

Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>



Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

Learning Bayesian networks for discrete data

Faming Liang^{a,*}, Jian Zhang^b^a Department of Statistics, Texas A&M University, College Station, TX 77843-3143, USA^b Department of Mathematics, University of York, York, YO10 5DD, UK

ARTICLE INFO

Article history:

Received 6 July 2008

Received in revised form 5 October 2008

Accepted 7 October 2008

Available online 17 October 2008

ABSTRACT

Bayesian networks have received much attention in the recent literature. In this article, we propose an approach to learn Bayesian networks using the stochastic approximation Monte Carlo (SAMC) algorithm. Our approach has two nice features. Firstly, it possesses the self-adjusting mechanism and thus avoids essentially the local-trap problem suffered by conventional MCMC simulation-based approaches in learning Bayesian networks. Secondly, it falls into the class of dynamic importance sampling algorithms; the network features can be inferred by dynamically weighted averaging the samples generated in the learning process, and the resulting estimates can have much lower variation than the single model-based estimates. The numerical results indicate that our approach can mix much faster over the space of Bayesian networks than the conventional MCMC simulation-based approaches.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

The use of graphs to represent statistical models has been one focus of research in recent years. In particular, researchers have directed interest in Bayesian networks and applications of such models to biological data, see e.g., [Friedman et al. \(2000\)](#) and [Ellis and Wong \(2008\)](#). The Bayesian network, as illustrated by [Fig. 1](#), is a directed acyclic graph (DAG) in which the nodes represent the variables in the domain and the edges correspond to direct probabilistic dependencies between them. As indicated by many applications, the Bayesian network is a powerful knowledge representation and reasoning tool under conditions of uncertainty that is typical of real-life applications.

Many approaches have been developed for learning of Bayesian networks in the literature. These approaches can be roughly grouped into three categories: the conditional independence test-based approaches, the optimization-based approaches, and the MCMC simulation-based approaches.

The approaches in the first category perform a qualitative study of dependence relationships between the nodes, and generate a network that represents most of the relationships. The approaches described in [Spirtes et al. \(1993\)](#), [Wermuth and Lauritzen \(1983\)](#) and [de Campos and Huete \(2000\)](#) belong to this category. The networks constructed by these approaches are usually asymptotically correct, but as pointed out by [Cooper and Herskovits \(1992\)](#) that the conditional independence tests with large condition-sets may be unreliable unless the volume of data is enormous. We note that due to limited research resources, the sample size of the biological data is often small, e.g., the gene expression data studied in [Friedman et al. \(2000\)](#) and the real examples studied in this paper.

The approaches in the second category attempt to find a network that optimizes a selected scoring function, which evaluates the fitness of each feasible network to the data. The scoring functions can be formulated based on different principles, such as entropy ([Herskovits and Cooper, 1990](#)), the minimum description length ([Lam and Bacchus, 1994](#)), and

* Corresponding author. Tel.: +1 979 845 8885; fax: +1 979 845 3144.

E-mail addresses: fliang@stat.tamu.edu (F. Liang), jz538@york.ac.uk (J. Zhang).

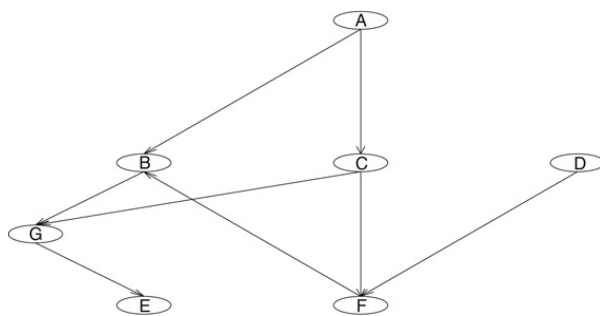


Fig. 1. An example of Bayesian networks.

Bayesian scores (Cooper and Herskovits, 1992; Heckerman et al., 1995). The optimization procedures employed are usually heuristic, such as tabu search (Bouckaert, 1995) and evolutionary computation (de Campos and Huete, 2000; Neil and Korb, 1999). Unfortunately, the task of finding a network structure that optimizes the scoring function is known to be a NP-hard problem (Chickering, 1996). Hence, the optimization process often stops at a local optimal structure.

The approaches in the third category work by simulating a Markov chain over the space of feasible network structures with the stationary distribution being the posterior distribution of the network. The work belonging to this category include Madigan and Raftery (1994), Madigan and York (1995), and Giudici and Green (1999), among others. In these works, the simulation is done using the Metropolis–Hastings (MH) algorithm, and the network features are inferred by averaging over a large number of networks simulated from the posterior distribution. Averaging over different networks can significantly reduce the variation of estimation suffered by the single network-based inference procedure. Although the approaches seem attractive, they can only work well for the problems with a very small number of variables. This is because the energy landscape of the Bayesian network can be quite rugged, with a multitude of local energy minima being separated by high energy barriers, especially when the network size is large. Here, the energy function refers to the negative log-posterior distribution function of the Bayesian network. As known by many researchers, the MH algorithm is prone to get trapped in a local energy minimum indefinitely in simulations from a system for which the energy landscape is rugged. To alleviate this difficulty, Friedman and Koller (2003) introduce a two-stage algorithm: use the MH algorithm to sample a temporal order of the nodes, and then sample a network structure compatible with the given node order. As discussed in Friedman and Koller (2003), for any Bayesian networks, there exists a temporal order of the nodes such that for any two nodes X and Y , if there is an edge from X and Y , then X must be preceding to Y in the order. For example, for the network shown in Fig. 1, a temporal order compatible with the network is ACDFBGE. The two-stage algorithm improves the mixing over the space of network structures, however, the structures sampled by it does not follow the correct posterior distribution, because the temporal order does not induce a partition of the space of network structures. A network may be compatible with more than one order. For example, the network shown in Fig. 1 is compatible with both the orders ACDFBGE and ADCFBGE.

In this article, we propose to learn Bayesian networks using the stochastic approximation Monte Carlo (SAMC) algorithm (Liang et al., 2007). A remarkable feature of the SAMC algorithm is that it possesses the self-adjusting mechanism and is thus less likely trapped by local energy minima. This is very important for learning of Bayesian networks. In addition, SAMC belongs to the class of dynamic weighting algorithms (Wong and Liang, 1997; Liu et al., 2001; Liang, 2002), and the samples generated in the learning process can be used to infer the network features via a dynamically weighted estimator. Like Bayesian model averaging estimators, the dynamically weighted estimator can have much lower variation than the single model-based estimator.

The remainder of this article is organized as follows. In Section 2, we give the formulation of Bayesian networks. In Section 3, we first give a brief review of the SAMC algorithm and then describe its implementation for Bayesian networks. In Section 4, we present the numerical results on a simulated example and two real biological data example. In Section 5, we conclude the paper with a brief discussion.

2. Bayesian networks

A Bayesian network model can be defined as a pair $B = (\mathcal{G}, \rho)$, where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a directed acyclic graph that represents the structure of the network, \mathcal{V} denotes the set of nodes, \mathcal{E} denotes the set of edges, and ρ is a vector of conditional probabilities as described below. For a node $V \in \mathcal{V}$, a parent of V is a node from which there is a directed link to V . The set of parents of V is denoted by $pa(V)$. In this article, we study only the discrete case where V is a categorical variable taking values in a finite set $\{v_1, \dots, v_{r_i}\}$. There are $q_i = \prod_{V_j \in pa(V_i)} r_j$ possible values for the joint state of the parents of V_i . Each element of ρ represents a conditional probability. For example, ρ_{ijk} is the probability of variable V_i in state j conditioned on that $pa(V_i)$ is in state k . Naturally, ρ is restricted by the constraints $\rho_{ijk} \geq 0$ and $\sum_{j=1}^{r_i} \rho_{ijk} = 1$. The joint distribution of the variables $\mathbf{V} = \{V_1, \dots, V_d\}$ can be specified by the decomposition

$$P(\mathbf{V}) = \prod_{i=1}^d P(V_i | pa(V_i)). \tag{1}$$

Let $\mathcal{D} = \{\mathbf{V}_1, \dots, \mathbf{V}_N\}$ denote a set of independently and identically distributed samples drawn from the distribution (1). Let n_{ijk} denote the number of samples for which V_i is in state j and $pa(V_i)$ is in state k . Then, the counts $(n_{i1k}, \dots, n_{ir_ik})$ follows a multinomial distribution; that is,

$$(n_{i1k}, \dots, n_{ir_ik}) \sim \text{Multinomial} \left(\sum_{j=1}^{r_i} n_{ijk}, \boldsymbol{\rho}_{ik} \right), \tag{2}$$

where $\boldsymbol{\rho}_{ik} = (\rho_{i1k}, \dots, \rho_{ir_ik})$. Thus, the likelihood function of the Bayesian network model can be written as

$$P(\mathcal{D}|\mathcal{G}, \boldsymbol{\rho}) = \prod_{i=1}^d \prod_{k=1}^{q_i} \binom{\sum_{j=1}^{r_i} n_{ijk}}{n_{i1k}, \dots, n_{ir_ik}} \rho_{i1k}^{n_{i1k}} \dots \rho_{ir_ik}^{n_{ir_ik}}. \tag{3}$$

To carry out a Bayesian analysis for the model, we have the following prior specification for the network structure and parameters. Since a network with a large number of edges is often less interpretable and there is a risk of over-fitting, it is important to use priors over the network space that encourage sparsity. For this reason, we let \mathcal{G} be subject to the prior

$$P(\mathcal{G}|\beta) \propto \left(\frac{\beta}{1-\beta} \right)^{\sum_{i=1}^d |pa(V_i)|}, \tag{4}$$

where $0 < \beta < 1$ is a user-specified parameter. In this article, we set $\beta = 0.1$ for all examples. The parameters $\boldsymbol{\rho}$ is subject to a product Dirichlet distribution

$$P(\boldsymbol{\rho}|\mathcal{G}) = \prod_{i=1}^d \prod_{k=1}^{q_i} \frac{\Gamma(\sum_{j=1}^{r_i} \alpha_{ijk})}{\Gamma(\alpha_{i1k}) \dots \Gamma(\alpha_{ir_ik})} \rho_{i1k}^{\alpha_{i1k}-1} \dots \rho_{ir_ik}^{\alpha_{ir_ik}-1}, \tag{5}$$

where $\alpha_{ijk} = 1/(r_i q_i)$ as suggested by Ellis and Wong (2008). Combining with the likelihood function and the prior distributions and integrating out $\boldsymbol{\rho}$, we get the posterior distribution (up to a multiplicative constant):

$$P(\mathcal{G}|\mathcal{D}) \propto \prod_{i=1}^d \left(\frac{\beta}{1-\beta} \right)^{|pa(V_i)|} \prod_{k=1}^{q_i} \frac{\Gamma(\sum_{j=1}^{r_i} \alpha_{ijk})}{\Gamma(\sum_{j=1}^{r_i} (\alpha_{ijk} + n_{ijk}))} \prod_{j=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + n_{ijk})}{\Gamma(\alpha_{ijk})}, \tag{6}$$

which contains all the network structure information provided by the data.

We note that the Bayesian network is conceptually different from the causal Bayesian network. In the causal Bayesian network, each edge can be interpreted as a direct causal relation between a parent node and a child node, relative to the other nodes in the network (Pearl, 1998). The formulation of Bayesian networks, as described above, is not sufficient for causal inference. To learn a causal Bayesian network, one needs a dataset obtained through experimental interventions. In general, one cannot learn a causal Bayesian network from the observational data alone. Refer to Cooper and Yoo (1999) and Ellis and Wong (2008) for more discussions on this issue.

3. Learning Bayesian networks using SAMC

3.1. A review of the SAMC algorithm

Suppose that we are working with the following Boltzmann distribution,

$$f(x) = \frac{1}{Z} \exp\{-U(x)/\tau\}, \quad x \in \mathcal{X}, \tag{7}$$

where Z is the normalizing constant, τ is the temperature, \mathcal{X} is the sample space, and $U(x)$ is called the energy function in terms of physics. In the context of Bayesian networks, $U(x)$ corresponds to $-\log P(\mathcal{G}|\mathcal{D})$, the negative logarithm of the posterior distribution (6), and the sample space \mathcal{X} is finite. Furthermore, we suppose that the sample space has been partitioned according to the energy function into m disjoint subregions denoted by $E_1 = \{x : U(x) \leq u_1\}$, $E_2 = \{x : u_1 < U(x) \leq u_2\}$, \dots , $E_{m-1} = \{x : u_{m-2} < U(x) \leq u_{m-1}\}$, and $E_m = \{x : U(x) > u_{m-1}\}$, where u_1, \dots, u_{m-1} are real numbers specified by the user. Let $\psi(x)$ be a non-negative function defined on the sample space with $0 < \int_{\mathcal{X}} \psi(x) dx < \infty$, and $\theta_i = \log(\int_{E_i} \psi(x) dx)$. In practice, we often set $\psi(x) = \exp\{-U(x)/\tau\}$.

SAMC seeks to draw samples from each of the subregions with a pre-specified frequency. If this goal can be achieved, then the local-trap problem can be avoided essentially as explained in Liang et al. (2007). Let $x^{(t+1)}$ denote a sample drawn

from a MH kernel $K_{\theta^{(t)}}(x^{(t)}, \cdot)$ with the proposal distribution $q(x^{(t)}, \cdot)$ and the stationary distribution

$$f_{\theta^{(t)}}(x) \propto \sum_{i=1}^{m-1} \frac{\psi(x)}{e^{\theta_i^{(t)}}} I(x \in E_i) + \psi(x) I(x \in E_m), \tag{8}$$

where $\theta^{(t)} = (\theta_1^{(t)}, \dots, \theta_{m-1}^{(t)})$ is an $(m - 1)$ -vector in a space Θ . For convenience, we set $\theta_m^{(t)} = 0$. Here, without loss of generality, we assume that E_m is non-empty; that is, $\int_{E_m} \psi(x) dx > 0$. In practice, E_m can be replaced by any subregion which is known to be non-empty.

Let $\pi = (\pi_1, \dots, \pi_m)$ be an m -vector with $0 < \pi_i < 1$ and $\sum_{i=1}^m \pi_i = 1$, which defines a desired sampling frequency for the subregions. Henceforth, π will be called the desired sampling distribution. Define $H(\theta^{(t)}, x^{(t+1)}) = (e^{(\theta^{(t+1)} - \pi)})$, where $e^{(\theta^{(t+1)})} = (e_1^{(\theta^{(t+1)})}, \dots, e_m^{(\theta^{(t+1)})})$ and $e_i^{(\theta^{(t+1)})} = 1$ if $x^{(t+1)} \in E_i$ and 0 otherwise. Let $\{\gamma_t\}$ be a positive, non-decreasing sequence satisfying the conditions,

$$(i) \sum_{t=0}^{\infty} \gamma_t = \infty, \quad (ii) \sum_{t=0}^{\infty} \gamma_t^\delta < \infty, \tag{9}$$

for some $\delta \in (1, 2)$. In the context of stochastic approximation (Robbins and Monro, 1951), $\{\gamma_t\}_{t \geq 0}$ is called the gain factor sequence. In this article, we set

$$\gamma_t = \frac{t_0}{\max(t_0, t)}, \quad t = 0, 1, 2, \dots \tag{10}$$

for a pre-specified values of $t_0 > 1$. A large value of t_0 will allow the sampler to reach all subregions very quickly even for a large system. Let $J(x)$ denote the index of the subregion that the sample x belongs to. Let $\{\mathcal{K}_s, s \geq 0\}$ be a sequence of compact subsets of Θ such that

$$\bigcup_{s \geq 0} \mathcal{K}_s = \Theta, \quad \text{and} \quad \mathcal{K}_s \subset \text{int}(\mathcal{K}_{s+1}), \quad s \geq 0, \tag{11}$$

where $\text{int}(A)$ denotes the interior of set A . Let \mathcal{X}_0 be a subset of \mathcal{X} , and let $\mathcal{T} : \mathcal{X} \times \Theta \rightarrow \mathcal{X}_0 \times \mathcal{K}_0$ be a measurable function which maps a point in $\mathcal{X} \times \Theta$ to a random point in $\mathcal{X}_0 \times \mathcal{K}_0$. Let σ_k denote the number of truncations performed until iteration k . Let \mathcal{J} denote the collection of the indices of the subregions from which a sample has been proposed; that is, \mathcal{J} contains the indices of all subregions which are known to be non-empty. With above notations, one iteration of SAMC can be described as follows.

The SAMC algorithm

(a) (Sampling) Simulate a sample $x^{(t+1)}$ by a single MH update with the target distribution as defined in (8).

(a.1) Generate y according to a proposal distribution $q(x_t, y)$. If $J(y) \notin \mathcal{J}$, set $\mathcal{J} \leftarrow \mathcal{J} \cup \{J(y)\}$.

(a.2) Calculate the ratio

$$r = e^{\theta_{J(x^{(t)})}^{(t)} - \theta_{J(y)}^{(t)}} \frac{\psi(y)q(y, x^{(t)})}{\psi(x^{(t)})q(x^{(t)}, y)}.$$

(a.3) Accept the proposal with probability $\min(1, r)$. If it is accepted, set $x^{(t+1)} = y$; otherwise, set $x^{(t+1)} = x^{(t)}$.

(b) (Weight updating) For all $i \in \mathcal{J}$, set

$$\theta_i^{(t+\frac{1}{2})} = \theta_i^{(t)} + a_{t+1} (I_{\{x^{(t+1)} \in E_i\}} - \pi_i) - a_{t+1} (I_{\{x^{(t+1)} \in E_m\}} - \pi_m). \tag{12}$$

(c) (Varying truncation) If $\theta^{(t+\frac{1}{2})} \in \mathcal{K}_{\sigma_t}$, then set $(\theta^{(t+1)}, x^{(t+1)}) = (\theta^{(t+\frac{1}{2})}, x^{(t+1)})$ and $\sigma_{t+1} = \sigma_t$; otherwise, set $(\theta^{(t+1)}, x^{(t+1)}) = \mathcal{T}(\theta^{(t)}, x^{(t)})$ and $\sigma_{t+1} = \sigma_t + 1$.

The self-adjusting mechanism of the SAMC algorithm is obvious: If a proposal is rejected, the weight of the subregion that the current sample belongs to will be adjusted to a larger value, and thus the proposal of jumping out from the current subregion will less likely be rejected in the next iteration. This mechanism enables the algorithm to escape from local energy minima very quickly. The SAMC algorithm represents a significant advance in simulations of complex systems for which the energy landscape is rugged.

The proposal distribution $q(x, y)$ used in the MH updates satisfies the following condition: For every $x \in \mathcal{X}$, there exist $\epsilon_1 > 0$ and $\epsilon_2 > 0$ such that

$$|x - y| \leq \epsilon_1 \implies q(x, y) \geq \epsilon_2, \tag{13}$$

where $|x - y|$ denotes a certain distance measure between x and y . This is a natural condition in the study of MCMC theory (Roberts and Tweedie, 1996). In practice, this kind of proposals can be easily designed for both discrete and continuum systems as discussed in Liang et al. (2007). Since \mathcal{X} is compact in the context of Bayesian networks, it is easy to verify that the proposal distributions described in Section 3.2 satisfy condition (13).

SAMC falls into the category of varying truncation stochastic approximation algorithms (Chen, 2002; Andrieu et al., 2005). Following Liang et al. (2007), we have the following convergence result: Under conditions (9) and (13), for all non-empty subregions,

$$\theta_i^{(t)} \rightarrow C + \log \left(\int_{E_i} \psi(x) dx \right) - \log (\pi_i + \pi_0), \tag{14}$$

as $t \rightarrow \infty$, where $\pi_0 = \sum_{j \in \{i: E_i = \emptyset\}} \pi_j / (m - m_0)$, $m_0 = \#\{i : E_i = \emptyset\}$ is the number of empty subregions, and $C = -\log \left(\int_{E_m} \psi(x) dx \right) + \log (\pi_m + \pi_0)$. In SAMC, the sample space partition can be made blindly by simply specifying some values of u_1, \dots, u_{m-1} . This may lead to some empty subregions.

Let $\widehat{\pi}_i^{(t)} = P(x^{(t)} \in E_i)$ be the probability of sampling from the subregion E_i at iteration t . Eq. (14) implies that as $t \rightarrow \infty$, $\widehat{\pi}_i^{(t)}$ will converge to $\pi_i + \pi_0$ if $E_i \neq \emptyset$ and 0 otherwise. With an appropriate specification of π , sampling can be biased to the low energy subregions to increase the chance of locating the global energy optimizer.

Let $(x^{(1)}, \theta^{(1)}), \dots, (x^{(n)}, \theta^{(n)})$ denote a set of samples generated by SAMC. Let $y^{(1)}, \dots, y^{(n')}$ denote the distinct samples among $x^{(1)}, \dots, x^{(n)}$. Generate a random variable/vector Y such that

$$P(Y = y^{(i)}) = \frac{\sum_{t=1}^n e^{\theta^{(t)}} I_{J(x^{(t)})}(x^{(t)} = y^{(i)})}{\sum_{t=1}^n e^{\theta^{(t)}}}, \quad i = 1, \dots, n', \tag{15}$$

where $I(\cdot)$ is the indicator function, and $J(x^{(t)})$ denote the index of the subregion the sample $x^{(t)}$ belongs to. Since the number of truncations in the varying truncation algorithm can only occur a finite number of times (Andrieu et al., 2005), $\theta_{J(x^{(t)})}^{(t)}$ can be bounded in a compact set and is thus finite. By calling some results from the literature of non-homogeneous Markov chains, Liang (in press) showed that the random variable/vector Y generated in (15) is asymptotically distributed as $f(\cdot)$. Note that the samples $(x^{(1)}, \theta^{(1)}), \dots, (x^{(n)}, \theta^{(n)})$ form a non-homogeneous Markov chain. Therefore, for an integrable function $h(x)$, the expectation $E_f h(x)$ can be estimated by

$$\widehat{E_f h(x)} = \frac{\sum_{t=1}^n e^{\theta^{(t)}} h(x^{(t)})}{\sum_{t=1}^n e^{\theta^{(t)}}}. \tag{16}$$

As $n \rightarrow \infty$, $\widehat{E_f h(x)} \rightarrow E_f h(x)$ for the same reason that the usual importance sampling estimate converges (Geweke, 1989).

3.2. Learning Bayesian networks using SAMC

In this subsection, we first describe how to make the MH moves over the space of feasible Bayesian networks, and then discuss some practical issues on the implementation of SAMC. Let \mathcal{G} denote a feasible Bayesian network for the data \mathcal{D} . At each iteration of SAMC, the sampling step can be performed as follows:

- (a) Uniformly randomly choose between the following possible changes to the current network $\mathcal{G}^{(t)}$ producing \mathcal{G}' :
 - (a.1) Temporal order change: Swap the order of two neighboring models. If there is an edge between them, reverse its direction.
 - (a.2) Skeletal change: Add (or delete) an edge between a pair of randomly selected nodes.
 - (a.3) Double skeletal change: Randomly choose two different pairs of nodes, and add (or delete) edges between each pair of the nodes.
- (b) Calculate the ratio

$$r = e^{\theta_{J(\mathcal{G}')}^{(t)} - \theta_{J(\mathcal{G}^{(t)})}^{(t)}} \frac{\psi(\mathcal{G}')}{\psi(\mathcal{G}^{(t)})} \frac{T(\mathcal{G}') \rightarrow T(\mathcal{G}^{(t)})}{T(\mathcal{G}^{(t)}) \rightarrow T(\mathcal{G}')},$$

where $\psi(\mathcal{G})$ is defined as the right-hand side of (6), and the ratio of the proposal probabilities $T(\mathcal{G}') \rightarrow T(\mathcal{G}^{(t)}) / T(\mathcal{G}^{(t)}) \rightarrow T(\mathcal{G}') = 1$ for all of the three types of the changes. Accept the new network structure \mathcal{G}' with probability $\min(1, r)$. If it is accepted, set $\mathcal{G}^{(t+1)} = \mathcal{G}'$; otherwise, set $\mathcal{G}^{(t+1)} = \mathcal{G}^{(t)}$.

It is easy to see that this proposal satisfies condition (13). We note that a similar proposal has been used by Wallace and Korb (1999) in a Metropolis sampling process of Bayesian networks. Later we will show by numerical examples that the SAMC sampling process can mix much faster over the space of Bayesian networks than the Metropolis sampling process. Note that the double changes are not necessary for the algorithm to work, but are included to help accelerate the sampling process.

Table 1
The definition of the distribution P_1 .

X	Z	$P(Z X)$	X	Z	$P(Z X)$
0	0	0.8	0	1	0.2
1	0	0.2	1	1	0.8

Let $h(\mathcal{G})$ denote a quantity of interest for a Bayesian network, such as the presence/absence of an edge or a future observation. It follows from (16) that $E_p h(\mathcal{G})$, the expectation of $h(\mathcal{G})$ with respect to the posterior (6), can be estimated by

$$\widehat{E_p h(\mathcal{G})} = \frac{\sum_{k=n_0+1}^n h(\mathcal{G}_k) e^{\theta_{J(\mathcal{G}_k)}^{(k)}}}{\sum_{k=n_0+1}^n e^{\theta_{J(\mathcal{G}_k)}^{(k)}}}, \tag{17}$$

where $(\mathcal{G}_{n_0+1}, \theta_{J(\mathcal{G}_{n_0+1})}^{(n_0+1)}), \dots, (\mathcal{G}_n, \theta_{J(\mathcal{G}_n)}^{(n)})$ denotes a set of samples generated by SAMC, and n_0 denotes the number of burn-in iterations.

For an effective implementation of SAMC, several issues need to be considered.

- Sample space partitioning. For learning Bayesian networks, the sample space is usually partitioned according to the energy function. The maximum energy difference in each subregion should be bounded by a reasonable number, say, 2, which ensures that the MH moves within the same subregion have a reasonable acceptance rate. Note that within the same subregion, the SAMC move is reduced to the conventional MH move.
- Choice of the desired sampling distribution π . Since π controls the sampling frequency of each subregion, intuitively, one may choose it to bias sampling to the low energy subregions to increase the chance of finding the global energy minima. In this article, we set

$$\pi_i \propto (m - i + 1)^2, \quad i = 1, 2, \dots, m, \tag{18}$$

in all computations.

- Choice of t_0 and N , where N denotes the total number of iterations. Since a large value of t_0 will force the sampler to reach all subregions quickly, even in the presence of multiple local energy minima, t_0 should be set to a large value for a complex problem. The appropriateness of the choice of t_0 and N can be diagnosed by checking the convergence of multiple runs (starting with different points) through an examination for the variation of $\hat{\theta}$ or $\hat{\pi}$, where $\hat{\theta}$ and $\hat{\pi}$ denote, respectively, the estimates of θ and π obtained at the end of a run. A rough examination for $\hat{\theta}$ is to see visually whether the $\hat{\theta}$ vectors produced in the multiple runs follow the same pattern. Existence of different patterns implies that the gain factor is still large at the end of the runs or some parts of the sample space are not yet visited in all runs. An examination for $\hat{\pi}$ can be based on the following statistic

$$\epsilon_f(E_i) = \begin{cases} \frac{\hat{\pi}_i - (\pi_i + \hat{d})}{\pi_i + \hat{d}} \times 100\%, & \text{if } E_i \text{ is visited,} \\ 0, & \text{otherwise,} \end{cases} \tag{19}$$

which measures the deviation of the realized sampling distribution from the desired one, where $\hat{d} = \sum_{j \in \{i: E_i \text{ is not visited}\}} \pi_j / (m - m'_0)$ and m'_0 is the number of subregions not visited in the run. It is said that $\{\epsilon_f(E_i)\}$, output from all runs and for all subregions, matches well if the following two conditions are satisfied: (i) there does not exist such a subregion which is visited in some runs but not in others, and (ii) $\max_{i=1}^m |\epsilon_f(E_i)|$ is less than a threshold value, say, 10%, for all runs. A group of $\{\epsilon_f(E_i)\}$ which does not match well implies that the runs have not yet converged. In this case, SAMC should be re-run with a large value of N or a larger value of t_0 .

4. Numerical examples

4.1. An illustrative example

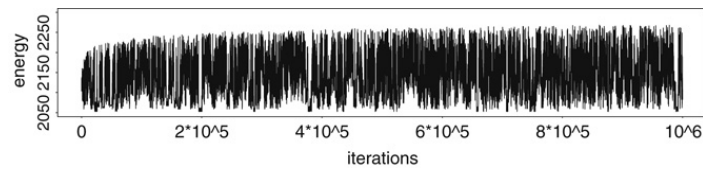
Consider the Bayesian network shown in Fig. 1 again. Suppose that a dataset, consisting of 500 independent observations, has been generated from the network according to the following distributions: $V_A \sim \text{Bernoulli}(0.7)$, $V_D \sim \text{Bernoulli}(0.5)$, $V_C|V_A \sim P_1$, $V_F|V_C, V_D \sim P_2$, $V_B|V_A, V_F \sim P_2$, $V_G|V_B, V_C \sim P_2$, and $V_E|V_G \sim P_1$, where P_1 and P_2 are defined as in Tables 1 and 2, respectively.

SAMC was first applied to this example, we partitioned the sample space into 501 subregions with an equal energy bandwidth, $E_1 = \{x : U(x) \leq 2000\}$, $E_2 = \{x : 2000 < U(x) \leq 2001\}$, ..., $E_{500} = \{x : 2498 < U(x) \leq 2499\}$, and $E_{501} = \{x : U(x) > 2499\}$, and set other parameters as follows: $\psi(x) = e^{-U(x)}$ and $t_0 = 5000$. SAMC was run for 10^6 iterations, and 10 000 samples were collected at equally spaced time points. Each run costs about 90 s CPU time on a 2.8 GHz computer (all computations reported in this article were done on the same computer). The overall acceptance rate of the

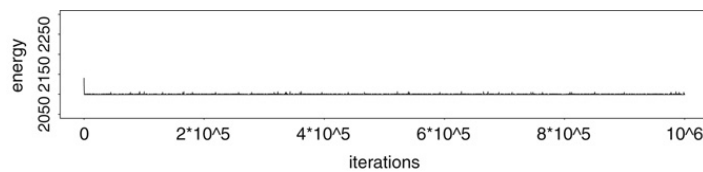
Table 2

The definition of the distribution P_2 .

X	Y	Z	$P(Z X, Y)$	X	Y	Z	$P(Z X, Y)$
0	0	0	0.9	0	0	1	0.1
0	1	0	0.5	0	1	1	0.5
1	0	0	0.5	1	0	1	0.5
1	1	0	0.1	1	1	1	0.9



(a) Evolving path of SAMC samples.



(b) Evolving path of MH samples.

Fig. 2. The sample paths (in the space of energy) produced by SAMC (upper panel) and MH (lower panel) for the simulated example.

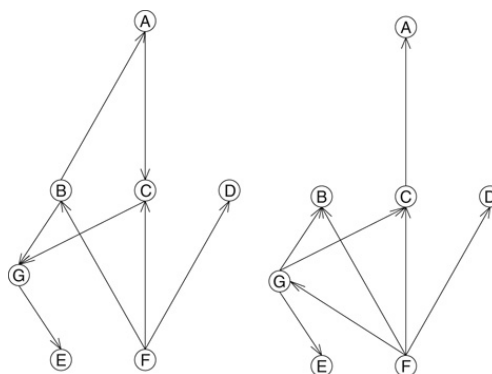


Fig. 3. The highest posteriori network structures produced by SAMC (left panel) and MH (right panel) for the simulated example.

SAMC moves is about 0.18. For comparison, MH was also applied to this example. MH was run for 10^6 iterations with the same proposal as used by SAMC, and 10 000 samples were collected at equally spaced time points. Each run costs about 81 s CPU time. The overall acceptance rate of the MH moves is 0.006, which is extremely low. Fig. 2(a) & (b) show, respectively, the sample paths (in the space of energy) produced by SAMC and MH. The sample paths indicate that SAMC can move very fast over the space of Bayesian networks, while MH tends to get trapped in a local energy minimum. The minimum energy values found by SAMC and MH are 2052.88 and 2099.19, respectively. The corresponding network structures are shown in Fig. 3. It is easy to see that the network produced by SAMC has the same skeleton with the true network, but with three edges reversed. The network structure produced by MH is quite different from the true one. We call the network produced by SAMC the putative maximum *a posteriori* (MAP) network. We note that the energy value of the true network is 2106.6, which is much higher than that of the putative MAP network.

Regarding the edge direction of Bayesian networks, we note again that the direction of an edge, in our formulation, does not necessarily imply the causal relation between the parent node and the child node. Therefore, for Bayesian networks, inference of structures should focus on the probability of presence/absence of the edges instead of the direction of the edges.

Later, SAMC was re-run 5 times with each run being lengthened to 2.5×10^6 iterations. In each of the five runs, the putative MAP network shown in Fig. 3(a) was re-located, and no networks with lower energy values were found. For the purpose of estimation, in each of the five runs, we discarded the first 5×10^5 iterations for the burn-in process, and retained the remaining iterations for the network inference. Table 3 shows the estimates of the presence probabilities of all possible edges of the Bayesian network, which are calculated using (17) based on five independent runs. Table 3 can be viewed as

Table 3

Estimates of the presence probabilities of the edges for the Bayesian network shown in Fig. 1. The numbers in parentheses show the standard errors of the estimates.

	A	B	C	D	E	F	G
A	–	0 (0)	0.9997 (0.0001)	0 (0)	0 (0)	0 (0)	0 (0)
B	1 (0)	–	0 (0)	0 (0)	0.0046 (0.0009)	0.4313 (0.0552)	1 (0)
C	0.0003 (0.0001)	0 (0)	–	0 (0)	0 (0)	0 (0)	0.9843 (0.0036)
D	0 (0)	0 (0)	0 (0)	–	0.0002 (0)	0.0476 (0.0233)	0 (0)
E	0 (0)	0 (0)	0 (0)	0 (0)	–	0 (0)	0.0044 (0.0009)
F	0 (0)	0.5687 (0.0552)	1 (0)	0.9524 (0.0233)	0.1638 (0.0184)	–	0 (0)
G	0 (0)	0 (0)	0.0003 (0.0001)	0 (0)	0.9956 (0.0009)	0 (0)	–

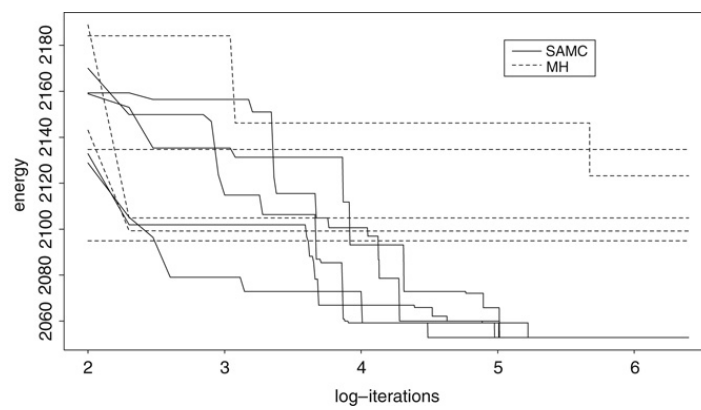


Fig. 4. Progression paths of minimum energy values produced in the five runs of SAMC (solid lines) and in the five runs of MH (dashed lines) for the simulated example.

a random graph. Further inference of the Bayesian network, for example, prediction for a future observation, can then be made from the random graph based on its Markov property.

For comparison, MH was also run 5 times with each run consisting of 2.5×10^6 iterations. Fig. 4 compares the progression paths of minimum energy values produced by SAMC and MH. It indicates again that SAMC is superior to MH for the learning of Bayesian networks. SAMC can locate the putative MAP network in each of the five runs, while the MH algorithm failed to locate the putative MAP network in all of the five runs.

4.2. Wisconsin breast cancer data

The Wisconsin Breast Cancer dataset, collected by Dr. W.H. Wolberg at the University of Wisconsin Hospitals, has 683 samples, which consist of visually assessed nuclear features of fine needle aspirates taken from patients' breasts. Each sample was assigned a 9-dimensional vector of diagnostic characteristics: clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses. Each component is in the interval 1–10, with 1 corresponding to the normal state and 10 to the most abnormal state. The samples were classified into two classes, benign and malignant. The classification was confirmed by either biopsy or further examinations. For a detailed description for the dataset, see Mangasarian and Wolberg (1990). In this article, we try to build a Bayesian network for the diagnostic characteristics and the overall diagnosis of the patient.

SAMC was first applied to this example. We partitioned the sample space into 17 001 subregions with an equal energy bandwidth, $E_1 = \{x : U(x) \leq 8000\}$, $E_2 = \{x : 8000 < U(x) \leq 8001\}$, ..., $E_{17000} = \{x : 24\,998 < U(x) \leq 24\,999\}$, and $E_{17001} = \{x : U(x) > 24\,999\}$, and set other parameters as follows: $\psi(x) = e^{-U(x)}$ and $t_0 = 10^5$. In the simulations, we also restricted the number of parents for each node to be no more than 5. SAMC was run 5 times. Each run consisted of 5×10^6 iterations, and cost about 35 m CPU time. The overall acceptance rate of the SAMC moves is about 0.2. For comparison, MH was also run 5 times for this example with the same proposal as used by SAMC. Each run consisted of 5×10^5 iterations, and cost about 15 m CPU time. The overall acceptance rate of the MH moves is about 0.15. Fig. 5 compares the progression paths of minimum energy values produced by SAMC and MH in the respective runs. It indicates that all of the five SAMC runs converge to the putative MAP network with an energy of 8373.9, while no MH runs converge to that value and all the MH runs got stuck at the very beginning of the simulations.

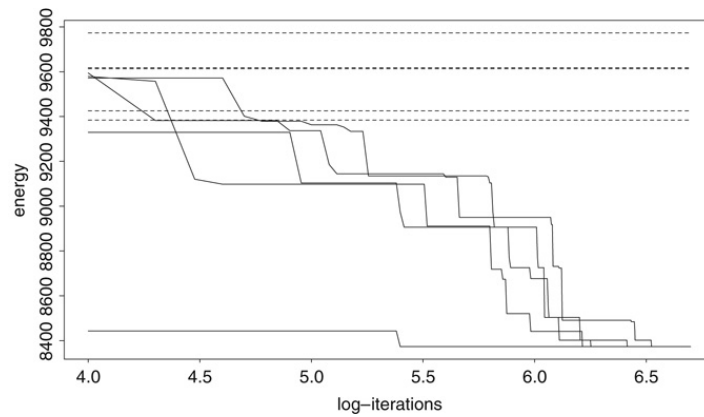


Fig. 5. Progression paths of minimum energy values produced in five runs of SAMC (solid lines) and five runs of MH (dashed lines) for the Wisconsin Breast Cancer example.

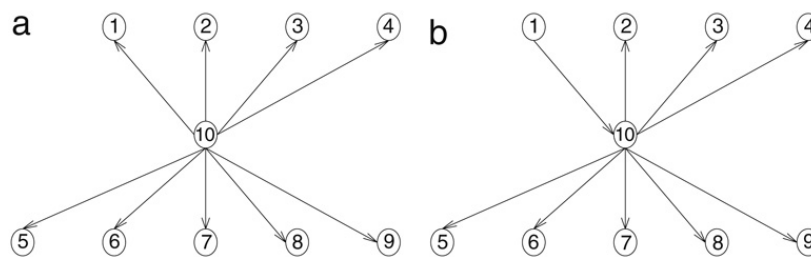


Fig. 6. The putative MAP Bayesian network (left panel, with an energy value of 8373.9) and a suboptimal Bayesian network (right panel, with an energy value of 8441.73) produced by SAMC for the Wisconsin Breast Cancer data.

Fig. 6(a) shows the putative MAP network produced by SAMC. It indicates that the overall diagnoses (node 10) of the patients depend on all 9 diagnostic characteristics, while the diagnostic characteristics are independent of each other conditional on the overall diagnoses of the patients. The presence probabilities of all possible edges of the network were also evaluated based on the five runs of SAMC. As a result, each edge in the putative MAP network has a presence probability of 1.0, and all other edges have a presence probability of zero. In simulations, SAMC also produced some networks with direction-reversed edges, but these networks all have higher energy values than the putative MAP network. For example, the network shown in Fig. 6(b) has one edge reversed from the putative MAP network, but has an energy value of 8441.73. Note that the network with all edges being reversed from the MAP network is not allowed, as we have restricted the number of parents for each node to be no more than 5 in simulations.

4.3. SPECT heart data

This dataset is available at machine learning repository <http://archive.ics.uci.edu/ml>. It describes diagnosing of cardiac Single Proton Emission Computed Tomography (SPECT) images. Each of the patients is classified into two categories: normal and abnormal. The database of 267 SPECT image sets (patients) was processed to extract features that summarize the original SPECT images. As a result, 22 binary feature patterns were created for each patient. In the past, the SPECT dataset has been used by a number of authors, including Cios et al. (1997) and Kurgan et al. (2001), to demonstrate their machine learning algorithms. In this article, we try to build a Bayesian network for the features and the overall classification of the patient.

SAMC was first applied to this example. We partitioned the sample space into 2001 subregions with an equal energy bandwidth, $E_1 = \{x : U(x) \leq 2000\}$, $E_2 = \{x : 2000 < U(x) \leq 2001\}$, ..., $E_{2000} = \{x : 3998 < U(x) \leq 3999\}$, and $E_{2001} = \{x : U(x) > 3999\}$, and set other parameters as follows: $\psi(x) = e^{-U(x)}$ and $t_0 = 50\,000$. SAMC was run 5 times. Each run consisted of 2×10^8 iterations, and cost about 325m CPU time. The overall acceptance rate of the SAMC moves is about 0.13. For comparison, MH was also run 5 times for this example with the same proposal as used by SAMC. Each run consisted of 2.0×10^8 iterations, and cost about 167m CPU time. The overall acceptance rate of the MH moves is only about 0.006. Fig. 7 compares the progression paths of minimum energy values produced by SAMC and MH in the respective runs. It is obvious that SAMC outperforms MH for this example; the minimum energy value produced by SAMC in any of the five runs is much lower than that produced by MH in all of the five runs.

Fig. 8 shows the putative MAP Bayesian network learned by SAMC over the five runs, where the node 23 corresponds to the overall classification of the patients. The plot indicates that conditioned on the features 17 and 21, the classification of the patients is independent of other features. Fig. 9 shows the consensus network for which each edge presents in the posterior network samples with a probability higher than 0.5. For example, the edge from 17 to 23 has a probability of 0.67

can lead to poor accuracy of the network structure, while discretization with a large number of states can lead to excessive computation efforts. Recently, some researchers seek to discretize the data non-uniformly according to a certain criterion, such as the conditional entropy (Fayyad and Irani, 1993) and the minimum description length (Wang et al., 2006).

Due to the extra weight updating step, SAMC costs a little more time than MH in each iteration. We note that the extra CPU cost can be significantly reduced by choosing the desired sampling distribution π to be uniform, especially when the number of non-empty subregions is large. When the desired sampling distribution is uniform, the weight updating step can be replaced by the following step,

(b') Set

$$\theta_i^{(t+\frac{1}{2})} = \theta_i^{(t)} + a_{t+1} (I_{\{x^{(t+1)} \in E_i\}} - I_{\{x^{(t+1)} \in E_m\}}), \quad (20)$$

where the weight is only updated for a single subregion that the current sample $x^{(t+1)}$ belongs to instead of all non-empty subregions. In our experience, the uniformly desired sampling distribution does not significantly deteriorate the performance of SAMC.

In this article, the inference of Bayesian networks is done via the dynamic importance estimator (17) for which the weights of many terms are very low. Alternatively, the inference can be done using the Occam's window approach as described in Madigan and Raftery (1994). That is, estimating $E_p h(\mathcal{G})$ by

$$\widetilde{E_p h(\mathcal{G})} = \frac{\sum_{\mathcal{G} \in \mathcal{A} \setminus \mathcal{B}} h(\mathcal{G}) P(\mathcal{G} | \mathcal{D})}{\sum_{\mathcal{G} \in \mathcal{A} \setminus \mathcal{B}} P(\mathcal{G} | \mathcal{D})}, \quad (21)$$

where

$$\mathcal{A} = \left\{ \mathcal{G}_k : \frac{\max_l P(\mathcal{G}_l | \mathcal{D})}{P(\mathcal{G}_k | \mathcal{D})} \leq c \right\},$$

for some constant c , say, a value between 10 and 100 as suggested by Jeffreys (1961, app. B); and

$$\mathcal{B} = \left\{ \mathcal{G}_k : \exists \mathcal{G}_l \in \mathcal{A}', \mathcal{G}_l \subset \mathcal{G}_k, \frac{P(\mathcal{G}_l | \mathcal{D})}{P(\mathcal{G}_k | \mathcal{D})} > 1 \right\}.$$

Two basic principles underly the estimator (21). Firstly, if a model predicts the data far less well than the best model in the class, it should be discarded, so the models not belonging to the set \mathcal{A} should be excluded from estimation. Secondly, appealing to Occam's razor, a model receiving less support from the data than any of its simpler sub-models should no longer be considered, so the models belonging to \mathcal{B} should also be excluded. Calculation of (21) is more expensive than calculation of estimator (17), as it includes a series of storages and comparisons of the posterior network samples.

Acknowledgment

Liang's research was supported in part by the grant (DMS-0607755) of the National Science Foundation and the award (KUS-C1-016-04) given by King Abdullah University of Science and Technology (KAUST). The authors thank Professor S.P. Azen, the associate editor, and the referee for their comments which have led to significant improvement of this paper.

References

- Andrieu, C., Moulines, É., Priouret, P., 2005. Stability of stochastic approximation under verifiable conditions. *SIAM J. Control Optim.* 44, 283–312.
- Bouckaert, R.R., 1995. Bayesian belief networks: From construction to inference. Ph.D. Thesis, University of Utrecht.
- Chen, H.F., 2002. Stochastic Approximation and its Applications. Kluwer Academic Publishers, Dordrecht.
- Chickering, D.M., 1996. Learning Bayesian networks is NP-complete. In: Fisher, D., Lenz, H.-J. (Eds.), *Learning from Data: Artificial Intelligence and Statistics V*. Springer-Verlag, New York, pp. 121–130.
- Cios, K.J., Wedding, D.K., Liu, N., 1997. CLIP3: Cover learning using integer programming. *Kybernetes* 26, 513–536.
- Cooper, G.F., Herskovits, E., 1992. A Bayesian method for the induction of probabilistic networks from data. *Mach. Learn.* 9, 309–347.
- Cooper, G.F., Yoo, C., 1999. Causal discovery from a mixture of experimental and observational data. In: *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, CA, pp. 116–125.
- de Campos, L.M., Huete, J.F., 2000. A new approach for learning belief networks using independence criteria. *Int. J. Approx. Reason* 24, 11–37.
- Ellis, B., Wong, W.H., 2008. Learning causal Bayesian network structures from experimental data. *J. Amer. Statist. Assoc.* 103, 778–789.
- Fayyad, U., Irani, K., 1993. Multi-interval discretization of continuous-valued attributes for classification learning. In: *Proceedings of International Joint Conference on Artificial Intelligence*. Chambery, France, pp. 1022–1027.
- Friedman, N., Linial, M., Nachman, I., Pe'er, D., 2000. Using Bayesian network to analyze expression data. *J. Comput. Biol.* 7, 601–620.
- Friedman, N., Koller, D., 2003. Being Bayesian about network structure: A Bayesian approach to structure discovery in Bayesian networks. *Mach. Learn.* 50, 95–125.
- Geweke, J., 1989. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica* 57, 1317–1339.
- Giudici, P., Green, P., 1999. Decomposable graphical Gaussian model determination. *Biometrika* 86, 785–801.
- Heckerman, D., Geiger, D., Chickering, D.M., 1995. Learning Bayesian networks: The combination of knowledge and statistical data. *Mach. Learn.* 20, 197–243.

- Herskovits, E., Cooper, G.F., 1990. Kutató: An entropy-driven system for the construction of probabilistic expert systems from datasets. In: Bonissone P. (Ed.), *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, Cambridge, pp. 54–62.
- Jeffreys, H., 1961. *Theory of Probability*, 3rd ed. Oxford University Press, Oxford.
- Kurgan, L.A., Cios, K.J., Tadeusiewicz, R., Ogiela, M., Goodenday, L.S., 2001. Knowledge discovery approach to automated cardiac SPECT diagnosis. *Artif. Intell. Med.* 23, 149–169.
- Lam, W., Bacchus, F., 1994. Learning Bayesian belief networks: An approach based on the MDL principle. *Comput. Intell.* 10, 269–293.
- Liang, F., 2002. Dynamically weighted importance sampling in Monte Carlo computation. *J. Amer. Statist. Assoc.* 97, 807–821.
- Liang, F., 2008. On the use of stochastic approximation Monte Carlo for Monte Carlo Integration. *Statist. Probab. Lett.*, in press (doi:10.1016/j.spl.2008.10.007).
- Liang, F., Liu, C., Carroll, R.J., 2007. Stochastic approximation in Monte Carlo computation. *J. Amer. Statist. Assoc.* 102, 305–320.
- Liu, J.S., Liang, F., Wong, W.H., 2001. A theory for dynamic weighting in Monte Carlo. *J. Amer. Statist. Assoc.* 96, 561–573.
- Madigan, D., Raftery, E., 1994. Model selection and accounting for model uncertainty in graphical models using Occam's window. *J. Amer. Statist. Assoc.* 89, 1535–1546.
- Madigan, D., York, J., 1995. Bayesian graphical models for discrete data. *Internat. Statist. Rev.* 63, 215–232.
- Mangasarian, O.L., Wolberg, W.H., 1990. Cancer diagnosis via linear programming. *SIAM News* 23, 1–18.
- Neil, J.R., Korb, K.B., 1999. The evolution of causal models. In: Zhong, N., Zhou, L. (Eds.), *Third Pacific Asia Conference on Knowledge Discovery and Data Mining*. Springer-Verlag, pp. 432–437.
- Pearl, J., 1998. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo.
- Robbins, H., Monro, S., 1951. A stochastic approximation method. *Ann. Math. Statist.* 22, 400–407.
- Roberts, G.O., Tweedie, R.L., 1996. Geometric convergence and central limit theorems for multidimensional Hastings and Metropolis algorithms. *Biometrika* 83, 95–110.
- Spirites, P., Glymour, C., Scheines, R., 1993. *Causation, Prediction and Search*. Springer-Verlag, New York.
- Wallace, C.S., Korb, K.B., 1999. Learning linear causal models by MML sampling. In: Gammernan, A. (Ed.), *Causal Models and Intelligent Data Management*. Springer-Verlag, Heidelberg.
- Wang, S., Li, X., Tang, H., 2006. Learning Bayesian networks structure with continuous variables. In: Li, X., Zaiane, O.R., Li, Z. (Eds.), *Lecture Notes in Computer Science*, vol. 4093. Springer-Verlag, Heidelberg, pp. 448–456.
- Wermuth, N., Lauritzen, S., 1983. Graphical and recursive models for contingency tables. *Biometrika* 72, 537–552.
- Wong, W.H., Liang, F., 1997. Dynamic weighting in Monte Carlo and optimization. *Proc. Natl. Acad. Sci. USA* 94, 14220–14224.