Ancestor Relations in the Presence of Unobserved Variables

Pekka Parviainen and Mikko Koivisto

Helsinki Institute for Information Technology HIIT, Department of Computer Science
University of Helsinki, Finland
{pekka.parviainen, mikko.koivisto}@cs.helsinki.fi

Abstract. Bayesian networks (BNs) are an appealing model for causal and non-causal dependencies among a set of variables. Learning BNs from observational data is challenging due to the nonidentifiability of the network structure and model misspecification in the presence of unobserved (latent) variables. Here, we investigate the prospects of Bayesian learning of ancestor relations, including arcs, in the presence and absence of unobserved variables. An exact dynamic programming algorithm to compute the respective posterior probabilities is developed, under the complete data assumption. Our experimental results show that ancestor relations between observed variables, arcs in particular, can be learned with good power even when a majority of the involved variables are unobserved. For comparison, deduction of ancestor relations from single maximum a posteriori network structures or their Markov equivalence class appears somewhat inferior to Bayesian averaging. We also discuss some shortcomings of applying existing conditional independence test based methods for learning ancestor relations.

1 Introduction

Directed acyclic graphs (DAGs) provide a convenient formalism for representing relationships among a set of variables in terms of *conditional independencies* (CIs) [17, 18]. To enable quantitative reasoning, a DAG is often attached to a probability measure that obeys exactly the CIs represented by the DAG. While the probability measure alone would of course suffice for probabilistic inference on the variables, the DAG contains additional structure that supports particularly causal interpretations: an arc between two variables represents a direct cause–effect relationship. The pair of the DAG and the measure is sometimes called a *Bayesian network*; the modifier "Bayesian" suggests a degree-of-belief interpretation of probability, which is applicable also when, for instance, the causal mechanisms are believed to be deterministic but just unknown to the modeller. Often a single Bayesian network is used for simultaneous modelling of several "similarly behaving" vectors of variables; then a node of the DAG corresponds to several random variables that are often treated as observations. If the nodes are observed, that is, the values of the respective random variables are known, standard principles of statistical inference can be implemented to derive more or less uncertain conclusions about the Bayesian network model, especially the DAG.

While automatic construction, or *learning*, of such DAGs from observational data is desirable, the task is notoriously challenging. First, a set of CIs can be represented by a number of different DAGs that form a so-called Markov equivalence class. Thus, the assumed "data generating DAG" cannot be identified by the represented CIs only. Second, if there are unobserved nodes at work, it may be that no DAG on the observed nodes can represent exactly the CIs among them. Then, the DAG model is misspecified in a way that directly affects the end result of statistical inference: the DAG. Third, the combinatorial and constrained nature of the DAG model brings major challenges concerning modeling complexity and, in particular, computational complexity.

To address these challenges, the art of learning DAGs from data has been developed in two rather distinct directions. Constraint-based methods [17, 18] rely on testing CIs. While the approach is not particularly suitable for importing prior knowledge, nor for efficient use of data, nor for managing nonidentifiability issues, it has given arise to a profound theory for dealing with unobserved variables. On the other hand, score-based methods [1, 11], particularly Bayesian ones [9, 15], excel in flexibility and statistical efficiency in the translation of what was known prior the observations to what is known *a posteriori*, including a proper treatment of nonidentifiability. For example, in the Bayesian approach there is no need to infer a single *maximum a posteriori* (MAP) DAG or its Markov equivalence class when there are many other almost equally good DAGs—instead, one may report *structural features*, e.g., arcs, that have a high posterior probability. As a drawback, it seems difficult to extend the Bayesian approach to handle the issue of unobserved nodes in a computationally efficient manner. Indeed, the score-based methods are often applied ignoring unobserved nodes altogether: either one refuses to make any conclusions, especially causal, about the DAG; or, one makes such conclusions at an unquantified risk of erroneous claims. While there are some notable exceptions that employ various score-driven heuristics to discover unobserved nodes [3–5, 8], principled methods are yet to be developed.

Motivated by these concerns, this paper investigates the potential of Bayesian learning of structural features of DAGs *on the observed nodes only*. Are there structural features that can be reliably learned from observational data, even if there may be some unobserved nodes at work? We find this question highly relevant and interesting, since the popular score-based methods for structure learning ignore unobserved nodes, which, however, are expected to be present in typical practical scenarios.

As a natural candidate for such a structural feature we consider *ancestor relations*. A node s is an ancestor of another node t if there is at least one directed path from s to t. An arc from s to t can be viewed as a special case of ancestor relations. The idea of learning ancestor relations from data is, of course, not new. Spirtes et al. [19] investigate constraint-based learning of ancestor relations using the FCI algorithm in a small-case empirical study. Their results suggest that reliable learning of ancestor relations is possible in the presence of unobserved nodes; however, direct comparison to our methods is not reasonable, as the predictions by FCI are unquantified and predictions are not necessarily made for all pairs of nodes. A Bayesian treatment is given by Friedman and Koller [9]: under the supposition that there are no unobserved nodes, DAGs are sampled (via node orderings) from their posterior distribution using a Markov chain Monte Carlo simulation and the posterior probabilities of ancestor relations, also called path

features, are estimated based on the sampled DAGs; based on the posterior probabilities, the ancestor relation is either claimed to hold or not to hold, potentially depending on the relative costs of making incorrect positive or negative claims.

Our present work contributes to this line of research by (a) giving a dynamic programming algorithm that computes the *exact* posterior probabilities of ancestor relations and by (b) studying the statistical power of learning such relations in the presence of *unobserved nodes*. From a computational point of view, ancestor relations present a new algorithmic challenge, as they do not fall in the class of modular features [9, 15]; see also a recent discussion by Tian et al. [21]. As can be expected, the computational complexity of the exact algorithm is exponential; the algorithm runs in $O(3^n n^2)$ time and $O(3^n)$ space on n-node instances. While such exponential complexity, of course, renders the algorithm computationally feasible only for relatively small instances, one should note that such moderately exponential time algorithms, that is, algorithms whose base constant is quite small, have attracted substantial interest in the context of Bayesian networks; see, e.g., Tian and He [20] and Kang et al. [12]. Both our algorithm and the power study assume that the prior over DAGs is of a restricted form, namely order-modular in the sense of Koivisto and Sood [15]; see also Friedman and Koller [9]. An order-modular prior generally assigns different prior probabilities to different DAGs within a Markov equivalence class. Compared to the uniform prior, this is, however, neither a disadvantage nor an advantage in general (besides the computational advantage), since the modeller's subjective prior may well be better represented with an order-modular prior than with the uniform prior. We also stress that, while our approach is fully Bayesian, the model is misspecified (does not fully represent the modeller's beliefs regarding unobserved nodes). Thus, the present work should be viewed as a study of the robustness of Bayesian averaging to model misspecification.

The remainder of the paper is structured as follows. We begin in Section 2 by reviewing a modular Bayesian network model [9, 15]. Then we give a dynamic programming algorithm for exact computation of the target posterior probabilities. Section 3 reports on empirical results concerning the statistical efficiency of learning ancestor relations and directed or undirected arcs with a varying number of observed nodes and data points per node. As an obvious (heuristic) alternative to Bayesian averaging, we also consider the deduction of ancestor relations from single MAP DAGs or their Markov equivalence classes. We also report on and discuss results obtained by the constraint-based algorithm, FCI [19] Finally, we summarize in Section 4.

2 Bayesian Discovery of Ancestor Relations

Our Bayesian network model relates a DAG on n nodes with m random variables per node (often treated as the observations or data; see below) by defining a joint probability measure on them.[1] Without any loss in generality we let the node set be $N = \{1, 2, \ldots, n\}$ and identify a DAG with its arc set $A \subseteq N \times N$; the set of *parents* of node v is $A_v = \{u : uv \in A\}$. If a DAG contains a directed path from u to v, then

[1] Note that while the m variables will be independent and identically distributed given a fully specified model, a Bayesian model also includes priors over the parameters and operates on exchangeability, not on independence.

u is called an *ancestor* of v, and v a *descendant* of u. With each node v we associate a sequence of random variables $D_v = D_{v1}D_{v2}\cdots D_{vm}$; we write D for $D_1D_2\cdots D_n$. A joint probability measure $p(A, D)$ is composed as $p(A, D) = p(A)p(D|A)$ with the following structure. By standard interpretation of conditional independencies on a DAG we have

$$p(D|A) = \prod_{v\in N} p(D_v|D_{A_v}, A_v) \ ;$$

for our purposes it is irrelevant how the local conditional measures $p(D_v|D_{A_v}, A_v)$ are further specified. For computational convenience, we define an *order-modular* prior for the DAG A. To this end, the joint prior of the DAG A and a linear order $L \subseteq N \times N$ on N is specified by

$$p(A, L) = \prod_{v\in N} \rho_v(L_v)q_v(A_v) \ ,$$

where $L_v = \{u : uv \in L\}$ consists of the predecessors of v in L and ρ_v and q_v are non-negative functions. The prior for the DAG is obtained by marginalizing the joint prior, that is, $p(A) = \sum_{L\supseteq A} p(A, L)$. Note that the sum is over all topological orderings of the DAG and that the set inclusion notation is valid (L is a superset of A). Note also that in practice the functions need to be specified only up to some normalization constant, e.g., $\rho_v(L_v) \propto 1$ and $q_v(A_v) \propto 1/\binom{n-1}{|A_v|}$, as the normalization constant will cancel in the quantities of our interest.

We consider a setting where the values of D, called the *data*, are observed, and we are interested in the posterior probability that the DAG A contains some specified structural feature. We will focus on two kinds of events that relate two nodes: uv is an arc in A, denoted $u{\to}v$; s is an ancestor of t in A, denoted $s{\rightsquigarrow}t$.

2.1 Computation

From an algorithmic point of view, it is convenient to compute the posterior probability of a structural feature $f(A)$ given the data D as the ratio $p(f(A), D)/p(D)$. Letting f be a 0–1-valued indicator function, we have $p(f(A), D) = \sum_A f(A)p(D|A)p(A)$, where the sum is over all DAGs on N. Koivisto and Sood [15] show that if $f(A)$ factorizes into a product of family-wise indicators $f_v(A_v)$, then the probabilities can be computed by dynamic programming (DP) across the node subsets of N in time $O(n^2 2^n)$ and space $O(n2^n)$; furthermore, the arc events $u{\to}v$ can be handled simultaneously for all the $n(n-1)$ node pairs uv within the same bounds [14].

The computation of the posterior probabilities of ancestor–descendant relationships seems more challenging, as the existence of directed path between two fixed nodes is a global property that does not factorize into independent local properties. We next give a DP algorithm that for every node subset S computes its contribution to the target probability, $p(s{\rightsquigarrow}t, D)$, assuming the nodes in S are the first $|S|$ nodes in the linear order L; the contribution is a sum over all possible DAGs, A_S, on the node set S. The key difference to the existing DP algorithms for arc probabilities or for the maximum posterior probability is that, aside from the set S, we need to keep a handle on the nodes

in S that are descendants of the source node s. To this end, we define a set $T \subseteq S$ such that $t \in T$ if and only if s is an ancestor of t or $t = s$. Thus, every DAG on S determines exactly one such set $T \subseteq S$.

Furthermore, for sets S and $T \subseteq S$ and a linear order $L_S \subseteq S \times S$ on the respective node set $S \subseteq N$, we use the shorthand

$$\mathcal{A}(L_S, S, T) = \{ A_S \subseteq L_S : \forall v \in S \, (s \rightsquigarrow v \text{ in } A_S \text{ iff } v \in T) \} ;$$

in words, $\mathcal{A}(L_S, S, T)$ contains a particular DAG A_S on S if and only if A_S is compatible with L_S and A_S contains a path from s to every node $v \in T$, and not to any other node in S.

Our dynamic programming algorithm will compute a function $g_s(S, T)$, defined for all $S \subseteq N$ and $T \subseteq S$ by

$$g_s(S, T) = \sum_{L_S} \sum_{A_S \in \mathcal{A}(L_S, S, T)} \prod_{v \in S} \rho_v(L_v) \beta_v(A_v) \ ,$$
$$\beta_v(A_v) = q_v(A_v) p(D_v | D_{A_v}, A_v) \ ,$$

where the outer summation is over all linear orders L_S on S. Intuitively, $g_s(S, T)$ is the sum of $p(A, D, L)$ over all DAGs A_S and linear orders L, with $A_S \subseteq L$, such that S are the first nodes in the order L and there is a path from s to $v \in S$ in A_S if and only if $v \in T$. That the values $g_s(S, T)$ are sufficient for computing the target quantity $p(s \rightsquigarrow t, D)$ is shown by the following result.

Lemma 1.

$$p(s \rightsquigarrow t, D) = \sum_{T : s, t \in T} g_s(N, T) \ .$$

Proof. The definitions directly yield

$$p(s \rightsquigarrow t, D) = \sum_{L} \sum_{\substack{A \subseteq L \\ s \rightsquigarrow t \text{ in } A}} \prod_{v \in N} \rho_v(L_v) \beta_v(A_v) \, ,$$

the outer summation being over all linear orders L on N.

We next break the inner summation into two nested summations by observing that the sets $\mathcal{A}(L, N, T)$, for $s, t \in T$, form a partition of the set $\mathcal{A}(L) = \{A \subseteq L : s \rightsquigarrow t \text{ in } A\}$: indeed, each DAG $A \in \mathcal{A}(L)$ determines precisely one node set T such that A contains a path from s to v for exactly the nodes in $v \in T$. Thus we have

$$\begin{aligned} p(s \rightsquigarrow t, D) &= \sum_{L} \sum_{T : s, t \in T} \sum_{A \in \mathcal{A}(L, S, T)} \prod_{v \in N} \rho_v(L_v) \beta_v(A_v) \\ &= \sum_{T : s, t \in T} \sum_{L} \sum_{A \in \mathcal{A}(L, S, T)} \prod_{v \in N} \rho_v(L_v) \beta_v(A_v) \\ &= \sum_{T : s, t \in T} g_s(N, T) \ . \end{aligned}$$

This completes the proof. □

From the algorithmic point of view, the pair (S,T) is sufficient for enabling a factorization of the sum over the A_S into independent sums over the parent sets A_v, for $v \in S$. Indeed, we have the following recurrence.

Lemma 2.

$$g_s(S,T) = 1 \qquad \textit{for } S \setminus \{s\} = \emptyset \textit{ and } s \in T\ ,$$
$$g_s(S,T) = 0 \qquad \textit{for } S \setminus \{s\} = \emptyset \textit{ and } s \notin T\ ,$$
$$g_s(S,T) = \sum_{v \in S} g_s(S \setminus \{v\}, T \setminus \{v\}) \rho_v(S \setminus \{v\}) \bar{\beta}_v(S,T) \qquad \textit{for } S \setminus \{s\} \neq \emptyset,$$

where

$$\bar{\beta}_v(S,T) = \begin{cases} \sum\limits_{\substack{A_v \subseteq S \setminus \{v\} \\ A_v \cap T \neq \emptyset}} \beta_v(A_v) & \textit{if } v \in T, \\ \sum\limits_{A_v \subseteq (S \setminus \{v\}) \setminus T} \beta_v(A_v) & \textit{if } v \in S \setminus T. \end{cases}$$

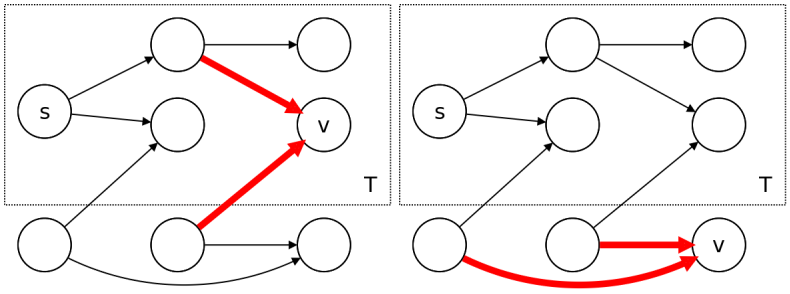
Proof. Proof is by straightforward induction on the size of S. First, observe that the sum over L_S in the definition of $g_s(S,T)$ breaks into a double-summation, in which the outer summation is over the last node $v \in S$ in the order L_S and the inner summation is over all linear orders, $L_{S \setminus \{v\}}$, on the remaining nodes $S \setminus \{v\}$. Second, observe that the summation over $A_S \in \mathcal{A}(L_S, S, T)$ breaks into a double-summation, in which the outer summation is over the DAGs $A_{S \setminus \{v\}} \in \mathcal{A}\big(L_{S \setminus \{v\}}, S \setminus \{v\}, T \setminus \{v\}\big)$ and the inner summation is over the parent sets $A_v \subseteq S \setminus \{v\}$ satisfying the requirement that (a) if there is no path from s to v (i.e., $v \notin T$), then there must be no path from s to u for any parent $u \in A_v$ of v, and (b) if there exists a path from s to v (i.e., $v \in T$), then there must exist a path from s to u for at least one parent u of v. □

Figure 1 illustrates the requirements on choosing parent sets for the node v in the last equation in Lemma 2. In Figure 1(a) $v \in T$, that is, it is required that there is a path from s to v. Now, we can choose any parent set for v as long as at least one of the parents is in T. On the other hand, in Figure 1(b) $v \notin T$, that is, it is required that there is no path from s to v. Now, we have to choose the parents of v from $S \setminus T$.

The evaluation of the values $g_s(S,T)$ using the recurrence is complicated by the fact that the inner summation, $\bar{\beta}_v(S,T)$, is over exponentially many sets A_v and, furthermore, there is a condition that depends not only on the set S but the set T. Fortunately, the inner summation can be precomputed for each $v \in N$ and $S \in N \setminus \{v\}$. Indeed, if $v \notin T$, then the sum is over all subsets A_v of $(S \setminus \{v\}) \setminus T$; if $v \in T$, then the sum is over all the remaining subsets of $S \setminus \{v\}$. Thus, it suffices to precompute

$$\hat{\beta}_v(U) = \sum_{A_v \subseteq U} \beta_v(A_v)$$

for all $U \subseteq N \setminus \{v\}$; the sums for the cases $v \notin T$ and $v \in T$ are then obtained as $\hat{\beta}_v((S \setminus \{v\}) \setminus T)$ and $\hat{\beta}_v(S \setminus \{v\}) - \hat{\beta}_v((S \setminus \{v\}) \setminus T)$, respectively. The function $\hat{\beta}_v$ is known as the zeta transform of β_v (over the subset lattice of $N \setminus \{v\}$), which can be



(a) (b)

Fig. 1. Choosing parent sets for a node $v \in S$ when (a) $v \in T$ and (b) $v \notin T$.

computed, given β_v, by the so-called fast zeta transform algorithm (see, e.g., [13, 15]) in time $O(n2^n)$ and space $O(2^n)$.

In summary, the values $g_s(S,T)$ for all $S \subseteq N$ and $T \subseteq S$ can be computed in time $O(n3^n)$ and space $O(3^n)$. The precomputation of the inner sum takes time $O(n^2 2^n)$ and space $O(n2^n)$ as noted above. Thus, the posterior probability that there exist a path from s to t, where s and t are two fixed nodes, can be computed in time $O(n3^n)$ and space $O(3^n)$. To compute the posterior probabilities for all node pairs st, it suffices to repeat the computations for each possible $s \in N$, for the values $g_s(S,T)$ actually contain the sufficient information regarding all possible descendant nodes t. Thus, in total, the time requirement is $O(n^2 3^n)$.

3 Experiments

Next we study how learning ancestor relations performs in practice. Our approach is to generate data from a known Bayesian network, called the *ground truth*, and compare the learned arcs and ancestor relations to the ground truth. Obviously, the learning performance is not expected to be perfect: when there are unobserved nodes at work, we easily learn arcs that are not present in the ground truth; this happens especially when an unobserved node is a common parent of two nodes that are not connected by an arc; namely, the two nodes are marginally dependent, and thus, in absence of the common parent, it is likely that we learn an arc between them, a false positive. On the other hand, we may expect that much of the structure can be learned even in presence of unobserved nodes. For example, if an unobserved node has exactly one child and one parent in the ground truth, both observed, then it is likely that the two arcs through the unobserved node in the middle will be just contracted to a single arc, which encodes a correct ancestor relation. We call the graph obtained from the ground truth by such contractions—that is, by connecting each parent of an unobserved node to every child of the node—the *shrunken ground truth*.

We have implemented the algorithm of Section 2.1 for Bayesian learning of ancestor relations in Matlab. In the experiments discussed next, we have used the BDeu score

with the equivalent sample size of 1, a uniform prior over linear orders on the nodes, and a uniform prior over parent sets of size at most a user-defined bound, which we set to 6.

3.1 Challenges of Learning Ancestor Relations

It is instructive to examine some representative challenges we face when learning ancestor relations and arcs. We consider a Bayesian network whose DAG is shown in Figure 2(a). All 14 variables are binary. The parameters of the network, that is, the probability of a node taking the value 1 given a particular value combination of its parents was drawn uniformly at random from the range $[0, 1]$ for each node and value combination of its parents. We generated 10 000 samples from the Bayesian network and learned ancestor relations from the data. Note that there are 16 arcs and 39 ancestor–descendant pairs in the ground truth. The DAG has quite a large Markov equivalence class, 140 graphs in total, and so one cannot expect reliable deduction of ancestor relations from a single MAP DAG.

For clarity of presentation, we discuss our findings mainly in terms of arcs instead of ancestor relations. Figure 2(b) shows arcs that are assigned a posterior probability of 0.5 or larger. Suppose we claim every arc or ancestor relation with probability 0.5 or larger to be present. Then, in total there are 12 true positive arcs, 4 false positive arcs, 20 true positive ancestor relations, and 4 false positive ancestor relations. Inspection reveals that the ancestor relation errors are due to a few flipped arcs. For example, in the ground truth there is a path from node 1 to eight different nodes. Thus, flipping the arc from 1 to 2 causes one false positive and eight false negative ancestor relations. While arc errors are rather independent, one flipped arc can lead to numerous ancestor relation errors, as seen earlier. It should also be noted that arc flips that are prone to cause a larger number of ancestor relation errors are also more probable. Namely, an arc is easily flipped when it does not break or create any v-structure, which is typically the case when one of the nodes is a source node in the ground truth.

The presence of unobserved nodes leads to claiming arcs between nodes that are only marginally dependent. In Figure 2(c) we see a DAG constructed from the arcs with probability 0.5 or larger when nodes 1, 4, 7, and 11 are discarded. Node 1 does not have children, so its disappearance should not affect the structure among the rest of the nodes. However, the removal of nodes 4, 7, and 11 affects the rest of the nodes: For instance, node 11 is a common cause of nodes 13 and 14, and so an arc appears between nodes 13 and 14. Also, nodes 5 and 6, which are parents of node 7 in the ground truth, have become parents of node 10, a child of node 7 in the ground truth. Similarly, the removal of node 4 leads also to appearance of some direct arcs from its parents to its children. After discarding the unobserved nodes, the shrunken ground truth contains 14 arcs and 18 ancestor relations. The algorithm finds 8 true positive arcs, 6 false positive arcs, 11 true positive ancestor relations, and 7 false positive ancestor relations. This suggests that ancestor relations can sometimes be learned as well as individual arcs.

For comparison, we also learned a *partial ancestral graph* (PAG) from the data with unobserved nodes using the fast causal inference (FCI) algorithm [18], which is designed for causal discovery with unobserved variables. The output graph is shown in Figure 2(d). An arc marked with two arrowheads indicates that the algorithm claims the

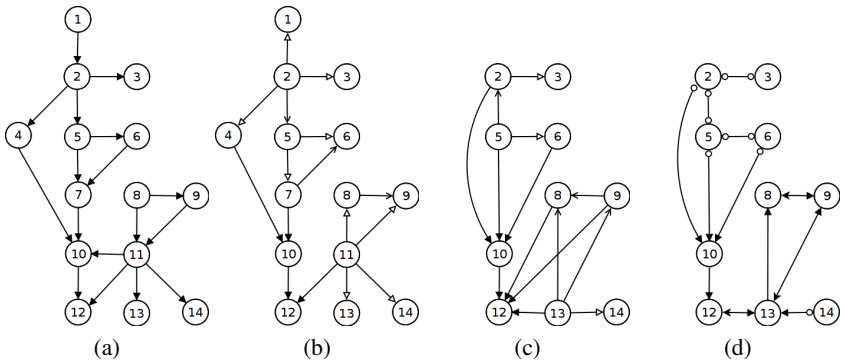


Fig. 2. Graphs. (a) The ground truth, from which 10 000 samples were generated. (b) Arcs with posterior probability at least 0.5. The arrowheads $>$, $\triangleright$, and $\blacktriangleright$ indicate that the probability is in the interval $(0.5, 0.8]$, $(0.8, 0.99]$, or $(0.99, 1]$, respectively. (c) Arcs with posterior probability at least 0.5 when nodes 1, 4, 7, and 11 are not observed. (d) A partially directed graph learned using the FCI algorithm when nodes 1, 4, 7, and 11 are not observed.

two nodes have a common (unobserved) cause; the symbol $\circ$ is a wildcard, indicating that there can be an arrowhead or there is no arrowhead. The results are generally in good agreement with the ground truth. The FCI algorithm is able to detect the unobserved parent of nodes 12 and 13. However, it is not sure whether there is an unobserved parent between nodes 13 and 14, and it is unable to detect the unobserved parent between nodes 12 and 14. It also finds an unobserved parent between nodes 8 and 9, which is not in agreement with the ground truth. As the wildcards assigned by the FCI algorithm do not quantify the uncertainty about the associated arcs, but the algorithm is ignorant regarding some ancestor relations, the algorithm may loose statistical power in detecting such relations; we will examine and discuss this issue further in the next section.

3.2 A Simulation Study

We generated synthetic data by a procedure adopted from Koivisto [14]. One hundred BNs on 14 binary nodes and maximum indegree 4, each with 10000 data points were obtained as follows.

1. Draw a linear order L on the node set $\{1, 2, \ldots, 14\}$ uniformly at random (u.a.r.).
2. For each node v independently:
   (a) let d_v be the number of predecessors of v in L;
   (b) draw the number of parents of v, denoted as n_v, from $\{0, 1, \ldots, \min\{4, d_v\}\}$ u.a.r.;
   (c) draw the n_v parents of v from the predecessors of v in L u.a.r.;
   (d) for each value configuration of the parents: draw the probability of a sample getting the value 1 from the uniform distribution on range $[0, 1]$.
3. Draw 10000 samples independently from the BN.

From each data set 24 subsets were generated by discarding $\ell = 0, 2, 4, 6, 8, 10$ randomly picked nodes and the associated data, and by including the first $m = 100, 500, 2000, 10000$ data points.

Our Bayesian method was applied to each data set and the performance of learning arcs and ancestor relations was summarized by ROC curves; see Figure 2. The ROC curve is obtained by setting a threshold for the posterior probability (of arcs or ancestor relations), and every time the posterior probability exceeds the threshold, we claim the respective arc or ancestor relation is present. Comparing these claims to the arc and ancestor relations that actually hold in the (shrunken) ground truth, we obtain true positives (TP) and false positives (FP) rates. By varying the threshold the pairs of these rates form a ROC curve, which shows the learning power (TP rate) as a function of FP rate.

As expected, the more data we have, the easier it is to learn both ancestor relations and arcs. Likewise, the task becomes harder as the number of unobserved nodes grows. (We note that the results for undirected arcs in the case of no unobserved nodes are in good agreement with Koivisto's [14] results for this particular setting.) The results (Figure 2) also suggest that the power of learning directed and undirected arcs is about the same, however, the power of learning ancestor relations being slightly smaller. The running times of the algorithm for 10, 12, and 14 observed nodes were roughly 3 minutes, 40 minutes, and 8 hours, respectively.

We then compared our Bayesian averaging approach to the deduction of structural features from a single MAP DAG. Two ways to pick a MAP DAG were considered: an optimistic and a random approach. In the optimistic approach we chose a member of the Markov equivalence class of a MAP DAG that yields the largest true positives rate, and used its true and false positives rates. This approach is arguably unrealistic in practice but serves as an upper bound for any approach based on a single MAP DAG. In the random approach we averaged the true and false positives rates over all DAGs in the Markov equivalence class of a MAP DAG; the averaged rates correspond to the respective expectations if one picks such a DAG at random. The true and false positives rates for these two approached are shown in Tables 1 and 2; column "diff." shows the difference between the true positives rates of the MAP DAG approach and the Bayesian averaging approach (the averages of the false positives rate being matched, of course); a negative value indicates that the Bayesian averaging approach is more powerful.

The results suggest that the random MAP DAG approach performs significantly worse than Bayesian averaging. On the other hand, the optimistic MAP DAG approach performs sometimes better than Bayesian averaging, especially when the data are abundant and there are many unobserved nodes.

Furthermore, we compared our method to the deduction of ancestor relations from the arc probabilities. To this end, we constructed a graph that consisted of the arcs whose posterior probability was larger than 0.5, that is, the arc is more likely to be present than absent, and deducted the ancestor relations from this graph. The results (Tables 1 and 2) show that the performance of the deduction of ancestor relations from arcs does not differ significantly from learning ancestor relations directly. We further compared the aforementioned approach to direct learning of ancestor relations. To this end, we assumed that exactly the ancestor relations whose probability is more than 0.5 exist and

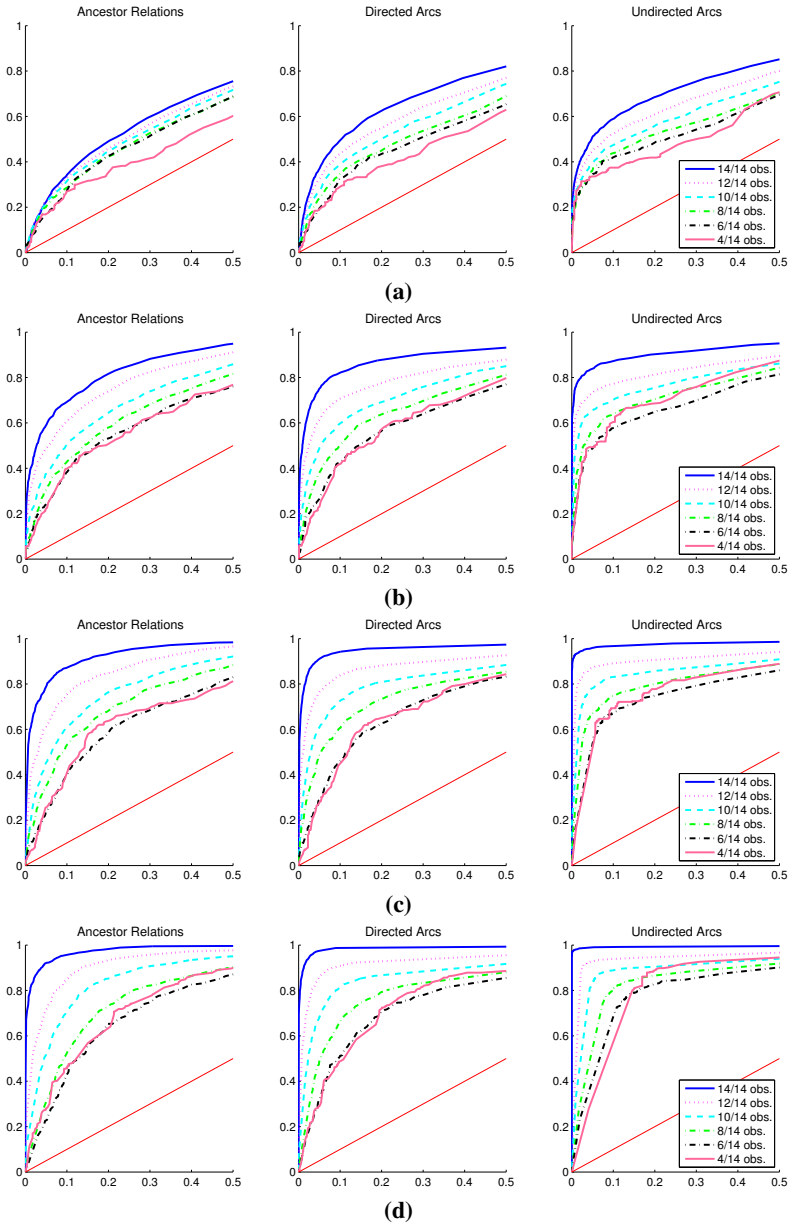


Fig. 3. ROC curves. The data contain (a) 100, (b) 500, (c) 2000 or (d) 10000 samples over 14 nodes. The straight red line is the curve obtained by random guess. The data-generating graphs contained on average 23.7 arcs and the shrunken ground truths on average 19.7, 15.9, 11.1, 7.0, and 3.3 arcs for 12, 10, 8, 6, and 4 observed nodes, respectively.

Table 1. Comparison of TP and FP rates for ancestor relations

		Opt. MAP DAG			Rand. MAP DAG			Arcs > 0.5			FCI		
m	ℓ	TP	FP	diff.	TP	FP	diff.	TP	FP	diff.	TP	FP	diff.
100	0	0.41	0.19	-0.09	0.37	0.21	-0.14	0.20	0.04	-0.01	0.002	0.000	0.002
100	2	0.35	0.16	-0.07	0.30	0.18	-0.14	0.17	0.03	-0.01	0.001	0.000	-0.001
100	4	0.34	0.11	0.01	0.27	0.14	-0.11	0.16	0.03	-0.01	0.002	0.000	0.002
100	6	0.34	0.08	0.07	0.24	0.12	-0.08	0.15	0.03	-0.00	0.002	0.000	0.002
100	8	0.36	0.05	0.19	0.23	0.09	-0.01	0.12	0.03	0.00	0.003	0.000	0.003
100	10	0.31	0.04	0.16	0.17	0.09	-0.06	0.12	0.02	0.01	0.000	0.000	0.000
500	0	0.65	0.10	-0.04	0.61	0.13	-0.12	0.58	0.04	0.01	0.014	0.002	-0.218
500	2	0.61	0.11	-0.02	0.54	0.14	-0.13	0.50	0.05	0.01	0.014	0.002	-0.143
500	4	0.53	0.11	0.01	0.45	0.14	-0.12	0.42	0.06	0.02	0.010	0.001	-0.073
500	6	0.50	0.10	0.06	0.40	0.14	-0.11	0.35	0.06	-0.01	0.011	0.000	-0.028
500	8	0.48	0.07	0.19	0.33	0.12	-0.10	0.27	0.06	0.00	0.014	0.000	0.014
500	10	0.51	0.05	0.26	0.32	0.12	-0.10	0.27	0.06	-0.02	0.005	0.000	0.005
2000	0	0.84	0.06	0.02	0.78	0.08	-0.08	0.78	0.05	0.01	0.048	0.004	-0.482
2000	2	0.76	0.10	0.02	0.70	0.12	-0.09	0.69	0.07	0.02	0.047	0.005	-0.329
2000	4	0.67	0.12	0.01	0.60	0.15	-0.11	0.60	0.09	0.02	0.041	0.007	-0.217
2000	6	0.64	0.12	0.06	0.54	0.16	-0.10	0.51	0.09	0.01	0.037	0.004	-0.090
2000	8	0.59	0.12	0.14	0.45	0.17	-0.09	0.40	0.10	-0.00	0.020	0.002	-0.033
2000	10	0.69	0.07	0.36	0.44	0.18	-0.18	0.41	0.09	0.07	0.005	0.000	0.005
10000	0	0.93	0.02	0.07	0.86	0.06	-0.06	0.87	0.02	0.00	0.129	0.011	-0.660
10000	2	0.86	0.08	0.06	0.79	0.11	-0.08	0.79	0.07	-0.00	0.121	0.010	-0.410
10000	4	0.80	0.11	0.07	0.70	0.15	-0.11	0.70	0.09	0.01	0.100	0.015	-0.326
10000	6	0.73	0.13	0.11	0.62	0.18	-0.10	0.60	0.13	0.00	0.086	0.010	-0.199
10000	8	0.72	0.14	0.19	0.57	0.20	-0.09	0.54	0.14	0.02	0.037	0.006	-0.125
10000	10	0.84	0.09	0.38	0.57	0.21	-0.11	0.54	0.14	-0.03	0.014	0.002	-0.025

cross-tabulated the ancestor relations predictions for deducting the ancestor relations from arcs and the direct computation of ancestor relations; see Table 3. Table 3 shows the average number of the node pairs for which either both methods, only the deduction from arc probabilities, only the direct computation of ancestor relation probabilities or neither method claims an ancestor relation to be present. Table 3 also shows the probability that the claim made by the direct computation is correct; NaN denotes that no claims falling into the particular category were made. Most of the time, both methods make the same predictions. Whenever the predictions differ, the prediction by direct computation is usually slightly more probable to be correct. We also notice that the two methods follow each other closely with larger datasets.

We also compared our method to the fast causal inference (FCI) method [18]; see Tables 1 and 2. We found it quite challenging to make a fair comparison because FCI outputs a partial ancestral graph (PAG) that cannot be directly compared to a DAG. We decided to ignore the wildcard arcs and claim only arcs and ancestor relations that FCI is sure about; this follows the approach of Spirtes et al. [19]. The results (Tables 1 and 2) show that FCI is very conservative: it does not make many mistakes but it often answers "don't know". This results in a relatively low statistical power of discovering arcs and

Table 2. Comparison of TP and FP rates for arcs

		Opt. MAP DAG			Rand. MAP DAG			Arcs > 0.5			FCI		
m	ℓ	TP	FP	diff.	TP	FP	diff.	TP	FP	diff.	TP	FP	diff.
100	0	0.34	0.05	-0.05	0.31	0.06	-0.10	0.26	0.03	0.00	0.003	0.000	0.003
100	2	0.30	0.06	-0.04	0.26	0.06	-0.11	0.22	0.02	0.00	0.002	0.000	-0.001
100	4	0.28	0.05	0.01	0.23	0.06	-0.08	0.18	0.02	0.00	0.003	0.000	0.003
100	6	0.28	0.04	0.05	0.21	0.06	-0.06	0.16	0.03	0.00	0.002	0.000	0.002
100	8	0.30	0.03	0.15	0.20	0.06	-0.02	0.13	0.03	0.00	0.003	0.000	0.003
100	10	0.31	0.03	0.15	0.18	0.07	-0.05	0.14	0.03	0.00	0.000	0.000	0.000
500	0	0.64	0.03	-0.03	0.60	0.03	-0.09	0.62	0.02	0.00	0.023	0.001	-0.273
500	2	0.55	0.03	0.00	0.50	0.04	-0.09	0.52	0.03	0.00	0.020	0.002	-0.153
500	4	0.46	0.04	0.02	0.41	0.05	-0.09	0.42	0.03	0.00	0.014	0.001	-0.076
500	6	0.42	0.04	0.06	0.34	0.06	-0.08	0.35	0.04	0.00	0.012	0.000	-0.026
500	8	0.42	0.04	0.18	0.30	0.07	-0.04	0.28	0.05	0.00	0.016	0.000	0.016
500	10	0.49	0.04	0.28	0.32	0.08	-0.08	0.30	0.07	0.00	0.005	0.000	0.005
2000	0	0.83	0.02	0.03	0.78	0.02	-0.06	0.81	0.02	0.00	0.075	0.003	-0.511
2000	2	0.71	0.03	0.02	0.66	0.04	-0.06	0.69	0.03	0.00	0.067	0.003	-0.319
2000	4	0.60	0.05	0.00	0.55	0.06	-0.08	0.59	0.04	0.00	0.054	0.005	-0.206
2000	6	0.55	0.05	0.05	0.47	0.07	-0.07	0.50	0.06	0.00	0.042	0.004	-0.088
2000	8	0.51	0.07	0.16	0.40	0.10	-0.05	0.41	0.08	0.00	0.023	0.002	-0.029
2000	10	0.65	0.06	0.33	0.43	0.12	-0.09	0.44	0.10	0.00	0.005	0.000	0.005
10000	0	0.93	0.01	0.08	0.87	0.02	-0.04	0.89	0.01	0.00	0.173	0.006	-0.672
10000	2	0.83	0.03	0.07	0.77	0.04	-0.05	0.80	0.03	0.00	0.146	0.006	-0.395
10000	4	0.75	0.05	0.08	0.67	0.06	-0.06	0.70	0.05	0.00	0.111	0.011	-0.304
10000	6	0.67	0.07	0.09	0.58	0.09	-0.06	0.61	0.08	0.00	0.090	0.009	-0.190
10000	8	0.64	0.09	0.15	0.52	0.12	-0.04	0.54	0.11	0.00	0.040	0.006	-0.118
10000	10	0.79	0.08	0.34	0.56	0.15	-0.09	0.58	0.14	0.00	0.015	0.002	-0.027

ancestor relations, sometimes significantly lower than that of the Bayesian averaging approach (at matched FP rates).

One should notice, though, that FCI can discover unobserved nodes with some success. However, usually the unobserved nodes that FCI "finds," do not seem to match the ones in the ground truth. For example, when the sample size is 2000 and there are no unobserved nodes, FCI finds on average 6.0 unobserved nodes. And when there are two unobserved nodes, only 11% of the "found" 4.8 unobserved nodes match the ground truth. In general, as the number of unobserved nodes increases, the number of found unobserved nodes decreases, but the percentage of correctly detected unobserved nodes increases; for example, when there are 8 unobserved nodes 54% of the claimed 0.7 unobserved nodes are correct.

3.3 Real Life Data

We tested our algorithm on two real life datasets found in UCI machine learning repository [7]: ADULT (15 variables, 32561 samples) and HOUSING (14 variables, 506 samples). We discretized all continuous variables to binary variables using the median as

Table 3. Ancestor Relations predicted by arcs and direct computation

		Predicted Ancestor Relations				Correct Predictions by dir. comp.			
m	ℓ	both	arcs	direct	none	both	arcs	direct	none
100	0	13.6	1.1	1.8	165.5	0.61	0.64	0.50	0.79
100	2	8.3	0.4	0.9	122.4	0.63	0.59	0.61	0.79
100	4	5.3	0.3	0.5	84.0	0.63	0.52	0.59	0.78
100	6	3.1	0.2	0.2	52.5	0.62	0.56	0.57	0.78
100	8	1.4	0.1	0.0	28.4	0.59	0.25	1.00	0.77
100	10	0.6	0.0	0.0	11.4	0.68	NaN	1.00	0.77
500	0	30.5	0.5	1.3	149.7	0.82	0.48	0.53	0.88
500	2	20.7	0.4	0.6	110.2	0.76	0.77	0.48	0.86
500	4	12.7	0.5	0.6	76.3	0.70	0.65	0.35	0.84
500	6	7.0	0.2	0.3	48.5	0.66	0.81	0.63	0.82
500	8	3.3	0.1	0.1	26.5	0.61	0.56	0.45	0.79
500	10	1.3	0.0	0.0	10.6	0.60	0.33	NaN	0.79
2000	0	39.7	0.2	0.4	141.8	0.85	0.82	0.39	0.93
2000	2	28.4	0.2	0.4	103.0	0.76	0.55	0.61	0.91
2000	4	18.6	0.2	0.4	70.8	0.69	0.47	0.30	0.88
2000	6	10.6	0.2	0.3	44.9	0.64	0.70	0.47	0.86
2000	8	5.2	0.1	0.1	24.6	0.58	0.89	0.54	0.82
2000	10	2.0	0.1	0.1	9.9	0.61	0.25	0.60	0.82
10000	0	40.9	0.1	0.4	140.7	0.92	0.60	0.61	0.96
10000	2	31.2	0.2	0.3	100.3	0.79	0.78	0.32	0.94
10000	4	21.6	0.1	0.2	68.0	0.71	0.60	0.36	0.91
10000	6	13.3	0.2	0.2	42.3	0.60	0.68	0.53	0.88
10000	8	7.2	0.0	0.1	22.7	0.57	0.75	0.44	0.85
10000	10	2.8	0.0	0.0	9.1	0.58	0.67	0.00	0.84

the cutpoint. Furthermore, we transformed the variable "native-country", which had 40 distinct values, of the ADULT dataset to a binary variable where 0 corresponded to value "USA" and 1 to all other values. For both datasets we set the maximum indegree to be 4.

Here, we do not know the ground truth and thus we have to resort to other comparisons. We investigate whether learning ancestor relations uncovers some information that cannot be obtained simply analyzing the arc probabilities. To this end, we deduct ancestor relations both from the ancestor relations probabilities and the arc probabilities. For ADULT, we have 210 potential ancestor relations. Both methods imply the presence of the same 79 ancestor relations. For HOUSING the methods are in almost as good agreement as for the ADULT. For 71 ordered pairs, both methods claim that an ancestor relation is present and for 110 pairs that an ancestor relation is not present. There is, however, one node pair for which the deduction from arcs suggests that there is no ancestor relation while the deduction from ancestor probabilities claims the opposite. This discrepancy is, though, due to the arbitrariness of the threshold. We notice that the posterior probability of an arc between the two nodes in question was 0.49 while the probability of an ancestor relation was 0.53.

4 Discussion

A key assumption in Bayesian network models is that all nodes relevant for capturing the dependencies of the associated variables are included in the model. One can argue that this assumption rarely holds in practice, and so the model is misspecified. Note, however, that in practice, every complex enough model is misspecified in one way or another. The issue, in general, calls for robustness studies, disregarding whether the adopted statistical paradigm is a frequentist or a Bayesian one. In this paper we have studied the power of Bayesian structure discovery in Bayesian networks that do not explicitly model latent variables.

We contributed with two positive findings. First, we showed that Bayesian learning of ancestor relationships is computationally feasible when the number of observed nodes is moderate, say, fewer than 20. The algorithm resembles the dynamic programming algorithm of Koivisto and Sood [15] for computing the posterior probabilities of *modular* features, the main difference being in handling the *nonmodularity* of ancestor relations, which explains the somewhat larger computational complexity; this suggests that recent discussions of the limitation of the "exact Bayesian approach" to modular features [2, 21] may be overly pessimistic. For larger networks on, say, more than 20 nodes, the dynamic programming algorithm becomes computational infeasible and one has to resort to heuristic methods, in particular, Markov chain Monte Carlo [6, 9, 10, 16].

Second, our simulation study shows that ancestor relations can be discovered with reasonable power even when a large fraction of the nodes in the underlying data generating model are unobserved. For instance, with a sample of 10000 data points on 10 nodes, around 75 % of the ancestor relations that hold on the data generating network on 14 nodes are correctly detected at a false positive fraction of 12 %.

We also found that the presented Bayesian averaging approach outperforms some of its obvious rivals: the deduction of ancestor relations from a single MAP DAG and the popular constraint-based algorithm, FCI [19]. On the other hand, we found that full Bayesian averaging performs only marginally better than partial Bayesian averaging, that is, first inferring arcs based on their marginal posterior probabilities, with some fixed threshold, and then deducing ancestor relations from the so constructed DAG; this suggests that partial Bayesian averaging should be the method of choice when the number of nodes is about 20–30. Although someone may perceive the competitiveness of partial Bayesian averaging as a drawback for full Bayesian averaging, it should be noted that the insight about the competitiveness of partial Bayesian averaging was gained by being able to perform full Bayesian averaging. An intriguing open question we did not address this work is, how well some existing score-based heuristics [4] to discover unobserved nodes perform in terms of learning arcs and ancestor relations.

Acknowledgements

Authors like to thank anonymous reviewers for insightful comments and suggestions. This research was supported in part by the Academy of Finland, Grant 125637.

References

1. G. F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347, 1992.
2. D. Eaton and K. Murphy. Bayesian structure learning using dynamic programming and MCMC. In *Proceedings of the Twenty-Third Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, 2007.
3. G. Elidan and N. Friedman. Learning the dimensionality of hidden variables. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 144–151, 2001.
4. G. Elidan and N. Friedman. Learning hidden variable networks: The information bottleneck approach. *Journal of Machine Learning Research*, 6:81–127, 2005.
5. G. Elidan, N. Lotner, N. Friedman, and D. Koller. Discovering hidden variables: A structure-based approach. In *Advances in Neural Information Processing Systems 13 (NIPS) 2000*, 2000.
6. B. Ellis and W. H. Wong. Learning causal Bayesian network structures from data. *Journal of American Statistical Association*, 103:778–789, 2008.
7. A. Frank and A. Asuncion. UCI machine learning repository, 2010.
8. N. Friedman. Learning belief networks in the presence of missing values and hidden variables. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML)*, 1997.
9. N. Friedman and D. Koller. Being Bayesian about network structure: A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning*, 50(1–2):95–125, 2003.
10. M. Grzegorczyk and D. Husmeier. Improving the structure MCMC sampler for Bayesian networks by int roducing a new edge reversal move. *Machine Learning*, 71:265–305, 2008.
11. D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995.
12. E. Y. Kang, I. Shpitser, and E. Eskin. Respecting Markov equivalence in computing posterior probabilities of causal graphical features. In *Proceeding of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)*, pages 1175–1180, 2010.
13. R. Kennes. Computational aspects of the Möbius transformation of graphs. *IEEE Transaction on Systems, Man, and Cybernetics*, 22(2):201–223, 1992.
14. M. Koivisto. Advances in exact Bayesian structure discovery in Bayesian networks. In *Proceedings of the Twenty-Second Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, 2006.
15. M. Koivisto and K. Sood. Exact Bayesian structure discovery in Bayesian networks. *Journal of Machine Learning Research*, 5:549–573, 2004.
16. D. Madigan and J. York. Bayesian graphical models for discrete data. *International Statistical Review*, 63:215–232, 1995.
17. J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge university Press, 2000.
18. P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. Springer Verlag, 2000.
19. P. Spirtes, C. Meek, and T. Richardson. Causal inference in the presence of latent variables and selection bias. In *Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, 1995.
20. J. Tian and R. He. Computing posterior probabilities of structural features in Bayesian networks. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2009.
21. J. Tian, R. He, and L. Ram. Bayesian model averaging using the k-best Bayesian network structures. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2010.