A Simulated Annealing-Based Method for Learning Bayesian Networks from Statistical Data

Martin Janžura,* Jan Nielsen†
Institute of Information Theory and Automation, Prague, Czech Republic

The problem of learning Bayesian networks from statistical data is described and reformulated as a discrete optimization problem. For a solution we employ the stochastic algorithm that is known as simulated annealing and that is based on the Markov Chain Monte Carlo approach. Numerical examples are included to illustrate the efficiency of the method. © 2006 Wiley Periodicals, Inc.

1. INTRODUCTION

The problem of learning Bayesian networks can be understood as a statistical estimation problem, with the unknown "parameter" given by the network structure to be estimated from a sequence of observed data configurations. The problem has been rather widely treated (see, e.g., Refs. 1–4, or, recently, Refs. 5 and 6). On the other hand, the simulated annealing is an optimization method based on the Markov Chain Monte Carlo (MCMC) idea. It is extensively employed in the area of image processing (see, e.g., Refs. 7, 8, and 9), whenever the Maximum Aposterior Probability (MAP) principle is followed, as well as in the standard discrete optimization large-scale problems like the "traveling salesman" and related ones.

The aim of the present article is to show that the simulated annealing method can be applied to the problem of learning Bayesian networks if it is reformulated as an optimization problem. The idea is surely not pioneering and the article is oriented more to the question of practical feasibility of the approach. From the computational point of view, the problem is characteristic and therefore difficult because of an extremely large number of hypothetical models, many of them equivalent or at least hardly distinguishable, and unsuitable properties of the objective function with many local optima.

*Author to whom all correspondence should be addressed: e-mail: janzura@utia.cas.cz.
†e-mail: jnielson@kaktus.cz.

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 21, 335–348 (2006)
 Published online in Wiley InterScience (www.interscience.wiley.com). • DOI 10.1002/int.20138

The feasibility of our method is demonstrated with the aid of rather simple but nontrivial examples with simulated data. Then we know the true model, and, moreover, the equivalence between the true and the estimated models can be exactly verified. A comparison with other competing methods, which could strongly justify the proposed approach, would need much more effort and is beyond the intention and scope of this article.

2. REPRESENTATION OF BAYESIAN NETWORK

By a Bayesian network we understand, as usual, a multivariate probability distribution that admits a recursive factorization according to a directed acyclic graph (see, e.g., Refs. 10 and 11).

In particular, let (G, E) be a (finite) directed acyclic graph, where G is a finite set of vertices (nodes, sites) and E is a collection of oriented edges. Further, let us consider the state space $X_G = \otimes_{g\in G} X_g$, with $|X_g| < \infty$ for every $g \in G$. A probability distribution P, defined on X_G, admits the recursive factorization according to (G, E) if there exists a system of probability kernels $\{Q_g(x_g|x_{pa(g)})\}_{g\in G}$ so that

$$P(x_G) = \prod_{g\in G} Q_g(x_g|x_{pa(g)}) \tag{1}$$

for every $x_G \in X_G$. Here $pa(g)$ denotes the set of parents of the vertex g in the directed graph (G, E).

By an arbitrary fixed enumeration of the graph vertices, we may assume $G = \{1, \ldots, K\}$. Then the directed acyclic graph can be alternatively described by a sequence of pairs $\{(R(k), A(k))\}_{k=1,\ldots,K}$ where $\{R(k)\}_{k=1,\ldots,K}$ is an ordering of the vertices and

$$A(k) \subset R(k)^* = \{j : R(j) < R(k)\}$$

for every $k = 1, \ldots, K$ denotes now the set of parents. The sequence

$$M = \{(R(k), A(k))\}_{k=1,\ldots,K} \tag{2}$$

will be quoted as a (graphical) model, and we shall denote by $\mathcal{M}$ the set of all admissible models.

For a visualization of the model, we prefer the form as used in Figure 1, which is more convenient for our definition of the graphical model. Here parents are always placed to the left of their child; thus the orientation of edges is obvious

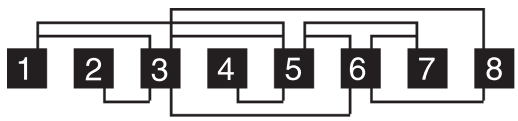

Figure 1. Example of a directed acyclic graph.

and can be suppressed in the picture. In the particular case in Figure 1, we have $R(k) = k$ for $k = 1, \ldots, 8$ and

$$M = \{(1,\varnothing),(2,\varnothing),(3,\{1,2\}),(4,\varnothing),(5,\{1,3,4\}),(6,\{3,5\}),(7,\{5,6\}),(8,\{3,6\})\} \tag{3}$$

The Bayesian network P is fully determined by a graphical model M together with the system of probability kernels $\mathcal{Q} = \{Q_g(.|.)\}_{g\in G} = \{Q_{k|A(k)}(.|.)\}_{k\in 1,\ldots,K}$. The pair $(M, \mathcal{Q})$ will be quoted as a probabilistic model, and the set of all probabilistic models will be denoted by $\mathcal{P}$. Nevertheless, as is well known, such a representation of a Bayesian network by its probabilistic model may not be defined uniquely, that is, different probabilistic models may yield the same Bayesian network. In such a way an equivalence relation on the set $\mathcal{P}$ of probabilistic models is induced. For a comprehensive study of the problem of equivalence, see, for example, Ref. 5 and the references therein.

3. PROBLEM

The problem of learning a Bayesian network, which is the problem to be solved, consists of identifying the probabilistic model $(M, \mathcal{Q})$ from a sequence of independent statistical data $x_G^{(1)}, \ldots, x_G^{(N)} \in X_G$. This sequence determines the empirical distribution

$$\hat{P}^{(N)}(y_G) = \frac{1}{N}\sum_{i=1}^{N} \delta(y_G, x_G^{(i)}) \tag{4}$$

which is a sufficient statistics here; namely, it holds

$$\Pr(x_G^{(1)}, \ldots, x_G^{(N)}) = \prod_{y_G\in X_G} \Pr(y_G)^{N.\hat{P}^{(N)}(y_G)} \tag{5}$$

Thus, it contains the same amount of information as the original data, and, in what follows, we shall deal directly rather with the empirical distributions.

3.1. Solution

In principle, we shall solve the problem by minimizing a suitable objective function, namely

$$\min_{(M,\mathcal{Q})\in\mathcal{P}} F(M, \mathcal{Q}) \tag{6}$$

where we set

$$F(M, \mathcal{Q}) = D(\hat{P}^{(N)}, P^{M,\mathcal{Q}}) + T_N(M) \tag{7}$$

with some appropriate measure of distance or dissimilarity $D(.,.)$, and a penalty term $T_N(M)$ reflecting our prior knowledge or belief on the structure of the graphical model. Here $\hat{P}^{(N)}$ is the empirical distribution defined above and $P^{M,\mathcal{Q}} = \prod_{k=1}^{K}\{Q_{k|A(k)}(.|.)\}$ is the unknown Bayesian network.

As a standard choice, we apply the information (Kullback–Leibler) divergence

$$D(p,q) = I(p|q) = \int \log \frac{dp}{dq}\, dp \tag{8}$$

as a reasonable measure of dissimilarity.

Let us denote by $\hat{P}^M$ the estimate of the probability distribution within the graphical model M, that is,

$$\hat{P}^M(x_G) = \prod_{k=1}^{K} \hat{P}_{k|A(k)}(x_k|x_{A(k)}) \tag{9}$$

with the empirical conditional marginals $\hat{P}_{\bullet|\bullet}$.

THEOREM 1. *Let* $P^{\hat{M},\hat{\mathcal{Q}}} = \arg\min_{(M,\mathcal{Q})\in\mathcal{P}} F(M,\mathcal{Q})$. *Then* $P^{\hat{M},\hat{\mathcal{Q}}} = \hat{P}^{\hat{M}}$, *that is,* $\hat{Q}_{k|\hat{A}(k)} = \hat{P}_{k|\hat{A}(k)}$ *for* $k = 1,\ldots,K$.

Proof. The statement follows directly from the Pythagorean equality

$$I(\hat{P}^{(N)}|P^{M,\mathcal{Q}}) = I(\hat{P}^{(N)}|\hat{P}^M) + I(\hat{P}^M|P^{M,\mathcal{Q}}) \qquad \blacksquare$$

The above proposition implies the minimum of $F(M,\mathcal{Q})$ to be attained at some $\hat{P}^M$ with $M \in \mathcal{M}$, and therefore we may restrict our considerations to solving the problem

$$\min_{M\in\mathcal{M}} \tilde{F}(M) \tag{10}$$

where

$$\tilde{F}(M) = I(\hat{P}^{(N)}|\hat{P}^M) + T_N(M) \tag{11}$$

Now, a convenient choice of the *penalty term* $T_N(M)$ is needed. Usually, no prior information about the model is available but a vaguely formulated belief that the graph structure "should not be as much complex." More scholarly, we could speak about an application of the Occam's razor principle. In the present article, we deal with the penalty term in the form

$$T_N(M) = c_N\left\{\beta \sum_{k=1}^{K}\sum_{j\in A(k)} |R(k) - R(j)| + \gamma \sum_{k=1}^{K} |A(k)|\right\} \tag{12}$$

where β and γ are (positive) constants expressing the strength of our prior belief, and c_N is a constant depending only on the sample size, that is, the number of data N. Such a penalty prefers a "small number" of parents (second term) that are "not too far" from the child (first term). Although the "small number of parents"

objective is obvious, the "short distance between parents and children" may be discussed. But, at least, among many equivalent or nearly equivalent models it helps to choose those that are better arranged. An unlimited number of modifications of the penalty term can be figured; for example, instead of $\gamma \sum_{k=1}^{K} |A(k)|$ we could insert $\sum_{k=1}^{K} \sum_{j=1}^{|A(k)|} \gamma_j$ with some growing sequence γ_j and thus penalize a large number of parents for each individual child. In principle, let us emphasize once again that the penalty should reflect the prior information in each individual problem, and no universal penalty can be proposed. In the case of missing information, we have no other chance but to deal with various hypotheses, each of them expressed by a particular penalty term.

3.2. Bayesian Context

We may also interpret our approach within the standard Bayesian framework. Then, by the Maximum Aposterior Probability (MAP) principle, we obtain a good solution to the problem by maximizing the joint distribution

$$\Pr(x_G^{(1)}, \ldots, x_G^{(N)}, M, \mathcal{Q}) = \prod_{i=1}^{N} P^{M,\mathcal{Q}}(x_G^{(i)}) \Pr(M, \mathcal{Q}) \tag{13}$$

over the set $\mathcal{P}$ of all probabilistic models. Because by Equation 5 we have

$$\prod_{i=1}^{N} P^{M,\mathcal{Q}}(x_G^{(i)}) = \prod_{y_G \in X_G} P^{M,\mathcal{Q}}(y_G)^{N\hat{P}^{(N)}(y_G)} \tag{14}$$

we obtain directly

$$\frac{1}{N} \log \prod_{i=1}^{N} P^{M,\mathcal{Q}}(x_G^{(i)}) = -I(\hat{P}^{(N)} | P^{M,\mathcal{Q}}) - H(\hat{P}^{(N)}) \tag{15}$$

where $H(\hat{P}^{(N)})$ is the entropy of empirical distribution and may be omitted as a constant not depending on the unknown model.

For the prior distribution $\Pr(M, \mathcal{Q})$ we suppose it depends only on the graphical model M, that is,

$$\Pr(M, \mathcal{Q}) = \Pr(M) \tag{16}$$

and, moreover, it will assume a special product form, namely,

$$\Pr(M) = \frac{1}{K!} \prod_{k \in \{1, \ldots, K\}} \Pr(A(k)) \tag{17}$$

with

$$\Pr(A(k)) = \prod_{j \in A(k)} \Pr(j \in A(k)) \prod_{j \in R(k)^* \setminus A(k)} [1 - \Pr(j \in A(k))] \tag{18}$$

And, if we choose logistic formulas for the "parental" probabilities, namely, for $k, j \in \{1, \ldots, K\}$ where $R(j) < R(k)$ we set

$$Pr(j \in A(k)) = \frac{1}{1 + e^{\beta(R(k) - R(j)) + \gamma}} \tag{19}$$

then we obtain precisely

$$\frac{1}{N} \log \Pr(M) = -T_N(M) + const. \quad \text{with } c_N = 1/N \tag{20}$$

Thus, obviously, maximizing the posterior probability is, in this case, equivalent to minimizing our objective function. Obviously, another prior distribution would yield another penalty term.

4. METHOD

For the numerical solution of the optimization problem, we apply *simulated annealing*, which is an MCMC technique (see Ref. 12) used to maximize a given function on a discrete state space with a large number of states.

4.1. Simulated Annealing

Let us briefly recall the basic general idea. Suppose we have to solve the problem

$$\max_{s \in S} f(s) \tag{21}$$

where f is an objective function and S with $|S| < +\infty$ is the state space. If S is really large, then any "brute force" solution is numerically infeasible.

If we set

$$Q^{\tau}(s) = \frac{e^{\tau f(s)}}{\sum_{t \in S} e^{\tau f(t)}} \tag{22}$$

then, obviously, this expression converges for $\tau \to \infty$ to Q^{∞}, where

$$Q^{\infty}(s) = \begin{cases} |\mu|^{-1} & \text{for } s \in \mu = \arg\max f(s) \subset S \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

Therefore, maximizing the function f is equivalent to sampling from the distribution Q^{∞}, which we can approximate by Q^{τ} for sufficiently large $\tau \gg 0$.

We have $Q^\tau > 0$. Then, according to Markov Chains Theory, if R^τ is an irreducible stochastic matrix satisfying $Q^\tau R^\tau = Q^\tau$, then

$$[R^\tau]^n_{s_0\bullet} \to Q^\tau(\bullet) \quad \text{for } n \to +\infty \tag{24}$$

for any initial state $s_0 \in S$. The simulated annealing procedure is based on merging both the convergences, and, if the condition

$$\tau_i < C \log n \tag{25}$$

with an appropriate constant is satisfied, then for any $s_0 \in S$ (see, e.g., Theorem 5.2.1 in Ref. 9)

$$[R^{\tau_1} R^{\tau_2} \ldots R^{\tau_n}]_{s_0\bullet} \to Q^\infty(\bullet) \quad \text{for } n \to +\infty \tag{26}$$

4.2. Algorithm

For the matrix R^τ we use Metropolis–Hastings construction:

$$R^\tau_{st} = \tilde{R}_{st} \min\left(1, \frac{Q^\tau(t)\tilde{R}_{ts}}{Q^\tau(s)\tilde{R}_{st}}\right) \quad \text{for } s \neq t \tag{27}$$

$$R^\tau_{ss} = 1 - \sum_{t \neq s} R^\tau_{st} \tag{28}$$

where $\tilde{R}$ is an arbitrary irreducible stochastic matrix (preferably a sparse one) satisfying

$$\tilde{R}_{st} = 0 \quad \text{if and only if } \tilde{R}_{ts} = 0 \tag{29}$$

The latter condition is obviously satisfied for symmetric matrices.

Now, we may introduce an algorithm that implements the principle described above. Let us fix a *cooling scheme* $(\tau_n)_{n=1\ldots}$, a *stopping time* $\bar{n}$, and a *proposal matrix* $\tilde{R}$.

(1) We start from an arbitrary initial configuration $s_0 \in S$.
(2) At the nth step we sample s_n from $R^{\tau_n}_{s_{n-1}\bullet}$.
(3) For $n = \bar{n}$ we stop and return $s_{\bar{n}}$.

The particular sampling from $R^{\tau_n}_{s_{n-1}\bullet}$ can be performed in two substeps. In the first one, which is known as the "proposal" phase, we sample ("propose") a new state t with the distribution $\tilde{R}^{\tau_n}_{s_{n-1}\bullet}$. In the second phase, denoted as "acceptance," we compute

$$\Delta_n = \min\left(1, \frac{Q^{\tau_n}(t)\tilde{R}_{ts_{n-1}}}{Q^{\tau_n}(s_{n-1})\tilde{R}_{s_{n-1}t}}\right) \tag{30}$$

and set ("accept") $s_n = t$ with the probability Δ_n, or keep the current state, that is, $s_n = s_{n-1}$, with the probability $1 - \Delta_n$.

The crucial advantage of the method consists of its numerical feasibility. For a detailed treatment see, for example, Ref. 9.

4.3. Implementation for the Bayesian Networks Learning

Within our optimization problem, as described in Section 2, we have

$$S = \mathcal{M} = \{(R(k), A(k))_{k\in\{1,\ldots,K\}}\} \tag{31}$$

$$f(M) = \tilde{F}(M) = D(\hat{P}^{(N)}|\hat{P}^{M}) + T_N(M) \tag{32}$$

Now, let us discuss particularities of the above algorithm for this specific optimization problem.

4.3.1. Cooling Scheme

Condition 25 implies very slow convergence and requires an extremely large stopping time. This inconvenience can be avoided (in practice) by choosing a faster cooling scheme; in our case, we deal with

$$\tau_i = (1 + \varepsilon)^i \text{ where } \varepsilon \text{ is a positive number close to 0} \tag{33}$$

As a standard choice we set $\varepsilon = 10^{-4}$ or $\varepsilon = 10^{-5}$. In principle, τ_i should be very small, nearly equal to 0, at the beginning, and rather large, at least, let us say, $\tau_i > 10^2$, at the end of the procedure.

4.3.2. Stopping Time

In principle, the stopping time $\bar{n}$ should be as large as possible. First of all, let us realize that the algorithm is based on a *random visiting scheme*, and we should "give a chance to the randomness." Moreover, as indicated in the preceding subsection, we have to guarantee that $\tau_{\bar{n}}$ is large enough. But, because we apply the fast cooling scheme described above, we have τ_n pretty large already for "small" n. Then, as a matter of fact, the algorithm terminates at some local optimum wherefrom it cannot escape because the random acceptance of a "worse" state is nearly forbidden ($\Delta_n \doteq 0$). Therefore, increasing the stopping time further makes no sense, and we may stop the procedure whenever, for some time, there are no changes in its output. For our purposes, it was sufficient to set $\bar{n} = 10^6$ or $\bar{n} = 2\cdot 10^6$ in dependence on the system size.

4.3.3. Proposal Matrix

The proposal matrix $\tilde{R}$ must be irreducible, with symmetrically placed zeros (preferably completely symmetric), but, for computational reasons, it should be as simple as possible, which standardly means to be sparse. We define

$$\tilde{R}_{M\overline{M}} = \begin{cases} \dfrac{1}{2(K-1)} & \text{if there exist } k,j \text{ with } R(j) = R(k)+1 \\ & R(u) = \bar{R}(u), A(u) = \bar{A}(u) \quad \text{for } u \neq k,j \\ & R(k) = \bar{R}(j), A(j)\backslash\{k\} = \bar{A}(j) \\ & R(j) = \bar{R}(k), A(k) = \bar{A}(k)\backslash\{j\} \\ 0 & \text{otherwise} \end{cases} \tag{34}$$

The essence of the "proposal" matrix $\tilde{R}$ is the following. We choose at random a pair of successive vertices (k,j) and exchange them in their ordering. Then, after the exchange, k cannot be a parent of j whereas j becomes a parent of k with probability $\frac{1}{2}$. Thus, in one step the model can be modified only locally, into some of $2\cdot(K-1)$ neighboring models (with equal probability $1/(2\cdot(K-1))$).

PROPOSITION 1. *The above defined stochastic matrix $\tilde{R}_{M\overline{M}}$ is irreducible and symmetric.*

Proof. The symmetry is obvious. For irreducibility, we have to show that for each pair of models $M, \overline{M}$ the transition $M \to \overline{M}$ can be performed with positive probability at some finite time. The idea of the proof consists of two subprocedures:

(1) killing old parents,
(2) choosing new parents when passing them.

Let $\bar{R}(j_0) = K$ for some $j_0 \in \{1,\ldots,K\}$. Then, by subsequent interchanging the model M can be modified (with positive probability in $R(j_0)-1$ steps) to M^0 with $R^0(j_0) = 1$ and $A^0(j_0) = \emptyset$ (killing old parents). Then, again with positive probability in $K-1$ steps, M^0 can be modified to M^1 with $R^1(j_0) = K$ and $A^1(j_0) = \bar{A}(j_0)$ (choosing new parents). Just j_0 has its proper new ordering and parents. Then the whole procedure must be repeated for j_1 with $\bar{R}(j_1) = K-1$ without affecting $(\bar{R}(j_0), \bar{A}(j_0))$, and so forth. ■

4.3.4. Penalty Term

Obviously (see the examples below), the method is sensitive to the constants in the penalty term, and, naturally, the sensitiveness is stronger for larger systems (larger K). In general, the penalty can be reasonably efficient, if its order of magnitude for the estimated model is approximately the same as that of the "distance" term. This can be understood as a recommendation in the case of missing prior information. But the constants cannot be set in advance; they must be adjusted by subsequent performances of the algorithm. We can start with small values (i.e., at the beginning of the analysis we more believe the data) and then to enlarge them gradually, until some kind of balance is reached.

4.3.5. Calculating Entropies

The critical point in the performance of the method is the necessity of calculating the entropies of various empirical marginals $\{H(\hat{P}_A^{(N)})\}_{A\subset\{1,\ldots,K\}}$, where $\hat{P}_A^{(N)}$ for $A \subset \{1,\ldots,K\}$ is the restriction of $\hat{P}^{(N)}$ to the space X_A. These entropies are involved in the Kullback–Leibler distance term $I(\hat{P}^{(N)}|\hat{P}^M)$. For small systems ($K \leq 10$), it is worthwhile to calculate all the marginal entropies in advance and to store them for further use. For larger systems, it is not possible (or, at least, it would be too time-consuming), and therefore the actual collection of entropies must be calculated at each step (in fact, the already calculated entropies could be stored consequently during the performance). Because $H(\hat{P}_A^{(N)}) = \log N - (1/N)\sum_{j=1}^{N} \log \#(x_A^{(j)})$, where $\#(x_A^{(j)})$ is the number of occurrences of $x_A^{(j)}$ in the data sequence $x_A^{(1)},\ldots,x_A^{(N)} \in X_A$, has a polynomial (in the data size N) computational complexity, the procedure is still feasible, but, consequently, the performance is slower.

4.3.6. System Size

For the sake of simplicity, the illustrative examples below deal with models of the size $K = 8$ or $K = 12$. But, with the modification in calculating entropies, as described in the preceding subsection, the tasks with $K = 25$ or $K = 30$ can be performed on a standard PC in a reasonable time, that is, in the range of minutes.

5. NUMERICAL EXPERIMENTS

We tested the algorithm with simulated data, sampled from given Bayesian networks. For simplicity we dealt with binary state spaces, that is, $X_k = \{0,1\}$ for every $k = \{1,\ldots,K\}$. Then the collection of probability kernels $\{Q_g(.|.)\}_{g\in G}$ is completely given by the values

$$Q^M_{k|A(k)}(1|x_{A(k)}) \in (0,1) \quad \text{for all } k = \{1,\ldots,K\} \text{ and } x_{A(k)} \in X_{A(k)} \tag{35}$$

For every particular example, the graphical model M, as well as the conditional probabilities $Q^M_{k|A(k)}(1|x_{A(k)})$, were chosen also at random with distributions corresponding to the priors as described in Equations 16–19.

Because the original Bayesian network enters the formulas only through the empirical distribution $\hat{P}^{(N)}$, we dealt with a rather large sample size N, typically $N = 10^6$. Then the empirical distribution was close enough to the original Bayesian network, and the comparison between the original and the estimated BNs made good sense. As far as the parameters of the simulated annealing are concerned, the cooling scheme was $\tau_n = (1+\varepsilon)^n$ with $\varepsilon = 10^{-5}$ and stopping time $\bar{n} = 10^6$ (see the preceding section).

Thus, the number of vertices K and the constants in the penalty term T_N are the only remaining parameters to be specified in the examples below.

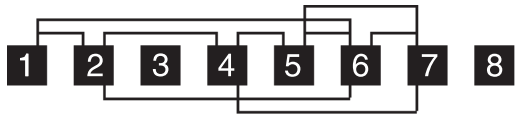

Figure 2. Original (input) model.

Example 1. Let us fix $K = 8$, and for the penalty term we set

$$c_N \beta \approx 10^{-4}, \qquad c_N \gamma \approx 10^{-4}$$

The original (input) graphical model M_0 under consideration in this example is displayed in Figure 2. The value of the objective function for this model is

$$\tilde{F}(M_0) = 0.000178 + 0.0056 = 0.005778$$

Let us recall that the first term corresponds to the distance whereas the second one is the penalty.

By applying the above described optimization procedure, we obtain an estimated model $\hat{M}$ (see Figure 3). The value of the objective function for the estimated model is again

$$\tilde{F}(\hat{M}) = 0.000178 + 0.0056 = 0.005778$$

The equal values of the objective function indicate that the models might be equivalent. The distribution of the input model is

$$P^{M_0} = \prod_{k=1}^{K} P_{k|A(k)} = P_1 P_{2|1} P_3 P_{4|2} P_{5|4} P_{6|\{5,2,1\}} P_{7|\{6,5,4\}} P_8 \tag{36}$$

and the distribution of the estimated model is

$$P^{\hat{M}} = P_5 P_{4|5} P_{2|4} P_{1|2} P_{6|\{1,2,5\}} P_{7|\{6,5,4\}} P_3 P_8 \tag{37}$$

Thanks to the large enough sample size N, we have effectively $\hat{P}_{k|A(k)}(x_k|x_{A(k)}) = P^{M_0,\mathcal{Q}}_{k|A(k)}(x_k|x_{A(k)})$, and therefore the equivalence problem may be considerably simplified. Namely, if two Bayesian networks are obtained by restrictions from the same global distribution, here $\hat{P}_G$, then, obviously, the equivalence

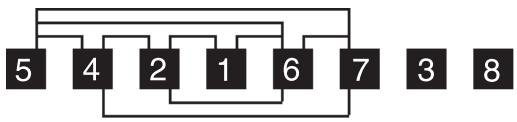

Figure 3. Estimated model.

of their respective graphical models is sufficient for the equivalence of the probabilistic models. And graphical models $M, \overline{M} \in \mathcal{M}$ are equivalent if

$$\mathcal{U}_M \backslash \mathcal{L}_M = \mathcal{U}_{\overline{M}} \backslash \mathcal{L}_{\overline{M}}$$

and

$$\mathcal{L}_M \backslash \mathcal{U}_M = \mathcal{L}_{\overline{M}} \backslash \mathcal{U}_{\overline{M}}$$

where

$$\mathcal{U}_M = \{\{k\} \cup A(k)\}_{k \in \{1,\ldots,K\}} \quad \text{and} \quad \mathcal{L}_M = \{A(k)\}_{k \in \{1,\ldots,K\}}$$

Thus, by comparing the original and estimated models (i.e., M_0 and $\hat{M}$), we obtain that they are, in our sense, equivalent.

Example 2. In this example we use the same input model and data, but we put more stress on the prior information, namely, we set

$$c_N \beta \approx 10^{-2}, \qquad c_N \gamma \approx 10^{-2}$$

Thus we have the same graphical model M_0 as in Figure 2, but the objective function for this case is (notice the penalty term)

$$\tilde{F}(M_0) = 0.000178 + 0.87 = 0.870178$$

whereas the objective function for the estimated model $\hat{M}$ (see Figure 4) is now

$$\tilde{F}(\hat{M}) = 0.225411 + 0.18 = 0.405411$$

Further, for the input model, we have again the distribution

$$P^{M_0} = P_1 P_{2|1} P_3 P_{4|2} P_{5|4} P_{6|\{5,2,1\}} P_{7|\{6,5,4\}} P_8 \tag{38}$$

but the distribution of the estimated model is

$$P^{\hat{M}} = P_3 P_8 P_7 P_{6|7} P_{1|6} P_5 P_{4|5} P_2 \tag{39}$$

Apparently, the two distributions differ substantially and therefore cannot be equivalent. Nevertheless, the objective function is smaller for the estimated model (so the estimated model should be "better"), due to overestimated values of the constants in the penalty term.

This example illustrates the importance of proper choice of the constants in the penalty term. Precise values are not necessary, but the order of magnitude should be adjusted correctly.

Figure 4. Estimated model.

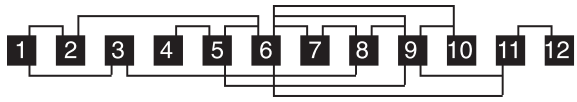

Figure 5. Original model.

Example 3. In this example, another input model with $K = 12$ vertices is presented. The penalty term is constructed in the same way as in the first example, namely, with

$$c_N \beta \approx 10^{-4}, \qquad c_N \gamma \approx 10^{-4}$$

The objective function for this input model (as displayed in Figure 5) attains the value

$$\tilde{F}(M) = 0.002263 + 0.00201 = 0.004273$$

whereas for the estimated model (Figure 6), it is

$$\tilde{F}(\hat{M}) = 0.002259 + 0.00212 = 0.004379$$

Now the distribution of the input model is

$$P^{M_0} = P_1 P_{2|1} P_{3|1} P_4 P_{5|4} P_{6|\{5,2\}} P_{7|6} P_{8|\{3,7\}} P_{9|\{5,6,8\}} P_{10|\{6,9\}} P_{11|\{6,9\}} P_{12|11} \tag{40}$$

and the distribution of the estimated model is

$$P^{\hat{M}} = P_2 P_{1|2} P_{3|1} P_4 P_{5|4} P_{6|\{5,2\}} P_{7|6} P_{8|\{3,7\}} P_{9|\{5,6,7,8\}} P_{10|\{6,9\}} P_{11|\{6,9\}} P_{12|11} \tag{41}$$

By comparing the respective terms, we observe that the models are not equivalent in the strict sense. The estimated model is better fitted to the actual data (notice the first terms of the objective function) but contains one more edge that makes the penalty term larger. But, in fact, there is only the conditional distribution $P_{9|\{5,6,8\}}$ substituted by $P_{9|\{5,6,7,8\}}$ in the estimated distribution.

Therefore we may deduce that these models are "nearly equivalent."

6. CONCLUDING REMARK

In principle, the approach proves its relevance for solving the problem of learning Bayesian networks. Due to the stochastic nature of the optimization procedure, the main advantage of the method is given by the possibility of handling

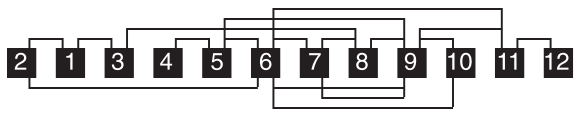

Figure 6. Estimated model.

graphs with a rather large number of vertices (much larger than in the illustrative examples above). On the other hand, from the rigorously statistical point of view the problem is overparametrized, with an extremely large number of hardly distinguishable models; therefore the method is sensitive to the quality of the prior information, and there exists a danger of misuse.

Acknowledgment

This work was supported by GAČR Grant No. 201/03/0478.

References

1. Buntine WL. A guide to the literature on learning probabilistic networks from data. IEEE Trans Knowl Data Eng 1996;8:195–210.
2. Gelman A, Carlin JB, Stern HS, Rubin DS. Bayesian data analysis. London: Chapman and Hall; 1995.
3. Heckerman D, Geiger D, Chickering DM. Learning Bayesian networks: The combination of knowledge and statistical data. Mach Learn 1995;20:197–243.
4. Jordan MI, editor. Learning in graphical models. Dordrecht: Kluwer; 1998.
5. Chickering DM. Optimal structure identification with greedy search. J Mach Learn Res 2002;3:507–554.
6. Chickering DM, Meek C. Finding optimal Bayesian networks. In: Proc 18th Conf on Uncertainty in Artificial Intelligence, August 1–4, 2002, Univerity of Alberta, Edmonton, Alberta, Canada; 2002. pp 94–102.
7. Geman D, Geman S. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans Patt Anal Mach Intell 1984;6:721–741.
8. Lakshmanan S, Derin H. Simultaneous parameter estimation and segmentation of Gibbs random fields using simulated annealing. IEEE Trans Patt Anal Mach Intell 1989;11: 799–813.
9. Winkler G. Image analysis, random fields and dynamic Monte Carlo methods. Berlin: Springer-Verlag; 1995.
10. Jensen FV. Introduction to Bayesian networks. London: UCL Press; 1996.
11. Lauritzen SL. Graphical models. Oxford: Oxford University Press; 1996.
12. Gilks WR, Richardson S, Spiegelhalter DJ, editors. Markov chain Monte Carlo in practice. London: Chapman and Hall; 1996.