12
Learning Bayesian Networks is NP-Complete

David Maxwell Chickering

Computer Science Department
University of California at Los Angeles
dmax@cs.ucla.edu

ABSTRACT
Algorithms for learning Bayesian networks from data have two components: a scoring metric and a search procedure. The scoring metric computes a score reflecting the goodness-of-fit of the structure to the data. The search procedure tries to identify network structures with high scores. Heckerman et al. (1995) introduce a Bayesian metric, called the BDe metric, that computes the relative posterior probability of a network structure given data. In this paper, we show that the search problem of identifying a Bayesian network—among those where each node has at most K parents—that has a relative posterior probability greater than a given constant is NP-complete, when the BDe metric is used.

12.1 Introduction

Recently, many researchers have begun to investigate methods for learning Bayesian networks. Many of these approaches have the same basic components: a scoring metric and a search procedure. The scoring metric takes a database of observed cases D and a network structure B_S, and returns a score reflecting the goodness-of-fit of the data to the structure. A search procedure generates networks for evaluation by the scoring metric. These approaches use the two components to identify a network structure or set of structures that can be used to predict future events or infer causal relationships.

Cooper and Herskovits (1992)—herein referred to as CH—derive a Bayesian metric, which we call the BD metric, from a set of reasonable assumptions about learning Bayesian networks containing only discrete variables. Heckerman et al. (1995)—herein referred to as HGC—expand upon the work of CH to derive a new metric, which we call the BDe metric, which has the desirable property of *likelihood equivalence*. Likelihood equivalence says that the data cannot help to discriminate equivalent structures.

We now present the BD metric derived by CH. We use B_S^h to denote the hypothesis that B_S is an I-map of the distribution that generated the database.[2] Given a belief-network structure B_S, we use Π_i to denote the parents of x_i. We use r_i to denote the number of states of variable x_i, and $q_i = \prod_{x_l \in \Pi_i} r_l$ to denote the number of instances of Π_i. We use the integer j to index these instances. That is, we write $\Pi_i = j$ to denote the observation of the jth instance of the parents of x_i.

---

[1] *Learning from Data: AI and Statistics V.* Edited by D. Fisher and H.-J. Lenz.

[2] There is an alternative causal interpretation of network structures not discussed here. See HGC for details.

Using reasonable assumptions, CH derive the following Bayesian scoring metric:

$$p(D, B_S^h|\xi) = p(B_S^h|\xi) \cdot \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(N'_{ij})}{\Gamma(N'_{ij} + N_{ij})} \cdot \prod_{k=1}^{r_i} \frac{\Gamma(N'_{ijk} + N_{ijk})}{\Gamma(N'_{ijk})} \tag{12.1}$$

where ξ is used to summerize all background information, N_{ijk} is the number of cases in D where $x_i = k$ and $\Pi_i = j$, $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$, $N'_{ij} = \sum_{k=1}^{r_i} N'_{ijk}$, and $\Gamma(\cdot)$ is the *Gamma* function. The parameters N'_{ijk} characterize our prior knowledge of the domain. We call this expression or any expression proportional to it the BD (*B*ayesian *D*irichlet) metric.

HGC derive a special case of the BD metric that follows from likelihood equivalence. The resulting metric is the BD metric with the prior parameters constrained by the relation

$$N'_{ijk} = N' \cdot p(x_i = k, \Pi_i = j|B_{S_C}^h, \xi) \tag{12.2}$$

where N' is the user's *equivalent sample size* for the domain, and $B_{S_C}^h$ is the hypothesis corresponding to the complete network structure. HGC note that the probabilities in Equation 12.2 may be computed from a *prior network*: a Bayesian network encoding the probability of the first case to be seen.

HGC discuss situations when a restricted version of the BDe metric should be used. They argue that in these cases, the metric should have the property of *prior equivalence*, which states that $p(B_{S1}^h|\xi) = p(B_{S2}^h|\xi)$ whenever B_{S1} and B_{S2} are equivalent.

HGC show that the search problem of finding the l network structures with the highest score among those structure where each node has at most one parent is polynomial whenever a decomposable metric is used. In this paper, we examine the general case of search, as described in the following decision problem:

K-LEARN
INSTANCE: Set of variables U, database $D = \{C_1, \ldots, C_m\}$, where each C_i is an instance of all variables in U, scoring metric $M(D, B_S)$ and real value p.
QUESTION: Does there exist a network structure B_S defined over the variables in U, where each node in B_S has at most K parents, such that $M(D, B_S) \geq p$?

Höffgen (1993) shows that a similar problem for PAC learning is NP-complete. His results can be translated easily to show that K-LEARN is NP-complete for $k > 1$ when the BD metric is used. In this paper, we show that K-LEARN is NP-complete, even when we use the BDe metric and the constraint of prior equivalence.

12.2 K-LEARN is NP-Complete

In this section, we show that K-LEARN is NP-complete, even when we use the likelihood-equivalent BDe metric and the constraint of prior equivalence.

The inputs to K-LEARN are (1) a set of variables U, (2) a database D, (3) the relative prior probabilities of all network structures where each node has no more than K parents, (4) parameters N'_{ijk} and N'_{ij} for some node–parent pairs and some values of i, j, and k, and (5) a value p.

The input need only include enough parameters N'_{ijk} and N'_{ij} so that the metric score can be computed for all network structures where each node has no more than K parents.

Consequently, we do not need the N'_{ijk} and N'_{ij} parameters for nodes having more than K parents, nodes with parent configurations that always have zero prior probabilities, and values of i, j, and k for which there is no corresponding data in the database. Also, we emphasize that the parameters N'_{ijk} must be derivable from some joint probability distribution using Equation 12.2.

Given these inputs, we see from Equation 12.1 that the BDe metric for any given network structure and database can be computed in polynomial time. Consequently, K-LEARN is in NP. In the following sections, we show that K-LEARN is NP-hard. In Section 12.2.1, we give a polynomial time reduction from a known NP-complete problem to 2-LEARN. In Section 12.2.2, we show that 2-LEARN is NP-hard using the reduction from Section 12.2.1, and then show that K-LEARN for $K > 2$ is NP-hard by reducing 2-LEARN to K-LEARN. In this discussion, we omit conditioning on background information ξ to simplify the notation.

12.2.1 Reduction from DBFAS to 2-LEARN

In this section we provide a polynomial time reduction from a restricted version of the feedback arc set problem to 2-LEARN. The general feedback arc set problem is stated in Garey and Johnson (1979) as follows:

FEEDBACK ARC SET
INSTANCE: Directed graph $G = (V, A)$, positive integer $K \leq |A|$.
QUESTION: Is there a subset $A' \subset A$ with $|A'| \leq K$ such that A' contains at least one arc from every directed cycle in G?

It is shown in Garvill (1977) that FEEDBACK ARC SET remains NP-complete for directed graphs in which no vertex has a total in-degree and out-degree more than three. We refer to this restricted version as DEGREE BOUNDED FEEDBACK ARC SET, or DBFAS for short.

Given an instance of DBFAS consisting of $G = (V, A)$ and K, our task is to specify, in polynomial time, the five components of an instance of 2-LEARN. To simplify discussion, we assume that in the instance of DBFAS, no vertex has in-degree or out-degree of zero. If any such vertex exists, none of the incident edges can participate in a cycle and we can remove the vertex from the graph without changing the answer to the decision problem.

To help distinguish between the instance of DBFAS and the instance of 2-LEARN, we adopt the following convention. We use the term *arc* to refer to a directed edge in the instance of DBFAS, and the term *edge* to refer to a directed edge in the instance of 2-LEARN.

We construct the variable set U as follows. For each node v_i in V, we include a corresponding binary variable v_i in U. We use $\mathcal{V}$ to denote the subset of U that corresponds to V. For each arc $a_i \in A$, we include five additional binary variables $a_{i1}, \ldots, a_{i5}$ in U. We use $\mathcal{A}_i$ to denote the subset of U containing these five variables, and define $\mathcal{A}$ to be $\mathcal{A}_1 \cup \ldots \cup \mathcal{A}_{|A|}$. We include no other variables in U.

The database D consists of a single case $C_1 = \{1, \ldots, 1\}$.

The relative prior probability of every network structure is one. This assignment satisfies our constraint of prior equivalence. From Equation 12.1 with database $D = C_1$ and relative

prior probabilities equal to one, the BDe metric—denoted $M_{BDe}(D, B_S)$—becomes

$$M_{BDe}(C_1, B_S) = \prod_i \frac{N'_{ijk}}{N'_{ij}} \tag{12.3}$$

where k is the state of x_i equal to one, and j is the instance of Π_i such that the state of each variable in Π_i is equal to one. The reduction to this point is polynomial.

To specify the necessary N'_{ijk} and N'_{ij} parameters, we specify a prior network and then compute the parameters using Equation 12.2, assuming an arbitrary equivalent sample size of one.[3] From Equation 12.3, we have

$$M_{BDe}(C_1, B_S) = \prod_i p(x_i = 1 | \Pi_i = 1, \ldots, 1, B^h_{S_C}) \tag{12.4}$$

To demonstrate that the reduction is polynomial, we show that the prior network can be constructed in polynomial time. In Section 12.2.3 (Theorem 12), we show that each probability in Equation 12.4 can be inferred from the prior network in constant time due to the special structure of the network.

We denote the prior Bayesian network $\mathcal{B} = (\mathcal{B}_S, \mathcal{B}_P)$. The prior network $\mathcal{B}$ contains both *hidden* nodes, which do not appear in U, and *visible* nodes which do appear in U. Every variable x_i in U has a corresponding visible node in $\mathcal{B}$ which is also denoted by x_i. There are no other visible nodes in $\mathcal{B}$. For every arc a_k from v_i to v_j in the given instance of DBFAS, $\mathcal{B}$ contains ten hidden binary nodes and the directed edges as shown in Figure 1 at the end of this subsection.

In the given instance of DBFAS, we know that each node v_i in V is adjacent to either two or three nodes. For every node v_i in V which is adjacent to exactly two other nodes in G, there is a hidden node h_i in $\mathcal{B}$ and an edge from h_i to x_i. There are no other edges or hidden nodes in $\mathcal{B}$.

We use h_{ij} to denote the hidden node parent common to visible nodes x_i and x_j. We create the parameters $\mathcal{B}_P$ as follows. For every hidden node h_{ij} we set

$$p(h_{ij} = 0) = p(h_{ij} = 1) = \frac{1}{2}$$

Each visible node in $\mathcal{B}$ is one of two types. The type of a node is defined by its conditional probability distribution. Every node a_{i5} in $\mathcal{B}$ (corresponding to the fifth variable created in U for the ith arc in the instance of DBFAS) is a *type II* node, and all other nodes are *type I* nodes. A type I node has the conditional probability distribution shown in Table 12.1.

We say that two variables in U are *prior siblings* if the corresponding nodes in the prior network $\mathcal{B}$ share a common hidden parent. We use S_{x_i} to denote the set of all variables in U which are prior siblings of x_i.

For each type II node a_{i5}, we define the *distinguished siblings* as the set $D_{a_{i5}} = \{a_{i3}, a_{i4}\} \subset S_{a_{i5}}$. Table 12.2 shows the conditional probability distribution of a type II node x_i with distinguished siblings $\{x_j, x_k\}$.

---

[3] Because there is only one case in the database, only the ratios $\frac{N'_{ijk}}{N'_{ij}}$ are needed (see Equation 12.3), and from Equation 12.2 the equivalent sample size is irrelevant. In general the equivalent sample size will need to be specified to uniquely determine the parameter values.

TABLE 12.1. Conditional probability distribution for a type I node.

h_{ij}	h_{ik}	h_{il}	$p(x_i = 1 \mid h_{ij}, h_{ik}, h_{il})$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

TABLE 12.2. Conditional probability distribution for a type II node x_i with $D_{x_i} = \{x_j, x_k\}$.

h_{ij}	h_{ik}	h_{il}	$p(x_i = 1 \mid h_{ij}, h_{ik}, h_{il})$
0	0	0	$\frac{1}{3}$
0	0	1	1
0	1	0	$\frac{2}{3}$
0	1	1	0
1	0	0	$\frac{2}{3}$
1	0	1	0
1	1	0	$\frac{1}{3}$
1	1	1	0

There are $|V|+5|A|$ visible nodes in $\mathcal{B}$, each visible node has at most three hidden node parents, and each probability table has constant size. Thus, the construction of $\mathcal{B}$ takes time polynomial in the size of the instance of DBFAS.

We now derive the value for p. From Equation 12.4, we obtain

$$\begin{aligned} M_{BDe}(C_1, B_S) &= \prod_i p(x_i = 1 \mid \Pi_i = 1, \ldots, 1, B^h_{S_C}) \\ &= \prod_i \delta^{3-|\Pi_i \cap S_{x_i}|} \cdot \frac{p(x_i = 1 \mid \Pi_i = 1, \ldots, 1, B^h_{S_C})}{\delta^{3-|\Pi_i \cap S_{x_i}|}} \\ &= \delta^{\left(3n - \sum_i |\Pi_i \cap S_{x_i}|\right)} \prod_i s'(x_i \mid \Pi_i, S_i) \end{aligned} \tag{12.5}$$

where $\delta < 1$ is a positive constant that we shall fix to be $15/16$ for the remainder of the paper.

Let σ be the total number of prior sibling pairs as defined by $\mathcal{B}$, and let γ be the number of prior sibling pairs which are not adjacent in B_S. The sum $\sum_i |\Pi_i \cap S_{x_i}|$ is the number of edges in B_S which connect prior sibling pairs and is therefore equal to $\sigma - \gamma$. Rewriting Equation 12.5, we get

$$M_{BDe}(C_1, B_S) = \delta^{(3n-(\sigma-\gamma))} \prod_i s'(x_i \mid \Pi_i, S_i) = c' \cdot \delta^{\gamma} \prod_i s'(x_i \mid \Pi_i, S_i) \tag{12.6}$$

We now state 3 lemmas, postponing their proofs to Section 12.2.3. A network structure B_S is a *prior sibling graph* if all pairs of adjacent nodes are prior siblings. (Not all pairs of prior siblings in a prior sibling graph, however, need be adjacent.)

Lemma 1 *Let B_S be a network structure, and let $B_{S'}$ be the prior sibling graph created by removing every edge in B_S which does not connect a pair of prior siblings. Then it follows that $M_{BDe}(C_1, B_{S'}) \geq M_{BDe}(C_1, B_S)$*

Throughput the remainder of the paper, the symbol α stands for the constant $24/25$.

Lemma 2 *If B_S is a prior sibling graph, then for every type I node x_i in B_S, if Π_i contains at least one element, then $s'(x_i \mid \Pi_i, S_i)$ is maximized and is equal to $m_1 = 64/135$. If $\Pi_i = \emptyset$, then $s'(x_i \mid \Pi_i, S_i) = \alpha \cdot m_1$.*

Lemma 3 *If B_S is a prior sibling graph, then for every type II node x_i in B_S, if $\Pi_i = D_{x_i}$, where D_{x_i} is the set of two distinguished siblings of x_i, then $s'(x_i|\Pi_i, S_i)$ is maximized and is equal to $m_2 = 40/81$. If $\Pi_i \neq D_{x_i}$ then $s'(x_i|\Pi_i, S_i) \leq \alpha \cdot m_2$.*

Finally, we define p in the instance of 2-LEARN as

$$p = c' m_1^{|V|} \left(m_1^4 m_2\right)^{|A|} \alpha^K \tag{12.7}$$

where m_1 and m_2 are defined by Lemma 2 and 3 respectively, and c' is the constant from Equation 12.6.

The value for p can be derived in polynomial time. Consequently, the entire reduction is polynomial.

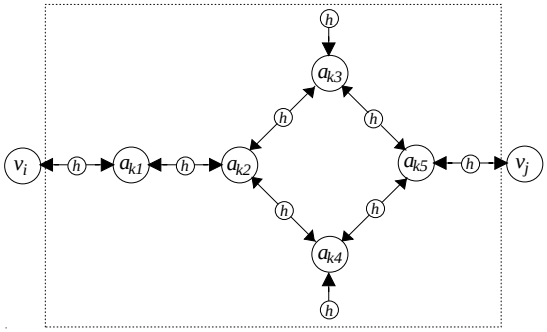

FIGURE 1. Subgraph of the prior net $\mathcal{B}$ corresponding to the *kth* arc in A from v_i to v_j.

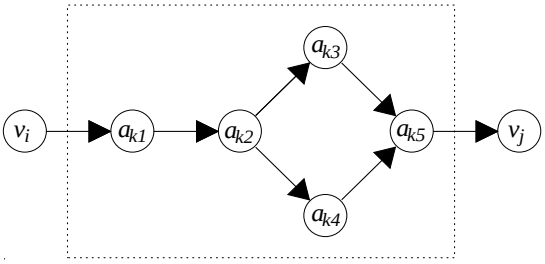

FIGURE 2. Optimal configuration of the edges incident to the nodes in $\mathcal{A}_k$ corresponding to the arc from v_i to v_j.

12.2.2 Proof of NP-Hardness

In this section, we first prove that 2-LEARN is NP-hard using the reduction from the previous section. Then, we prove that K-LEARN is NP-hard for all $k > 1$, using a reduction from 2-LEARN.

The following lemma explains the selection of p made in Equation 12.7, which in turn facilitates the proof that 2-LEARN is NP-hard. Let γ_k be the number of prior sibling pairs $\{x_i, x_j\}$ which are not adjacent in B_S, where at least one of $\{x_i, x_j\}$ is in $\mathcal{A}_k$. It follows that $\sum_k \gamma_k = \gamma$, and we can express Equation 12.6 as

$$M_{BDe}(C_1, B_S) \;=\; c' \left[\prod_{x_i \in \mathcal{V}} s'(x_i|\Pi_i, S_i)\right] \left[\prod_j t(\mathcal{A}_j, \gamma_j)\right] \tag{12.8}$$

where $t(\mathcal{A}_j, \gamma_j) = \delta^{\gamma_j} \prod_{x_i \in A_j} s'(x_i|\Pi_i, S_i)$.

Lemma 4 *Let B_S be a prior sibling graph. If each node in $\mathcal{A}_k$ is adjacent to all of its prior siblings, and the orientation of the connecting edges are as shown in Figure 2, then $t(\mathcal{A}_k, \gamma_k)$ is maximized and is equal to $m_1^4 \cdot m_2$. Otherwise, $t(\mathcal{A}_k, \gamma_k) \leq \alpha \cdot m_1^4 \cdot m_2$.*

Proof: In Figure 2, every type I node in $\mathcal{A}_k$ has at least one prior sibling as a parent, and the single type II node has its distinguished siblings as parents. Thus, by Lemmas 2 and

3, the score $s'(x_i|\Pi_i, S_i)$ for each node $x_i \in \mathcal{A}_k$ is maximized. Furthermore, every pair of prior siblings are adjacent. Thus, we have

$$t(\mathcal{A}_j, \gamma_j) = \delta^{\gamma_j} \prod_{x_i \in A_j} s'(x_i|\Pi_i, S_i) = \delta^0 \cdot m_1 \cdot m_1 \cdot m_1 \cdot m_1 \cdot m_2$$

Suppose there exists another orientation of edges incident to the nodes in $\mathcal{A}_k$ such that that $t(\mathcal{A}_k, \gamma_k) > \alpha \cdot m_1^4 \cdot m_2$. Because $\delta < \alpha$ ($\frac{15}{16} < \frac{24}{25}$), every pair of prior siblings must be adjacent in this hypothetical configuration. Furthermore, every node in $\mathcal{A}_k$ must achieve its maximum score, else the total score will be bounded above by $\alpha \cdot m_1^4 \cdot m_2$. From Lemma 3 and Lemma 2, it follows that the resulting configuration must be identical to Figure 2 □

The next two theorems prove that 2-LEARN is NP-hard.

Theorem 5 *There exists a solution to the 2-LEARN instance constructed in Section 12.2.1 with $M_{BDe}(C_1, B_S) \geq p$ if there exists a solution to the given* DBFAS *problem with $A' \leq K$.*

Proof: Given a solution to DBFAS, create the solution to 2-LEARN as follows: For every arc $a_k = (v_i, v_j) \in A$ such that $a_k \notin A'$, insert the edges in B_S between the corresponding nodes in $\mathcal{A}_k \cup v_i \cup v_j$ as shown in Figure 2. For every arc $a_k = (v_i, v_j) \in A'$, insert the edges in B_S between the corresponding nodes in $\mathcal{A}_k \cup v_i \cup v_j$ as shown in Figure 2, except for the edge between a_{k1} and a_{k2} which is reversed and therefore oriented from a_{k2} to a_{k1}.

To complete the proof, we must first show that B_S is a solution to the 2-LEARN instance, and then show that $M_{BDe}(C_1, B_S)$ is greater than or equal to p. Because each node in B_S has at most two parents, we know B_S is a solution as long as it is acyclic. By construction, B_S cannot contain a cycle unless there is a cycle in G for which none of the edges are contained in A'. Because G is a solution to DBFAS, this implies B_S is acyclic. We now derive $M_{BDe}(C_1, B_S)$. Let $\mathcal{A}^{opt}$ be the subset of $\mathcal{A}_k$ sets which correspond to the arcs in $A \setminus A'$. Rewriting Equation 12.8 we get

$$M_{BDe}(C_1, B_S) = c' \left[\prod_{x_i \in \mathcal{V}} s'(x_i|\Pi_i, S_i) \right] \cdot \left[\prod_{\mathcal{A}_j \in \mathcal{A}^{opt}} t(\mathcal{A}_j, \gamma_j) \right] \cdot \left[\prod_{\mathcal{A}_k \in \mathcal{A} \setminus \mathcal{A}^{opt}} t(\mathcal{A}_k, \gamma_k) \right]$$

Every node $x_i \in \mathcal{V}$ has at least one prior sibling node as a parent because each node in the instance of DBFAS has an in-degree of at least one. Furthermore, Lemma 4 guarantees that for every $\mathcal{A}_k$ in $\mathcal{A}^{opt}$, $t(A_j, \gamma_j)$ equals $m_1^4 \cdot m_2$. Now consider any $\mathcal{A}_k$ in $\mathcal{A} \setminus \mathcal{A}^{opt}$. All prior sibling pairs for which at least one node is in this set are adjacent in B_S, so γ_k is zero. Furthermore, every node in this set attains a maximum score, except for the type I node a_{k2} which by Lemma 2 attains a score of $\alpha \cdot m_1$. Plugging into Equation 12.9 we have

$$\begin{aligned} M_{BDe}(C_1, B_S) &= c' \left[m_1^{|V|} \right] \cdot \left[(m_1^4 \cdot m_2)^{|\mathcal{A}^{opt}|} \right] \cdot \left[(m_1^4 \cdot m_2 \cdot \alpha)^{|\mathcal{A} \setminus \mathcal{A}^{opt}|} \right] \\ &= c' m_1^{|V|} \left[m_1^4 \cdot m_2 \right]^{|A|} \alpha^{|A'|} \end{aligned}$$

Because $\alpha < 1$ and $|A'| \leq K$ we conclude that $M_{BDe}(C_1, B_S) \geq p$. □

Theorem 6 *There exists a solution to the given* DBFAS *problem with $A' \leq K$ if there exists a solution to the 2-LEARN instance constructed in Section 12.2.1 with $M_{BDe}(C_1, B_S) \geq p$.*

Proof: Given the solution B_S to the instance of 2-LEARN, remove any edges in B_S which do not connect prior siblings. Lemma 1 guarantees that the BDe score does not decrease due to this transformation.

Now create the solution to DBFAS as follows. Recall that each set of nodes $\mathcal{A}_k$ corresponds to an arc $a_k = (v_i, v_j)$ in the instance of DBFAS. Define the solution arc set A' to be the set of arcs corresponding to those sets $\mathcal{A}_k$ for which the edges incident to the nodes in $\mathcal{A}_k$ are *not* configured as shown in Figure 2.

To complete the proof, we first show that A' is a solution to DBFAS, and then show that $|A'| \leq K$. Suppose that A' is not a solution to DBFAS. This means that there exists a cycle in G that does not pass through an arc in A'. For every arc (v_i, v_j) in this cycle, there is a corresponding directed path from v_i to v_j in B_S (see Figure 2). But this implies there is a cycle in B_S which contradicts the fact that we have a solution to 2-LEARN. ¿From Lemma 4 we know that each set $\mathcal{A}_k$ that corresponds to an arc in A' has $t(\mathcal{A}_k, \gamma_k)$ bounded above by $\alpha \cdot m_1^4 \cdot m_2$. Because $M_{BDe}(C_1, B_S) \geq p$, we conclude from Equation 12.8 that there can be at most K such arcs. □

Theorem 7 *K-LEARN with $M_{BDe}(D, B_S)$ satisfying prior equivalence is NP-hard for every integer $K > 1$.*

Proof: Because 2-LEARN is NP-hard, we establish the theorem by showing that any 2-LEARN problem can be solved using an instance of K-LEARN.

Given an instance of 2-LEARN, an equivalent instance of K-LEARN is identical to the instance of 2-LEARN, except that the relative prior probability is zero for any structure that contains a node with more than two parents[4]. It remains to be shown that this assignment satisfies prior equivalence. We can establish this fact by showing that no structure containing a node with more than two parents is equivalent to a structure for which no node contains more than two parents.

Chickering (1995) shows that for any two equivalent structures B_{S_1} and B_{S_2}, there exists a finite sequence of arc reversals in B_{S_1} such that (1) after each reversal B_{S_1} remains equivalent to B_{S_2}, (2) after all reversals $B_{S_1} = B_{S_2}$, and (3) if the edge $v_i \rightarrow v_j$ is the next edge to be reversed, then v_i and v_j have the same parents with the exception that v_i is also a parent of v_j. It follows that after each reversal, v_i has the same number of parents as v_j did before the reversal, and v_j has the same number of parents as v_i did before the reversal. Thus, if there exists a node with l parents in some structure B_S, then there exists a node with l parents in any structure that is equivalent to B_S. □

12.2.3 Proof of Lemmas

To prove Lemmas 1 through 3, we derive $s'(x_i|\Pi_i, S_{x_i})$ for every pair $\{x_i, \Pi_i\}$. Let x_i be any node. The set Π_i must satisfy one of the following mutually exclusive and collectively exhaustive assertions:

[4]Note that no new parameters need be specified.

Assertion 1 For every node x_j which is both a parent of x_i and a prior sibling of x_i (i.e. $x_j \in \Pi_i \cap S_{x_i}$), there is no prior sibling of x_j which is also a parent of x_i.

Assertion 2 There exists a node x_j which is both a parent of x_i and a prior sibling of x_i, such that one of the prior siblings of x_j is also a parent of x_i.

The following theorem shows that to derive $s'(x_i|\Pi_i, S_{x_i})$ for any pair $\{x_i, \Pi_i\}$ for which Π_i satisfies Assertion 1, we need only compute the cases for which $\Pi_i \subseteq S_{x_i}$.

Theorem 8 *Let x_i be any node in B_S. If Π_i satisfies Assertion 1, then $s'(x_i|\Pi_i, S_{x_i}) = s'(x_i|\Pi_i \cap S_{x_i}, S_{x_i})$.*

Proof: From Equation 12.5, we have

$$s'(x_i|\Pi_i, S_{x_i}) = \frac{p(x_i|\Pi_i, B^e_{S_C})}{\delta^{3-|\Pi_i \cap S_{x_i}|}} \tag{12.9}$$

Because Π_i satisfies Assertion 1, it follows by construction of $\mathcal{B}$ that x_i is d-separated from all parents that are not prior siblings once the values of $\Pi_i \cap S_{x_i}$ are known. □

For the next two theorems, we use the following equalities.[5]

$$p(h_{ij}, h_{ik}, h_{il}) = p(h_{ij})p(h_{ik})p(h_{il}) \tag{12.10}$$

$$p(h_{ij}, h_{ik}, h_{il}|x_j) = p(h_{ij}|x_j)p(h_{ik})p(h_{il}) \tag{12.11}$$

$$p(h_{ij}, h_{ik}, h_{il}|x_j, x_k) = p(h_{ij}|x_j)p(h_{ik}|x_k)p(h_{il}) \tag{12.12}$$

$$p(h_{ij} = 0|x_i = 1) = \frac{2}{3} \tag{12.13}$$

Equation 12.10 follows because each hidden node is a root in $\mathcal{B}$. Equation 12.11 follows because any path from x_j to either h_{ik} or h_{il} must pass through some node $x \neq x_j$ which is a sink. Equation 12.12 follows from a similar argument, noting from the topology of $\mathcal{B}$ that $x \notin \{x_j, x_k\}$. Equation 12.13 follows from Tables 1 and 2, using the fact that $p(h_{ij} = 0)$ equals $1/2$.

Theorem 9 *Let x_i be any type I node in B_S for which Π_i satisfies Assertion 1. If $|\Pi_i \cap S_{x_i}| = 0$ then $s'(x_i|\Pi_i, S_{x_i}) = \alpha \cdot m_1$. If $|\Pi_i \cap S_{x_i}| = 1$ then $s'(x_i|\Pi_i, S_{x_i}) = m_1$. If $|\Pi_i \cap S_{x_i}| = 2$ then $s'(x_i|\Pi_i, S_{x_i}) = m_1$.*

Proof: Follows by solving Equation 12.9, using Equations 12.10 through 12.13 and the probabilities given in Table 12.1. □

Theorem 10 *Let x_i be any type II node in B_S for which Π_i satisfies assertion 1. If $|\Pi_i \cap S_{x_i}| = 0$ then $s'(x_i|\Pi_i) = \alpha^2 \cdot m_2$. If $|\Pi_i \cap S_{x_i}| = 1$ then $s'(x_i|\Pi_i) = \alpha \cdot m_2$. If $|\Pi_i \cap S_{x_i}| = 2$ and $\Pi_i \neq D_{x_i}$ then $s'(x_i|\Pi_i) = \alpha \cdot m_2$. If $\Pi_i = D_{x_i}$ then $s'(x_i|\Pi_i) = m_2$.*

[5] We drop the conditioning event $B^e_{S_C}$ to simplify notation.

Proof: Follows by solving Equation 12.9, using Equations 12.10 through 12.13 and the probabilities given in Table 12.2. □

Now we show that if Assertion 2 holds for the parents of some node, then we can remove the edge from the parent which is not a sibling without decreasing the score. Once this theorem is established, the lemmas follow.

Theorem 11 *Let x_i be any node. If $\Pi_i = \{x_j, x_k\}$, where $x_j \in S_{x_i}$ and $x_k \in S_{x_j}$, then $s'(x_i|x_j) \geq s'(x_i|x_j, x_k)$.*

Proof: For any node we have

$$p(x_i = 1|x_j = 1, x_k = 1) = \frac{p(x_i = 1)p(x_k = 1|x_i = 1)p(x_j = 1|x_i = 1, x_k = 1)}{p(x_k = 1)p(x_j = 1|x_k = 1)}$$

Because x_i and x_k are not prior siblings, it follows that $p(x_k|x_i) = p(x_k)$. Expressing the resulting equality in terms of $s'(x_i|\Pi_i, S_{x_i})$, noting that x_i has only one prior sibling as a parent, and canceling terms of δ, we obtain

$$s'(x_i|\{x_j, x_k\}, S_{x_i}) = s'(x_i|\emptyset, S_{x_i})\frac{s'(x_j|\{x_i, x_k\}, S_{x_j})}{s'(x_j|\{x_k\}, S_{x_j})} \tag{12.14}$$

If x_j is a type I node, or if x_j is a type II node and x_i and x_k are *not* its distinguished siblings, then $s'(x_j|\{x_i, x_k\}, S_{x_j})$ equals $s'(x_j|\{x_k\}, S_{x_j})$, which implies that we can improve the local score of x_i by removing the edge from x_k. If x_j is a type II node, and $D_{x_j} = \{x_i, x_k\}$, then $s'(x_j|\{x_i, x_k\}, S_{x_j})$ equals $(1/\alpha) \cdot s'(x_j|\{x_k\}, S_{x_j})$, which implies we can remove the edge from x_k without affecting the score of x_i. □

The preceding arguments also demonstrate the following theorem.

Theorem 12 *For any pair $\{x_i, \Pi_i\}$, where $|\Pi_i| \leq 2$, the value $p(x_i = 1|\Pi_i)$ can be computed from $\mathcal{B}$ in constant time when the state of each of the variable in Π_i is equal to one.*

12.3 References

[Chickering, 1995] Chickering, D. M. (1995). A Transformational characterization of Bayesian network structures. In *Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence,* Montreal, QU. Morgan Kaufman.

[Cooper and Herskovits, 1992] Cooper, G. and Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347.

[Garey and Johnson, 1979] Garey, M. and Johnson, D. (1979). *Computers and intractability: A guide to the theory of NP-completeness.* W.H. Freeman.

[Garvil, 1977] Garvil, F. (1977). Some NP-complete problems on graphs. In *Proc. 11th Conf. on Information Sciences and Systems,* Johns Hopkins University, pages 91–95. Baltimore, MD.

[Heckerman et al., 1995] Heckerman, D., Geiger, D., and Chickering, D. (1995). Learning discrete Bayesian networks. *Machine Learning*, 20:197-243.

[Höffgen, 1993] Höffgen, K. (revised 1993). Learning and robust learning of product distributions. Technical Report 464, Fachbereich Informatik, Universität Dortmund.