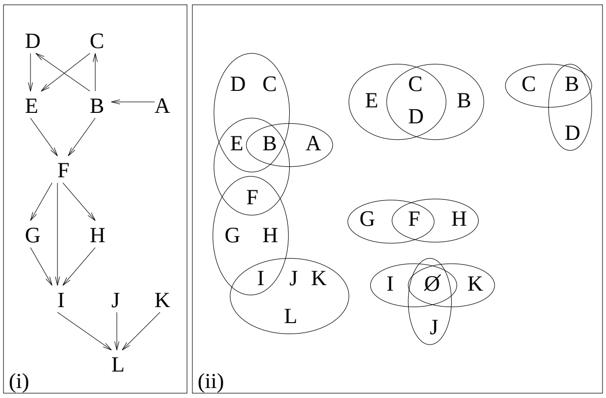

Figure 5: (i) A BN $\mathcal{D}$. (ii) $\mathbf{H}$ = BN2HMN($\mathcal{D}$).

Lemma 1 Let $\mathcal{D}$ be a BN and $\mathbf{H}$ = BN2HMN($\mathcal{D}$). Then $\mathbf{H}$ is a *unique* HMN.

Proof: The acyclic hypergraph generated by MAKEACYHYP for any BN is unique. Thus, the root MN $\mathcal{H}_0$ in $\mathbf{H}$ is unique and condition (i) is satisfied. For each hyperedge $h \in \mathcal{H}_0$, if $\mathcal{D}_h$ encodes a nontrivial CI, then a unique acyclic hypergraph is constructed and added to $\mathbf{H}$. Thus, condition (ii) is satisfied. Consider two hyperedges h_i and h_j in $\mathcal{H}_0$. Let $\mathcal{H}_i$ and $\mathcal{H}_j$ be children of $\mathcal{H}_0$ assigned to h_i and h_j, respectively. If $\mathcal{H}_i$ and $\mathcal{H}_j$ can only be assigned to h_i and h_j, respectively, then $\mathbf{H}$ is unique and condition (iii) is satisfied. Suppose then that $\mathcal{H}_i$ can also be assigned to h_j. By definition, *context*$(\mathcal{H}_i)$ is contained by $h_i \cap h_j$. Since the child acyclic hypergraph for any hyperedge is unique, this means that $\mathcal{H}_i$ and $\mathcal{H}_j$ are identical. Thus, $\mathbf{H}$ is a unique HMN. □

We now present the main result of this paper.

Theorem 1 Let $\mathcal{D}$ be a BN and $CIL_{\mathcal{D}}$ be a CIL for $\mathcal{D}$. Let $\mathbf{H}$ be the HMN $\mathbf{H}$ = BN2HMN($\mathcal{D}$). Then $CIL_{\mathcal{D}} \equiv CI(\mathbf{H})$.

Theorem 1 indicates that the algorithm BN2HMN will transform a BN into an *equivalent* HMN. Lemma 1 states that the constructed HMN is *unique*. Henceforth, we will refer to $\mathbf{H}$ = BN2HMN($\mathcal{D}$) as the *canonical* HMN for $\mathcal{D}$.

6 ADVANTAGES OF HMNs

Here we emphasize some of the desirable properties that canonical HMNs do not share with other probabilistic networks.

6.1 Undirected Representations

In this subsection, we examine four kinds of undirected representations of probabilistic knowledge. We scrutinize MNs, MUGs, NJTs, and HMNs based on compactness of representation, BN faithfulness, and uniqueness. While each of MNs, MUGs, and NJTs possess exactly one of these desirable properties, only our canonical HMN exhibits all three.

Transforming a BN into a MN may necessitate that some CIs be sacrificed. For example, the BN $\mathcal{D}$ in Figure 1 (i) can be transformed into the traditional MN shown in Figure 6 (i). As already mentioned, both the embedded CI $I(B, A, C)$ and the full CI $I(AC, BE, DF)$ are represented in the BN $\mathcal{D}$ but not in the MN $\mathcal{H}$. Thus, MNs are a compact representation but are not a faithful representation of BNs. Alternative methods of representing BNs has subsequently been explored.

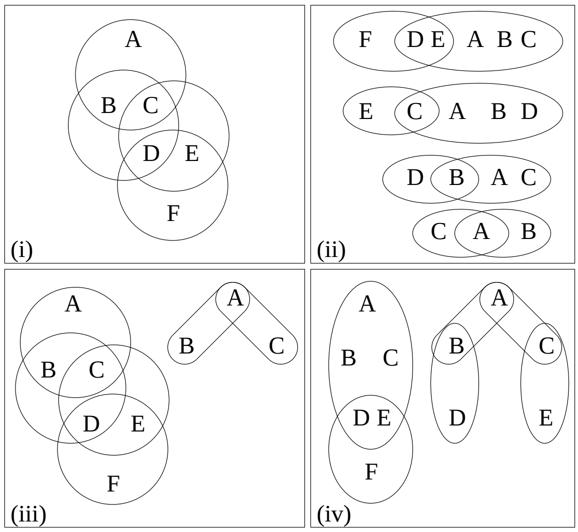

Figure 6: The BN $\mathcal{D}$ in Figure 1 (i) can be transformed into a traditional MN (i), MUGs (ii), a NJT (iii), and the canonical HMN (iv).

MUGS [2, 9] are a faithful representation of BNs but are definitely not canonical. Following the topological ordering A, B, C, D, E, F of the BN $\mathcal{D}$ in Figure 1 (i), the BN $\mathcal{D}$ can be transformed into the MUGs shown in Figure 6 (ii). (The two CIs $I(A, \emptyset, \emptyset)$ and $I(B, A, \emptyset)$ are not depicted.) While MUGs are a faithful representation of BNs, they are definitely not a compact representation of probabilistic knowledge.

In contrast to MUGs, NJTs [5] are a more compact representation but are not a faithful representation of BNs. The root MN in a NJT is a traditional MN obtained by applying the moralization and triangulation procedures to a BN. Thereby, the traditional MN in

Figure 6 (i) is a valid root MN for the NJT constructed for the BN $\mathcal{D}$ in Figure 1 (i). There is an *embedded* CI $I(B, A, C)$ holding on the hyperedge ABC in the root MN. However, there are no other embedded CIs holding on the other hyperedges BCD, CDE, and DEF in the root MN. By definition, the BN in Figure 1 (i) can be transformed into the NJT shown in Figure 6 (iii). It can be seen that the NJT is a compact representation. However, the NJT representation is not faithful, since the CI $I(D, B, ACE)$ is represented in the BN $\mathcal{D}$ but not in the constructed NJT. More formally, the BN in Figure 1 (i) can be transformed into the NJT $\{\mathcal{H}'_0 = \{ABC,\ BCD,\ CDE,\ DEF\},\ \mathcal{H}'_1 = \{AB,\ AC\}\}$. However,

$$CI(\mathcal{H}'_0) \cup CI(\mathcal{H}'_1) \quad \not\models \quad I(D, B, ACE),$$

where $\models$ stands for *logical implication* [8].

Our algorithm BN2HMN will transform the BN $\mathcal{D}$ in Figure 1 (i) into the canonical HMN $\mathbf{H} = \{\mathcal{H}_0 = \{ABCDE,\ DEF\},\ \mathcal{H}_1 = \{DB,\ AB,\ AC,\ CE\}\}$, depicted in Figure 6 (iv). By Theorem 1, this canonical HMN is also *faithful* to the BN $\mathcal{D}$. That is,

$$CIL_{\mathcal{D}} \quad \equiv \quad CI(\mathcal{H}_0) \cup CI(\mathcal{H}_1).$$

Now consider the uniqueness of transforming a BN into an undirected representation. Traditional MNs are not unique since they depend on the triangulation procedure. MUGs are not unique since they depend on the topological ordering of the BN. NJTs are not unique since they depend on the traditional MN chosen as the root, which again depends on the triangulation procedure. By Lemma 1, only the canonical HMN is *unique* for a given BN.

The important point in this subsection is that our canonical HMN serves as the first undirected representation of BNs in terms multiple MNs that is faithful, economical, and unique.

6.2 Query Optimization

Empirical studies on ten large real-world BNs explicitly demonstrate how a query can be processed more efficiently using a NJT instead of a traditional MN [5]. However, it is important to realize that NJTs are not a *faithful* representation of BNs. By definition, the root MN in a NJT is a traditional MN. Some CIs, sacrificed in the transformation from a BN into a root MN, may not be recoverable by simply looking at the internal structure of the hyperedges in the root MN. By Theorem 1, it immediately follows that a query may be optimized using independencies in a HMN representation that otherwise would have gone unrepresented in a NJT approach.

Example 6 Consider again the BN $\mathcal{D}$ in Figure 1 (i). One possible NJT and the canonical HMN are shown in Figure 6 (iii) and (iv), respectively. Since the NJT sacrificed the CI $I(D, B, ACE)$, the query $p(ABE)$ must be answered as:

$$\begin{aligned} p(ABE) &= \sum_{CD} p(ABCDE) \\ &= \sum_{CD} \frac{[\frac{p(AB)\cdot p(AC)}{p(A)}] \cdot p(BCD) \cdot p(CDE)}{p(BC) \cdot p(CD)}. \end{aligned}$$

Since the HMN is faithful to the BN, the same query $p(ABE)$ can be answered more efficiently as:

$$\begin{aligned} p(ABE) &= \sum_{C} p(ABCE) \\ &= \sum_{C} \frac{p(AB) \cdot p(AC) \cdot p(CE)}{p(A) \cdot p(C)}. \end{aligned}$$

In Ex. 9, the query was answered efficiently because it was local. Xiang et al. [11] have argued that in practice queries tend to be *local*, i.e., queries tend to focus on one part of the network. However, a non-local query involving variables in different parts of a HMN may require a lot of joining. Any additional join, required as a direct consequence of multiple levels of nesting, may be viewed as undesirable. One solution suggested by Kjaerulff [5] is to decide upon a reasonable level of nesting. An alternative solution is to keep a copy of those higher level marginals that are *feasible* [10]. This approach saves joining but requires additional storage and more work during belief update.

6.3 Directed Representations

A *succinct* cover of all independency information is mandatory in probabilistic reasoning. By definition, a CIL is an example of such a canonical cover for a BN. For example, CIL_1 is a cover of all CIs represented in the BN $\mathcal{D}$ in Figure 1 (i):

$$\begin{aligned} CIL_1 = \{&I(A, \emptyset, \emptyset), I(B, A, \emptyset), I(C, A, B), I(D, B, AC), \\ &I(E, C, ABD), I(F, DE, ABC)\}, \end{aligned}$$

where CIL_1 is defined following the topological ordering A, B, C, D, E, F. By the same token,

$$\begin{aligned} CIL_2 = \{&I(A, \emptyset, \emptyset), I(C, A, \emptyset), I(E, C, A), I(B, A, CE), \\ &I(D, B, ACE), I(F, DE, ABC)\}, \end{aligned}$$

and

$$\begin{aligned} CIL_3 = \{&I(A, \emptyset, \emptyset), I(B, A, \emptyset), I(D, B, A), I(C, A, BD), \\ &I(E, C, ABD), I(F, DE, ABC)\}, \end{aligned}$$

are also succinct covers of $(CIL_1)^+$, where CIL_2 are CIL_3 are defined following the topological orderings A, C, E, B, D, F and A, B, D, C, E, F, respectively. This demonstrates that the canonical cover for a given BN in terms of a CIL is not necessarily unique, since there may be more than one topological ordering of the variables in $\mathcal{D}$.

On the contrary, the algorithm BN2HMN will transform the BN $\mathcal{D}$ in Figure 1 (i) into the *canonical* HMN in Figure 6 (iv). By Theorem 1, all CIs holding in the BN $\mathcal{D}$ can be succinctly represented by the *canonical* cover:

$$\begin{aligned} CI(\mathbf{H}) &= CI(\mathcal{H}_0) \cup CI(\mathcal{H}_1) \\ &= \{I(F, DE, ABC),\ I(D, B, ACE), \\ &\quad I(BD, A, CE),\ I(ABD, C, E)\}. \end{aligned}$$

By Lemma 1, the set of all CIs holding in a BN $\mathcal{D}$ can either be succinctly represented by possibly more than one CIL or by the *unique* canonical basis CI($\mathbf{H}$), where $\mathbf{H}$ = BN2HMN($\mathcal{D}$).

Even more impressive is the fact that the canonical HMN is a unique cover for each *equivalence class* of BNs. That is, if $\mathcal{D}_1$ and $\mathcal{D}_2$ are two *equivalent* BNs, then BN2HMN($\mathcal{D}_1$) = BN2HMN($\mathcal{D}_2$).

7 CONCLUSION

Kjaerulff [5] has clearly demonstrated using ten large real-world BNs how a *nested jointree* (NJT) can be used to process a query more efficiently than in a single MN alone. This can be understood by the fact that a NJT representation can utilize independencies not encoded in a MN representation. However, NJTs are not a *faithful* representation of BNs. In this paper, we introduced *hierarchical Markov networks* (HMNs) as a new kind of probabilistic network. We gave an algorithm to transform a BN into a *unique* and *equivalent* HMN. This means that a query may be optimized using independencies in a HMN that otherwise would have gone unrepresented in a NJT approach. Thereby, BNs and HMNs form a highly desirable pair for the acquisition and inference of probabilistic knowledge.

References

[1] C. Beeri, R. Fagin, D. Maier, and M. Yannakakis. On the desirability of acyclic database schemes. *J. ACM*, 30(3):479–513, 1983.

[2] D. Geiger. Towards the formalization of informational dependencies. Technical Report CSD-880053, University of California, 1988.

[3] P. Hajek, T. Havranek, and R. Jirousek. *Uncertain Information Processing in Expert Systems*, CRC Press, 1992.

[4] F.V. Jensen. Junction tree and decomposable hypergraphs. Technical report, JUDEX, Aalborg, Denmark, 1988.

[5] U. Kjaerulff. Nested junction trees. In D. Geiger and P.P. Shenoy, editors, *Proc. 13th Conf. on Uncertainty in Artificial Intelligence*, pages 302–313, Providence, Rhode Island, 1997.

[6] S.L. Lauritzen and D.J. Spiegelhalter. Local computation with probabilities on graphical structures and their application to expert systems. *J. Royal Statistical Society, B*, (50):157–244, 1988.

[7] S.L. Lauritzen, A.P. Dawid, B.N. Larsen, and H.G. Leimer. Independence properties of directed markov fields. *Networks*, (20):491–505, 1990.

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

[9] R.D. Shachter. A graph-based inference method for conditional independence. In P. Bonnisone, M. Henrion, L.N. Kanal, and J.F. Lemmer, editors, *Proc. 7th Conf. on Uncertainty in Artificial Intelligence*, pages 353–360, 1991.

[10] G. Shafer. An Axiomatic Study of Computation in Hypertrees. School of Business Working Paper Series, No. 232, University of Kansas, 1991.

[11] Y. Xiang, D. Poole, and M. Beddoes. Multiply sectioned Bayesian networks and junction forests for large knowledge-based systems. *Computational Intelligence*, 9(2):171–220, 1993.

[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.