Bayesian Covariance Selection[*]

Adrian Dobra
Duke University,
Durham, NC 27708,
adobra@isds.duke.edu

Mike West
Duke University,
Durham, NC 27708,
mw@isds.duke.edu

July 28, 2004

Abstract

We present a novel structural learning method called HdBCS that performs covariance selection in a Bayesian framework for datasets with tens of thousands of variables. HdBCS is based on the intrinsic connection between graphical models on undirected graphs and graphical models on directed acyclic graphs (Bayesian networks). We show how to produce and explore the corresponding association networks by Bayesian model averaging across the models identified. We illustrate the use of HdBCS with an example from a large-scale gene expression study of breast cancer.

Keywords: Bayesian inference; Bayesian model averaging; Bayesian networks; Covariance selection; Directed acyclic graphs; Gene expression data; Graphical models.

1 INTRODUCTION

We are concerned with datasets d in which a large number p (e.g., tens of thousands) of variables are recorded and the sample size n is relatively small (e.g., tens or possibly hundreds of observations). Through suitable transformations, d can sometimes be assumed to roughly follow a multivariate Gaussian distribution $N_p(0, \Sigma)$. Directly attempting to fit this apparently simple model with $p(p+1)/2$ parameters represented by the entries of Σ raises challenging questions of structuring and dimensionality reduction in parameter space.

Dempster (1972) introduced the idea of reducing the number of parameters that need to be estimated by setting to zero selected elements of the precision matrix $\Omega = \Sigma^{-1}$. This can, and generally will, lead to more robust estimates of Σ if Ω is required to have a substantial number of structural zeros. In addition, the dependency patterns among the variables in d can be visually summarized by means of an undirected independence graph $\mathcal{G}$ in which each variable is associated with a vertex and the edges that link the vertices are the off-diagonal elements of Ω that are not constrained to be zero. The resulting Gaussian distribution satisfies a set of conditional independence relations encoded by $\mathcal{G}$. These relations are called the *pairwise*, *local* and *global Markov properties*, while the pair $M = (\Sigma, \mathcal{G})$ is called a *Gaussian graphical model*; see, for example, Lauritzen (1996). This model is undirected since the edges in $\mathcal{G}$ are lines that represent symmetric associations.

We are interested in performing covariance selection (Dempster, 1972) with the objective of identifying a number of Gaussian graphical models that are best supported by the data and, in a Bayesian framework, by the prior information available. Inference on the parameters of Σ (equivalently, Ω) can consequently be done by Bayesian model averaging (Madigan et al., 1996) across the pool of models selected.

[*] Adrian Dobra is Research Assistant Professor of Molecular Genetics & Microbiology and of Statistics & Decision Sciences, Duke University, Durham, NC 27708. Mike West is Arts & Sciences Professor of Statistics & Decision Sciences, Duke University, Durham, NC 27708. The research of Adrian Dobra was partially supported by the NIH on Grant HL73042-01. The research of Mike West was partially supported by the NSF on Grant NSF-DMS-0102227 and by the NIH on Grant HL73042-01. This work has benefited from discussions with our colleagues Beatrix Jones, Joseph R. Nevins, Jennifer Pittman, Quanli Wang and Guang Yao.

Searching for graphical models with tens of thousands of nodes is an extremely difficult task, statistically and computationally, due to the vast space of possible graphs that needs to be explored. The majority of the structural learning methods developed so far involve exploring the target space by sequentially adding (deleting) one or more edges to (from) the current graph. In the special case when decomposable graphs $\mathcal{G}$ are the only graphs considered, the search space is considerably reduced and there exist conjugate prior distributions for the parameters of $M = (\Sigma, \mathcal{G})$ (Giudici, 1996; Lauritzen, 1996) that lead to exact formulas for the marginal likelihood of M, $p(d|M)$. Unfortunately there are two major shortcomings that make decomposable graphs less desirable: (i) the learning procedure is slowed down by the need of determining what edges can be changed so that the resulting graph is still decomposable; and, much more importantly, (ii) the decomposability constraint is simply too severe to yield models that are representative for the complex dependency patterns that exist among the variables in d in other than rather small dimensional problems. In the most general case when all the possible graphs are considered, numerical or stochastic methods for approximately computing the posterior probability of M need to be employed (Roverato, 2002; Atay-Kayis and Massam, 2003; Dellaportas et al., 2004) which result in search procedures that cannot efficiently cover huge sets of graphs. For a comprehensive review of learning Gaussian graphical models in moderately large datasets, see Jones et al. (2004).

An alternative method of performing covariance selection is to exploit the connection between graphical models on undirected graphs and graphical models on directed acyclic graphs (DAGs, henceforth). The latter distributions follow the *order Markov property* relative to their underlying DAGs and further obey the Markov properties with respect to the moralized undirected versions of these DAGs (Lauritzen, 1996). A DAG is a convenient graphical structure that induces a recursive factorization of the joint density as a product of univariate regressions associated with each variable. Variables are linked in a DAG with arrows instead of lines. An arrow points from an explanatory variable (the parent) to the response (the child). The decomposition of a multivariate joint distribution induced by a DAG is a straightforward generalization of the usual chain rule and yields exact formulas for computing the marginal likelihood of the corresponding model (Heckerman and Geiger, 1995; Geiger and Heckerman, 2002). Thus DAG models have properties similar to those of graphical models on decomposable graphs. Actually, any decomposable graph can be transformed in a DAG using an ordering generating by the maximum cardinality search algorithm (Lauritzen, 1996) which implies that the class of decomposable graphs is included in the class $\mathcal{S}$ of undirected graphs that can be obtained by moralizing (i.e., ensuring an edge exists between the parents of each child) and replacing the arrows with undirected edges (Madigan et al., 1996).

Searching the space of DAGs can be done using local moves involving the addition, deletion or reversal of arrows. Unfortunately, methods based on local moves can spend much time traversing DAGs that are statistically equivalent (Heckerman et al., 1994). Two equivalent DAGs describe the same joint distribution and consequently the same Markov relations. Madigan et al. (1996) and Chickering (1995) present characterizations of equivalent DAGs and introduce search algorithms that “jump” between equivalence Markov classes. These search methods, although proven to be better than “simple” local moves-based algorithms, are, unfortunately, simply not efficient enough to scale to datasets with tens of thousands of variables.

Our aim is to present a novel framework for constructing high-dimensional Gaussian graphical models by searching for graphical models on DAGs. This approach builds on the methods introduced in Dobra and West (2004a) and is guaranteed to eventually converge to local optima in the space of undirected graphs $\mathcal{S}$, in the sense of identifying local modes in posterior distributions over $\mathcal{S}$ based on the dataset d.

2 PRIORS AND MARGINAL LIKELIHOODS FOR DAGS

Assume that the dataset d contains n samples $(x_{1j}, x_{2j}, \ldots, x_{pj})'$, $j = 1, 2, \ldots, n$, from a random normal vector $X = (X_1, X_2, \ldots, X_p)' \sim p(x) = N_p(0, \Sigma)$. Here $\Sigma = \{\sigma_{ij}\}$ is positive definite with precision matrix $\Omega = \Sigma^{-1} = \{\omega_{ij}\}$. The univariate regression models having each X_j as response and the remaining variables $X_{i \neq j}$ as covariates can be written in structural form as

$$x = \Gamma x + \epsilon, \tag{1}$$

where $\epsilon = (\epsilon_1, \ldots, \epsilon_p)'$ are the error terms for the p regressions and $\Gamma = \{\gamma_{ij}\}$ is a $p \times p$ matrix of regression coefficients.

This linear system can be transformed to define a multivariate normal distribution $p(x)$ as given by the chain rule by conditioning on an ordering T of the index set $V = \{1, 2, \ldots, p\}$. In this case Γ is an upper triangular matrix with zero diagonal elements and the regression errors terms are independent $\epsilon \sim N_p(0, \Psi)$ with $\Psi = \mathrm{diag}(\psi_1, \ldots, \psi_p)$.

For every $v \in V$, $T(v)$ represents the regression equation associated with X_v. The equation $T(v)$ is said to *explain* variable X_v while X_v is said to be *explained* in regression $T(v)$. Given an ordering T, the possible set of predictors of X_v is constrained to be $\pi(v, T) = \{i | T(i) > T(v)\}$. Since some of the variables X_i, $i \in \pi(v, T)$, might not be relevant for predicting X_v, introduce the indicator vectors $\xi^v = (\xi^v_1, \xi^v_2, \ldots, \xi^v_p)$ where $\xi^v_i = 1$ if X_i is selected in the regression model for X_v and $\xi^v_i = 0$ otherwise. The regression coefficient γ_{vi} is equal to zero if $i \notin \pi(v, T)$.

Let $\mathrm{pa}(v) = \{i | \gamma_{vi} = 1\}$ be the indices of the variables that appear in regression $T(v)$. The sets $\{\mathrm{pa}(v) | v \in V\}$ define a DAG $\mathcal{D}$ with vertex set V and arrows going from any $j \in \mathrm{pa}(v)$ to v. The variables X_i, $i \in \mathrm{pa}(v)$, are the parents of X_v in $\mathcal{D}$, while X_v is their child. The ordering T is the reverse of a well-ordering for $\mathcal{D}$; see, for example, Chickering (1995). Although T and ξ completely identify the sets $\{\mathrm{pa}(v)\}_v$, there might exist several orderings T that induce the same model $\mathcal{D} = \{\mathrm{pa}(v)\}_v$ for the same inclusion/exclusion patterns ξ.

It follows that the joint distribution $p(x)$ conditional on $\mathcal{D}$ can be written as:

$$p(x|\mathcal{D}) = p(x|T, \xi) = \prod_{v \in V} p(x_v | x_{\mathrm{pa}(v)}). \tag{2}$$

Let $\eta_v = \{\gamma_{vi} | i \in \pi(v, T)\} \cup \{\psi_v\}$, $v \in V$, be the non-overlapping parameter sets associated with the regression equations (1). Dobra and West (2004a), following the original ideas presented in Geiger and Heckerman (2002), showed that an encompassing inverse Wishart prior for the covariance matrix Σ induces independent, consistent normal/inverse gamma priors for the regression parameters $\{\eta_v\}$. This choice of priors leads to exact formulas for calculating the marginal likelihood $p(x_v|\xi^v)$ with respect to the parameters of the regression of X_v on $X_{\mathrm{pa}(v)}$. It follows that the marginal likelihood of d is given by:

$$p(d|\mathcal{D}) = \prod_{v \in V} p(x_v | \xi^v),$$

Across DAGs $\mathcal{D}$, the posterior of $\mathcal{D}$ is then given by

$$p(\mathcal{D}|d) \propto p(d|\mathcal{D}) p(\mathcal{D}). \tag{3}$$

In most of the graphical models literature, a uniform prior on the space of DAGs is preferred, but such a prior corresponds to uniform priors on the individual regression models which induce graphical models that often overfit the data since too many regressors are included in each equation. Parsimonious priors that induce sparsity – that say that each variable is expected to have a relatively small number of predictors – are often more relevant. One class of such priors is constructed by assuming that $\{\xi^v_j | j, v \in V\}$ are *apriori* independent within and across regressions with $p(\xi^v_j = 1) = 1 - p(\xi^v_j = 0) = \beta$ (Chipman et al., 2001). Here β is chosen to be small since only a few covariates are expected to be present in each regression. The prior probability of a 0/1 pattern ξ^v that defines the regression of X_v on $X_{\mathrm{pa}(v)}$ is then

$$p(\xi^v) \propto [\beta/(1 - \beta)]^{\#\,\mathrm{pa}(v)}.$$

It follows that the prior weight of any DAG $\mathcal{D}$ is:

$$p(\mathcal{D}) = p(\xi) = \prod_{v \in V} p(\xi^v) \propto [\beta/(1 - \beta)]^{\#\mathcal{D}}, \tag{4}$$

where $\#\mathcal{D} = \sum\limits_{j=1}^{p} \#\,\mathrm{pa}(j)$ represents the number of directed edges in $\mathcal{D}$.

The prior in (4) gives the same weight to equivalent DAGs since two such DAGs are required to have the same skeleton, i.e. the undirected graph obtained by removing directions (Chickering, 1995). Moreover, if the priors $\{\pi(\eta_v)\}_{v \in V}$ for the regression parameters are induced by an inverse Wishart prior on the covariance matrix Σ as described above, any two equivalent DAGs have the same marginal likelihood (Geiger and Heckerman, 2002; Heckerman et al., 1994). Consequently, two equivalent DAGs receive the same posterior weight (3) calculated as the product of contributions for each of the regression equations in the compositional representation.

3 STRUCTURAL LEARNING ON DAGS

A Bayesian model selection criterion is to search for DAGs with large scores given by their posterior probabilities (3); see, for example, Chipman et al. (2001). In this section we describe a novel procedure – termed HdBCS (High-dimensional Bayesian Covariance Selection) – that identifies high-scoring DAGs. Our method has three distinct stages, as follows. The first step determines a restricted set of candidate predictors for the univariate regression associated with each variable. The second step consists of a heuristic for finding DAGs with large posterior probabilities. At the third step, these models are sequentially improved until convergence to local optima. Earlier versions of the first two steps can be found in Dobra et al. (2004a).

HdBCS is specifically designed to exploit the architecture of a parallel computing environment since it has to handle datasets with tens of thousands of variables. We comment on the efficiency of our method (e.g., number of compute nodes required, increase in the search speed if more resources are allocated to the program) as each step is presented. A full implementation is freely available for download as a C++ package (Dobra, 2004) that makes extensive use of Message Passing Interface (MPI) libraries.

3.1 First step of HdBCS

The main idea behind HdBCS is to produce good regression models for each variable and to combine these models in a joint multivariate normal distribution using the chain rule (which is essentially equivalent to constructing a DAG or a Bayesian network). It is not reasonable to believe that any variable can be a possible predictor for any other variable. Continuously searching through a large number of predictors can be extremely costly (if not prohibitive) in terms of computing time. The search can be speed up by several orders of magnitude by selecting a relatively rich (e.g., hundreds) set of possible predictors for each variable instead of repeatedly performing regression model selection with tens of thousands of candidate covariates.

We want to identify, for each $v \in V$, a subset of variables $\mathcal{V}^v \subset V \setminus \{v\}$ that are the most likely variables to have strong associations with X_v. Although it would be desirable to have sets $\mathcal{V}^v$ that are as small as possible, we need to make sure that, once we further restrict the set of possible predictors by conditioning on an ordering of V, most if not all of the variables still have a non-empty list of potential predictors.

We move in the space of regression models for X_v using Gibbs sampling (George and McCulloch, 1993; Gelfand et al., 1990). Stochastic algorithms, such as Gibbs sampling, for visiting candidate models lead to much better results than deterministic methods (e.g., forward/backward selection). The huge number of candidate predictors prohibit the use of Gibbs sampling to assess the importance of each predictor or to make inferences on the regression parameters. In this context Gibbs sampling can only be run for a relatively small number of iterations.

We start the search at a random regression model ξ^v for X_v for each v. We set $\mathcal{V}^v \leftarrow \emptyset$ and denote

$$\xi^v_{(j)} = (\xi^v_1, \ldots, \xi^v_{j-1}, \xi^v_{j+1}, \ldots, \xi^v_p).$$

The vector ξ^v is updated component-wise by sequentially sampling in a random order, for each $j \in V \setminus \{v\}$, from a Bernoulli distribution with probability

$$p(\xi^v_j = 1 | \xi^v_{(j)}) = \frac{a}{a+b},$$

where $a = p(x_v | \xi^v_{(j)}, \xi^v_j = 1) p(\xi^v_{(j)}, \xi^v_j = 1)$, and $b = p(x_v | \xi^v_{(j)}, \xi^v_j = 0) p(\xi^v_{(j)}, \xi^v_j = 0)$. Gibbs sampling tends to move towards models with large posterior weight $p(x_v|\xi^v)p(\xi^v)$. Every time a variable X_j ($j \neq v$) is included in a regression, we add it to the set of predictors: $\mathcal{V}^v \leftarrow \mathcal{V}^v \cup \{j\}$. A Gibbs sampling iteration is a full cycle through all the possible predictors in $V \setminus \{v\}$.

This step of HdBCS can be run independently for each variable and hence it is perfectly suitable for parallel computing on a cluster. The time required for each iteration of Gibbs sampling is a function of the ratio between the total number of predictor variables and the number of available processors. Since, in our case, the number of predictor variables may be of the order of tens of thousands, we cannot run Gibbs sampling for many iterations irrespective of the number of available processors, thus there exists the danger that we will not visit potential strong predictors.

Fortunately, much computation can be saved by exploiting the duality between being a predictor and a predictee. If X_v is in the set of possible predictors for another variable X_{v_1} (i.e., $v \in \mathcal{V}^{v_1}$), then v_1 is also included in the set of possible predictors for v: $\mathcal{V}^v \leftarrow \mathcal{V}^v \cup \{v_1\}$. This means that each $\mathcal{V}^v$ is enriched from the searches performed for all the variables in the dataset and the resulting predictor sets $\mathcal{V}^v$, $v \in V$, are consistent with each other. As a consequence, Gibbs sampling can safely be run for hundreds instead of thousands of iterations to select relevant sets of predictors.

3.2 Second step of HdBCS

This step consists of a heuristic for finding high-scoring DAGs that was first suggested in Dobra et al. (2004a) and has its roots in the dependency networks model introduced by Heckerman et al. (2000). Initial regressions given by $\{\xi^v\}_v$ are generated by employing Gibbs sampling to visit candidate models with regressors in $\{\mathcal{V}^v\}_v$ – the sets of potential predictors determined at the first step of HdBCS. Reducing the available predictors for each $v \in V$ from $V \setminus \{v\}$ to $\mathcal{V}_v$ leads to a significant decrease in the running time necessary to find the starting regressions.

An ordering T is sequentially produced such that the indicator vectors $\{\xi^v\}_{v \in V}$ define a DAG $\mathcal{D}$. Let $C \subset V$ be the indices of the variables that have not been ordered yet. For each such variable X_v, $v \in C$, assign its *explanatory score*:

$$s_v^C = \prod_{j \in C} p(x_j | \xi^j_{(v)}, \xi^j_v = 0) p(\xi^j_{(v)}, \xi^j_v = 0).$$

This score is calculated by removing X_v from all the regressions in which X_v appears as a predictor and by taking the product over the posterior probabilities of the resulting regressions. The smaller the score s_v^C, the greater is the "importance" of X_v as a predictor for the other variables. At each iteration, the next variable in the ordering that is being produced is that variable with the smallest contribution as an explanatory variable for the remaining variables.

Set iterate counter $h = 0$ and the candidate variable index set to $C = V$. For each $h = 1, 2, \ldots, p-1$ repeat the following:

1. Sample a variable $v \in C$ according to probabilities proportional to $\left(s_j^C\right)^a$, $j \in C$, for some annealing parameter $a > 0$. Variables with a larger score s_j^C are more likely to be selected.

2. Set $T(h) = v$, i.e. X_v becomes the next variable in the ordering.

3. Remove v from the candidate variable index set by an update of C to $C \setminus \{v\}$.

4. Update the regressions ξ^j for all $j \in C$ such that $\xi^j_v = 1$ by using Gibbs sampling to visit models and by selecting the model with the highest posterior probability. In other words, we attempt to determine new models for the variables that had X_v selected as a regressor. The set of candidate predictors for each $j \in C$ is $C \cap \mathcal{V}^j$.

5. Update the explanatory scores s_j^C for $j \in V$.

At $h = p$, there is a single variable remaining in C that completes the ordering T. We note that this heuristic procedure is not fully parallelizable. Although finding initial regressions can be done independently for each variable, the rest of the procedure is inherently sequential as the main loop that assigns an ordering step-by-step cannot be avoided. The parallel component of the algorithm is represented by the updates of the regression models after removing a variable from the set of possible predictors.

It is important to point out that Gibbs sampling gives a satisfying level of randomization of the entire procedure that leads to different DAGs when the method is applied repeatedly. Thus multiple models can be generated by running this heuristic a number of times.

3.3 Third step of HdBCS

The sets of DAGs found at the second step of HdBCS provide starting points for a procedure that converges to local modes of the posterior distribution over the space of DAGs. Assume that such a DAG is induced by an ordering T of V and by the indicator vectors $\{\xi^v\}_v$. There are p regressions numbered $1, 2, \ldots, p$ because each variable X_v has a regression associated with it even if this equation has no covariates.

The regression equation j explains variable X_v with $v = T^{-1}(j)$. The ordering T reduces the set of possible predictors for X_v from $\mathcal{V}^v$ to $\mathcal{V}^v \cap \pi(v, T)$. Ideally, every X_v would like to access those variables in $\mathcal{V}^v$ that induce the best linear model for X_v, but some of these covariates might unavailable due to the ordering T. Conditioning on a certain ordering can be seen as a competing process among the variables in V to "obtain" their best predictors. Since the posterior weight of a DAG is the product of the posterior weights of each regression model, "better" regression models for each variable translate directly into "better" DAGs. Therefore the key to finding better DAGs is to find improved orderings that make available as covariates the best predictors for each variable.

The orderings are improved by defining a set of local moves on $\mathcal{S}(V)$ – the set of permutations of V. Take two equations $j_1, j_2 \in V$ ($j_1 < j_2$) with $T(v_1) = j_1$ and $T(v_2) = j_2$. Consider the ordering T' with $T'(v_1) = j_2$, $T'(v_2) = j_1$ and $T'(v) = T(v)$ if $v \in V \setminus \{v_1, v_2\}$. The move in $\mathcal{S}(V)$ that transforms T into T' could change the set of available predictors for all the variables X_v such that $j_1 \leq T(v) \leq j_2$ in the following way: X_{v_2} is no longer a possible predictor for X_v, while X_{v_1} could enter the set of predictors for X_v (this happens only if $v_1 \in \mathcal{V}^v$). In order to find a good DAG given the new ordering T', we would have to update at most $(j_2 - j_1 + 1)$ regressions; this will be computationally expensive if j_1 and j_2 are far from each other. Moreover, we need a way to choose pairs of equations (j_1, j_2) that lead to models with higher posterior weight.

For each equation $j \in V \setminus \{p\}$, we define a local move that switches equations j and $j + 1$. In the previous notation, we take $j_1 = j$ and $j_2 = j + 1$. There are only two variables whose set of available predictors change: X_{v_2}, $v_2 = T^{-1}(j + 1)$, is removed from the set of predictors for X_{v_1}, $v_1 = T^{-1}(j)$, while X_{v_1} becomes available as a possible predictor for X_{v_2} – again, only if $v_1 \in \mathcal{V}^{v_2}$. Using Gibbs sampling, we determine updated regression models for X_{v_1} and X_{v_2}. The score $s(j)$ for equation j is defined as the ratio of, in the numerator, the product of the posterior weights of the updated regressions j and $j + 1$, and, in the denominator, the product of the posterior weights of the original regressions j and $j + 1$. Since the other regressions have not changed, $s(j)$ also represents the ratio of the posterior weights of the updated and original DAGs.

The search method we propose is now as follows. Calculate the scores $s(j)$ for every equation $j \in V$. The $p - 1$ possible switches (or local moves) define a set of neighbors in $\mathcal{S}(V)$ for the current ordering. The updated regression models corresponding to these scores are also retained. Choose to make the switch (i.e., move to a new ordering and hence to a new DAG) by sampling from the discrete distribution induced on $\{1, 2, \ldots, p - 1\}$ by the scores $s(j)$. More specifically, the weight associated with regression equation j is proportional to $s^a(j)$, where $a > 0$ is an annealing constant. The larger the posterior weight of a DAG that corresponds to a switch, the more likely it is to choose the regression that induced that switch. This stochastic search method always moves in the space $\mathcal{S}(V)$; that is, a proposed move is never rejected.

Assume we chose the switch corresponding to the regression j_0 and have updated the two regression models associated with $X_{T^{-1}(j_0)}$ and $X_{T^{-1}(j_0+1)}$. We have to update the scores for at most three equations: $j_0 - 1$, j_0 and $j_0 + 1$. Since this corresponds to searching with Gibbs sampling for at most six new regression models (two for each equation), the updating can be done fast and hence moving in the space of DAGs is also very fast. One could argue that the regression models for the updated equation j_0 do not have to be searched again since we can simply revert back to the previous "old" models. However, we want to perform the search again because new models might be found by Gibbs sampling which could lead to more highly weighted DAGs.

This procedure is well-suited for a parallel implementation. Computing the scores associated with each regression can be done independently on separate processors. After this initial computation, only six processors are needed to update and determine the scores for the regressions whose "neighbors" were affected by the switch. We emphasize that this procedure is guaranteed to improve the orderings and hence to lead to better and better models in terms of posterior probabilities over DAGs $\mathcal{D}$.

4 BAYESIAN MODEL AVERAGING

We account for model uncertainty by selecting from the set $\mathcal{B}$ of all models generated only the DAGs with posterior probabilities comparable with the posterior probability of $\mathcal{D}_0$, the highest scoring model identified, in the sense of restricting to:

$$\mathcal{A} = \{\mathcal{D} \in \mathcal{B} : p(\mathcal{D}|d) \geq Cp(\mathcal{D}_0|d)\}, \tag{5}$$

where $0 < C < 1$ is usually a small constant whose increase leads to a decrease of the number of models selected. This is one of the two principles that underlie the Occam's Window algorithm of Madigan and Raftery (1994). The second Occam's Window principle that excludes less parsimonious models with small posterior weights is already taken into account by the prior probability (4) assigned to each model.

The set of models $\mathcal{A}$ might still contain DAGs that correspond to the same Markov equivalence class and hence to the same joint multivariate normal distribution. In order to identify such models, we transform each DAG $\mathcal{D}$ in $\mathcal{A}$ in the corresponding equivalence class representatives called PDAGs (Chickering, 1995; Chickering, 2002) or essential graphs (Andersson et al., 1997). If the same PDAG is created more than once, its duplicates are removed from $\mathcal{A}$. The arrows in a DAG that can be reversed without changing the implied joint distribution are replaced by a line in the associated PDAG, while the other arrows are left unchanged. Therefore a PDAG contains a combination of arrows and lines that translates into a probability of an undirected symmetric association and a probability of a directed forward or backward association for each pair of variables (Madigan et al., 1996). These probabilities are calculated by adding the posterior probabilities of the PDAGs in $\mathcal{A}$ that contain a line, a forward arrow, a backward arrow or no link between two variables.

A network of pairwise associations among the variables X_v, $v \in V$, can be created by displaying the most probable link identified for each pair $(v, v') \in V \times V$. An alternative way to construct an association network using the DAGs $\{\mathcal{D}_0, \mathcal{D}_1, \ldots, \mathcal{D}_m\}$ from (5) is to consider the independence graph $\mathcal{G} = (V, E)$ where the set of undirected edges are the non-zero off-diagonal elements of the precision matrix Ω corresponding to the implied mixture over graphs

$$\sum_{i=0}^{m} \pi_i N_p(0, \Sigma_i), \tag{6}$$

with $\pi_i = p(\mathcal{D}_i|d)/[\sum_{j=1}^{m} p(\mathcal{D}_j|d)]$. Here $N_p(0, \Sigma_i)$ is the distribution induced by $\mathcal{D}_i$ that further defines a linear system $x = \Gamma_i x + \epsilon$ as in (1), where Γ_i is upper triangular with zero diagonal elements and $\epsilon \sim N_p(0, \Psi_i)$. The prior weight of $\mathcal{D}_i$ from (4) ensures for small values of β that the predictor sets of each regression model involves a relatively small number of regressors, which means that the lower triangular matrix $\mathrm{L}_i = (\mathrm{I}_p - \Gamma_i)'\Psi_i^{-1/2}$ is sparse. This gives us immediate access to the precision matrix Ω_i through the implied Cholesky decomposition: $\Omega_i = \mathrm{L}_i\mathrm{L}_i'$. Although the full covariance matrix $\Sigma_i = (\mathrm{L}_i^{-1})'\mathrm{L}_i^{-1}$ may not be sparse, covariance matrices of subsets of variables $U \subset V$ can readily be obtained via a simple matrix multiplication:

$$\Sigma_{i;U} = \mathrm{S}_{i;U}'\mathrm{S}_{i;U}, \tag{7}$$

where $\mathrm{S}_{i;U}$ are the rows of L_i^{-1} corresponding to the variables in U.

Quantities of interest can be evaluated by directly sampling from the posterior mixture (6) and the corresponding posteriors for the Σ_i, and then averaging across the samples obtained. For example, if we are interested in the precision matrix Ω of the mixture (6) or in the variance-covariance matrix Σ_U, $U \subset V$, we need to proceed as follows:

1. Sample a model $\mathcal{D}_{i_0}$ using the weights $\{\pi_0, \pi_1, \dots, \pi_m\}$.
2. Sample the regression parameters from the corresponding closed form posteriors over all regressions in the implied compositional DAG (Dobra et al., 2004a) to obtain the matrices $(\Gamma_{i_0}, \Psi_{i_0})$.
3. Calculate $\mathrm{L}_{i_0} = (\mathrm{I}_p - \Gamma_{i_0})' \Psi_{i_0}^{-1/2}$, producing the posterior sampled value of Ω as simply $\mathrm{L}_{i_0}\mathrm{L}_{i_0}'$.
4. Take the inverse of the lower-triangular matrix L_{i_0} and compute Σ_U as in (7).

From a posterior sample, we can then compute, for example, the corresponding Monte Carlo approximation to $E(\Omega|d)$ by simple averaging element-wise. Similar concepts apply to $E(\Sigma_U|d)$. By exploring the posterior samples for Ω we can identify the approximate posterior probabilities of inclusion of any edge.

The estimated precision matrix Ω leads to an independence graph $\mathcal{G}$ with tens of thousands of vertices. This addresses inference on graph structure. The consequent issue of producing visual representations (images) of $\mathcal{G}$, is extremely challenging when p is high, due to the immense number of variables and edges present in the graph. One strategy is to explore and graphically render relevant smaller portions of $\mathcal{G}$. More specifically, given a subset of variables $U \subset V$, we would like to identify another subset $U' \subset V$ of variables that are related with the variables in U and graphically produce the subgraph associated with $U \cup U'$. Such a subgraph may contain tens or hundreds of variables, but its generation is possible in practice using tools such as *GraphExplore* (Wang et al., 2004).

A straightforward solution is to include in U' variables found on shortest paths of length 2,3,4,. . . that connect pairs of variables in U – see, for example, Zhou et al. (2002). Such an approach based on graph theory alone has the disadvantage that there is no consistent way to rank the relative relevance of the variables in U' with respect to the target variables in U. In addition, it is not clear how to interpret the significance of the length of a path in a coherent statistical manner because two variables X_{v_1} and X_{v_2} might be connected by an edge in the independence graph $\mathcal{G}$ even if they are only weakly correlated. This could happen if X_{v_1} and X_{v_2} are both predictors in the regression model associated with another variable X_v, but X_{v_2} (X_{v_1}) is not a predictor in the regression model for X_{v_1} (X_{v_2}); in such a case, the edge between X_{v_1} and X_{v_2} (actually, between v_1 and v_2) is generated through the process of “moralization” (Lauritzen, 1996; Madigan et al., 1996).

To focus on statistical relevance of paths, Jones and West (2004) have defined a decomposition of the covariance between two variables X_{v_1} and X_{v_2} into weights that measure the strengths of the relationships between variables along paths that link X_{v_1} and X_{v_2} in $\mathcal{G}$. Their construction makes use of the covariance matrix of the best DAG $\mathcal{D}_0$ to determine the variables that are most relevant in mediating correlation between X_{v_1} and X_{v_2}. This decomposition arises in any given graph and so may be explored across multiple graphs sampled, as here, from the approximate posterior over graphs. Developing this idea methodologically is an open research question currently.

A specific, direct method of evaluating paths is as follows. Consider a variable X_v with $v \notin U$. The variance covariance matrix associated with $\{v\} \cup A$, that is,

$$\left[\begin{array}{cc} \sigma_{vv} & \Sigma_{vU} \\ \Sigma_{Uv} & \Sigma_{UU} \end{array} \right],$$

can be estimated using Bayesian model averaging as we have already described. Here $\Sigma_{Uv} = \Sigma_{vU}'$, $\Sigma_{UU} = \Sigma_U$, while σ_{vv} represents the variance of X_v. The percent of the variance of X_v that is explained by the variables in U is given by:

$$\rho(v|U) = 100 * \left(\Sigma_{vU}\Sigma_{UU}^{-1}\Sigma_{vU}'\right)/\sigma_{vv}. \tag{8}$$

The score $\rho(v|U)$ in (8) is a generalized version of correlation coefficient between X_v and X_U and it is equivalent to the familiar R^2 statistic from simple linear regression models. Higher values of $\rho(v|U)$ indicate that the pattern of variation of X_v is closely related to the patterns of variation of the variables X_U. Therefore we can order the variables in $V \setminus U$ in decreasing order with respect to (8); the set U' is taken to be the top variables in the resulting list.

The following specific example shows that the variables in U' connect most if not all variables in U through paths of various lengths in $\mathcal{G}$. In this example, we simply evaluate the Monte Carlo approximation to the posterior mean of $\rho(v|U)$ in (8).

5 EXAMPLE

We illustrate illustrate our methods through an analysis of gene expression data from breast cancer samples obtained at biopsy at the Koo Foundation Sun Yat-Sen Cancer Center in Taipei. This dataset is available as supplemental material in Huang et al. (2003). Gene expression assays were performed on the Affymetrix Human U95Av2 GeneChip. Expression intensities were produced from the resulting CEL files using Robust Multi-chip Average (Bolstad et al., 2003). After the removal of the control probes, we have $p = 12,558$ genes as variables.

We employed HdBCS to search for sparse Gaussian graphical models fitted to the (centered and scaled) expression data on $n = 89$ samples. At the first step of HdBCS we ran Gibbs sampling for 250 iterations with a burn-in time equal of 25 iterations. The prior probability of inclusion of a variable as a predictor in a regression model was taken to be $\beta = 1/(p-1)$. Figure 1 shows the reduction in the number of potentially relevant predictors as percentages relative to the initial number of predictors ($12,557$ in this case). All the variables retain fewer than 5% of the possible explanatory covariates. We chose to generate five models at the second step of HdBCS, and these models were further improved in $6,000,000$ iterations at the third step of HdBCS. The number of Gibbs sampling iterations performed to determine updated regression models after each equation switch was sequentially increased from 100 to 350 iterations with a constant burn-in time of 25 iterations. The annealing parameter used to smooth the weights associated with each regression was sequentially increased from 1 to 3. Each regression model was constrained to have no more than ten explanatory variables.

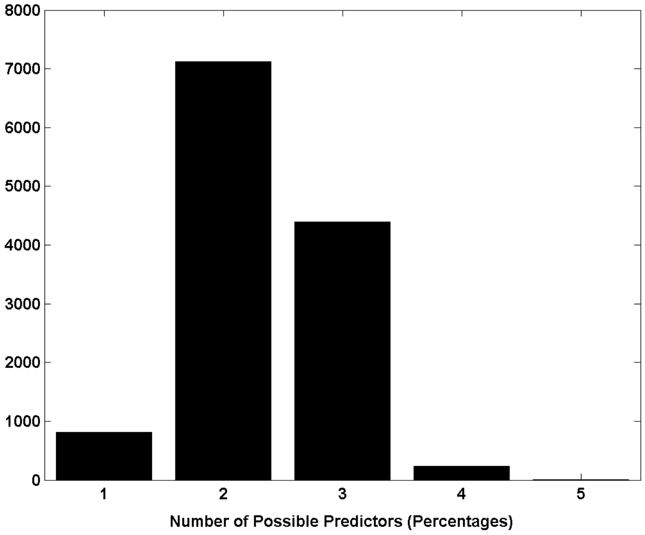

Figure 1: Reduction in the number of possible predictors of the $12,558$ genes/variables as a result of the first step of HdBCS. The plot gives the frequencies of the number of possible predictors as percentages.

The left-hand panel of Figure 2 shows the posterior probabilities of the DAGs visited at the last $5,000,000$ iterations from each of the five starting points. HdBCS significantly improves the scores of the DAGs it generates by more than $2,500$ units on a log-scale. The right-hand panel of Figure 2 gives the probabilities of the last $250,000$ DAGs generated in the run that leads to the "best" models. Here the scores of the DAGs do not constantly increase as they did in the beginning of the run. We decided to stop the third step of HdBCS because a number of 162 DAGs with distinct PDAGs and comparable posterior weights assessed by taking $C = 0.001$ in (5) were identified.

We employed Bayesian model averaging to estimate the precision matrix corresponding to the mixture of multivariate normal distributions associated with the 162 DAGs. The corresponding independence graph $\mathcal{G}$ on

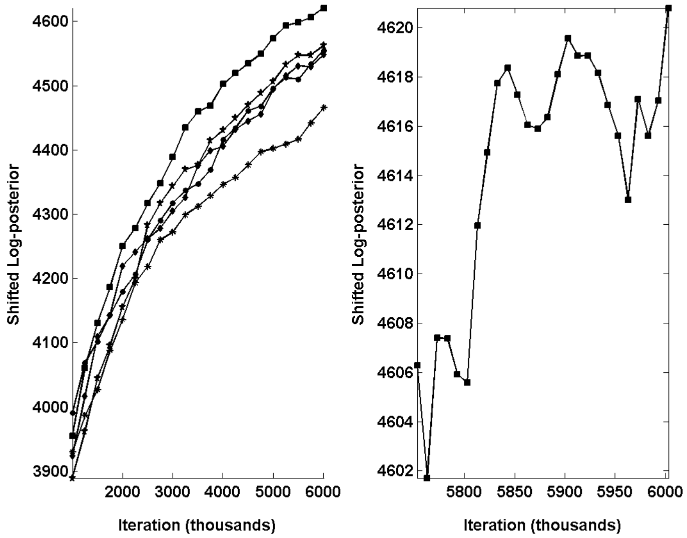

Figure 2: Posterior probabilities of the DAGs generated by HdBCS from five different starting points. The left-hand panel shows the last $5,000$ iterations while the right-hand panel shows the last 250 iterations. The posterior probabilities are represented on a log scale and are shifted with respect to the smallest weight of a DAG included in the plot.

$12,558$ genes has edges given by the non-zero off-diagonal elements of the precision matrix as estimated by its Monte-Carlo posterior mean.

Assume that we want to determine the genes whose expression patterns are related to the expression patterns of GATA3 and FOXA1 (also known as HNF3A). These genes encode transcription factors and are known to be expressed in closed association with the estrogen receptor-α gene (Lacroix and Leclercq, 2004). We estimated the multiple correlation coefficient (8) of each gene with respect to GATA3 and FOXA1. The sub-graph of $\mathcal{G}$ corresponding to these target genes and some of the remaining genes with the largest values of the estimated multiple correlation coefficient is given in Figure 3. This sub-graph is connected, as expected. It is important to point out the presence of two oligonucleotide sequences associated with XBP1 (XBP_1 and XBP_2 in Figure 3). XBP1 is a gene that encodes a transcription factor and is also known to have strong expression association with estrogen receptor-α (Lacroix and Leclercq, 2004), thus there is no surprise that XBP1 shows up in this context; the appearance of two probe sets for XBP1 simply reflects the collinearity of expression levels of these two representatives of the gene on the Affymetrix array. Any of the genes in Figure 3 has the potential to play a role in the estrogen receptor-α pathway; one of the key uses of these models and methods applied to such data is the generation of clues to biological function of identified genes (Dobra et al., 2004a; DeLong et al., 2004).

6 DISCUSSION

This works represents advances in our capacity to understand and perform structural learning of Gaussian graphical models with tens of thousands of variables. We describe how to identify a set of relevant models and how to eliminate those models that lead to the same joint distribution. By employing Bayesian model averaging we account for the inherent uncertainties introduced by conditioning on a particular graphical structure (DAG) and produce estimated networks of association between the variables involved.

We also describe a constructive, model-based method for exploring these networks and producing relevant sub-

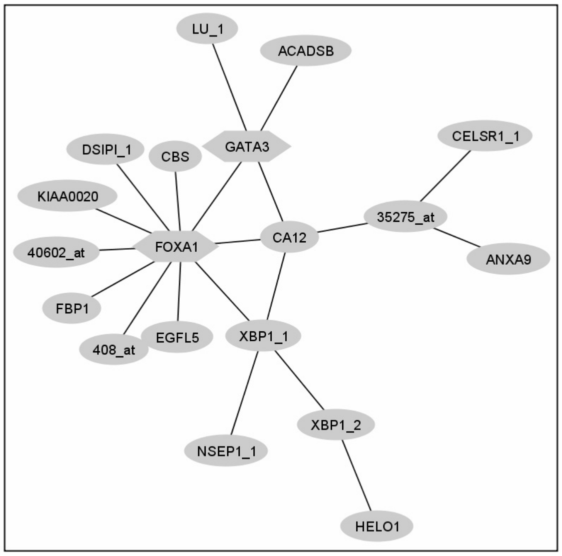

Figure 3: Sub-graph of $\mathcal{G}$ showing the expression associations between two target genes, GATA3 and FOXA1, and the genes with expression levels "explained" by the expression levels of the target genes. These genes (ellipses) are located on paths of various lengths between the target genes (hexagons). This image was produced using *GraphExplore* (Wang et al., 2004).

graphs. This method is based on a score which we call the multiple correlation coefficient that can be extended to a coherent method for generating *overlapping* clusters – see Dobra et al. (2004b). The variables in each of these clusters can be strongly correlated with each other or they can be strongly correlated with a subset of variables in the same group. This clustering algorithm turns out to have a very special significance in the context of one of our key motivating applications – gene expression data analysis – since it identifies and groups genes that may well share biological functional relationships. In this area, the approach offers potential for display and interrogation of associations among genes in useful and informative ways.

Finally, the approach – including model search, summary and visualization – has been implemented in software that is available (Dobra, 2004; Dobra et al., 2004b; Wang et al., 2004).

7 ACKNOWLEDGMENTS

Computations were performed in part on a Beowulf cluster under grant support from NSF and NIH, and with administrative support from the Center for Computational Science, Engineering, and Medicine at Duke University.

References

Andersson, S. A., Madigan, D., and Perlman, M. D. (1997). "A Characterization of Markov Equivalence Classes for Acyclic Digrahs." *Annals of Statistics*, 25, 505–541.

Atay-Kayis, A. and Massam, H. (2003). "A Monte Carlo Method to Compute the Marginal Likelihood in Nondecomposable Graphical Gaussian Models." Manuscript.

Bolstad, B. M., Irizarry, R. A., Astrand, M., and Speed, T. P. (2003). "A Comparison of Normalization Methods for High Density Oligonucleotide Array Data Based on Bias and Variance." *Bioinformatics*, 19, 185–193.

Chickering, D. M. (1995). “A Transformational Characterization of Equivalent Bayesian Network Structures.” In *Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence*, 87–98. Morgan Kaufmann, San Francisco.

— (2002). “Learning Equivalence Classes of Bayesian-network Structures.” *Journal of Machine Learning Research*, 2, 445–498.

Chipman, H., George, E. I., and McCulloch, R. E. (2001). “The Practical Implementation of Bayesian Model Selection.” In *Model Selection*, ed. P, Lahiri, Vol. 38 of *IMS Lecture Notes Monograph Series*, 67–116. Institute of Mathematical Statistics.

Dellaportas, P., Giudici, P., and Roberts, G. (2004). “Bayesian Inference for Non-decomposable Graphical Gaussian Models.” *Sankhya, Series A*. To appear.

DeLong, M., Yao, G., Wang, Q., Dobra, A., Black, E. P., Chang, J. T., Bild, A., West, M., Nevins, J. R., and Dressman, H. (2004). “DIG – A System for Gene Annotation and Functional Discovery.” Manuscript submitted for publication.

Dempster, A. P. (1972). “Covariance Selection.” *Biometrics*, 28, 157–175.

Dobra, A. (2004). “HdBCS: Bayesian Covariance Selection in High Dimensions.” Available for download at http://www.stat.duke.edu/∼adobra/hdbcs.html.

Dobra, A., Hans, C., Jones, B., Nevins, J. R., and West, M. (2004a). “Sparse Graphical Models for Exploring Gene Expression Data.” *Journal of Multivariate Analysis*, 90, 196–212.

Dobra, A., Wang, Q., and West, M. (2004b). “Graphical Model-based Gene Clustering and Metagene Expression Analysis.” ISDS Discussion Paper #04-24.

Geiger, D. and Heckerman, D. (2002). “Parameter Priors for Directed Acyclic Graphical Models and the Characterization of Several Probability Distributions.” *The Annals of Statistics*, 30, 1412–1440.

Gelfand, A. E., Hills, S. E., Racine-Poon, A., and Smith, A. F. M. (1990). “Illustration of Bayesian Inference in Normal Data Models Using Gibbs Sampling.” *Journal of the American Statistical Association*, 85, 972–985.

George, E. I. and McCulloch, R. E. (1993). “Variable Selection via Gibbs Sampling.” *Journal of the American Statistical Association*, 88, 881–889.

— (1997). “Approaches for Bayesian Variable Selection.” *Statistica Sinica*, 7, 339–373.

Giudici, P. (1996). “Learning in graphical Gaussian models.” In *Bayesian Statistics 5*, eds. J. M, Bernardo, J. O, Berger, A. P, Dawid, and A. F. M, Smith, 621–628. Oxford University Press.

Heckerman, D., Chickering, D. M., Meek, C., Rounthwaite, R., and Kadie, C. (2000). “Dependency Networks for Infererence, Collaborative Filtering, and Data Visualization.” *Journal of Machine Learning Research*, 1, 49–75.

Heckerman, D. and Geiger, D. (1995). “Likelihood and Parameter Priors for Bayesian Networks.” Tech. MSR-TR-95-54. Microsoft Research.

Heckerman, D., Geiger, D., and Chickering, D. M. (1994). “Learning Bayesian Networks: the Combination of Knowledge and Statistical Data.” In *Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence*, 293–301. Morgan Kaufmann, San Francisco.

Huang, E., Cheng, S. H., Dressman, H., Pittman, J., Tsou, M. H., Horng, C. F., Bild, A., Iversen, E. S., Lio, M., Chen, C. M., West, M., Nevins, J. R., and Huang, A. T. (2003). “Gene Expression Predictors of Breast Cancer Outcomes.” *The Lancet*, 361, 1590–1596.

Jones, B., Carvalho, C., Dobra, A., Hans, C., Carter, C., and West, M. (2004). “Experiments in Stochastic Computation for High-Dimensional Graphical Models.” ISDS Discussion Paper 04–01.

Jones, B. and West, M. (2004). “Covariance Decomposition in Multivariate Analysis.” ISDS Discussion Paper 04–15.

Lacroix, M. and Leclercq, G. (2004). “About GATA3, HNF3A, and XBP1, Three Genes Co-expressed With the Oestrogen Receptor-α Gene (ESR1) in Breast Cancer.” *Molecular and Cellular Endocrinology*, 219, 1–7.

Lauritzen, S. L. (1996). *Graphical Models*. Oxford: Clarendon Press.

Madigan, D., Andersson, S. A., Perlman, M. D., and Volinsky, C. T. (1996). “Bayesian Model Averaging and Model Selection for Markov Equivalence Classes of Acyclic Digraphs.” *Commumications in Statistics: Theory and Methods*, 25, 2493–2520.

Madigan, D. and Raftery, A. E. (1994). “Model Selection and Accounting for Model Uncertainty in Graphical Models Using Occam’s Window.” *Journal of the American Statistical Association*, 89, 1535–1546.

Raftery, A. E., Madigan, D., and Hoeting, J. A. (1997). “Bayesian Model Averaging for Linear Regression Models.” *Journal of the American Statistical Association*, 92, 179–191.

Roverato, A. (2002). “Hyper-inverse Wishart Distribution for Non-decomposable Graphs and its Application to Bayesian inference for Gaussian Graphical Models.” *Scandinavian Journal of Statistics*, 29, 391–411.

Wang, Q., Dobra, A., and West, M. (2004). “GraphExplore: A Software Tool for Graph Visualization.” ISDS Discussion Paper #04-22.

Zhou, X., Kao, M. C. J., and Wong, W. H. (2002). “Transitive Functional Annotation by Shortest-path Analysis of Gene Expression Data.” *Proceedings of the National Academy of Sciences*, 99, 12783–12788.