Triangulation of Bayesian Networks: a Relational Database Perspective

S.K.M. Wong, D. Wu, and C.J. Butz

Department of Computer Science
University of Regina
Regina, Saskatchewan, Canada, S4S 0A2
{wong, danwu, butz}@cs.uregina.ca

Abstract. In this paper, we study the problem of triangulation of Bayesian networks from a relational database perspective. We show that the problem of triangulating a Bayesian network is equivalent to the problem of identifying a maximal subset of conflict free conditional independencies. Several interesting theoretical results regarding triangulating Bayesian networks are obtained from this perspective.

1 Introduction

The Bayesian network model [5] has been well established as an effective and efficient tool for managing uncertain information. A *Bayesian network* (BN) consists of (i) a qualitative component, namely, *directed acyclic graph* (DAG), which encodes *conditional independence* (CI) information existing in a problem domain, and (ii) a quantitative component, namely, a set of *conditional probability distributions* (CPDs) whose product defines a *joint probability distribution* (jpd). A BN is normally transformed into a (decomposable) Markov network for probabilistic inference. This transformation consists of two separate and sequential graphical operations, namely, *moralization* and *triangulation*. The moralization of a given BN is unique, while there may exist multiple choices of triangulation. The particular choice of triangulation is important since efficiency of probabilistic reasoning is affected by the chosen triangulated graph [3].

On the other hand, the relational database model [4] has been established for designing data management systems. Historically, the relational database model was proposed for processing data consisting of records (or tuples); it was not designed as a reasoning tool. In our recent research, it has been emphasized that there exists an intriguing relationship between the relational database model and the Bayesian network model [6, 7, 8, 9].

In this paper, we provide an analytical study of the triangulation problem from a relational database perspective. In particular, we show that this problem is equivalent to that of identifying a maximal subset of conflict free conditional independencies. This new perspective is not only consistent with those graphical methods developed for triangulation, but also enables us to immediately obtain several interesting theoretical results regarding the triangulation of BNs.

The paper is organized as follows. We review triangulation of BNs in Section 2. In Section 3, constructing acyclic database schemes is discussed. Section 4 investigates the relationship between triangulation in BNs and the construction of acyclic database schemes. The conclusion is given in Section 5.

2 Triangulation in Bayesian networks

A BN is usually transformed into a (decomposable) Markov network [5] for inference. During this transformation, two graphical operations are performed on the DAG of a BN, namely, moralization and triangulation.

Given a BN, we use $\mathcal{D} = (U,\ E)$ to denote its associated DAG, where U represents the nodes in $\mathcal{D}$, and $E \subseteq U \times U$ represents the set of directed edge of $\mathcal{D}$. Moralizing a DAG simply means for each node of $\mathcal{D}$, adding undirected edges between every pair of its parents if they are not connected in $\mathcal{D}$, and then dropping the directionality of all directed edges of $\mathcal{D}$. We use $\mathcal{M}^{\mathcal{D}}$ to denote the moralized graph of $\mathcal{D}$, and omit the superscript when it is understood from context. The moralized graph $\mathcal{M}$ of $\mathcal{D}$, by definition, is undirected.

A *cycle* in a undirected graph G means a sequence of nodes $x_1,\ x_2,\ \ldots,\ x_m$ such that $x_j,\ j = 2,\ \ldots,\ m-1$, are distinct, $x_1 = x_m$ and $(x_i,\ x_{i+1}),\ i = 1,\ \ldots,\ m-1$, is an edge in G. The *length* of this cycle is m. A undirected graph G is said to be *complete* if every pair of its nodes are connected by an edge. A subset S of nodes in G is said to be complete if there are edges between every pair of nodes in S. A subset S of nodes in G is said to be a *maximal clique* if S is complete and there does not exist another subset $S^{'}$ of nodes which is complete and $S \subset S^{'}$.

The moralized DAG is transformed into a triangulated graph. A graph is *triangulated* if and only if every cycle of length four or greater contains an edge between two nonadjacent nodes in the cycle. If a graph is not triangulated, one can make it triangulated by adding some edges, called *fill-in* edges. It is noted that one may have many choices of adding fill-in edges to make a graph triangulated. We will use $\mathcal{T}^{\mathcal{D}}$ to represent a triangulated graph from a DAG $\mathcal{D}$ and omit its superscript if no confusion arises.

Example 1. Consider the DAG $\mathcal{D}$ shown in Fig 1 (i). Its unique moralized graph $\mathcal{M}$ is shown in Fig 1 (ii). $\mathcal{M}$ is not triangulated since there is a cycle, i.e., $A,\ B,\ D,\ E,\ C,\ A$. We may have multiple choices to triangulate this moralized graph, including, for instance, we may add fill-in edges, $(B,\ C),\ (C,\ D)$ as shown in Fig 1 (iii), or $(B,\ C),\ (B,\ E)$ as shown in Fig 1 (iv), or $(A,\ D),\ (A,\ E)$ as shown in Fig 1 (v). The graphs in Fig 1 (iii), (iv), (v) are triangulated.

3 Construction of Acyclic Database Schemes

In this section, we divert our discussion to the construction of acyclic database schemes in the relational database model [4]. We first briefly introduce some pertinent notions.

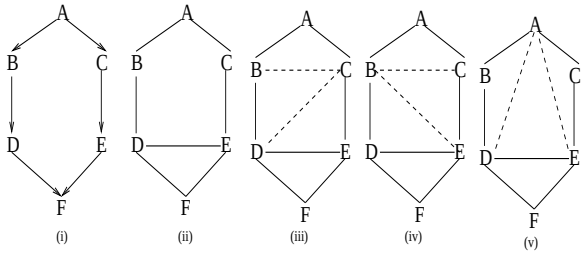

Fig. 1. A DAG $\mathcal{D}$ in (i) and its moralized graph $\mathcal{M}$ in (ii). Three possible triangulations are in (iii), (iv) and (v), where fill-in edges are indicated as dotted lines.

Let $\mathcal{N}$ be a finite set of symbols, called *attributes*. We define a *database scheme* $\mathbf{R} = \{R_1,\ R_2,\ \ldots,\ R_n\}$ to be a set of subsets of $\mathcal{N}$. By XY, we mean $X \cup Y$. For each i, the set R_i is called a *relation scheme*. If $r_1,\ r_2, \ldots,\ r_n$ are relations, where r_i is a relation over the scheme R_i, $1 \leq i \leq n$, then we call $r = \{r_1,\ r_2,\ \ldots,\ r_n\}$ a *database* over $\mathbf{R}$. We also use $r_i[R_i]$ to explicitly indicate that the relation r_i is over the scheme R_i. A relation $r[R]$ is said to satisfy the (full) *multivalued dependency* (MVD) [4] $Y \rightarrow\!\!\!\rightarrow X|Z$, if $r[R] = r[XY] \bowtie r[YZ]$, where $R = XYZ$, $r[XY]$ and $r[YZ]$ are projections [4] of $r[R]$ onto schemes XY and YZ, respectively, and Y is called the *key* of this MVD. A MVD $Y \rightarrow\!\!\!\rightarrow X|Z$ is said to *split* a set W of attributes if $W \cap X \neq \emptyset$, and $W \cap Z \neq \emptyset$. Given a set M of MVDs, the left hand sides of the MVDs in M are called the *keys* of M. A set M of MVDs are said to be *conflict free* [2] if (i) the keys of M are not split by any MVD in M, and (ii) M satisfies the *intersection property* [2].

A database scheme can be conveniently represented by a hypergraph [2]. A hypergraph is a pair $(\mathcal{N}, \mathbf{S})$, where $\mathcal{N}$ is a finite set of nodes (attributes) and $\mathbf{S}$ is a set of edges (hyperedges) which are arbitrary subsets of $\mathcal{N}$. If the nodes are understood, we will use $\mathbf{S}$ to denote the hypergraph. A hypergraph $\mathbf{S}$ is *acyclic* (or a *hypertree*) if its elements can be ordered, say $S_1, S_2, ..., S_N$, such that $(S_i \cap \bigcup_{k=1}^{i-1} S_k) \subseteq S_j$, where $1 \leq j \leq i-1$, $i = 2,\ \ldots,\ N$. We call any such ordering a *tree (hypertree) construction ordering* for $\mathbf{S}$.

There is a one to one correspondence between a hypergraph and a relational database scheme. For a database scheme $\mathbf{R} = \{R_1,\ R_2,\ \ldots,\ R_n\}$, its corresponding hypergraph representation has as its set of nodes those attributes that appear in one or more of the R_i's, and as its set $\mathbf{R} = \{R_1,\ R_2,\ \ldots,\ R_n\}$ of hyperedges. In other words, we treat $\mathbf{R}$ as a hypergraph, each of hyperedge is one of the relation schemes in $\mathbf{R}$. On the other hand, for a hypergraph, we can treat each of its hyperedge as a relation scheme and all the hyperedges compose a database scheme. For instance, given a database scheme $\mathbf{R} = \{R_1 = \{A,\ B,\ C\},\ R_2 = \{B,\ C,\ E\},\ R_3 = \{B,\ D,\ E\},\ R_4 = \{D,\ E,\ F\}\}$, we can represent it as a hypergraph as shown in Fig 2, and vice versa. A database scheme $\mathbf{R}$ is called *acyclic* if its corresponding hypergraph is acyclic [2]. Therefore, we will use the terms acyclic database scheme and acyclic hypergraph interchangeably.

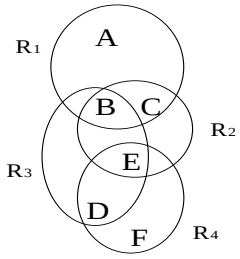

Fig. 2. A hypergraph example.

Relational database scheme design has been extensively studied [4]. A culminating result [2] is the desirability of an *acyclic database scheme*, since acyclic schemes possess a number of desirable properties. One of the properties that is relevant to our discussion is that an acyclic hypergraph is equivalent to a set of conflict free MVDs. Furthermore, an efficient algorithm [2] was developed to construct an acyclic hypergraph from a set of conflict free MVDs. A graphical method [2] was also developed to identify all the MVDs implied by an acyclic hypergraph.

4 Triangulation From a Relational Database Perspective

In this section, we first review the relationship between a triangulated graph and an acyclic hypergraph. We then study the triangulation problem from the relational database perspective.

In [2], many equivalent definitions of acyclic database scheme were suggested.

Theorem 1. [2] There is a one-to-one correspondence between triangulated graphs and acyclic hypergraphs.

That is, for each triangulated undirected graph, denoted $\mathcal{G}$, there is a corresponding equivalent acyclic hypergraph, denoted $\mathcal{H}$, where each hyperedge is a *maximal* clique of $\mathcal{G}$; and for each acyclic hypergraph $\mathcal{H}$, there is a corresponding undirected $\mathcal{G}$, which has the same nodes as $\mathcal{H}$ and an edge between every pair of nodes that are in the same hyperedge of $\mathcal{H}$.

Example 2. Consider the triangulated undirected graph $\mathcal{G}$ in Fig 3 (i) and its corresponding acyclic hypergraph $\mathcal{H}$ shown in Fig 3 (ii). The maximal cliques of $\mathcal{G}$, i.e., ABC, BCE, BDE, and DEF, are exactly the four hyperedges of $\mathcal{H}$. On the other hand, if we draw edges between every pair of nodes that are contained by the same hyperedge of $\mathcal{H}$, we will obtain the triangulated undirected graph $\mathcal{G}$.

It immediately follows from theorem 1 that we can consider the problem of triangulating a Bayesian network as the equivalent problem of constructing an acyclic hypergraph. This connection also makes it possible to apply approaches

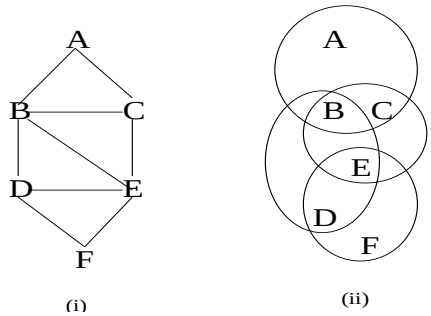

Fig. 3. A triangulated undirected graph $\mathcal{G}$ in (i) and its corresponding acyclic hypergraph $\mathcal{H}$ in (ii).

that were originally developed in relational databases model for constructing an acyclic hypergraph to the problem of triangulation in Bayesian networks.

In relational database theory, an algorithm was developed to construct an acyclic hypergraph from a set of MVDs which satisfies the condition of *conflict free* [2]. Moreover, the acyclic hypergraph constructed is equivalent to this input set of conflict free MVDs. In [8], the relationship between the notion of MVD in relational database model and the notion of CI in Bayesian network model has been thoroughly studied. Given a BN over set $U = \{X_1, \ \ldots, \ X_n\}$ of variables, we say X is *conditional independent* (CI) of Z given Y, denoted $I(X, \ Y, \ Z)$, if $p(X|YZ) = p(X|Y)$, where $X, \ Y, \ Z$ are disjoint subsets of U. A CI $I(X, \ Y, Z)$ is *full* if $XYZ = U$, and Y is the *key* of this CI. In this paper, we are only concerned with full CIs and we will use the term CI to refer to full CI unless otherwise explicitly mentioned. Since the logical implication for MVD and CI coincides [8], algorithms and notions developed for one can be safely applied to the other. For instance, the notion of a conflict free set of MVDs can be applied as the notion of a conflict free set of CIs [8]. Furthermore, in [7], we can construct an acyclic hypergraph from an input set of conflict free CIs.

Theorem 2. [7] There is a one-to-one correspondence between acyclic hypergraphs and conflict free CIs.

The algorithm in [7] suggests that for a given Bayesian network, if we can obtain a set of conflict free CIs, then we can construct an acyclic hypergraph. By theorem 1, constructing an acyclic hypergraph from an input set of conflict free CIs is equivalent to constructing a triangulated graph. By theorem 2, the problem of triangulation in Bayesian networks now turns out to be the problem of how to obtain a set of conflict free CIs from a Bayesian network. It is worth mentioning that the set of conflict free CIs obtained from a Bayesian network should be *maximal*, otherwise, although the triangulated graph of the Bayesian network obtained from a (not necessarily maximal) set of conflict free CIs is indeed triangulated, it will contain superfluous [3] fill-in edges in the triangulation.

All CIs holding in a DAG, denoted $\mathcal{C}$, can be identified using the d-separation method [5]. By using the method developed in [10], we can remove any redundant

CIs in $\mathcal{C}$ to obtain a *reduced* cover, denoted $\mathcal{C}^{'}$. The problem of triangulation in BNs now turns out to be the problem of obtaining a maximal conflict free subset of $\mathcal{C}^{'}$. A subset S of $\mathcal{C}^{'}$ is a *maximal conflict free* subset if (i) S is conflict free; and (ii) no other subset $S^{'}$ is also conflict free, where $S \subset S^{'} \subseteq \mathcal{C}^{'}$. For a reduced cover $\mathcal{C}^{'}$, we may have multiple maximal conflict free subsets. Since the intersection property is always satisfied by any CIs that hold in a DAG [5], therefore, to form a maximal conflict free subset S, we only need to check whether the CIs in S split its keys.

Based on the above discussion, we have the following theorem.

Theorem 3. Let $\mathcal{D}$ be a DAG of a BN and $\mathcal{C}^{'}$ be the reduced cover of all CIs $\mathcal{C}$ holding in $\mathcal{D}$. There is a one-to-one correspondence between the triangulations of $\mathcal{D}$ and the maximal conflict free subsets of $\mathcal{C}^{'}$.

Theorem 3 indicates that for each maximal conflict free subset of $\mathcal{C}^{'}$, there is a corresponding triangulation of $\mathcal{D}$, and vice versa.

Example 3. Consider the DAG $\mathcal{D}$ shown in Fig 1 (i). The reduced cover of all CIs holding in $\mathcal{D}$ is

$$C^{'} = \{c_1 = I(A,\ BC,\ DEF),\ c_2 = I(AC,\ BE,\ DF),\ c_3 = I(AB,\ CD,\ EF),$$
$$c_4 = I(B,\ AD,\ CEF),\ c_5 = I(C,\ AE,\ BDF),\ c_6 = I(F,\ DE,\ ABC)\}.$$

It can be verified that $S_1 = \{c_6,\ c_3,\ c_1\}$ is a maximal conflict free subset of $C^{'}$, from which the acyclic hypergraph $\mathcal{H}_1$ in Fig 4 (i) can be constructed. Similarly, it can be verified that $S_2 = \{c_6,\ c_2,\ c_1\}$ is a maximal conflict free subset of $C^{'}$, from which the acyclic hypergraph $\mathcal{H}_2$ in Fig 4 (ii) can be constructed. Similarly, it can also be verified that $S_3 = \{c_6,\ c_4,\ c_5\}$ is a maximal conflict free subset of $C^{'}$ as well, from which the acyclic hypergraph $\mathcal{H}_3$ in Fig 4 (iii) can be constructed. The corresponding triangulated undirected graphs for $\mathcal{H}_1,\ \mathcal{H}_2$, and $\mathcal{H}_3$ are shown in Fig 1 (iii), (iv), and (v), respectively.

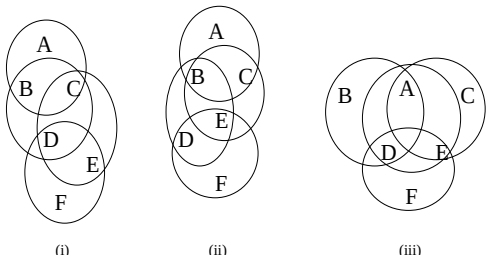

Fig. 4. Acyclic hypergraphs $\mathcal{H}_1$, $\mathcal{H}_2$ and $\mathcal{H}_3$ corresponding to maximal conflict free subsets S_1, S_2 and S_3.

It is noted that in the above example, in all three maximal conflict free subsets, namely, S_1, S_2 and S_3, the CI $c_6 = I(F,\ DE,\ ABC)$ appears in all of

them and this is not an coincidence. Another interesting result that is revealed by the new perspective of triangulation is as follows.

Theorem 4. Let $\mathcal{D}$ be a DAG of a BN, $\mathcal{C}^{'}$ be the set of reduced cover of all CIs $\mathcal{C}$ implied by $\mathcal{D}$. Let $\mathcal{M}^{\mathcal{D}}$ be the moralized graph of $\mathcal{D}$, and S_1, S_2, ..., S_n be all maximal conflict free subsets of $\mathcal{C}^{'}$. If $I(X,\ Y,\ Z) \in \mathcal{C}^{'}$ and Y is a subset of a maximal clique in $\mathcal{M}^{\mathcal{D}}$, then $I(X,\ Y,\ Z) \in \bigcap_{i=1}^{n} S_i$.

Proof. We prove this theorem by contradiction. Assume $I(X,\ Y,\ Z)$ is not in at least one of S_1, S_2, ..., S_n. Without loss of generality, suppose $I(X,\ Y,\ Z) \notin S_j$, where $1 \leq j \leq n$. Since S_j is a maximal conflict free set of CIs and all the CIs in $\mathcal{C}^{'}$ satisfy the intersection property, the only reason for $I(X,\ Y,\ Z)$ not being in S_j is that the key Y of $I(X,\ Y,\ Z)$ is split by some CI in S_j. However, since Y is a subset of a maximal clique in $\mathcal{M}^{\mathcal{D}}$, Y can not be split by any CI in S_j. A contradiction.

Theorem 4 states that any CI $I(X,\ Y,\ Z)$ in $\mathcal{C}^{'}$ whose key Y is contained by a maximal clique of the moralized graph of a DAG will appear in all maximal conflict free subsets of $\mathcal{C}^{'}$.

Theorem 3 shows that the triangulation problem of Bayesian networks can be solved by choosing a maximal conflict free subset of the reduced cover $\mathcal{C}^{'}$. This analytical view suggests that we can use the techniques developed for solving constraint satisfaction problem to help us choose a maximal conflict free subset. Theorem 4 shows that in the course of choosing a maximal conflict free subset, certain CIs must always be chosen.

5 Conclusion

In this paper, we have studied the problem of triangulation of Bayesian networks from a relational database perspective. This new perspective views the graphical problem of triangulation as an analytical one by utilizing the notion of maximal conflict free subset of CIs. Two interesting theoretical results have been presented and they show the potential of treating the problem of triangulation as a constraint satisfaction problem.

References

[1] Roman Bartak. Constraint programming: In pursuit of the holy grail. In *Proceedings of the Week of Doctoral Students (WDS)*, Czech Republic, 1999. Prague.

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

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

[4] D. Maier. *The Theory of Relational Databases*. Principles of Computer Science. Computer Science Press, Rockville, Maryland, 1983.

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

[6] S.K.M. Wong. An extended relational data model for probabilistic reasoning. *Journal of Intelligent Information Systems*, 9:181–202, 1997.

[7] 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.

[8] S.K.M. Wong, C.J. Butz, and D. Wu. On the implication problem for probabilistic conditional independency. *IEEE Transactions on System, Man, Cybernetics, Part A: Systems and Humans*, 30(6):785–805, 2000.

[9] S.K.M. Wong, Tao Lin, and Dan Wu. Construction of a bayesian dag from conditional independencies. In *The Seventh International Symposium on Artificial Intelligence and Mathematics*. Accepted, 2002.

[10] S.K.M. Wong, Tao Lin, and Dan Wu. Construction of a non-redundant cover for conditional independencies. In *The Fifteenth Canadian Conference on Artificial Intelligence*. Accepted, 2002.