



Population Markov Chain Monte Carlo

KATHRYN BLACKMOND LASKEY

klaskey@gmu.edu

*Department of Systems Engineering and Operations Research, MS 4A6, George Mason University,
Fairfax, VA 22030, USA*

JAMES W. MYERS

James.w.myers@trw.com

TRW, VARI9D02, 12011 Sunset Hills Rd., Reston, VA 20190

Editors: Nando de Freitas, Christophe Andrieu, Arnaud Doucet

Abstract. Stochastic search algorithms inspired by physical and biological systems are applied to the problem of learning directed graphical probability models in the presence of missing observations and hidden variables. For this class of problems, deterministic search algorithms tend to halt at local optima, requiring random restarts to obtain solutions of acceptable quality. We compare three stochastic search algorithms: a Metropolis-Hastings Sampler (MHS), an Evolutionary Algorithm (EA), and a new hybrid algorithm called Population Markov Chain Monte Carlo, or popMCMC. PopMCMC uses statistical information from a population of MHSs to inform the proposal distributions for individual samplers in the population. Experimental results show that popMCMC and EAs learn more efficiently than the MHS with no information exchange. Populations of MCMC samplers exhibit more diversity than populations evolving according to EAs not satisfying physics-inspired local reversibility conditions.

Keywords: Markov chain Monte Carlo, Metropolis-Hastings algorithm, graphical probabilistic models, Bayesian networks, Bayesian learning, evolutionary algorithms

1. Introduction

Markov chain Monte Carlo (MCMC) has become increasingly popular as a general purpose class of approximation methods for complex inference, search and optimization problems. An MCMC is a stochastic simulation that visits solutions with long term frequency equal to the Boltzmann, or free energy minimizing, distribution. A variety of MCMC samplers can be constructed for any given problem by varying the sampling distribution subject to conditions that ensure convergence to the Boltzmann distribution. Samplers with the same long-run frequency distribution can vary greatly in their short-term dynamics. An important engineering challenge is to design samplers that rapidly reach low energy solutions but resist becoming trapped in local basins of attraction.

Various approaches have been proposed to improve performance of MCMC samplers. Global information about the energy function can be used if available to inform sampling and thus increase efficiency. For example, adaptive samplers use information from the sampling history to adjust the sampling distribution as sampling progresses (e.g., Gilks, Roberts, & Sahu, 1998). Care must be taken to ensure that the adaptation process does not destroy

ergodicity or worsen the convergence rate. Some authors have suggested using a population of MCMC samplers to assess the variability in results from different runs of the sampler (e.g., Gelman et al., 1995). Multiple runs can be used to develop tests of convergence that compare within-sampler and between-sampler variation in the solution (Gelman & Rubin, 1992). It has been suggested that performance might be improved by exchanging information among multiple samplers running in parallel (e.g., Kass & Raftery, 1995; Geyer, 1991). This approach is subject to the same difficulty as any adaptive sampler—how to use the information in a way that ensures that convergence to the target stationary distribution and desirable asymptotic properties of estimators derived from the sampler.

In examining ways to incorporate information exchange, it seems natural to consider evolutionary algorithms (EAs), a class of stochastic algorithms modeled after biological systems (Back, 1996; Fogel, 1991; Schwefel, 1995; Holland, 1995). In an EA, a population of simulated solutions evolves according to a Darwinian process of survival of the fittest. Information exchange between pairs of solutions occurs in a manner analogous to genetic reproduction. Asymptotic behavior of EAs is also typically analyzed using Markov chains, but unlike MCMC, characterizing the stationary distribution of an EA can be difficult (deJong, 1975; Davis & Principe, 1993).

This paper describes a modification of an MCMC sampler to incorporate information exchange among solutions in a population of Metropolis-Hastings samplers. The individual samplers are adaptive, using information from other samplers in the population to adjust their sampling distributions. At the population level, however, transition probabilities are constant, thus ensuring ergodicity and geometric convergence to the stationary distribution. We compare our popMCMC algorithm with both a standard EA and a population of independent Metropolis-Hastings samplers (MHS). As an experimental testbed, we have chosen the problem of learning directed graphical models, or Bayesian Networks (BNs) (Pearl, 1988; Jensen, 2001). We consider problems with missing observations and hidden variables, because most current approaches to such problems, which rely on local deterministic search, are acknowledged to be prone to halting at local optima (Lauritzen, 1995; Friedman, 1998a, 1998b). The stochastic nature of our algorithms allows movement away from local optima, while information exchange allows building blocks of good solutions to percolate within a population, thus potentially biasing the search in favor of better solutions.

The remainder of the paper is organized as follows. In Section 2 we describe the problem of learning directed graphical models using mixtures of conjugate prior distributions. In Section 3 we present the learning algorithms considered in this study. Section 4 describes our research hypotheses and presents empirical results. The final section summarizes our work and identifies directions for future research.

2. The learning problem

A Bayesian network (BN), or directed graphical model, specifies a joint probability distribution over a collection of random variables as a graph encoding conditional independence relationships and a set of local distributions encoding probability information. Each node in the graph represents a random variable that is conditionally independent of its non-descendants given its parents. The local distributions at each node specify a set of

probability distributions for the associated random variable, one for each combination of values for the node's parents. The local distributions implicitly encode a joint distribution over configurations of the random variables that satisfies the independence assumptions implied by the graph (Pearl, 1988).

Formally, let $X = (X_1, \dots, X_n)$ denote a collection of random variables. Let G be a directed acyclic graph with n nodes, where each node is associated with one of the X_i . Let $X_{pa(i|G)}$ denote the subset of random variables that are parents of X_i in G . Let j index states x_{ij} of X_i and let c index configurations x_{ic} of states of the parents $X_{pa(i|G)}$ of X_i in G . The local distribution at node X_i assigns a probability $\theta_{ijc} = p(X_i = x_{ij} \mid X_{pa(i|G)} = x_{ic})$ to each state x_{ij} of X_i given each configuration x_{ic} of its parents $X_{pa(i|G)}$. The conditional independence assumptions and the local distributions imply a joint probability for each configuration (x_1, \dots, x_n) of all the variables. This distribution is given by:

$$p(x_{1j_1}, \dots, x_{nj_n} \mid G, \theta_G) = \prod_i p(X_{i_j_i} = x_{i_j_i} \mid X_{pa(i|G)} = x_{i_c_i}, \theta_G) = \prod_i \theta_{i_j_i c_i} \quad (1)$$

Here, it is understood that there is one "parent configuration," $c = \emptyset$, for root nodes in the graph, and configurations of parents of non-root nodes range over the cross product of the state spaces of the parents.

The learning problem is to infer a graph G and a set of local distributions $\theta_G = \{\theta_{ijc}\}$ from an independent and identically distributed sample of cases drawn from the distribution (1). In Bayesian learning, a prior distribution is defined over graph structures and local distributions, and the cases are used to infer a posterior distribution. The most common approach (e.g., Cooper & Herskovits, 1992) is to assign a prior probability $q(G)$ to each graph and independent Dirichlet distributions $g(\theta_{i1c}, \dots, \theta_{ik_i c} \mid G)$ for each of the local conditional distributions θ_{ijc} :

$$g(\theta_{i1c}, \dots, \theta_{ik_i c} \mid G) = \frac{\Gamma(\alpha_{ic})}{\Gamma(\alpha_{i1c}) \dots \Gamma(\alpha_{ik_i c})} \theta_{i1c}^{\alpha_{i1c}-1} \dots \theta_{ik_i c}^{\alpha_{ik_i c}-1}, \quad (2)$$

where k_i is the number of states of node x_i , $\alpha_{ic} = \alpha_{i1c} + \dots + \alpha_{ik_i c}$, and $\Gamma(\alpha)$ is the gamma function.¹ The distribution (2) is a natural conjugate distribution for θ_G when the observations are independent draws from (1). The assumption that the distributions (2) are independent for different i and c permits the Bayesian updating problem to be decomposed into separate tractable problems for each node (e.g., Spiegelhalter & Lauritzen, 1990). The parameter $(\alpha_{i1c}, \dots, \alpha_{ik_i c})$ can be interpreted as a vector of prior "virtual counts" for the different values of variable X_i when the parent variable is in configuration c . The posterior distribution on $\theta_{i(c)}$ given a sample of cases drawn from (1) is also a member of the natural conjugate family, and is obtained by incrementing the prior counts by the sample counts. That is, the posterior distribution is Dirichlet with parameter $(\alpha_{i1c} + n_{i1c}, \dots, \alpha_{ik_i c} + n_{ik_i c})$, where n_{ijc} is the number of sampled cases in which X_i took on its j th value x_j and its parents $X_{pa(i|G)}$ took on configuration x_{ic} . Thus, the virtual counts can be interpreted as memory traces of past samples. Specifically, the prior distribution (2) is appropriate when one's knowledge is like having observed that in α_{ijc} out of α_{ic} past cases when $X_{pa(i|G)}$ was in configuration c , variable X_i was in state x_j .

The prior and posterior expected values of θ_{ijc} are given by

$$E[\theta_{ijc} \mid \alpha_{i1c}, \dots, \alpha_{ik_i c}, G] = P(X_i = x_j \mid \alpha_{i1c}, \dots, \alpha_{ik_i c}, G) = \frac{\alpha_{ijc}}{\alpha_{ic}} \quad (3)$$

and

$$E[\theta_{ijc} \mid \alpha_{i1c}, \dots, \alpha_{ik_i c}, n_{i1c}, \dots, n_{i,k_i c}, G] = \frac{\alpha_{ijc} + n_{ijc}}{\alpha_{ic} + n_{ic}}, \quad (4)$$

respectively, where α_{ic} is as defined above and $n_{ic} = \sum_j n_{ijc}$. Often, graphs are assumed equally likely *a priori* and uniform distributions $(\alpha_{i1c}, \dots, \alpha_{ik_i c}) = (1, \dots, 1)$ are used for the parameters (e.g., Cooper & Herskovits, 1992). If stronger prior information is available, it can be incorporated by specifying a non-uniform member of the natural conjugate family (Heckerman & Geiger, 1995).

Another convenient feature of the above family of prior distributions is the existence of a local decomposition of the marginal likelihood, or the probability of the observations conditional only on graph structure, integrated over the parameters θ_G . The sample configuration score for configuration x_c of the parents of node X_i is defined as:

$$\begin{aligned} \sigma(ic \mid G, x) \\ = \log \left(\frac{\Gamma(\alpha_{i1c} + n_{i1c}) \dots \Gamma(\alpha_{ik_i c} + n_{ik_i c})}{\Gamma(\alpha_{ic} + n_{ic})} \frac{\Gamma(\alpha_{ic})}{\Gamma(\alpha_{i1c}) \dots \Gamma(\alpha_{ik_i c})} \right). \end{aligned} \quad (5)$$

The sample graph score for graph G is the sum of the configuration scores:

$$\sigma(G, x) = \sum_{i,c_i} \sigma(ic_i \mid G) \quad (6)$$

where i ranges over node indices and c_i ranges over configurations of the parents of X_i . The graph score is the logarithm of the marginal likelihood $p(x \mid G)$ of the observed sample under the hypothesis that the conditional independence structure encoded by G obtains. The value $\sigma(G, x)$ thus measures how well the graph G fits the data. It is a sum of local components, one for each configuration of the parents of each node. The ratio of posterior probabilities of two graphs G_1 and G_2 is given by

$$\frac{q(G_1 \mid x)}{q(G_2 \mid x)} = \frac{q(G_1)}{q(G_2)} \exp\{\sigma(G_1, x) - \sigma(G_2, x)\} \quad (7)$$

Obtaining the posterior probability of a graph explicitly would require computing a normalization constant that sums over all possible directed acyclic graphs on n nodes:

$$q(G \mid x) = \frac{q(G) \exp\{\sigma(G, x)\}}{\sum_{G'} q(G') \exp\{\sigma(G', x)\}}, \quad (8)$$

Clearly, this is infeasible. Moreover, there is no enumeration procedure that is guaranteed to find the most probable graphs quickly. Thus, heuristic search methods are necessary. The local decomposition (6) provides the basis for efficient computation in incremental search. If two structures differ by only a single arc, then the difference in graph scores that appears in (7) can be computed from the configuration scores of just the two nodes connected by the arc. This provides a rapid test of the direction and amount of improvement in incremental learning algorithms that proceed by adding or deleting single arcs. These search methods have seen wide application and appear to perform well when the sample consists of independent draws from (1). However, deterministic incremental search is likely to terminate at local optima on more complex problems. Performing several runs from different starting points is a common approach to this difficulty. Stochastic search is another attractive option.

The closed form expressions for the posterior distribution of θ_{ijc} and $\sigma(G, x)$ apply when data are independently drawn complete cases distributed according to (1). Incomplete data adds complexity to the learning problem. A version of the EM algorithm has been applied to the problem of learning BNs with missing observations under the assumption that observations are missing at random (Friedman, 1998a, 1998b). EM and its variants are local hill climbing searches, and thus can become trapped at local optima, especially for certain patterns of missing observations. Stochastic search is attractive as a way to escape local basins of attraction without requiring random restarts. We assume observations are missing at random (Little & Rubin, 1987) and treat the missing observations as an additional component of the distribution to be sampled over. If both graph and missing data are sampled, then the graph score function (6) depends not just on the structure but also on the values sampled for the missing data. That is, observed data x_{obs} are augmented by sampled values x_{mis} for the missing data, forming the complete data $x_{com} = (x_{mis}, x_{obs})$. The score for a graph G , missing data x_{mis} , and observed data x_{obs} is denoted by $\sigma(G, x_{com}) = \sigma(G, x_{mis}, x_{obs})$. Our search algorithms hold x_{obs} fixed and vary G and x_{mis} in an attempt to find structure and missing data combinations with high posterior probability given x_{obs} .

In addition to the ability to escape local optima, another advantage of stochastic search algorithms is explicit representation of all components of uncertainty in the posterior distribution. Algorithms that return a single structure may understate uncertainty about the conditional independence relationships encoded by the graph. If a sample of graphs is returned, frequencies of arcs can be used to estimate the posterior probability of the existence and direction of arcs between nodes.

3. Stochastic search algorithms

3.1. Population-based stochastic search

The population-based search algorithms considered in this paper fall under the category of evolutionary algorithms (EAs), broadly defined. EAs operate by selecting and modifying a population of individual solutions in order to discover and evolve the population toward better solutions in the search space. In a slight modification of the formal characterization of Back (1996), we define an EA as follows:

Definition 1. An evolutionary algorithm is an 8-tuple

$$EA = \{I, \Phi, \Sigma, \Omega, \mu, \lambda, \Psi, \tau\} \quad (9)$$

such that:

- I is a set of individuals;
- Φ is a fitness function that maps individuals to real numbers;
- Σ is a selection operator that maps the population of individuals into a subset chosen for reproduction;
- Ω is a set of genetic operators (such as mutation and crossover) that transform subsets of “parent” individuals into subsets of “offspring” individuals;
- μ is the number of individuals in the parent generation;
- λ is the number of individuals in the child generation;
- Ψ is the process by which selection and genetic operators are applied to transform a population of parent individuals into a population of child individuals; and
- τ is a termination rule.

The algorithm takes an initial population as input and proceeds by successively applying the transformation process until the termination criterion is met.

For the problem of BN learning with incomplete data, the search space consists of graph structures G and sampled values for the missing observations x_{mis} . For all algorithms we used the fitness function $\Phi = \sigma(G, x_{obs}, x_{mis})$ as defined in (6). All algorithms also use the same representation I for graph structures and missing observations. All algorithms also used a fixed common population size $\mu = \lambda$ and termination rule τ . The algorithms differed only in the features of direct interest to this study: the selection operator Σ , genetic operators Ω , and intergenerational transformation process Ψ .

In the field of EAs, the terms *genotype* and *phenotype* are borrowed from biology to refer to the internal representation used by the algorithm and the expressed characteristics of the individual, respectively. In our problem, the phenotype is a BN with a particular graph structure G and local probability distributions given by the posterior expected value of θ_G given $x_{com} = (x_{obs}, x_{mis})$.

The choice of genotype is an important consideration in an EA, and is closely tied to the operators used to modify parent solutions to obtain offspring. It is important to find operators that are likely to transform good solutions into good solutions and also operators that facilitate non-local jumps, to keep the search from becoming mired in local basins of attraction. In particular, the representation and operators should facilitate the preservation of good “schema” (modules, or configurations of features that serve as “building blocks”) from generation to generation (Holland, 1995). To facilitate comparison of algorithms, we used a common representation and the same mutation operator for all algorithms. We varied only the factors that were of direct interest to our study, the information exchange operators (none for MHS, crossover for standard EA and statistical model for popMCMC) and the selection rule (individual-level Metropolis-Hastings for MHS, fitness-based pre-selection for standard EA, and population-level Metropolis-Hastings for popMCMC).

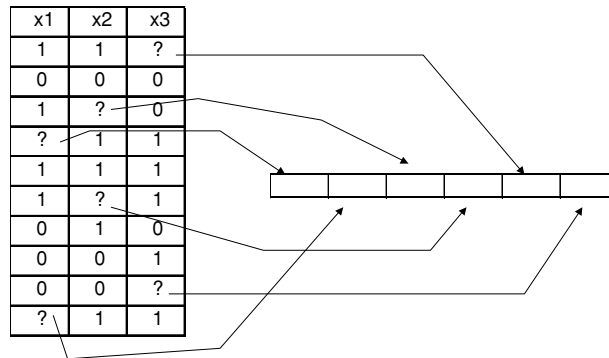


Figure 1. Missing data chromosome.

The genotype consists of two chromosomes, one for the missing data x_{mis} and one for the graph structure G . The missing data chromosome is represented as a vector as shown in figure 1. In figure 1, the observed data are shown as a table in which columns are variables in the BNs and rows are independent and identically distributed observations, in which some of the values, represented with the symbol “?”, are missing. Each position in the missing data chromosome corresponds to one of the “?” entries in the observed data table. For each individual in the population, the missing data chromosome is filled with imputed values for the corresponding missing entries in the data table.

The other chromosome specifies the network structure. We used the representation shown in figure 2 for network structures. The graph structure is represented as an adjacency list, where each row represents a variable in the network and the entries to the right of the vertical line represent the parents of the variable. We used this representation because it worked well in our initial experiments with EAs. However, it has the disadvantage that an adjacency list can represent an illegal structure (a graph with a directed cycle). The information exchange operators we implemented sometimes created illegal graph structures. We treated illegal graph structures by assigning them very low scores. We plan to run additional experiments using a representation that does not allow illegal structures.

Execution times for all algorithms we considered were very similar. Applying selection and genetic operators and detecting illegal structures were very fast. The most computationally intensive part of the algorithm was computing the structure scores, a step shared by

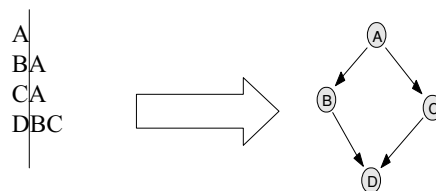


Figure 2. Graph structure chromosome.

Table 1. Comparison of algorithms.

	Independent MH	Evolutionary algorithm	popMCMC
Individuals I	x_{mis} as in figure 1; G as in figure 2	x_{mis} as in figure 1; G as in figure 2	x_{mis} as in figure 1; G as in figure 2
Fitness function Φ	$\sigma(G, x_{obs}, x_{mis})$	$\sigma(G, x_{obs}, x_{mis})$	$\sigma(G, x_{obs}, x_{mis})$
Selection Σ	Uniform pre-selection; MH post-selection	Binary tournament selection	Uniform pre-selection; MH post-selection
Operators Ω	Mutation	Mutation and crossover	Adaptive mutation
Parent population size μ	$\mu = 20$	$\mu = 20$	$\mu = 20$
Child population size λ	$\lambda = 20$	$\lambda = 20$	$\lambda = 20$
Transformation process Ψ	For each individual, apply mutation and accept or reject	Select pair; choose mutation or crossover; apply mutation to individuals or crossover to pairs; continue till next generation is populated	For each individual, apply adaptive mutation and accept or reject
Termination rule τ	500 iterations	500 iterations	500 iterations

all algorithms. Our algorithms were implemented as interpreted MATLAB functions. The slowest part of this computation was calculating the sufficient statistics n_{ijc} . We made no attempt to optimize this computation because our primary concern was comparing the algorithms. Improvements in this computation would provide a constant per-iteration speedup for all algorithms.

To summarize, we considered stochastic algorithms that search a space of solutions by evolving a population of solutions in a way that tends over time to improve the fitness of individuals in the population. In our problem, individuals in the population correspond phenotypically to BNs on a set of variables. The genotype is given by two chromosomes, one for the network structure and one for the missing observations. Table 1 compares the three algorithms considered in this study on the features identified in Definition 1. As noted above, the algorithms used the same representation for solutions, fitness function, parent and child population sizes, and termination rule. The fitness function was given a priori from the problem context as the logarithm of the marginal likelihood for the structure/missing data combination. Fitness therefore measures how well the phenotypic BN performs at predicting the observed data. The other fixed factors were varied in preliminary experiments to determine reasonable values for comparison runs. The features varied in the comparison study were the selection operator, the genetic operators, and the intergenerational transformation process. The algorithms we describe below correspond to different choices for these features.

There is a growing literature on the application of MCMC and EAs to complex learning, search and optimization problems. This literature is too numerous for a full review, but a few works are especially relevant. MCMC has become quite popular as a general-purpose tool

for Bayesian inference problems too complex to admit closed form solutions. The standard reference in this area is Gilks, Richardson, and Spiegelhalter (1996), which contains a good deal of useful practical advice. Madigan and York (1993) were first to apply Metropolis-Hastings sampling to search over structures for graphical models. The approach has been called MC³ for Markov Chain Monte Carlo Model Composition. We augment the standard MC³ sampler by imputing missing observations as we sample graph structures. Larrañaga et al. (1996) applied genetic algorithms to the problem of learning BN structures from a sample of observations and found that the genetic algorithm was able to find high-scoring structures. The treatment was limited to complete data and there was no direct comparison to other algorithms. Holmes and Mallick (1998) used genetic operators to inform the proposal distribution for a MHS. They demonstrated this algorithm on two difficult high dimensional problems: parameter estimation for a neural network regression model and knot selection for a spline interpolant. They found that their sampler converged quickly, consistently sampled from higher density areas than standard Metropolis-Hastings, and proposed larger changes without sacrificing acceptance probabilities. They did not compare their sampler with a standard evolutionary algorithm.

3.2. Metropolis-Hastings sampler

The first algorithm we considered was a Metropolis-Hastings sampler (MHS) (Metropolis et al., 1953; Hastings, 1970). For purposes of comparison with the other algorithms, we ran a population of independent samplers in parallel, but these samplers did not exchange information with each other.

A MHS is a Markov chain designed to simulate convergence of a system to the Boltzmann, or minimum free energy, distribution. An externally given potential field influences the motion of the system through configuration space. For our learning problem, configurations of the system are pairs $y = (G, x_{mis})$ of graph structures and missing observations. The potential energy of configuration y is equal to:

$$E(y) = E(G, x_{mis}) = -\sigma(G, x_{mis}, x_{obs}). \quad (10)$$

The Boltzmann distribution depends on a temperature parameter, which we set equal to unity. Under the unit temperature assumption, the Boltzmann distribution is the posterior distribution of graph structures and missing data given observed data:

$$\frac{1}{Z} \exp\{-E(y)\} = \frac{1}{Z} \exp\{\sigma(G, x_{mis}, x_{obs})\} = p(G, x_{mis} \mid x_{obs}), \quad (11)$$

where Z is a normalization constant. The MHS is designed to converge to stationary distribution (11), sampling structures and imputed values with long run frequency equal to their posterior probability given the observed data.

MH sampling proceeds as follows. Let the current configuration be denoted by $y^{(c)} = (G^{(c)}, x_{mis}^{(c)})$. A new configuration is proposed probabilistically according to a proposal distribution $R(y^{(n)} \mid y^{(c)})$. The proposed new configuration is either rejected or accepted according to a probabilistic rule. If the new configuration is accepted, it replaces the current

configuration; if it is rejected, the system stays at the current configuration. The acceptance probability is given by

$$A(y^{(n)} | y^{(c)}) = \min \left[1, \exp\{\sigma(y^{(n)}) - \sigma(y^{(c)})\} \frac{R(y^{(c)} | y^{(n)})}{R(y^{(n)} | y^{(c)})} \right]. \quad (12)$$

The transition probability from configuration $y^{(c)}$ to $y^{(n)}$ is therefore given by

$$T(y^{(n)} | y^{(c)}) = \begin{cases} R(y^{(n)} | y^{(c)}) A(y^{(n)} | y^{(c)}) & y^{(n)} \neq y^{(c)} \\ 1 - \sum_{y^{(n')} \neq y^{(c)}} R(y^{(n')} | y^{(c)}) A(y^{(n')} | y^{(c)}) & y^{(n)} = y^{(c)} \end{cases} \quad (13)$$

It is straightforward to verify that this transition distribution satisfies a condition known as detailed balance or local reversibility (Gilks, Richardson, & Spiegelhalter, 1996; Neal, 1993):

$$T(y^{(n)} | y^{(c)}) p(G^{(c)}, x_{mis}^{(c)} | x_{obs}) = T(y^{(c)} | y^{(n)}) p(G^{(n)}, x_{mis}^{(n)} | x_{obs}). \quad (14)$$

This implies (Feller, 1968) that the distribution $p(G, x_{mis} | x_{obs})$ is a stationary distribution for the chain. For our problem, the configuration space is finite. Thus, if all configurations are reachable from all other configurations and the transition distribution is time-independent, the system converges geometrically to a unique stationary distribution and satisfies a central limit theorem (Feller, 1968). While geometric convergence is guaranteed, the rate of convergence depends strongly on the proposal distribution and can be very slow in practice. An examination of (12) shows that if the proposal distribution is equal to the target stationary distribution, the acceptance probability is equal to 1, and the sampler reduces to independent draws from the target distribution. This of course is infeasible to carry out in practice, as the target distribution is unknown. However, it is well known that proposal distributions close to the target stationary distribution converge more quickly (e.g., Roberts, 1996). Proposal distributions that are too spread out relative to the target distribution suffer from low acceptance rates. Acceptance rates can usually be improved by concentrating samples near the current state, but this results in very slow traversal of the search space. The problem of local sampling and consequent correlation of successive observations in the Markov chain has been referred to as ‘‘poor mixing.’’

At each step of our algorithm, an arc was proposed for addition, deletion or reversal and a subset of the missing data was chosen for mutation. The proposal distributions were chosen so that the reversal probabilities needed for the Hastings adjustment were easily computed.

The missing data proposal distribution was constructed as follows. Each missing data site was selected for mutation with probability p_m or remained unchanged with probability $1 - p_m$, independent of which other sites were chosen for sampling. For each selected gene, the current value of the gene was replaced probabilistically by another value from the value set of the variable corresponding to that gene. For example, suppose a selected site maps to variable X with possible values $\{1, 2, 3, 4, 5\}$, and the current setting is 4. The proposed new value would be a random draw from the set $\{1, 2, 3, 5\}$.

The mutation operator for the structure chromosome is tailored to the structure representation of figure 2. There are two basic mutation operators: adding a variable to the list of parent variables and deleting a variable from the list of parent variables. These operators have the effect in the phenotype of adding and deleting arcs, respectively. We also include a third mutation operation, reversal of an arc, which is implemented genotypically by deleting the parent-child arc and adding the child-parent arc. To mutate structures, the algorithm first randomly selects whether to add, remove or reverse an arc. If an arc is to be added, it then selects randomly from among nodes with fewer parents than the maximum parent limit, and then selects an arc to add randomly from among the node's non-parents. If an arc is to be deleted or reversed, it selects randomly from among non-root nodes and then selects an arc at random from the node's parents to delete or reverse.

3.3. Evolutionary algorithm

The second algorithm we considered is an EA. This algorithm differs from the MCMC algorithm in two ways (Table 1: Comparison of Algorithms). First, the Metropolis-Hastings proposal and acceptance process is replaced by a process in which individuals are chosen for reproduction according to fitness, produce offspring, and are replaced in the population by their offspring. In the terminology of EAs, a standard EA uses *pre-selection*, whereas the MHS uses a *post-selection* rule that ensures convergence to a given stationary distribution. The second difference is that the MCMC samplers are independent of each other, whereas the EA includes a crossover operator in which information is exchanged between pairs of solutions.

We used binary tournament selection (Goldberg & Deb, 1991) to select pairs of individuals for reproduction. Two individuals are randomly selected from the population and compared. The one with the highest fitness is selected for reproduction. Then another two individuals are randomly selected and the most fit is kept as the mate to the first parent. This process continues μ times until the next generation is populated.

The crossover operator we used for both missing data and structure is called parameterized uniform crossover (Syswerda, 1989; DeJong & Spears, 1990). Parameterized uniform crossover selects a subset of the genes at random and exchanges the values of the genes between the parents. Figure 3 illustrates the crossover operation on graph structures. Note that it is possible to obtain illegal structures both from mutation and crossover. Illegal structures were replaced by a graph with no arcs, and thus received a very low score. As noted above, detection of illegal structures can be accomplished with low computational overhead in comparison with computation of structure scores.

3.4. Population Markov chain Monte Carlo

Many empirical studies have supported the use of EAs on problem domains with large, multi-dimensional, multi-modal search spaces. However, there is very little theory on dynamic behavior of EAs beyond special case results for canonical algorithms. The population of solutions in an EA is a Markov chain. When the mutation probability is non-zero, the population size is bounded, and the selection and intergenerational transformation processes are non-adaptive, the Markov chain is ergodic and converges geometrically to a unique

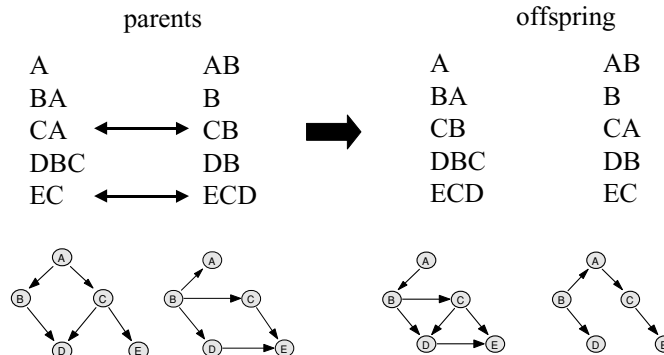


Figure 3. Uniform crossover for structures.

stationary distribution. However, this distribution is very difficult to characterize for problems of any complexity. A major advantage of MCMC algorithms is the explicit characterization of the stationary distribution up to a normalization constant. This can be a very useful theoretical tool. In Bayesian inference, this property is especially useful, because the algorithm can be designed to converge to a sample from the posterior distribution of interest, making the estimation of posterior moments quite straightforward. Even in optimization problems explicit characterization of the stationary distribution can be quite useful.

A persistent difficulty with EAs is the tendency to “genetic drift.” Once a good basin of attraction has been found, exploitation tends to win out over exploration, and the population tends to become homogeneous. For complex, multi-modal search spaces an EA may converge prematurely to a sub-optimal mode, leaving modes with better solutions unexplored. The EA community has tried many approaches to alleviate this problem such as niching (DeJong, 1975) speciation (Spears, 1994) and adaptive mutation (Kitano, 1990). To date, there is no agreed upon approach, but there are many promising prospects. We conjectured that the Hastings correction in (18) might be useful in mitigating the problem of genetic drift, because Hastings correction reduces the acceptance probability of jumps from which a return proposal is too unlikely relative to the probability of the nominated solution. Thus, proposals for jumps that would abandon a basin of attraction for a more probable basin might be rejected because finding and returning to the original basin would be too difficult.

As noted above, exchange of information between parallel samplers has been suggested as a way to improve mixing in MCMC samplers. Holmes and Mallick (1998) obtained encouraging results from applying an approach similar in many respects to ours. They focused on problems with continuous state spaces. Their information exchange operator selected two individuals for reproduction, exchanged some of the sites as in standard crossover, but then instead of using the result directly, the algorithm proposed a direction of change based on the newly generated solution. They then used Metropolis-Hastings post-selection to either move to the new solution or remain unchanged.

As noted above, efficiency of the MHS is improved by using a proposal distribution as close as possible to the target distribution. If a population of samplers is converging to a common target distribution, then it might be expected that global performance could be

improved if pooled information from the population were used to inform the individual proposal distributions. For example, we might expect the proportion of solutions with an arc between two variables to be higher for arcs with higher posterior probability. Making use of population arc frequencies in the proposal distribution might improve performance of the sampler. Similarly, the population proportion of missing values could be used to inform proposals for mutating the missing data chromosome.

To formalize this idea, consider a family of MH samplers, all with the same target distribution $p(y \mid x_{obs})$ but having different proposal distributions $R(y^{(n)} \mid y^{(c)}, \zeta)$ indexed by parameter ζ . A popMCMC algorithm uses a population of solutions to estimate features of the target stationary distribution in an attempt to select a proposal distribution as close as possible to the target distribution.

Definition 2. A population Markov Chain Monte Carlo (popMCMC) algorithm with target distribution $p(y)$, proposal distribution family $\{R(y^{(n)} \mid y^{(c)}, \zeta)\}_{\zeta \in \mathcal{Z}}$, population size μ , and proposal parameter function $\hat{\zeta}(y_1, \dots, y_\mu)$, is a component-wise Metropolis-Hastings sampler on tuples (y_1, \dots, y_μ) of states in which the proposal distribution for $y_i^{(n)}$ given $(y_1^{(c)}, \dots, y_\mu^{(c)})$ is $R(y_i^{(n)} \mid y_i^{(c)}, \hat{\zeta}(y_1^{(c)}, \dots, y_\mu^{(c)}))$.

Although we have not performed rigorous theoretical analysis of the properties of popMCMC samplers, we include a brief heuristic discussion of convergence properties and the choice of proposal parameter function $\hat{\zeta}$. The family $R(y_i^{(n)} \mid y_i^{(c)}, \zeta)$ of proposal distributions can be chosen as a rich semi-parametric family and the parameter function $\hat{\zeta}$ can be chosen to fit interesting features of $p(y)$, such as lower order marginal distributions of components of y . For some problems it may be possible to choose the family so that there exist ζ for which $R(y \mid y', \zeta)$ is nearly independent of y' and arbitrarily close to $p(y)$. For such families it may be possible to design parameter functions $\hat{\zeta}(y_1, \dots, y_\mu)$ so that that $R(y_i^{(n)} \mid y_i^{(c)}, \hat{\zeta})$ converges to $p(y)$ as the population size μ becomes large. Under appropriate regularity conditions, convergence rates of such samplers should approximate the convergence rate of the independence sampler.² It is well known that the independence sampler $R(y \mid \zeta)$ converges more rapidly as $\inf_y \{R(y \mid \zeta)/p(y)\}$ increases toward 1 (Roberts, 1996). Thus, under appropriate regularity conditions on the estimator $\hat{\zeta}$, examining $\inf_y \{R(y_i^{(n)} \mid y_i^{(c)}, \hat{\zeta})/p(y)\}$ when the y_i are independent draws from $p(y)$ should provide useful information about the convergence rate.

We based the popMCMC sampler for this study on population frequencies of arcs and missing values. Let π_{ij} be the probability of an arc from i to j under the distribution $p(y)$, and let A_{ij} be the number of individuals for which there is an arc from node i to node j in a population of size μ . Then $(A_{ij}, A_{ji}, \lambda - A_{ij} - A_{ji})$ follows a multinomial distribution with probability vector $(\pi_{ij}, \pi_{ji}, 1 - \pi_{ij} - \pi_{ji})$. If we assume a uniform prior distribution for the probability vector, then the posterior distribution given A_{ij} and A_{ji} is Dirichlet $(1 + A_{ij}, 1 + A_{ji}, 1 + \lambda - A_{ij} - A_{ji})$, and the posterior expected values for π_{ij} and π_{ji} are

$$\begin{aligned} \hat{\pi}_{ij} &= E[\pi_{ij} \mid A_{ij}] = \frac{A_{ij} + 1}{\lambda + 3} \quad \text{and} \\ \hat{\pi}_{ji} &= E[\pi_{ji} \mid A_{ji}] = \frac{A_{ji} + 1}{\lambda + 3} \end{aligned} \tag{15}$$

respectively. These expected values are used in place of the uniform sampling of arc additions, removals, and reversals in the independent MHS described in Section 3.2 above. Similarly, let B_{ij} be the number of individuals in the population for which the i th position in the missing data chromosome takes on the j th value, where j ranges from 1 to k , the number of states of the corresponding variable. Again using the Dirichlet conjugate prior distribution, the posterior expectation of the probability that the i th position in the missing data chromosome takes on value j is:

$$\hat{\rho}_{ij} = \frac{B_{ij} + 1}{\lambda + k} \quad (16)$$

We use these estimates to modify the MHS algorithm of Section 3.2 as follows. An individual to mutate is selected at random from the population. A subset of sites on the missing data chromosome is chosen by selecting each site with a constant probability. A pair of nodes is chosen at random to have an arc added, removed or reversed. The current values for arcs and missing values are replaced by a random draw from the distribution (15) in the case of the structure chromosome and (16) in the case of the missing data chromosome. The distributions (15) and (16) are then used in computing the Metropolis-Hastings acceptance probabilities.

At the level of the individual, the above proposal distribution is adaptive in that it depends on global information about the distribution of arcs and imputed observations in the current population. Our adaptation rule is based on a matching frequencies of arcs and missing observations. More complex adaptation strategies can be devised that assume more sophisticated patterns of correlation involving configurations of arcs or missing observations. Viewed at the level of the population of solutions, popMCMC is a Markov chain with fixed transition probabilities. The relative improvement of popMCMC over the non-adaptive MHS depends on how well $R(y_i^{(n)} | y_i^{(c)}, \hat{\zeta})$ approximates $p(y)$ as compared with an a priori fixed proposal distribution.

4. Empirical comparison of approaches

Comparisons among algorithms were run on the three networks shown in figure 4. The first is the one used in our exploratory studies. The first network is similar to a network used by Friedman (1998b) to investigate learning BNs with hidden variables. The second is a data set on college plans analyzed originally by Sewell and Shah (1968) and studied by Whittaker (1990) and Heckerman (1996). The third is the ALARM network (Beinlich et al., 1989), which has become a standard benchmark network for testing BN learning algorithms. We began by running a set of exploratory experiments on each algorithm to find a “good” set of parameters for that algorithm. We then conducted a set of experiments comparing the algorithms with each other. The exploratory experiments were conducted by generating cases from network in figure 4(a), from which training and test sets of 1000 observations each were generated. Design decisions such as the use of tournament selection for the EA, the rate of mutation, and the population sizes were made with the help of these exploratory experiments.

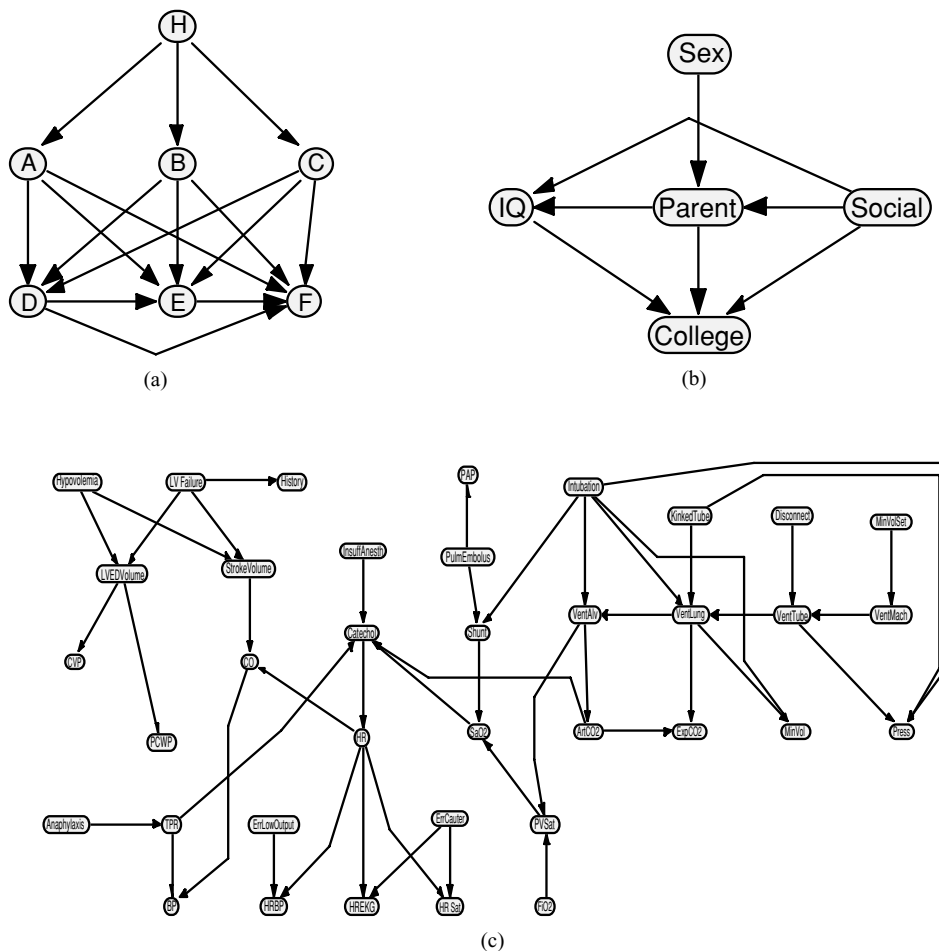


Figure 4. Networks used in comparison study. (a) $1 \times 3 \times 3$ network, (b) College plans network and (c) ALARM network.

Figure 5 compares a single run of the MHS with a single run of the EA on the $1 \times 3 \times 3$ network. We ran 500 generations with a population size of 20. The first plot shows best so far curves, which are used extensively in EA research to measure how quickly the search finds good solutions. The second plot shows the number of different network structures in the population in each generation. Solutions found by the EA improve at a much faster rate than those found by the MHS. On the other hand, the EA concentrates on only a few structures (different ones for each run), whereas the MHS retains a diverse population of structures for all 500 iterations. Although this figure shows only one run of each algorithm, a similar pattern was observed for multiple runs on each data set.

Figure 6 shows the same plot, this time comparing the EA with popMCMC. The adaptive mutation operator has enabled popMCMC to home in on good structures as quickly

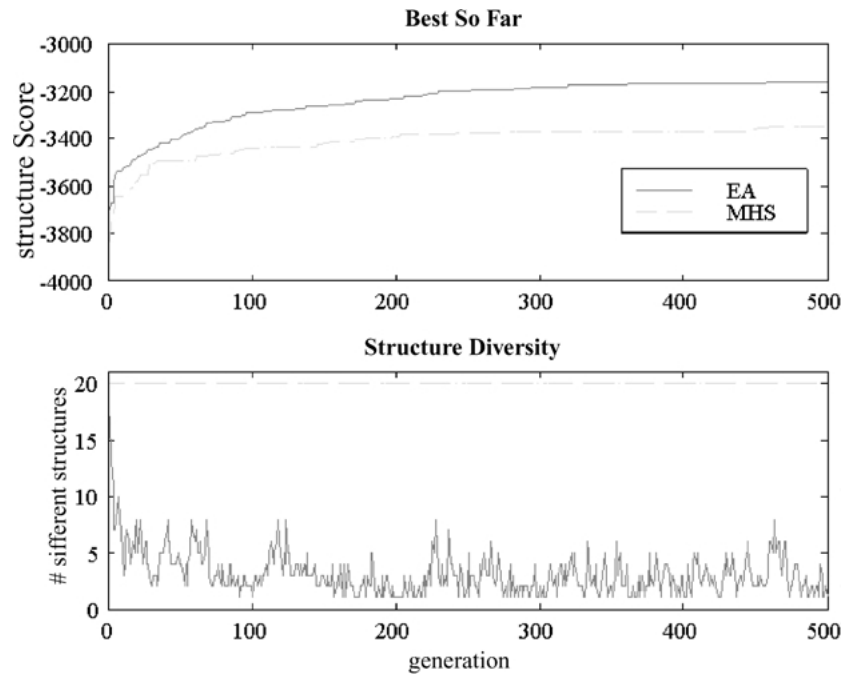


Figure 5. Comparison of EA with independent MHS.

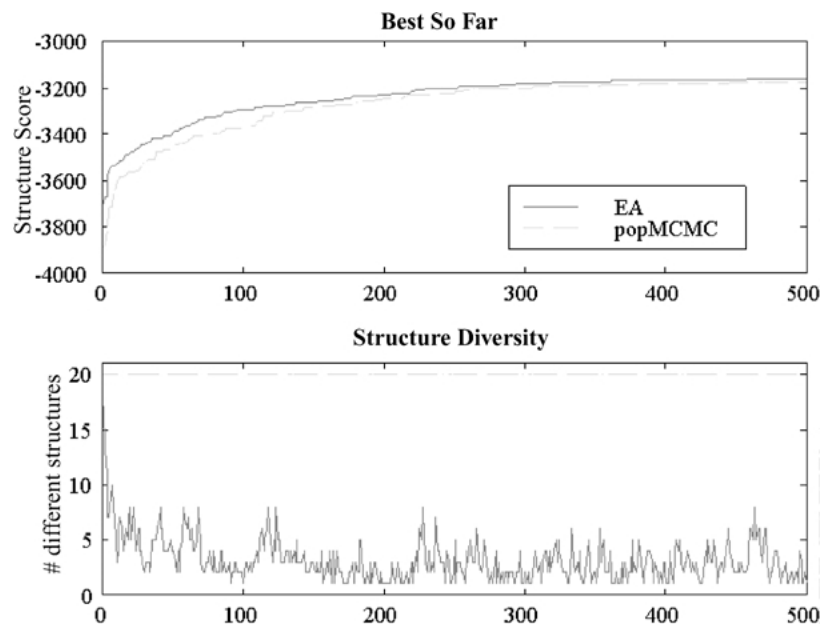


Figure 6. Comparison of EA with popMCMC.

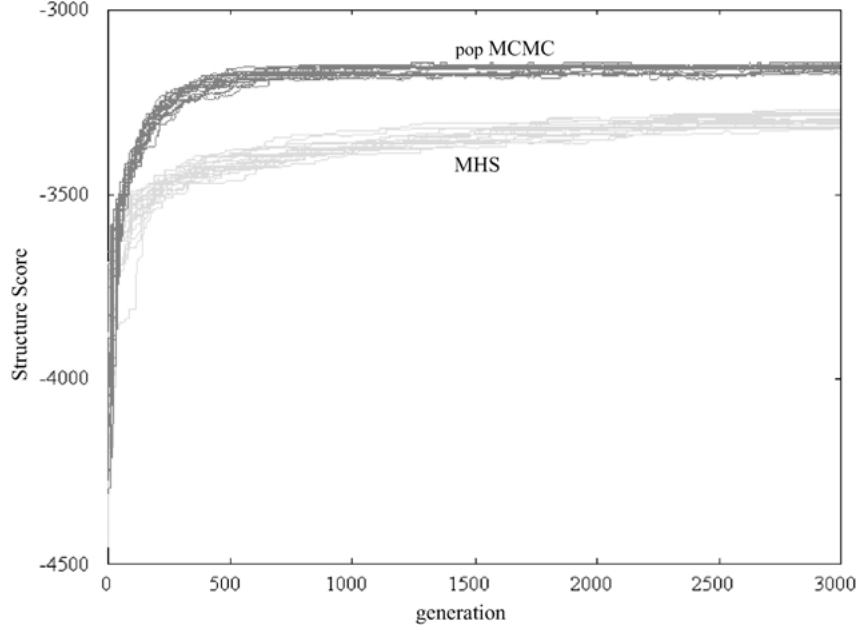


Figure 7. Trajectories for MHS and popMCMC.

as the EA, but the diversity is much greater. In a graphic illustration of the improvement in convergence rate from adaptive mutation, figure 7 plots the population of scores over a 500 generation history for popMCMC as compared with the standard MHS, again for the $1 \times 3 \times 3$ network. We ran the standard MHS for 5000 iterations and were unable to achieve scores as high as those attained by popMCMC and EA after 500 iterations. Interestingly, when we computed the scale reduction metric proposed by Gelman and Rubin (1992) as a convergence diagnostic, the standard MHS appeared to have converged after 500 iterations according to the criterion suggested by Gelman and Rubin. This suggests that sampling based approaches to diagnosing convergence may indicate convergence when the sampler remains far from the target stationary distribution.

Figure 8 compares five runs of each algorithm on the $1 \times 3 \times 3$ network. The first plot shows 95% credible intervals for the score $\sigma(G, x)$ after 500 runs. The second plot measures the ability of the models to do predictions on samples not seen during training. We took the learned structures, used the posterior expected value $\tilde{\theta}_G = E[\theta_G | G, x_o]$ to assign local distributions, and used the resulting network to predict each variable in each case of the holdout sample from the other variables in the case.

We scored these predictions using the log loss metric. Given a database of cases and a BN, the log loss for variable X_i is given by

$$\lambda_i(G, \tilde{\theta}_G) = \sum_k -\log\{p(x_{ik} | x_{(i)k}, G, \tilde{\theta}_G)\}, \quad (17)$$

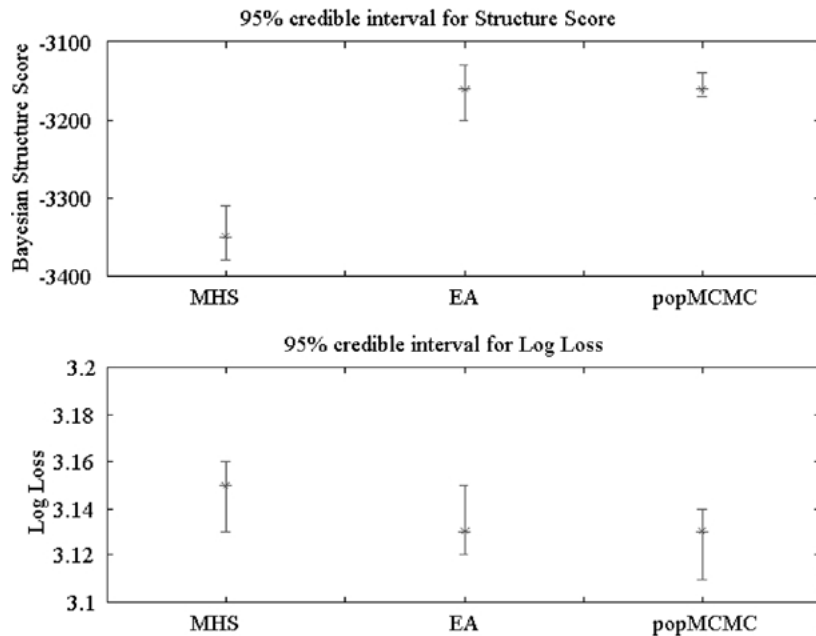


Figure 8. Structure scores and log loss for the three algorithms.

where k ranges over the cases in the holdout sample and $p(x_{ik} | x_{(i)k}, G, \tilde{\theta}_G)$ is the probability assigned by the BN $(G, \tilde{\theta}_G)$ to the value of variable i observed in case k , conditional on the values given in case k for the other variables. We added the log loss values for each variable to get an overall predictive score for the holdout sample.

From this figure it can be seen that the MHS algorithm has consistently lower final scores than the other algorithms. The average final score for the popMCMC is very near the average final score found for the EA, but the range of final scores is narrower. This would be expected if the EA had a greater tendency to become trapped in local basins of attraction.

When we consider predictive performance, the difference among algorithms is not as great, although the trend is the same.³ The credible intervals for average log loss for all three algorithms overlap considerably. This suggests that predictive performance is not too negatively impacted by using models that score somewhat less highly. The same pattern of results was obtained for the other data sets. The EA and popMCMC had better scores and better predictive performance than the standard MHS, but credible intervals for the latter overlapped considerably for all three algorithms.

There is evidence that at least in some applications averaging multiple models can improve predictive performance over use of the single best model found by the search process (Madigan & Raftery, 1994; Hoeting et al., 1996). We compared predictive performance of the single best model against predictive performance of a prediction constructed by averaging the predictions of all models in the population. Because the stationary distribution of an MCMC is a random sample from the posterior distribution of models, an equally weighted

average is the most appropriate way to combine models. For MHS and popMCMC, we found on average slightly better predictions from the averaged models than from the single best model, but the effect was not large enough to be statistically detectable from the small sample of 5 runs of each algorithm. There was no detectable difference in predictive performance with the EA, presumably because there was so little within-population variation in structures. More research is needed to determine the conditions under which statistically detectable improvements in predictive performance can be achieved by using multiple models.

5. Summary and discussion

All three algorithms examined in this study can be regarded as evolutionary algorithms according to the framework proposed by Back (1996). We compared a standard Metropolis-Hastings sampler and a standard evolutionary algorithm with a new hybrid algorithm called Population Markov Chain Monte Carlo. Like the EA, popMCMC exchanges information among solutions in a population. Like the MHS, popMCMC satisfies conditions ensuring ergodicity and convergence to the Boltzmann distribution on the given energy surface. On the problem of learning directed graphical models, or BNs, with missing observations and hidden variables, experimental results demonstrated that incorporating information exchange increased the rate of improvement in solutions from the initial solution, and that the MCMC algorithms had greater population diversity than the EA. The MCMC algorithms offer the additional advantages of explicit characterization of the stationary distribution and straightforward estimation of moments and other features of the stationary distribution.

Incorporation of ideas from EA research into the design of MCMC samplers may be a fruitful direction of research. In particular, there are useful concepts and results from the field of EAs in representation of solutions, transformation operators for given classes of problems, and how problem representation affects the choice of effective transformation operators. Other concepts that may prove useful are speciation, in which information exchange is limited to solutions that are similar to each other, and niching, in which solutions are adapted to particular modes of the fitness landscape. With appropriate modifications, some of these concepts might prove useful for improving the performance of MCMC algorithms for difficult learning and optimization problems.

Cross-fertilization in the other direction may also prove useful. Experimental results showed that population diversity increased dramatically when a post-selection step was incorporated that ensures that the transition distribution satisfies the balance condition implying convergence to the Boltzmann distribution. We conjecture that physics-inspired modifications to EAs might be useful in improving both exploration and exploitation. The balance condition is also useful theoretically for characterizing the stationary distribution of a sampler and for designing the sampler that converges rapidly to a given stationary distribution. The theory of non-equilibrium thermodynamics might be useful for analyzing the dynamic behavior of EAs designed to converge to specified Boltzmann distributions.

Experience has shown that hierarchical multi-resolution representations are key to intelligence and emergence of complexity. We conjecture that multi-level samplers in which

parameterized population models are used to inform sampling at lower levels may be generally useful for search, optimization and learning.

We also conducted experiments using crossover in a population of MH samplers. We replaced fitness-based pre-selection with Metropolis-Hastings post-selection. Results showed an increase in diversity, but convergence was no faster than the standard MCMC. Thus, turning an EA into a MHS does not automatically improve performance. A plausible explanation is that both fitness-based pre-selection in the EA and the statistical estimates in adaptive mutation have the effect of focusing sampling in promising directions. Our crossover-augmented MHS used random selection of individuals to cross over, and therefore did not benefit from this focusing effect. Moreover, the crossover operator incorporated information only from the two solutions participating in crossover. As discussed above, the improvement in performance of popMCMC over independent MHS appears to derive from the ability to incorporate statistical information from the entire population into the proposal distribution for each of the samplers. We are considering ways to incorporate crossover or other pair-based reproduction into popMCMC without sacrificing convergence speed.

Population based methods that evolve to a sample of independent draws from the posterior distribution are of general use in constructing statistical estimates of quantities from the posterior distribution. This approach is particularly useful if the objective is to draw inferences about the existence and direction of arcs in a BN, something that cannot be done with standard BN learning algorithms. The population of solutions can be treated as a sample of network structures drawn from the posterior distribution, and therefore the frequency with which an arc occurs in a particular direction is an estimate of the posterior probability of an arc in that orientation.

In conclusion, the results of our experimental comparison suggest that a marriage of ideas from EA and MCMC is a promising direction of research. We demonstrated this promise with a new hybrid algorithm that is a fixed transition MHS at the population level, an adaptive MHS at the individual solution level, and is an evolutionary algorithm according to Back's criteria. Our hybrid algorithm is only one of many promising possible approaches to combining physics and biology inspired sampling approaches.

Acknowledgments

The research reported in this paper was supported in part by the Defense Advanced Research Projects Agency under Contract # DACA-76-93-C-0025. The authors are grateful to Tod Levitt for many helpful discussions, and for useful comments and suggestions on an earlier draft of this paper.

Notes

1. The Gamma function is given by $\Gamma(\alpha) = \int_0^{\infty} u^{\alpha-1} e^{-u} du$. When α is an integer, $\Gamma(\alpha) = (\alpha - 1)!$
2. The independence sampler is a Metropolis-Hastings sampler in which the proposal distribution does not depend on the current state.
3. Note that lower log loss scores are better, whereas higher Bayesian Dirichlet scores are better.

References

- Back, T. (1996). *Evolutionary algorithms in theory and practice*. New York: Oxford University Press.
- Beinlich, I. A., & Suermondt, H. J., et al. (1989). The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proceedings of the Second European Conference on Artificial Intelligence in Medicine*.
- Cooper, G. F., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9, 309–347.
- Davis, T. E., & Principe, J. C. (1993). A Markov chain framework for the simple genetic algorithm. *Evolutionary Computation*, 1:3, 269–288.
- DeJong, K. A. (1975). An analysis of the behavior of a class of genetic adaptive systems. In *Computer and communication sciences* (p. 256). Ann Arbor, MI: University of Michigan.
- DeJong, K. A., & Spears, W. M. (1990). An analysis of the interacting roles of population size and crossover in genetic algorithms. In *Proceedings of the First International Conference on Parallel Problem Solving from Nature*, Dortmund, Germany.
- Feller, W. (1968). *An introduction to probability theory and its applications*. New York: Wiley.
- Fogel, L. J. (1991). *System identification through simulated evolution: A machine learning approach to modeling*. Needham Heights: Ginn Press.
- Friedman, N. (1998a). The Bayesian structural EM algorithm. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*. San Mateo, CA: Morgan Kaufmann.
- Friedman, N. (1998b). Learning belief networks in the presence of missing values and hidden variables. In *Fourteenth International Conference on Machine Learning (ICML-97)*. San Mateo, CA: Morgan Kaufmann.
- Gelman, A., Carlin, J., Stern, H., & Rubin, D. (1995). *Bayesian data analysis*. London: Chapman and Hall.
- Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7, 457–472.
- Geyer, C. J. (1991). Markov chain Monte Carlo maximum likelihood. In E. D. Keramidas (Ed.), *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface* (pp. 156–163). Fairfax Station: Interface Foundation.
- Gilks, W. R., Richardson, S., & Spiegelhalter, D. (Eds) (1996). *Markov chain Monte Carlo in practice*. London: Chapman and Hall.
- Gilks, W. R., Roberts, G. O., & Sahu, S. K. (1998). Adaptive Markov chain Monte Carlo through regeneration. *Journal of the American Statistical Association*, 93, 1045–1054.
- Goldberg, D. E., & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. In G. J. E. Rawlins (Ed.), *Foundations of genetic algorithms* (Vol. 1, pp. 69–93). San Mateo, CA: Morgan Kaufmann.
- Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:1, 97–109.
- Heckerman, D. (1996). *A tutorial on learning with Bayesian networks*. Redmond WA: Microsoft.
- Heckerman, D., Geiger, D., et al. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20, 197–243.
- Hoeting, J., Madigan, D., Raftery, A., & Volinsky, C. (1996). Bayesian model averaging, Technical Report #335, Department of Statistics, University of Washington, Seattle, WA.
- Holland, J. H. (1995). *Adaptation in natural and artificial systems*. Cambridge, MA: MIT Press.
- Holmes, C. C., & Mallick, B. K. (1998). *Parallel Markov chain Monte Carlo sampling: An evolutionary based approach*. London: Imperial College.
- Jensen, F. V. (2001). *Bayesian networks and decision graphs*. New York: Springer.
- Kass, R., & Raftery, A. (1995). Bayes factors. *Journal of the American Statistical Association*, 90:430, 773–795.
- Kitano, H. (1990). Designing neural networks using genetic algorithms with graph generation systems. *Complex Systems*, 4, 461–476.
- Larrañaga, P., & Poza, M., et al. (1996). Structure learning of Bayesian networks by genetic algorithms: A performance analysis of control parameters. *IEEE Journal on Pattern Analysis and Machine Intelligence*, 18:9, 912–926.
- Lauritzen, S. (1995). The EM algorithm for graphical association models with missing data. *Computational Statistics & Data Analysis*, 19, 191–201.

- Lauritzen, S. (1996). *Graphical models*. Oxford: Oxford Science Publications.
- Little, R., & Rubin, D. (1987). *Statistical analysis with missing data*. New York: John Wiley & Sons.
- Madigan, D., Raftery, A. E., et al. (1994). Strategies for graphical model selection. In P. Cheeseman, & W. Oldford (Eds.), *Selecting models from data: Artificial intelligence and statistics IV* (pp. 91–100). Berlin: Springer-Verlag.
- Madigan, D., & York, J. (1993). Bayesian graphical models for discrete data. Technical Report no. 259, Department of Statistics, University of Washington, Seattle, WA.
- Metropolis, N., Rosenbluth, A. W., et al. (1953). Equations of state calculation by fast computing machines. *Journal of Chemical Physics*, 21, 1087–1092.
- Neal, R. (1993). Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto.
- Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. San Francisco: Morgan Kaufmann.
- Roberts, G. O. (1996). The Gibbs sampler and Metropolis-Hastings algorithm. In W. R. Gilks, S. Richardson, & D. Spiegelhalter (Eds.), *Markov chain Monte Carlo in practice*. London: Chapman & Hall.
- Schwefel, H.-P. (1995). *Evolution and optimum seeking*. New York: John Wiley & Sons.
- Sewell, W., & Shah, V. (1968). Social class, parental encouragement, and educational aspirations. *American Journal of Sociology*, 73, 559–572.
- Spears, W. M. (1994). Simple subpopulation schemes. In *Proceedings of the Third Annual Conference on Evolutionary Programming*. San Diego: World Scientific.
- Spiegelhalter, D., & Lauritzen, S. (1990). Sequential updating of conditional probabilities on directed graphical structures. *Networks*, 20, 279–605.
- Syswerda, G. (1989). Uniform crossover in genetic algorithms. In *Proceedings of the 3rd International Conference on Genetic Algorithms*. San Mateo, CA: Morgan Kaufmann.
- Whittaker, J. (1990). *Graphical models in applied multivariate statistics*. Chichester: John Wiley & Sons.

Received July 17, 2000

Revised April 19, 2001

Accepted September 24, 2001

Final manuscript October 2, 2001