

Dependency networks for genome-wide data

Technical Report no. 547

Department of Statistics, University of Washington

ADRIAN DOBRA

*Department of Statistics, University of Washington,
Seattle, WA 98195, U.S.A.
adobra@u.washington.edu*

Abstract

We describe a new stochastic search algorithm for linear regression models called the bounded mode stochastic search (BMSS). We make use of BMSS to perform variable selection and classification as well as to construct sparse dependency networks. Furthermore, we show how to determine genetic networks from genome-wide data that involves any combination of continuous and discrete variables. We illustrate our methodology with several simulated and real-world datasets.

Some key words: Bayesian regression analysis; Dependency networks; Gaussian graphical models; Gene expression; Stochastic search; Variable selection.

1 Introduction

Nowadays the identification of biological pathways from genome-wide studies is the focus of a considerable research effort. The overarching goal is to use gene expression, genotype, clinical and physiological information to create a network of interactions that could potentially be representative for underlying biological processes. The numerous approaches to learning networks developed so far are quite diverse and, for this reason, are complementary to each other. Most of these techniques have been successful in unraveling various parts of the complex biology that induced the patterns of covariation represented in the observed data. The biggest challenge comes from the large number of biological entities that need to be represented in a network. The links (edges) between these entities (vertices) need to be determined from a relatively small number of available samples. Inducing sparsity in the resulting network is key both for statistical reasons (a small sample size can support a reduced number of edges) and for biological reasons (only a small set of regulatory factors are expected to influence a given entity). As such, each vertex is expected

to have a modest number of direct neighbors. The vertices that can be reached by following paths of length one, two or three starting from a given vertex are expected to define a functional module associated with that vertex.

Two co-expressed genes are likely to be involved in the same biological pathways, hence an association network in which missing edges correspond with genes having low absolute correlation of their expression levels could reveal groups of genes sharing the same functions (Butte et al., 2000; Steuer et al., 2003). Shortest-path analysis in such networks can uncover other genes that do not have the same expression pattern but are involved in the same biological pathway (Zhou et al., 2002). Another important type of networks for expression data is represented by Gaussian graphical models (Dobra et al., 2004; Schafer and Strimmer, 2005; Li and Gui, 2006; Castelo and Roverato, 2006; Wille and Bühlmann, 2006). The observed variables are assumed to follow a multivariate normal distribution. The edges in this network correspond with non-zero elements of the inverse of the covariance matrix. The biological relevance of paths in Gaussian graphical models networks is studied in Jones and West (2005). Other types of networks are derived from Bayesian networks whose graphical representation is a directed acyclic graph (Segal et al., 2003; Yu et al., 2004; Friedman, 2004).

A related question that appears in genome-wide studies is the identification of a reduced set of molecular and clinical factors that are related to a certain phenotype of interest. This is known as the variable selection problem and can be solved based on univariate rankings that individually measure the dependency between each candidate factor and the response – see, for example, Golub et al. (1999); Nguyen and Rocke (2002); Dudoit et al. (2002); Tusher et al. (2001). Other approaches consider regression that involve combinations of factors, which lead to a huge increase in the number of candidate models that need to be examined. The stepwise methods of Furnival and Wilson (1974) can only be used for very small datasets due to their inability to escape local modes created by complex patterns of collinear predictors. A significant step forward were Markov chain Monte Carlo (MCMC) algorithms that explore the models space by sampling from the joint posterior distribution of the candidate models and regression parameters – see, for example, George and McCulloch (1993, 1997); Green (1995); Raftery et al. (1997); Nott and Green (2004). Excellent review papers about Bayesian variable selection for Gaussian linear regression models are Carlin and Chib (1995), Chipman et al. (2001) and Clyde and George (2004). Lee et al. (2003) make use of MCMC techniques in the context of probit regression to develop cancer classifiers based on expression data. Theoretical considerations related to the choice of priors for regression parameters are discussed in Fernández et al. (2003) and Liang et al. (2008).

MCMC methods can have a slow convergence rate due to the high model uncertainty resulting from the small number of available samples. Yeung et al. (2005) recognize this problem and develop a multi-class classification method by introducing a stochastic search algorithm called the iterative Bayesian model averaging (iBMA). While this method performs very well in the context of gene selection in microarray studies, it is still based on an univariate ordering of the candidate predictors. Hans et al. (2007) make another step forward and propose the shotgun stochastic search (SSS) algorithm that is capable of quickly moving towards high-probable models while evaluating and recording complete neighborhoods around the current most promising models.

The aim of this paper is to combine variable selection based on linear regressions and the iden-

tification of biological networks in a coherent and comprehensive methodology. We introduce a new stochastic search algorithm for linear regressions called the bounded mode stochastic search (BMSS). We use BMSS to determine a reduced set of variables that renders the response conditionally independent of the remaining variables. We make further use of this algorithm to learn a dependency network (Heckerman et al., 2000) involving this small set of predictors and the response. This dependency network allows us to infer sparse networks of interactions. The initial variable selection step that eliminates most of the variables present in the data is critical because the networks we ultimately construct contain only vertices that are directly relevant for the phenotype of interest. We allow for the presence of any combination of continuous and binary variables and, as such, we are no longer restricted to the multivariate normal assumption required by the Gaussian graphical models. Moreover, the edges we identify are indicative of complex nonlinear relationships and generalize correlation-based networks.

The structure of this paper is as follows. In Section 2 we give three stochastic search algorithms for linear regressions: the Markov chain Monte Carlo model composition algorithm (MC³), SSS and BMSS. We compare their relative efficiency and further show the properties of BMSS in a simulated example in Section 3. In Section 4 we discuss dependency networks and in Section 5 we define focused genetic networks. In Section 6 we give a procedure to learn Gaussian graphical models using BMSS and dependency networks. The properties of this procedure are illustrated in a simulated example and a real-world dataset. In Section 7 we present the analysis of four gene expression datasets with our focused genetic networks. In Section 8 we make some concluding remarks.

2 Stochastic search algorithms for small subsets regressions

In order to keep the notation simple, in this section we assume that a response variable Y is associated with the first component of the random vector $X = (X_1, \dots, X_p)$, while the remaining components are the candidate explanatory covariates. Let $V = \{1, 2, \dots, p\}$ and denote by D the corresponding $n \times p$ design matrix. A regression model for $Y = X_1$ given a subset $X_A = (X_i)_{i \in A}$, $A \subset V \setminus \{1\}$ of the remaining variables is denoted by $[1|A]$. We follow the prior specification for regression parameters described in Appendixes A and B for Normal linear regression (Y continuous) and logistic regression (Y binary). We denote by $p(D|[1|A]) = p(D_{\{1\} \cup A}|[1|A])$ the marginal likelihood of the regression $[1|A]$ where D_B are the B columns of D . The posterior probability of $[1|A]$ is readily available up to a normalizing constant:

$$p([1|A]|D) \propto p(D|[1|A])p([1|A]),$$

where $p([1|A])$ is the prior probability of model $[1|A]$.

Gene expression datasets are characterized by a very large p/n ratio. As such, we are interested in regressions that contain a number of predictors much smaller than p . There are two ways to focus on these small subsets regressions. The first approach involves choosing a prior on the candidate regressions space that downweights richer regressions (Chipman, 1996; Kohn et al., 2001; Scott and Berger, 2006). While such priors encourage sparsity and seem to work reasonably well (Hans et al., 2007), we found that in practice it is not straightforward to calibrate them to completely

avoid evaluating the marginal likelihood of models with many predictors. Such calculations are prone to lead to numerical difficulties especially when there are no formulas available for the corresponding high-dimensional integrals. This is the case of logistic regressions whose marginal likelihoods are estimated using the Laplace approximation (see Appendix B). The second approach involves reducing the space of candidate models to $\mathcal{R}_{p_{max}}$ – the set of regressions with at most p_{max} predictors. This implies that only $|\mathcal{R}_{p_{max}}| = \sum_{j=1}^{p_{max}} \binom{p-1}{j}$ regressions need to be considered which represents a significant reduction compared to 2^{p-1} – the total number of regressions for Y . Nevertheless, $|\mathcal{R}_{p_{max}}|$ is still extremely large for the datasets we are interested in and hence its exhaustive enumeration is not feasible. There is no substantive need to further penalize for model complexity and we assume throughout that the models in $\mathcal{R}_{p_{max}}$ are a priori equally likely, i.e.

$$p([1|A]) = 1/|\mathcal{R}_{p_{max}}|, \quad (2.1)$$

which implies $p([1|A]|D) \propto p(D|[1|A])$. The uniform prior (2.1) favors models with more regressors if p_{max} is much smaller than p . If we let $j \in \{0, 1, \dots, p_{max}\}$ be the dimension of a regression in $\mathcal{R}_{p_{max}}$, then the prior for models of size j induced by (2.1) is

$$p(j) = \binom{p-1}{j} / |\mathcal{R}_{p_{max}}|.$$

This is simply a consequence of the fact that there are more regressions of one dimension than regressions of some other dimension. Since we do not make any inferences related to the size of regression models, we consider that the uniform prior (2.1) does not induce an undesirable bias in the results we ultimately report.

We record the highest posterior probability models identified by a stochastic search algorithm in a list $\mathcal{L} \subset \mathcal{R}_{p_{max}}$. We define

$$\mathcal{L}(c) = \{[1|A] \in \mathcal{L} : p([1|A]|D) \geq cp([1|A_h]|D)\},$$

where $c \in (0, 1)$ and $[1|A_h] = \operatorname{argmax}_{[1|A'] \in \mathcal{L}} p([1|A']|D)$. According to Kass and Raftery (1995), a choice of c in one of the intervals $(0, 0.01]$, $(0.01, 0.1]$, $(0.1, 1/3.2]$, $(1/3.2, 1]$ means that the models in $\mathcal{L} \setminus \mathcal{L}(c)$ have decisive, strong, substantial or “not worth more than a bare mention” evidence against them with respect to $[1|A_h]$. We further introduce the set $\mathcal{L}(c, m)$ that consists of the top m highest posterior probability models in $\mathcal{L}(c)$. This reduced set of models is needed because $\mathcal{L}(c)$ might still contain a large number of models for certain values of c especially if there are many models having almost the same posterior probability. We construct our algorithms such that the addition of a new model to \mathcal{L} is always followed by the pruning of the models in $\mathcal{L} \setminus \mathcal{L}(c, m)$. Therefore $\mathcal{L} = \mathcal{L}(c, m)$ after each update and consequently we never output more than m models. These will be the highest posterior probability models identified during a search. We assume that the list \mathcal{L} also records the values of the marginal likelihood of each regression.

We define the neighborhood of the regression $[1|A]$ as (Hans et al., 2007):

$$\operatorname{nbd}_{\mathcal{R}_{p_{max}}}([1|A]) = \operatorname{nbd}_{\mathcal{R}_{p_{max}}}^+([1|A]) \cup \operatorname{nbd}_{\mathcal{R}_{p_{max}}}^0([1|A]) \cup \operatorname{nbd}_{\mathcal{R}_{p_{max}}}^-([1|A]).$$

The three subsets of neighbors are obtained by including an additional predictor in the regression, by substituting a predictor with another predictor and by deleting a predictor from the regression:

$$\begin{aligned}\text{nb}d_{\mathcal{R}_{p_{max}}}^+([1|A]) &= \{[1|A \cup \{j\}] : j \in (2:p) \setminus A\} \cap \mathcal{R}_{p_{max}}, \\ \text{nb}d_{\mathcal{R}_{p_{max}}}^0([1|A]) &= \{[1|(A \setminus \{j_1\}) \cup \{j_2\}] : j_1 \in A, j_2 \in (2:p) \setminus A\}, \\ \text{nb}d_{\mathcal{R}_{p_{max}}}^-([1|A]) &= \{[1|A \setminus \{j\}] : j \in A\}.\end{aligned}$$

The regression neighborhoods are defined so that any regression in $\mathcal{R}_{p_{max}}$ can be connected with any other regression in $\mathcal{R}_{p_{max}}$ through a sequence of regressions in $\mathcal{R}_{p_{max}}$ such that any two consecutive regressions in this sequence are neighbors. We remark that the size of the neighborhoods of the regressions in $\mathcal{R}_{p_{max}}$ is not constant. If the regression $[1|A]$ contains the maximum number of predictors (i.e., $|A| = p_{max}$), no other variable can be added to the model (i.e., $\text{nb}d_{\mathcal{R}_{p_{max}}}^+([1|A]) = \emptyset$). As such, the neighborhood will be too constrained if we would not allow the substitution of a variable currently in the model with some variable currently outside the model.

We describe three stochastic techniques for visiting $\mathcal{R}_{p_{max}}$. The first procedure is called the Markov chain Monte Carlo model composition algorithm (MC³) and was introduced by Madigan and York (1995). It constructs an irreducible chain on $\mathcal{R}_{p_{max}}$ as follows:

procedure MC³ (p_{max}, c, m, k_{max})

► Start at a random regression $[1|A_1] \in \mathcal{R}_{p_{max}}$. Set $\mathcal{L} = \{[1|A_1]\}$.

► For $k = 1, 2, \dots, k_{max}$ do:

• Uniformly draw a regression $[1|\tilde{A}]$ from $\text{nb}d_{\mathcal{R}_{p_{max}}}([1|A_k])$, where $[1|A_k]$ is the current state of the chain. Set $[1|A_{k+1}] = [1|\tilde{A}]$ with probability

$$\min \left\{ 1, \frac{p([1|\tilde{A}]|D)/|\text{nb}d_{\mathcal{R}_{p_{max}}}([1|\tilde{A}]|)}{p([1|A_k]|D)/|\text{nb}d_{\mathcal{R}_{p_{max}}}([1|A_k]|)} \right\}.$$

Otherwise set $[1|A_{k+1}] = [1|A_k]$. Include $[1|\tilde{A}]$ in \mathcal{L} and prune it, such that $\mathcal{L} = \mathcal{L}(c, m)$.

□

MC³ moves around $\mathcal{R}_{p_{max}}$ by sampling from the posterior distribution $\{p([1|A]|D) : [1|A] \in \mathcal{R}_{p_{max}}\}$. As such, the probability of identifying the highest posterior probability regression $[1|A^*]$ in $\mathcal{R}_{p_{max}}$ is

$$p([1|A^*]|D) / \left\{ \sum_{[1|A] \in \mathcal{R}_{p_{max}}} p([1|A]|D) \right\}.$$

This probability could be almost zero if $|\mathcal{R}_{p_{max}}|$ is large and n is small, which means that MC³ could be very inefficient in finding models with large posterior probability. Hans et al. (2007) recognized this issue and proposed the shotgun stochastic search algorithm (SSS) that aggressively moves towards regions with high posterior probability in $\mathcal{R}_{p_{max}}$ by evaluating the entire neighborhood of the current regression instead of only one random neighbor.

procedure SSS(p_{max}, c, m, k_{max})

► Start at a random regression $[1|A_1] \in \mathcal{R}_{p_{max}}$. Set $\mathcal{L} = \{[1|A_1]\}$.

► For $k = 1, 2, \dots, k_{max}$ do:

• Let $[1|A_k]$ be the current regression. Sample three models $[1|A_k^+]$, $[1|A_k^0]$ and $[1|A_k^-]$ from the neighbors' sets $\text{nbr}_{\mathcal{R}_{p_{max}}}^+([1|A_k])$, $\text{nbr}_{\mathcal{R}_{p_{max}}}^0([1|A_k])$ and $\text{nbr}_{\mathcal{R}_{p_{max}}}^-([1|A_k])$, respectively. The probability of selecting a regression $[1|A]$ from a neighbors' set is proportional with its posterior probability $p([1|A]|D)$, normalized within that set. Sample a regression $[1|A_{k+1}]$ from $B = \{[1|A_k^+], [1|A_k^0], [1|A_k^-]\}$ with probability proportional with its posterior probability normalized within the set B . Include the regressions in $\text{nbr}_{\mathcal{R}_{p_{max}}}([1|A_k])$ in \mathcal{L} and prune \mathcal{L} , such that $\mathcal{L} = \mathcal{L}(c, m)$.

□

Hans et al. (2007) empirically show that SSS finds models with high probability faster than MC³. This is largely true if one does not need to make too many changes to the current model to reach $[1|A^*]$ or other models with comparable posterior probability. On the other hand, fully exploring the neighborhoods of all the models on a path connecting the current model with $[1|A^*]$ could be inefficient if this path is relatively long. In this case MC³ might end up reaching $[1|A^*]$ after visiting fewer models than SSS. Each iteration of SSS is computationally more expensive than an iteration of MC³ since the entire neighborhood of each regression needs to be visited and recorded. For this reason SSS does not stay at the same model for two consecutive iterations as MC³ does. While SSS can significantly benefit from cluster computing that allows a simultaneous examination of subsets of neighbors, it still moves around $\mathcal{R}_{p_{max}}$ by selecting the regression whose neighborhood will be studied at the next iteration from the neighbors of the current regression $[1|A_k]$. This constitutes a limitation of the algorithm because it is very likely that other models from the list \mathcal{L} could lead to $[1|A^*]$ faster than a model from $\text{nbr}_{\mathcal{R}_{p_{max}}}([1|A_k])$.

Berger and Molina (2005) pointed out that the model whose neighbor(s) could be visited at the next iteration should be selected from the list of models identified so far with probabilities proportional with the posterior model probabilities. More specifically, they record every model they identify, i.e. they take $\mathcal{L} = \mathcal{L}(0, \infty)$. A model $[1|A] \in \mathcal{L}$ is selected with probability:

$$p([1|A]|D) / \left\{ \sum_{[1|A'] \in \mathcal{L}} p([1|A']|D) \right\}. \quad (2.2)$$

After a large number of iterations, \mathcal{L} is likely to contain a large number of models, thus (2.2) converges to the probability that MC³ reaches $[1|A]$. If the list of recorded models is pruned such that $\mathcal{L} = \mathcal{L}(c, m)$ with $c > 0$ and m finite, some of the models with lower posterior probability will be discarded from \mathcal{L} . This leads to more aggressive moves in the models space. A stochastic search algorithm could therefore reach high posterior probability models faster than MC³, but could also end up being trapped in local modes. Consequently, algorithms that select models for exploration using (2.2) should use a smaller but strictly positive c and some larger value of m .

As an aside, Berger and Molina (2005) also suggest that one should select a neighbor of the current model for inclusion in \mathcal{L} based on the posterior inclusion probabilities of each variable calculated based on the list of models explored so far. Our experience shows that selecting a neighbor

with equal probability works very well, while using variable inclusion probabilities seems to guide the search towards regions of the models space that have been explored before.

We propose a novel stochastic search algorithm which we call the bounded mode stochastic search (BMSS). Our method combines MC³, SSS and some of the ideas of Berger and Molina (2005) in two different stages. In the first stage, we attempt to advance in the space of models fast by exploring only one model at each iteration. Once higher posterior probability models have been reached, we proceed to exhaustively explore their neighborhoods at the second stage to make sure we do not miss any relevant models that are close to the models already identified. There is no benefit of exploring the same model twice at the second stage, hence we keep track of the models explored at the previous iterations.

procedure BMSS($p_{max}, c, m, k_{max}^1, k_{max}^2$)

► Start at a random regression $[1|A_1] \in \mathcal{R}_{p_{max}}$. Set $\mathcal{L} = \{[1|A_1]\}$.

► *Stage One.* For $k = 1, 2, \dots, k_{max}^1$ do:

• Uniformly draw a regression $[1|\tilde{A}]$ from $\text{nbr}_{\mathcal{R}_{p_{max}}}([1|A_k])$, where $[1|A_k]$ is the current model. Include $[1|\tilde{A}]$ in \mathcal{L} and prune it, such that $\mathcal{L} = \mathcal{L}(c, m)$.

• Sample a regression $[1|A_{k+1}]$ from \mathcal{L} with probability proportional with $\{p([1|A]|D) : [1|A] \in \mathcal{L}\}$. □

► Mark all the models in \mathcal{L} as unexplored.

► *Stage Two.* For $k = k_{max}^1 + 1, \dots, k_{max}^1 + k_{max}^2$ do:

• Let $\mathcal{L}_U \subset \mathcal{L}$ the subset of unexplored models. If $\mathcal{L}_U = \emptyset$, STOP.

• Sample a model $[1|\tilde{A}]$ from \mathcal{L}_U with probability proportional with $\{p([1|A]|D) : [1|A] \in \mathcal{L}_U\}$. Mark $[1|\tilde{A}]$ as explored.

• Explore the entire neighborhood of $[1|\tilde{A}]$ into \mathcal{L} . Every model in $\text{nbr}_{\mathcal{R}_{p_{max}}}([1|\tilde{A}]) \setminus \mathcal{L}$ is marked as unexplored and its posterior probability is calculated and recorded.

• Prune \mathcal{L} so that $\mathcal{L} = \mathcal{L}(c, m)$.

□

At the second stage BMSS can end if no unexplored models are found in \mathcal{L} before completing k_{max}^2 iterations.

We compare the relative performance of these three stochastic search techniques using the gene expression datasets described in Sections 7.2, 7.3 and 7.4. We run one instance of BMSS starting from the null regression using the parameter values for p_{max}, c, m, s_1 and s_2 mentioned in the text. We counted the number of models evaluated by BMSS and run SSS and MC³ until they evaluate the same number of models. We pool the top models identified by all three methods for each dataset and report which models were missed by each algorithm – see Table 1. Two different marginal likelihood values correspond with two different models. We performed the computations on a Mac Pro desktop computer with two 3 GHz dual-core Intel Xeon processors.

SSS evaluates the largest number of models at each iteration and hence it is the fastest algorithm, followed by BMSS and MC³. However, BMSS does not seem to miss any models with the largest marginal likelihood, while SSS and MC³ do not identify the complete set of top ten models for any of the three datasets. In fact, SSS and MC³ fail to identify all top ten models for the leukemia data. We note that BMSS evaluated only a small fraction of the total number of candidate

Table 1: Comparison of the effectiveness of the three stochastic search methods. We report the logarithm of the marginal likelihoods of the top ten regressions identified by all three algorithms. The models that were not identified by an algorithm are marked with “no.” The time elapsed until the completion of each run is measured in seconds.

Dataset	Breast cancer			Leukemia			Lymph		
Total models	1.984×10^{11}			3.608×10^{12}			1.173×10^{19}		
Evaluated models	1575100			1319200			2805400		
Algorithm	BMSS	SSS	MC ³	BMSS	SSS	MC ³	BMSS	SSS	MC ³
Time	248	176	1496	181	134	808	1022	967	3027
Model 1	-31.99	-31.99	-31.99	-6.40	no	no	-42.33	-42.33	no
Model 2	-32.00	-32.00	-32.00	-6.41	no	no	-42.78	no	-42.78
Model 3	-32.51	no	-32.51	-6.42	no	no	-43.03	no	-43.03
Model 4	-33.10	-33.10	-33.10	-6.44	no	no	-43.17	no	-43.17
Model 5	-33.26	-33.26	-33.26	-6.45	no	no	-43.27	-43.27	no
Model 6	-33.27	-33.27	no	-6.47	no	no	-43.31	-43.31	-43.31
Model 7	-33.30	-33.30	-33.30	-6.480	no	no	-43.34	-43.34	no
Model 8	-33.39	-33.39	-33.39	-6.484	no	no	-43.49	-43.49	-43.49
Model 9	-33.64	-33.64	-33.64	-6.487	no	no	-43.57	-43.57	no
Model 10	-33.80	-33.80	-33.80	-6.49	no	no	-43.62	-43.62	no

models as the second and third columns of Table 1 show.

3 Simulation study: multicollinear candidate predictors

We follow an example suggested in Nott and Green (2004). As in George and McCulloch (1997), we generate $Z_1, \dots, Z_{15}, Z \sim N_{300}(0, I_{300})$. Let $X_i = Z_i + 2Z, i = 1, 3, 5, 8, 9, 10, 12, 13, 14, 15, X_2 = X_1 + 0.15Z_2, X_4 = X_3 + 0.15Z_4, X_6 = X_5 + 0.15Z_6, X_7 = X_8 + X_9 - X_{10} + 0.15Z_7$ and $X_{11} = X_{14} + X_{15} - X_{12} - X_{13} + 0.15Z_{11}$. George and McCulloch (1997) point out that this design matrix leads to correlations of about 0.998 between X_i and X_{i+1} for $i = 1, 3, 5$. There are also strong linear associations between (X_7, X_8, X_9, X_{10}) and $(X_{11}, X_{12}, X_{13}, X_{14}, X_{15})$. We let $\tilde{X} = [X^{(1)} X^{(2)}]$ be a 300×30 design matrix obtained by independently simulating two instances $X^{(1)}$ and $X^{(2)}$ of the 300×15 design matrix X . Consider the 30-dimensional vector of regression coefficients β defined by $\beta_j = 1.5$, if $j = 1, 3, 5, 7, 11, 12, 13, \beta_8 = -1.5$ and $\beta_j = 0$ otherwise. We generate a binary vector Y_b whose i -th component ($i = 1, \dots, 300$) is simulated from a Bernoulli distribution with success probability $[1 + \exp(-\tilde{Y}_i)]^{-1}$, where $\tilde{Y} = \tilde{X}\beta$.

We would like to study whether the predictors X_{16}, \dots, X_{30} do not appear in the logistic regressions corresponding with Y_b . We also want to learn whether Y_b appears as a relevant predictor in the linear regressions associated with each $X_i, i = 1, \dots, 30$. We employed BMSS with $p_{max} = 5, c = 0.1, m = 500, s_1 = 25000$ and $s_2 = 100$ to learn high-posterior regression models of each variable given the rest. Due to the complex correlation structure amongst the 31 covariates, variables may be selected even if their regression coefficients are zero. The upper panel of Figure 3

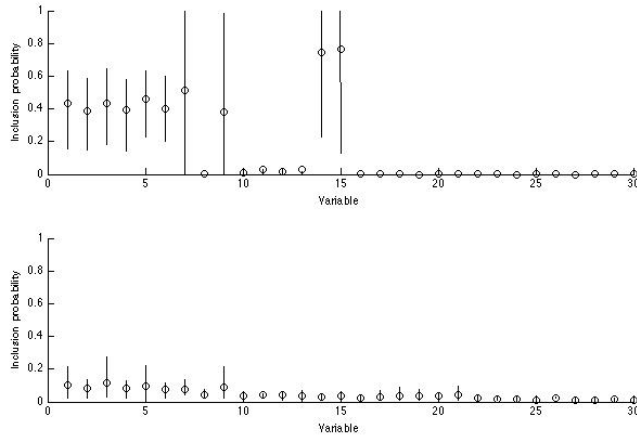


Figure 1: The upper panel gives the average posterior inclusion probabilities and 80% credibility intervals of X_i , $i = 1, \dots, 30$ for the logistic regression models of Y_b in the simulation study described in Section 3. The lower panel reports the average posterior inclusion probabilities of Y_b in the linear regression models of X_i , $i = 1, \dots, 30$.

shows the mean posterior inclusion probabilities together with the corresponding 80% credibility intervals for each of the 30 candidate predictors that appear in the highest posterior probability logistic regressions of Y_b across 100 simulated datasets. We see that the posterior inclusion probabilities of the variables X_{16}, \dots, X_{30} are indeed almost zero. Variable X_i ($i = 1, 3, 5$) consistently appears in the regressions selected and has a slightly larger posterior inclusion probability than X_{i+1} . Variables $\{X_2, X_4, X_6\}$ are selected due to their large correlations with $\{X_1, X_3, X_5\}$. Remark that the posterior inclusion probability of X_8 is almost zero and that X_7 and X_9 are alternatively selected instead of X_8 due to large correlations among these three variables. For the same reason, X_{14} and X_{15} are selected instead of X_{11}, X_{12} and X_{13} . The lower panel of Figure 3 gives the posterior inclusion probabilities of Y_b in the highest posterior probability regression models of each X_i . As expected, Y_b appears more often in regression models associated with X_1, \dots, X_9 and almost never appears in regression models associated with X_{16}, \dots, X_{30} . We also see that regression models do not seem to capture the dependency between Y_b and X_{11}, X_{12}, X_{13} due to the intricate structure of dependencies of this dataset.

The percentage of correctly predicted samples from Y_b across the 100 replicates is 95.86% with a standard deviation of 1.07%. This is indicative of an extremely good fit for the logistic regressions selected.

4 Learning and inference for dependency networks

We denote by $X_{-j} = X_{V \setminus \{j\}}$, for $j = 1, \dots, p$. A dependency network (Heckerman et al., 2000) is a collection of conditional distributions or regressions of each variable given the rest:

$$\mathcal{D} = \{p(X_j | X_{-j} = x_{-j}) : j = 1, \dots, p\}.$$

Each of these local probability distributions can be modeled and learned independently of the others. We can make use of one of the stochastic search algorithms from Section 2 to determine a set $\mathcal{L}^j = \{[j|A_l^j] : l = 1, \dots, |\mathcal{L}^j|\}$ of high posterior probability regressions of X_j given X_{-j} . It follows that

$$p(X_j|X_{-j} = x_{-j}) = p(X_j|X_{A^j} = x_{A^j}) = \sum_{l=1}^{|\mathcal{L}^j|} p(X_j|X_{A_l^j} = x_{A_l^j}) p^*([j|A_l^j]|D, \mathcal{L}^j), \quad (4.1)$$

where $A^j = \cup_{l=1}^{|\mathcal{L}^j|} A_l^j$ are the indices of all the regressors that appear in at least one regression in \mathcal{L}^j . The weight of each regression in the mixture (4.1) is given by its posterior probability normalized within \mathcal{L}^j :

$$p^*([j|A_l^j]|D, \mathcal{L}^j) = p([j|A_l^j]|D) / \left[\sum_{l'=1}^{|\mathcal{L}^j|} p([j|A_{l'}^j]|D) \right]. \quad (4.2)$$

If the number p of observed variables is extremely large and p_{max} is small, it is likely that the size of each A^j will also be much smaller than $p - 1$. Equation (4.1) implies that X_j is conditionally independent of $X_{V \setminus (\{j\} \cup A^j)}$ given X_{A^j} . Hence the dependency network \mathcal{D} is sparse and embeds conditional independence constraints that creates a parsimonious structure among the observed covariates. This structure reflects the uncertainty of a particular choice of regressions associated with each variable through Bayesian model averaging (Kass and Raftery, 1995). The parameters p_{max} , c and m of the three stochastic search algorithms from Section 2 control the size as well as the number of models in the lists \mathcal{L}^j .

We sample from \mathcal{D} using an ordered Gibbs sampling algorithm (Geman and Geman, 1984). Assume that the current state of the chain is $x^{(t)} = (x_1^{(t)}, \dots, x_p^{(t)})$. For each $j = 1, \dots, p$, simulate

$$x_j^{(t+1)} \sim p(X_j|X_{-j} = (x_1^{(t+1)}, \dots, x_{j-1}^{(t+1)}, x_{j+1}^{(t)}, \dots, x_p^{(t)})),$$

which gives the next state of the chain $x^{(t+1)}$. Simulating from each mixture (4.1) is performed by sampling a regression $[j|A_l^j]$ with probabilities proportional with (4.2), sampling a set of regression coefficients corresponding with $[j|A_l^j]$ and then sampling from the corresponding conditionals – see Appendixes A and B for details.

We remark that, given enough samples, we should have $j_1 \in A_{j_2}$ if and only if $j_2 \in A_{j_1}$ for any $j_1 \neq j_2$. This means that X_{j_1} appears in the conditional of X_{j_2} and vice-versa. Bayesian model averaging is key in this context because it eliminates the need to make an explicit decision relative to the choice of covariates that appear in each conditional distribution. As such, the order in which we sample from the local probability distributions \mathcal{D} should be irrelevant. We emphasize that the symmetry of the sets A_j is not explicitly enforced.

The most important question relates to the existence of a joint probability distribution $p(X_V)$ associated with the local probability distributions \mathcal{D} . Given a positivity condition usually satisfied

in practice, a dependency network \mathcal{D} uniquely identifies a joint distribution $p(X_V)$ up to a normalizing constant (Besag, 1974). If $p(X_V)$ exists, it is unique and \mathcal{D} is called consistent. If $p(X_V)$ does not exist, \mathcal{D} is called inconsistent (Heckerman et al., 2000). Hobert and Casella (1998) study the more general case when \mathcal{D} is inconsistent but still determines an improper joint distribution. Arnold et al. (2001) provide a comprehensive discussion related to conditionally specified distributions. Related results are presented in Gelman and Speed (1993) and Besag and Kooperberg (1995), among others.

The ordered Gibbs sampling algorithm can be used to sample from $p(X_V)$ if \mathcal{D} is consistent (Heckerman et al., 2000). Unfortunately the output from the Gibbs sampler will offer no indication whether \mathcal{D} is indeed consistent (Hobert and Casella, 1998). We make use of the samples generated from \mathcal{D} only to estimate relevant quantities of interest, such as bivariate dependency measures. These samples reflect the structure of \mathcal{D} and do not necessarily come from a proper joint distribution $p(X_V)$.

5 Focused genetic networks

We are interested to construct a network associated with $Y = X_1$ where $X_V = (X_1, \dots, X_p)$ is vector of continuous or binary random variables. Constructing a network that involves the entire X_V is likely to be a computationally expensive task if p is extremely large. Moreover, even if this network is constructed, exploring and displaying it might prove to be a challenge in itself due to its huge number of connections it involves. We develop a two step procedure that defines a network with respect to Y and involves only a small number of the other variables:

Step 1. Use a stochastic search algorithm for regression models (see Section 2) to determine a set of high posterior probability regressions \mathcal{L}^1 of Y given X_{-1} . Then Y is independent of $X_{V \setminus (\{1\} \cup A^1)}$ given X_{A^1} , where A^1 are the indices of the variables present in the regressions \mathcal{L}^1 . The direct implication is that the network we construct should not contain $X_{V \setminus (\{1\} \cup A^1)}$ since these variables do not bring any additional information about Y if X_{A^1} is known.

Step 2. Learn a dependency network \mathcal{D} that involves Y and X_{A^1} as described in Section 4. Use the ordered Gibbs sampler to generate a random sample $\tilde{D}_{\{1\} \cup A^1}$ from \mathcal{D} . This random sample embeds the structural constraints implied by \mathcal{D} . Identify two different types of networks as follows:

(a) *Association networks.* Estimate the pairwise associations $d(X_{j_1}, X_{j_2})$, $j_1, j_2 \in \{1\} \cup A^1$ based on $\tilde{D}_{\{1\} \cup A^1}$. Here $d(\cdot, \cdot)$ denotes Kendall's tau, Spearman's rho or the correlation coefficient. We prefer using Kendall's tau or Spearman's rho because they measure the concordance between two random variables (Nelsen, 1999). On the other hand, the correlation coefficient reflects only linear dependence. The edges in the resulting association network connect pairs of variables whose pairwise associations are different from zero.

(b) *Liquid association networks.* Li (2002) introduced the concept of liquid association to quantify the dynamics of the association between two random variables X_{j_1} and X_{j_2} given a third random variable X_{j_3} . We denote this measure with $d(X_{j_1}, X_{j_2} | X_{j_3})$. The liquid association is especially relevant for pairs of random variables with a low absolute value of their pairwise association

$d(X_{j_1}, X_{j_2})$. Such pairs will not be captured in an association network. However, the association between X_{j_1} and X_{j_2} could vary significantly as a function of X_{j_3} . In this paper we take $X_{j_3} = Y$ since we want to measure the change with respect to our designated target variable. If Y is continuous, the liquid association between X_{j_1} and X_{j_2} given Y is defined as the expected value of the derivative of $d_{Y=y}(X_{j_1}, X_{j_2})$ with respect to Y , i.e.

$$d(X_{j_1}, X_{j_2}|Y) = E \left[d'_{Y=y}(X_{j_1}, X_{j_2}|Y = y) \right],$$

where $d_{Y=y}(X_{j_1}, X_{j_2})$ is the measure of association $d(\cdot, \cdot)$ between X_{j_1} and X_{j_2} evaluated for the samples $Y = y$. Li (2002) proves that $d(X_{j_1}, X_{j_2}|Y) = E[X_{j_1}X_{j_2}Y]$ if X_{j_1} , X_{j_2} and Y are normal random variables with zero mean and unit variance, while $d(\cdot, \cdot)$ is the correlation coefficient, i.e. $d(X_{j_1}, X_{j_2}) = E[X_{j_1}X_{j_2}]$. If $Y \in \{0, 1\}$ is a binary random variable, we define the liquid association between X_{j_1} and X_{j_2} given Y as the absolute value of the change between the association of X_{j_1} and X_{j_2} for the samples with $Y = 1$ versus the samples with $Y = 0$:

$$d(X_{j_1}, X_{j_2}|Y) = |d_{Y=1}(X_{j_1}, X_{j_2}) - d_{Y=0}(X_{j_1}, X_{j_2})|. \quad (5.1)$$

As suggested in Li (2002), a permutation test can be used to assess the statistical significance of (5.1). We generate random permutations Y^* of the observed values of Y and compute the corresponding liquid association score. The p-value is given by the number of permutations that lead to a score higher than the observed score divided by the total number of permutations. The liquid association can be evaluated with respect to Kendall's tau, Spearman's rho or the correlation coefficient. The corresponding liquid association network involves only X_{A^1} and is formed by joining variables whose liquid association given Y is different from zero. The liquid association measures are estimated based on $\tilde{D}_{\{1\} \cup A^1}$.

We emphasize that estimating the strength of pairwise interactions based on the observed samples $D_{\{1\} \cup A^1}$ instead of $\tilde{D}_{\{1\} \cup A^1}$ leads to an increased number of edges in the resulting networks. Inducing sparsity in the network structure is the key to identify the most relevant associations. This is the reason why the samples simulated from the dependency network we learn from the data are used for estimation. We illustrate the difference in the number of edges found to be statistically significant in the numerical examples from Section 7.

6 Covariance selection

In this section we show that the dependency networks we identify are an effective tool to perform covariance selection. More specifically, we assume that $X_V \sim N_p(0, \Sigma)$. Dempster (1972) proposed reducing the number of parameters that need to be estimated by setting some of the off-diagonal elements of the precision matrix $\Omega = \Sigma^{-1}$ to zero. A pattern of zero constraints for Ω is called a Gaussian graphical model (GGM). Its independence graph has vertices V and edges associated with the non-zero elements of Ω . If $\Omega_{j_1 j_2} = 0$, X_{j_1} and X_{j_2} are conditional independent given the remaining variables $X_{V \setminus \{j_1, j_2\}}$. We learn a dependency network \mathcal{D} for X_V as described in Section 4. We simulate a sample D from \mathcal{D} . We remark that \tilde{D} are not necessarily sampled from

a multivariate normal distribution. Indeed, even for $p = 2$, it is known that there exist bimodal bivariate distributions with normal conditional distributions (Gelman and Meng, 1991). We learn a GGM from the sample partial correlations estimated from \tilde{D} whose significance was assessed at a false discovery rate of 1%. The precision matrix Ω is estimated from the resulting GGM using the iterative proportional scaling algorithm of Speed and Kiiveri (1986).

6.1 Simulation study

We follow an example suggested in Yuan and Lin (2007). We generate a sample of size 50 from multivariate normal distribution $N_{10}(0, \Omega^{-1})$, where $\Omega_{ii} = 1, i = 1, \dots, 10, \Omega_{i-1,i} = \Omega_{i,i-1} = 0.5, i = 2, \dots, 10$ and $\Omega_{1,10} = \Omega_{10,1} = 0.4$. The other elements of the precision matrix Ω are zero which leads to a GGM whose independence graph is a cycle of length 10. This is an relevant example since this independence graph is not decomposable and it is relatively sparse (only 22.2% of the possible edges are present). We employed BMSS with $p_{max} = 3, c = 0.0001, m = 10000, s_1 = 2500, s_2 = 100$ and five search replicates to learn high-posterior regression models of each of the ten covariates given the rest. We simulated 5000 samples from the resulting dependency network with a burn-in time of 250. To reduce the correlation between consecutive samples we discarded 9 out of 10 samples and ended up with 500 dependency network samples. We assess the accuracy of our estimator in terms of the Kullback-Leibler loss. We repeated this simulation 100 times to account for the sampling variability.

In Table 2 we give the Kullback-Leibler loss (KL) as well as the number of false positive (FP) and false negative (FN) edges associated with our estimation method (D). We determine an independence graph from partial correlations estimated from the data with a false discovery rate of 1%. The corresponding results are shown in column P of Table 2. We also give the results obtained using the penalized likelihood methods Lasso (L) and Garrote (G) of Yuan and Lin (2007), the approach (MB) of Meinshausen and Bühlmann (2006) and the SIN method of Drton and Perlman (2004) with two cut-off values 0.05 and 0.25 as they were reported in Yuan and Lin (2007). We see that the dependency networks give the second smallest Kullback-Leibler loss. Despite having the largest estimation standard error, method D clearly outperforms the SIN method of Drton and Perlman (2004). The accuracy of our approach in the recovery of the structure of the graph is comparable with the performance of the state of the art covariance selection approaches. Learning the structure of the GGM from the dependency networks is key since method P leads to the second largest estimation error.

6.2 Call center data

We analyze a large scale dataset originally described in Shen and Huang (2005) and further studied in Huang et al. (2006), Bickel and Levina (2008) and Rajaratnam et al. (2008). The number of calls n_{ij} for $i = 1, \dots, 239$ days and $j = 1, \dots, 102$ ten-minute daily time intervals were recorded in 2002 from the call center of a major financial institution. A transformation $x_{ij} = (n_{ij} + 0.25)^{1/2}$ was subsequently employed to assure the normality assumption. We need to predict the volume of calls from the second half of the day based on the volume of calls from the first half of the day. We

Table 2: Covariance selection results for the cycle Gaussian graphical model of length 10. The standard errors across the 100 replicates are shown in parentheses.

	D	P	L	G	MB	SIN(0.05)	SIN(0.25)
KL	0.67 (0.42)	4.90 (2.77)	0.89 (0.04)	0.65 (0.03)	0.93 (0.02)	6.83 (0.23)	4.03 (0.18)
FP	2.80 (2.54)	0.20 (0.72)	19.24 (0.63)	5.81 (0.30)	3.58 (0.19)	0.06 (0.02)	0.25 (0.06)
FN	0.40 (0.80)	6.74 (3.69)	0.02 (0.02)	0.03 (0.02)	0.00 (0.00)	4.50 (0.16)	2.55 (0.13)

write

$$x = \begin{pmatrix} x^{(1:51)} \\ x^{(52:102)} \end{pmatrix}, \quad \mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}, \quad (6.1)$$

where $x_i^{(1:51)} = (x_{i1}, \dots, x_{i51})^T \sim N_{51}(\mu_1, \Sigma_{11})$, $x_i^{(52:102)} = (x_{i52}, \dots, x_{i102})^T \sim N_{51}(\mu_2, \Sigma_{22})$ for $i = 1, \dots, 239$. The linear regression of $x_i^{(52:102)}$ based on $x_i^{(1:51)}$ is

$$\hat{x}_i^{(52:102)} = \mu_2 + \Sigma_{21} \Sigma_{11}^{-1} (x_i^{(1:51)} - \mu_1). \quad (6.2)$$

The dataset is divided into a training set $x_{1:205}$ (first 205 days) and a test set $x_{206:239}$ (the remaining 34 days). We use $x_{1:205}$ to learn the dependency structure among the 102 variables associated with each time interval and to further estimate the covariance matrix Σ .

We employed BMSS with $p_{max} = 2$, $c = 0.1$, $m = 500$, $s_1 = 25000$, $s_2 = 100$ and three search replicates to learn a dependency network from the training data. We also determined dependency networks for $p_{max} \in \{3, 4, 5\}$. We simulated 25000 samples from the resulting dependency network with a burn-in time of 2500 and saved only the 25-th sample. The covariance matrix Σ was estimated from the resulting GGM. We denote by $\hat{\Sigma}_j$ the estimators corresponding with $p_{max} = j$, $j = 2, 3, 4, 5$ and by Σ_{mle} the sample covariance matrix for the training data. We use the prediction equation (6.2) with these five estimators for Σ . We measure the prediction error on the test data by the average absolute forecast error. The resulting forecast error lines are shown in Figure 6.2. The sum of the prediction errors on the test data is 74.73 for $\hat{\Sigma}_{mle}$, 59.97 for $\hat{\Sigma}_5$, 55.33 for $\hat{\Sigma}_4$, 51.08 for $\hat{\Sigma}_4$ and 49.79 for $\hat{\Sigma}_2$. We see that the performance of the forecast improves as we determine a more parsimonious structure of the precision matrix corresponding with Σ . The number of edges in the independence graphs with $p_{max} = 2, 3, 4, 5$ are 214, 196, 180 and 180. We remark that the forecast performance of $\hat{\Sigma}_2$ and $\hat{\Sigma}_3$ is better than the performance of the other estimators for Σ previously reported in the literature – see Huang et al. (2006); Bickel and Levina (2008); Rajaratnam et al. (2008).

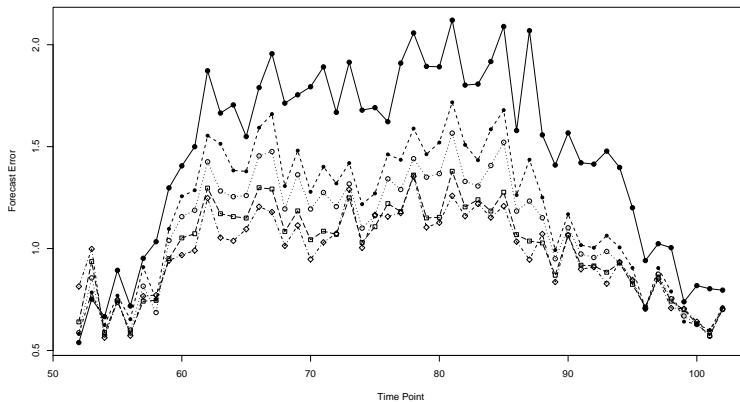


Figure 2: Forecast error associated with $\widehat{\Sigma}_{\text{mle}}$ (solid circles, solid lines), $\widehat{\Sigma}_5$ (smaller solid circles, dashed lines), $\widehat{\Sigma}_4$ (empty circles, dotted lines), $\widehat{\Sigma}_3$ (squares, long dashed lines) and $\widehat{\Sigma}_2$ (diamonds, dot dashed lines).

7 Real-world gene expression examples

7.1 Estrogen receptor genes

We make use of a breast cancer dataset that is publicly available as supplemental material in Pittman et al. (2004). Gene expression assays were performed on the Human U95Av2 GeneChip. The MASS5.0 signal measures of expression were transformed on a log2 scale and quantile normalized. After the removal of the 67 control probes and of the genes with small variation or with low levels, the resulting dataset comprises 7027 probe sets and 158 samples.

We use BMSS with $p_{\max} = 5$, $c = 0.001$, $m = 1000$, $s_1 = 250000$ and $s_2 = 250$ to determine genes that are potentially involved in the estrogen receptor (ER) pathway. The response variable is a probe (ESR1) associated with the estrogen receptor itself. BMSS identified 327 regressions which involve 180 probe sets (predictors). The corresponding model-averaged R^2 is estimated at 78.35% indicating a strong predictive relationship between the selected predictors and ESR1.

We apply BMSS with $p_{\max} = 3$, $c = 0.001$, $m = 1000$, $s_1 = 10000$ and $s_2 = 100$ to learn the structure of the dependency network that involves ESR1 and these 180 predictors. We simulate 25000 samples from the resulting dependency network with a burn-in of 2500 samples and use them to estimate Kendall's tau between any pair of the 181 covariates. There are 469 pairwise dependencies that are significant at a false discovery rate of 1%. These dependencies are represented as edges in a graph having vertices associated with each of the 181 probes. Figure 7.1 gives the direct neighbors of ESR1 in this graph together with the edges between them. The most important predictor of ESR1 that has a posterior inclusion probability equal to 1 is HG3125-HT3301 which is another probe associated with the estrogen receptor. If we would use the actual expression data to estimate Kendall's tau coefficients among the 16920 possible pairs of 181 covariates, 7230 coefficients (44.38%) will turn out to be significant at a false discovery rate of 1%. This is a considerable increase compared with the 469 pairs (0.03%) we find from the sparse dependency

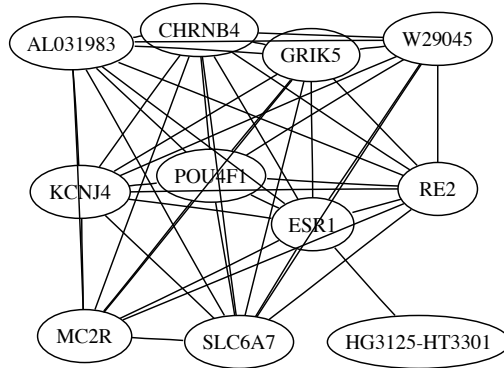


Figure 3: Probe sets whose corresponding expression levels are strongly related with ESR1 in the breast cancer data of Pittman et al. (2004).

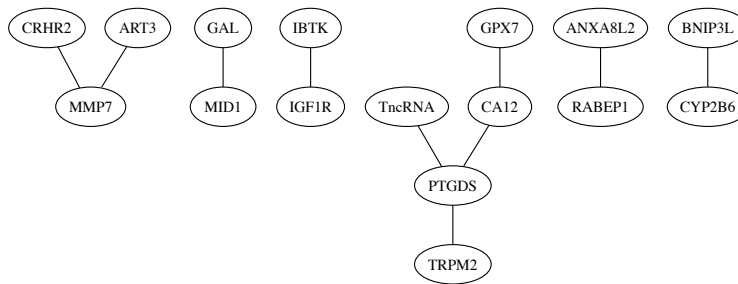


Figure 4: Liquid association network for ESR1 in the breast cancer data of Pittman et al. (2004).

network we learned using BMSS.

Figure 7.1 shows the ten pairs of genes whose liquid association given ESR1 is significant at a false discovery rate of 1%. We used the data simulated from the dependency network for estimation. We remark the presence of two genes (CA12 and IGF1R) that were also identified in the conditional independence gene expression networks determined in Dobra et al. (2004) using a richer version of this breast cancer dataset.

7.2 Breast cancer prognosis data

We analyze the breast cancer prognosis dataset from van't Veer et al. (2002). Here the goal is to develop a gene expression classifier to predict which patients are likely to develop metastases within five years vs. patients that remained disease-free for at least five years. Yeung et al. (2005) identified 4919 significantly regulated genes in the training set of 76 samples. The test set comprises 19 samples. van't Veer et al. (2002) selected 70 genes based on their high correlation with the response and reported that only two samples in the test set were incorrectly classified based on the expression levels of these genes. Yeung et al. (2005) used Bayesian model averaging to produce a classifier that involves only six genes. Their predictive model gives 3 classification errors in the test set.

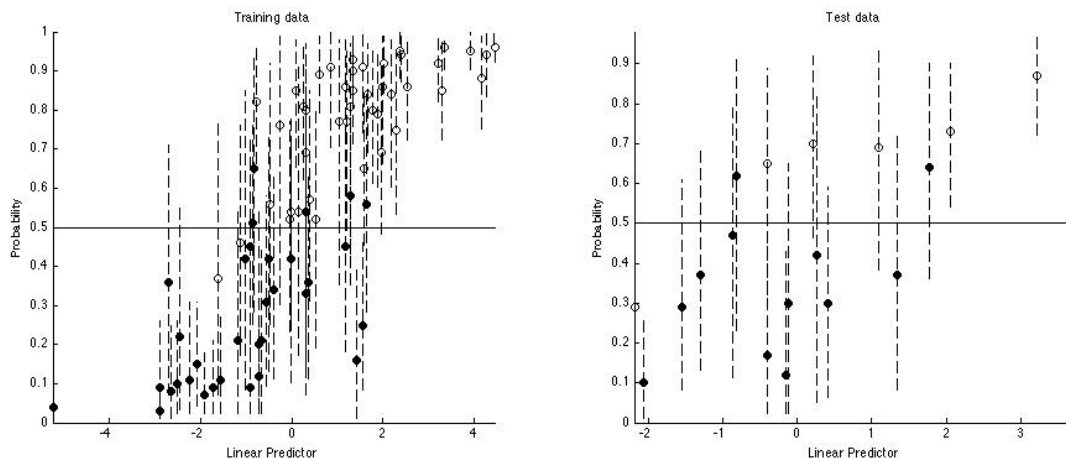


Figure 5: Prediction results for the breast cancer prognosis data. The solid circles represent disease-free patients, while the open circles represent patients with metastases within five years.

We use BMSS with $p_{max} = 3$, $c = 0.1$, $m = 500$, $s_1 = 1000000$, $s_2 = 100$ and three search replicates to identify high posterior probability logistic regressions associated with the binary variable five-year prognosis. BMSS identified 23 regressions involving 23 genes. Five of these genes (AL080059, Contig49670.RC, NM.012214, Contig59951, NM.003315) also appear in the list of six genes of Yeung et al. (2005). This set of genes gives excellent prediction results. In the training data, 69 samples (90.79%) were correctly predicted with a Brier score of 8.77 having a standard error of 0.59. In the test data, 16 samples (84.21%) were correctly classified with a Brier score of 3.56 having a standard error of 1.18. Throughout the paper we use the generalized version of the Brier score given in Yeung et al. (2005). Figure 7.2 shows the corresponding prediction probabilities together with their associated 80% intervals.

We use BMSS again with $p_{max} = 3$, $c = 0.001$, $m = 1000$, $s_1 = 10000$, $s_2 = 100$ and three search replicates to learn the structure of the dependency network involving the 23 genes and prognosis. We simulate 25000 samples from the resulting dependency network with a burn-in of 2500 samples and a gap of 100 between two consecutive saved samples. We use the resulting 250 samples to estimate Kendall's tau for any pair of the 24 covariates. There are only 27 pairs of variables having a value of Kendall's tau different than zero at a false discovery rate of 1% – see Figure 7.2. We remark that the three neighbors of prognosis are among the five genes that were also identified by Yeung et al. (2005). One of them (AL080059) also appears at the top of the list of the highest correlated genes with prognosis of van't Veer et al. (2002).

We note that the Kendall's tau of 35 pairs of covariates would have been found to be significantly different than zero if we would have done the estimation using the actual data. This drop of 23% in the number of significant pairwise dependencies is indicative of the parsimony of the dependency network we identified.

Figure 7.2 gives the nine gene-gene interactions that are influenced by prognosis. We used the data simulated from the dependency network to estimate the change in Kendall's tau and reported the pairs having a p-value smaller than 0.05.

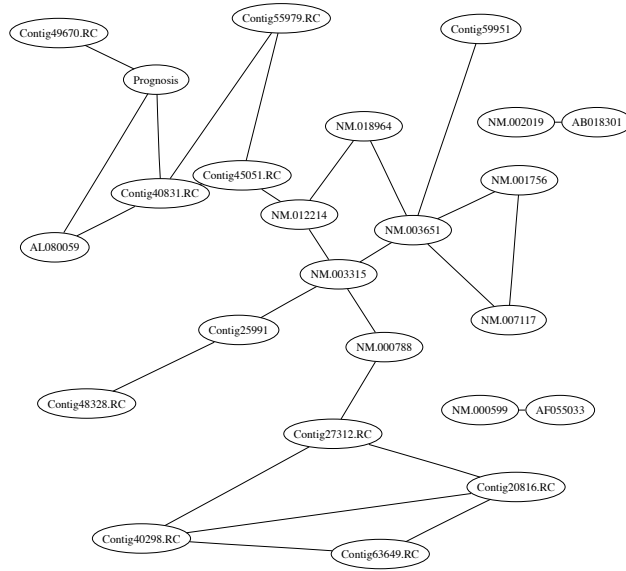


Figure 6: Pairs of covariates that show strong dependencies in the breast cancer prognosis data. We considered only the 23 predictive genes we identified together with the binary response variable prognosis.

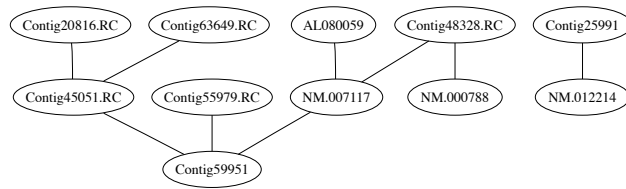


Figure 7: Liquid association network for prognosis in the breast cancer prognosis data.

7.3 Leukemia data

The leukemia dataset of Golub et al. (1999) comprises samples from patients with acute lymphoblastic leukemia (ALL) or acute myeloid leukemia (AML). The initial pre-processing of Yeung et al. (2005) leaves the expression levels of 3051 genes on 38 training samples and 34 test samples. We applied BMSS with $p_{max} = 4$, $c = 0.1$, $m = 500$, $s_1 = 1000000$, $s_2 = 100$ and three search replicates to identify high posterior probability logistic regressions associated with the binary response variable ALL/AML in the training data. BMSS returns a list of 563 regressions involving 94 genes. Based on this set of genes, all the 38 training samples are correctly predicted with a Brier score of 0.245 with a standard error of 0.14. In the test data, 33 samples (97.06%) are correctly classified with a Brier score of 1.45 and a standard error of 0.54. Figure 7.3 shows the prediction probabilities and their 80% credible intervals. Other results reported in the literature include Yeung et al. (2005) who make two classification errors in the test data based on 20 selected genes and a Brier score of 1.5. Lee et al. (2003) missclassifies one test sample based on the expression levels of five genes, while Nguyen and Rocke (2002) reports 1 – 3 missclassified test samples based on 50 – 1500 genes.

Next we employ BMSS with $p_{max} = 2$, $c = 0.001$, $m = 1000$, $s_1 = 10000$, $s_2 = 100$ and three search replicates to learn the structure of the dependency network involving the 94 genes and the binary variable ALL/AML. As before, we simulate from the resulting dependency network and estimate Kendall’s tau for any pair of the 95 covariates. There are 502 pairs of variables having a value of Kendall’s tau different than zero at a false discovery rate of 1% – see Figure 7.2. If we would have use the observed data to estimate Kendall’s tau, 3379 pairs of variables out of the total of 4465 pairs would have turned up significant at the same false discovery rate level. Figure 7.3 shows the eight neighbors of ALL/AML in the resulting graph as well as the edges among them. We remark that 35 edges are present out of the total of 36 possible edges indicating very strong dependencies in the variation of these nine variables.

Figure 7.3 gives the 55 pairs of genes that whose association in expression depends on ALL/AML. We measured association using Kendall’s tau estimated from the data simulated from the dependency network. The significance of the corresponding permutation tests was assessed at a false discovery rate of 1%.

7.4 Lymph node data

We predict lymph node positivity status (LNPos) in human breast cancer based on the expression levels of 4512 genes. This dataset has been previously analyzed in Hans et al. (2007) and Pittman et al. (2004). It comprises 100 low-risk (node-negative) samples and 48 high-risk (high node-positive). There are two additional predictors: estimated tumor size (in centimeters) and estrogen receptor status (binary variable determined by protein assays). We used BMSS with $p_{max} = 6$, $c = 0.25$, $m = 1000$, $s_1 = 100000$, $s_2 = 100$ and five search replicates to identify high posterior probability logistic regressions associated with LNPos involving the 4514 candidate predictors. BMSS returns 11 regressions in which 17 genes appear. The model-averaged prediction probabilities lead to 141 samples correctly predicted 95.27% with a Brier score of 9.03 having a standard error of 0.80. We further check the relevance of the regressions identified by performing leave-one-

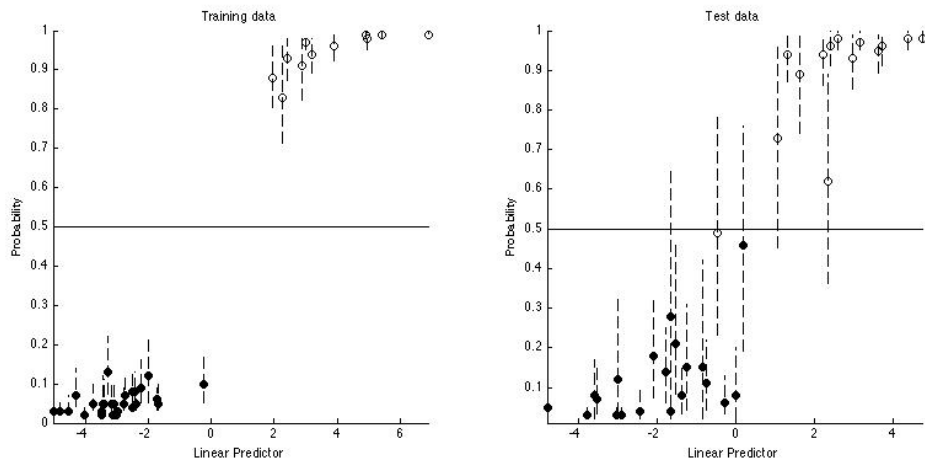


Figure 8: Prediction results for the leukemia data. The solid circles represent ALL patients, while the open circles represent AML patients.

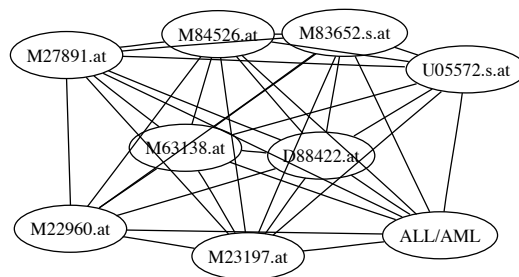


Figure 9: Eight genes having the strongest dependencies with the ALL/AML response variable in the leukemia data.

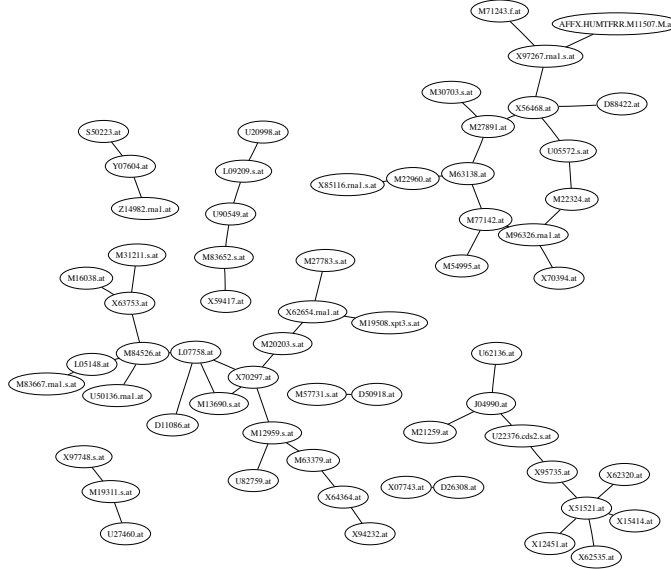


Figure 10: Liquid association network for ALL/AML in the leukemia data.

out cross-validation. The model probabilities as well as the prediction probabilities are recomputed after sequentially leaving out each of the 148 observations. The resulting model-averaged logistic regressions are used to predict the observation left out. The prediction results seem to hold: 132 samples (89.19%) are still correctly predicted. The corresponding Brier score is 13.47 with a standard error of 1.47. Figure 7.4 summarizes the observed prediction results.

Hans et al. (2007) show their prediction results based on the top ten logistic regression models they identify. These models involve 18 genes out of which four (RGS3, ATP6V1F, GEM, WSB1) are also present in our list of 17 genes. Their fitted prediction probabilities correctly predict 135 samples (91.22%), while their leave-one-out cross-validation predictive performance has a sensitivity of 79.2% and a specificity of 76%. Our leave-one-out predictive performance has a sensitivity of 85.42% and a specificity of 91% which indicates that we have discovered combinations of predictors with better predictive power.

We use BMSS with $p_{max} = 2$, $c = 0.001$, $m = 1000$, $s_1 = 10000$, $s_2 = 100$ and three search replicates to learn the structure of the dependency network involving the 17 predictive genes and LNPos. We simulate 25000 samples from this dependency network with a burn-in of 2500 samples and a gap of 100 between two consecutive saved samples. We use the resulting 250 samples to estimate Kendall's tau for any pair of the 18 covariates. There are 30 pairs of variables having a value of Kendall's tau different than zero at a false discovery rate of 1% – see Figure 7.4. We note that we would have identified 77 Kendall's tau coefficients if we would have used the observed data for estimation. The two neighbors of LNPos in Figure 7.4 are RGS3 and ATP6V1F that also belong to the list of top 18 genes of Hans et al. (2007).

We used the simulated data from the dependency network to identify the gene-gene interactions that were dependent on LNPos. Figure 7.4 shows the eight pairs whose p-value was below 0.05. We remark the triangle involving the genes ATP6V1F, GEM and WSB1. All three were also present

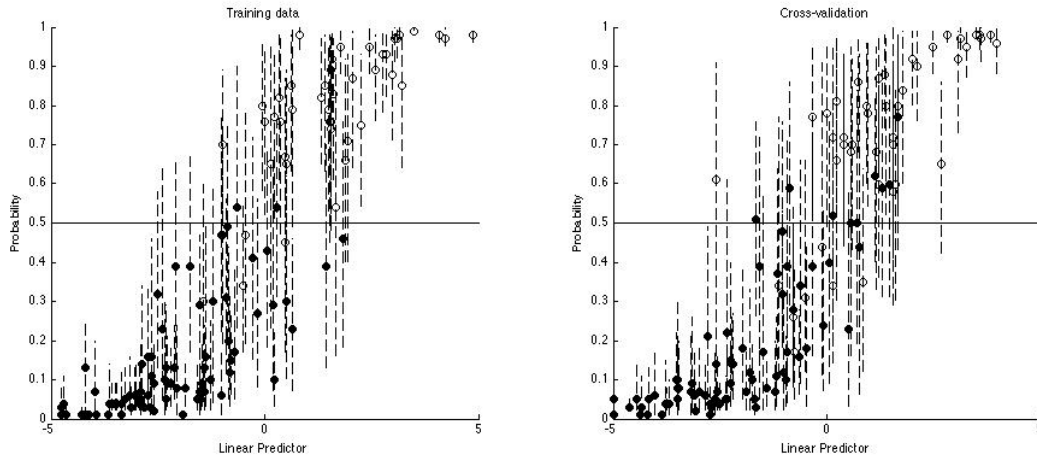


Figure 11: Prediction results for the leukemia lymph node data. The solid circles represent low-risk patients, while the open circles represent high-risk patients.

in the list of Hans et al. (2007). *ATP6V1F* is one of the two genes showing a strong dependence with LNPos in Figure 7.4. There is an edge between *GEM* and *WSB1* in both Figures 7.4 and 7.4 which implies that their dependence in expression is strong and is influenced by LNPos.

8 Discussion

The methodology we developed is relevant in two different albeit related areas. First of all, we proposed a stochastic search algorithm called BMSS for linear regression models that explores the space of candidate models more efficiently than other related model determination methods. We showed how BMSS performs for normal linear regressions and logistic regressions for several high-dimensional datasets. The classifiers constructed through Bayesian model averaging from the set of regressions identified by BMSS hold their performance for out-of-sample prediction. We do not discuss our choice of priors for regression parameters given in Appendixes A and B because we believe that this is only one choice among many other choices available in the literature. Our priors work well for the applications described in this paper, but we would not be surprised to see that other priors perform even better. We stress that our contribution is the BMSS algorithm. Our procedure can be employed with any other choice of prior parameters as long as the marginal likelihood of each regression model can be explicitly computed or at least accurately approximated in a reasonable amount of time. For example, regression models for discrete variables with more than two categories can also be determined using BMSS.

Second of all, we proposed using dependency networks to learn genetic networks from the data. We showed that the BMSS algorithm can be used to reduce the set of potential variables that should be included in the resulting focused network that is relevant for a given target variable. Focused networks alleviate the huge computational burden implied by constructing and exploring networks with hundreds or thousands of variables. BMSS is employed again when learning the dependency

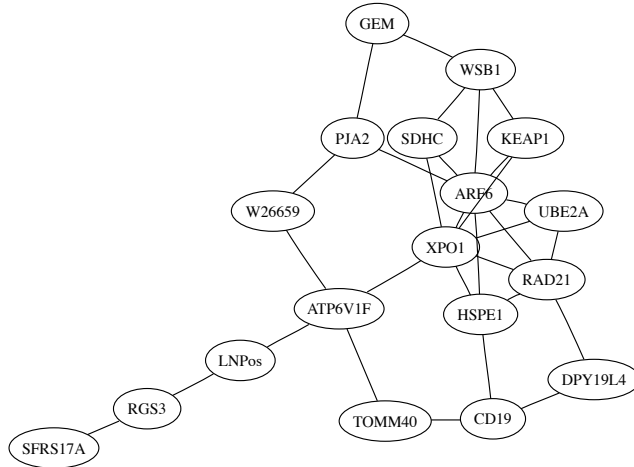


Figure 12: Pairs of covariates that show strong dependencies in the lymph node data. We considered only the 17 predictive genes we identified together with the binary response variable LNPos.

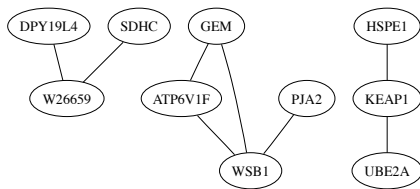


Figure 13: Liquid association network for LNPos in the leukemia lymph node data.

networks that involve the smaller set of variables associated with the target.

The advantages of our approach are as follows: (i) the edges of the network are no longer restricted to linear associations; (ii) any combination of continuous and discrete variables can be accommodated in a coherent manner; (iii) there is no need to assume a multivariate normal model for expression datasets as required by the Gaussian graphical models; (iv) the approach scales to high-dimensional datasets since the regression models associated with each variable can be learned independently – possibly on many computing nodes if a cluster of computers is available; (v) sparsity constraints can be specified in a straightforward manner. Model uncertainty is explicitly taken into account when sampling from the dependency network since we model the conditional distribution of each variable as a mixture of regressions. We showed that BMSS can be used to determine Gaussian graphical models on two datasets that are not relevant for biological applications, but have been recently analyzed with the best covariance selection techniques currently existent in the literature. Our dependency network approach can certainly be employed in the determination of Gaussian graphical models networks. Nevertheless, the association and liquid association networks defined in Section 5 are less restrictive in terms of the statistical assumptions they require. We hope that this will translate into an increased potential to provide valuable insights into the underlying biological phenomena.

Acknowledgments

The work of the author was supported in part by Grant R01 HL092071 from the National Institute of Health and by a seed grant from the Center of Statistics and the Social Sciences, University of Washington. The author would like to thank Ka Yee Yeung who provided the data for the breast cancer and leukemia examples, Chris Hans who provided the lymph node data and Jianhua Huang who provided the Call Center data.

Appendix A: Bayesian inference for normal regression

Let $Y = X_1$ be a continuous response variable and $X_{-1} = (X_2, \dots, X_p)$ be the vector of explanatory variables. Denote by D_1 the first column of the $n \times p$ design matrix D , by D_{-1} the columns $2, \dots, p$ of D . To keep the notation simple we assume that all the explanatory variables are present in the regression $[1|(2 : p)]$ with coefficients $\beta = (\beta_2, \dots, \beta_p)$. We center and scale the observed covariates such that their sample means are zero and their sample standard deviations are one. We assume $p(Y|X_{-1} = x) = N(x^T \beta, \sigma^2)$. The prior for σ^2 is $p(\sigma^2) = \text{IG}((p + 2)/2, 1/2)$ and, conditional on σ^2 , the regression coefficients have independent priors $p(\beta_j) = N(0, \sigma^2)$, $j = 2, \dots, p$. Dobra et al. (2004) show that the corresponding posterior distributions are

$$\begin{aligned} p(\sigma^2|D) &= \text{IG}\left((n + p + 2)/2, \left(1 + D_1^T D_1 - D_1^T D_{-1} M^{-1} D_{-1}^T D_1\right)\right), \\ p(\beta|\sigma^2, D) &= N_{p-1}\left(M^{-1} D_{-1}^T D_1, \sigma^2 M^{-1}\right), \end{aligned}$$

where $M = I_{p-1} + D_{-1}^T D_{-1}$. The marginal likelihood of $[1|(2 : p)]$ therefore given by

$$p(D|[1|(2 : p)]) = \frac{\Gamma((n+p+2)/2)}{\Gamma((p+2)/2)} (\det M)^{-1/2} \left(1 + D_1^T D_1 - D_{-1} M^{-1} D_{-1}^T D_1\right)^{-(n+p+2)/2}.$$

Appendix B: Bayesian inference for logistic regression

Let $Y = X_1$ be a binary response variable. We follow the notations from Appendix A. We denote by D^i the i -th row of the design matrix D and define $D_{i,p+1} = 1$, for $i = 1, \dots, n$. The coefficients of the regression $[1|(2 : p)]$ are $\beta = (\beta_2, \dots, \beta_p)$ and β_{p+1} – the intercept term. We center and scale the explanatory variables X_{-1} such that their sample means are zero and their sample standard deviations are one. We assume that $p(Y|X_{-1} = x) = \mathcal{B}(1, g(\beta, \beta_{p+1}, x))$ with $g(\beta, \beta_{p+1}, x) = (1 + \exp(-x^T \beta - \beta_{p+1}))^{-1}$ and that the regression coefficients have independent priors $p(\beta_j) = N(0, 1)$, $j = 2, \dots, p+1$. The posterior distribution of β is therefore given by

$$p(\beta, \beta_{p+1}|D) = \frac{1}{p(D|[1|(2 : p)])} \exp(l^D(\beta, \beta_{p+1})),$$

where

$$l^D(\beta, \beta_{p+1}) = -\frac{p}{2} \log(2\pi) - \frac{1}{2}(\beta^T \beta + \beta_{p+1}^2) + \sum_{i=1}^n [D_{i1} \log(g(\beta, \beta_{p+1}, D^i)) + (1 - D_{i1}) \log(1 - g(\beta, \beta_{p+1}, D^i))],$$

and $p(D|[1|(2 : p)]) = \int_{\mathfrak{R}^{p+1}} \exp(l^D(\beta, \beta_{p+1})) \prod_{j=2}^{p+1} d\beta_j$ is the marginal likelihood. The Laplace approximation (Tierney and Kadane, 1986) to $p(D|[1|(2 : p)])$ is

$$p(D|\widehat{[1|(2 : p)]}) = (2\pi)^{\frac{p}{2}} l^D(\widehat{\beta}, \widehat{\beta}_{p+1}) [H^D(\widehat{\beta}, \widehat{\beta}_{p+1})]^{-1/2},$$

where $(\widehat{\beta}, \widehat{\beta}_{p+1}) = \operatorname{argmax}_{(\beta, \beta_{p+1}) \in \mathfrak{R}^{p+1}} l^D(\beta, \beta_{p+1})$ is the posterior mode and H^D is the $p \times p$ Hessian matrix associated with l^D . The gradient of $l^D(\beta, \beta_{p+1})$ is $h^D(\beta, \beta_{p+1}) = \left(h_j^D(\beta, \beta_{p+1})\right)_{1 \leq j \leq p}$

where $h_j^D(\beta, \beta_{p+1}) = -\beta_{j+1} + \sum_{i=1}^n (D_{i1} - g(\beta, \beta_{p+1}, D^i)) D_{i,j+1}$, $j = 1, \dots, p$. It follows that the entries of $H^D(\beta, \beta_{p+1})$ are

$$H_{j,k}^D(\beta, \beta_{p+1}) = \sum_{i=1}^n (1 - g(\beta, \beta_{p+1}, D^i)) g(\beta, \beta_{p+1}, D^i) D_{i,j+1} D_{i,k+1} + \delta_{jk},$$

where $\delta_{jk} = 1$ if $j = k$ and $\delta_{jk} = 0$ if $j \neq k$.

The posterior mode $(\widehat{\beta}, \widehat{\beta}_{p+1})$ is determined using the Newton-Raphson algorithm that produces a sequence $(\beta^0, \beta_{p+1}^0) = 0, (\beta^1, \beta_{p+1}^1), \dots, (\beta^k, \beta_{p+1}^k), \dots$ such that

$$(\beta^{k+1}, \beta_{p+1}^{k+1})^T = (\beta^k, \beta_{p+1}^k)^T + \left[H^D(\beta^k, \beta_{p+1}^k)\right]^{-1} h^D(\beta^k, \beta_{p+1}^k), \quad k \geq 0.$$

Sampling from the posterior distribution $p(\beta, \beta_{p+1}|D)$ is done with the Metropolis-Hastings algorithm. At iteration k , generate $(\tilde{\beta}, \tilde{\beta}_{p+1})^T \sim N_{p+1} \left((\beta^k, \beta_{p+1}^k)^T, H^D(\hat{\beta}, \hat{\beta}_{p+1}) \right)$. Set $(\beta^{k+1}, \beta_{p+1}^{k+1}) = (\tilde{\beta}, \tilde{\beta}_{p+1})$ with probability

$$\min \left(1, \exp \left(l^D(\tilde{\beta}, \tilde{\beta}_{p+1}) - l^D(\beta^k, \beta_{p+1}^k) \right) \right).$$

Otherwise set $(\beta^{k+1}, \beta_{p+1}^{k+1}) = (\beta^k, \beta_{p+1}^k)$.

References

- Arnold, B. C., Castillo, E., and Sarabia, J. M. (2001). “Conditionally specified distributions: an introduction.” *Statistical Science*, 16, 249–274.
- Berger, J. O. and Molina, G. (2005). “Posterior model probabilities via path-based pairwise priors.” *Statistica Neerlandica*, 59, 3–15.
- Besag, J. (1974). “Spatial interaction and the statistical analysis of lattice systems (with Discussion).” *J. R. Statist. Soc. B*, 36, 192–236.
- Besag, J. and Kooperberg, C. (1995). “On conditional and intrinsic autoregressions.” *Biometrika*, 82, 733–746.
- Bickel, P. J. and Levina, E. (2008). “Regularized estimation of large covariance matrices.” *Ann. Statist.*, 36, 199–227.
- Butte, A. J., Tamayo, P., Slonim, D., Golub, T. R., and Kohane, I. S. (2000). “Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks.” *Proceedings of the National Academy of Sciences*, 97, 12182–12186.
- Carlin, B. P. and Chib, S. (1995). “Bayesian Model Choice via Markov Chain Monte Carlo.” *Journal of the Royal Statistical Society, Series B*, 57, 473–484.
- Castelo, R. and Roverato, A. (2006). “A robust procedure for Gaussian graphical model search from microarray data with p larger than n .” *Journal of Machine Learning Research*, 7, 2621–2650.
- Chipman, H. (1996). “Bayesian variable selection with related predictors.” *Canad. J. Statist.*, 24, 17–36.
- Chipman, H., George, E. I., and McCulloch, R. E. (2001). “The Practical Implementation of Bayesian Model Selection (with discussion).” In *Model Selection*, ed. P. Lahiri, 66–134. IMS: Beachwood, OH.
- Clyde, M. and George, E. I. (2004). “Model Uncertainty.” *Statistical Science*, 19, 81–94.

- Dempster, A. P. (1972). “Covariance selection.” *Biometrics*, 28, 157–75.
- Dobra, A., Hans, C., Jones, B., Nevins, J. R., Yao, G., and West, M. (2004). “Sparse graphical models for exploring gene expression data.” *Journal of Multivariate Analysis*, 90, 196–212.
- Drton, M. and Perlman, M. D. (2004). “Model selection for Gaussian concentration graphs.” *Biometrika*, 91, 591–602.
- Dudoit, S., Fridlyand, J., and Speed, T. P. (2002). “Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data.” *Journal of the American Statistical Association*, 97, 77–87.
- Fernández, C., Ley, E., and Steel, M. F. (2003). “Benchmark Priors for Bayesian Model Averaging.” *Journal of Econometrics*, 75, 317–343.
- Friedman, N. (2004). “Inferring cellular networks using probabilistic graphical models.” *Science*, 30, 799–805.
- Furnival, G. M. and Wilson, R. W. (1974). “Regression by Leaps and Bounds.” *Technometrics*, 16, 499–511.
- Gelman, A. and Meng, X.-L. (1991). “A note on bivariate distributions that are conditionally normal.” *The American Statistician*, 45, 125–126.
- Gelman, A. and Speed, T. P. (1993). “Characterizing a joint probability distribution by conditionals.” *J. R. Statist. Soc. B*, 55, 185–188.
- Geman, S. and Geman, D. (1984). “Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images.” *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 6, 721–742.
- 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.
- Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D., and Lander, E. S. (1999). “Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring.” *Science*, 286, 531–537.
- Green, P. J. (1995). “Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination.” *Biometrika*, 82, 711–732.
- Hans, C., Dobra, A., and West, M. (2007). “Shotgun Stochastic Search for “Large p” Regression.” *Journal of the American Statistical Association*, 102, 507–516.

- Heckerman, D., Chickering, D. M., Meek, C., Rounthwaite, R., and Kadie, C. (2000). “Dependency networks for inference, collaborative filtering and data visualization.” *Journal of Machine Learning Research*, 1, 1–48.
- Hobert, J. P. and Casella, G. (1998). “Functional compatibility, Markov chains, and Gibbs sampling with improper posteriors.” *Journal of Computational and Graphical Statistics*, 7, 42–60.
- Huang, J., Liu, N., Pourahmadi, M., and Liu, L. (2006). “Covariance matrix selection and estimation via penalized normal likelihood.” *Biometrika*, 93, 85–98.
- Jones, B. and West, M. (2005). “Covariance decomposition in undirected Gaussian graphical models.” *Biometrika*, 92, 779–786.
- Kass, R. and Raftery, A. E. (1995). “Bayes factors.” *J. Am. Statist. Assoc.*, 90, 773–95.
- Kohn, R., Smith, M., and Chan, D. (2001). “Nonparametric regression using linear combinations of basis functions.” *Statist. Comp.*, 11, 313–322.
- Lee, K. E., Sha, N., Dougherty, E. R., Vanucci, M., and Mallick, B. K. (2003). “Gene Selection: a Bayesian Variable Selection Approach.” *Bioinformatics*, 19, 90–97.
- Li, H. and Gui, J. (2006). “Gradient directed regularization for sparse Gaussian concentration graphs, with application to inference of genetic networks.” *Biostatistics*, 2, 302–317.
- Li, K.-C. (2002). “Genome-wide coexpression dynamics: theory and application.” *Proc. Nat. Acad. Sci.*, 99, 16875–16880.
- Liang, F., Paulo, R., Molina, G., Clyde, M., and Berger, J. O. (2008). “Mixtures of g-priors for Bayesian Variable Selection.” *Journal of the American Statistical Association*, 103, 410–423.
- Madigan, D. and York, J. (1995). “Bayesian Graphical Models for Discrete Data.” *International Statistical Review*, 63, 215–232.
- Meinshausen, N. and Bühlmann, P. (2006). “High-dimensional graphs with the Lasso.” *Ann. Statist.*, 34, 1436–62.
- Nelsen, R. B. (1999). *An Introduction to Copulas*, vol. 139 of *Lecture Notes in Statistics*. Springer-Verlag.
- Nguyen, D. V. and Rocke, D. M. (2002). “Tumor Classification by Partial Least Squares Using Microarray Gene Expression Data.” *Bioinformatics*, 18, 39–50.
- Nott, D. and Green, P. (2004). “Bayesian variable selection and the Swendsen-Wang algorithm.” *Journal of Computational and Graphical Statistics*, 13, 1–17.
- Pittman, J., Huang, E., Dressman, H., Horng, C. F., Cheng, S. H., Tsou, M. H., Chen, C. M., Bild, A., Iversen, E. S., Huang, A. T., Nevins, J. R., and West, M. (2004). “Integrated modeling of clinical and gene expression information for personalized prediction of disease outcomes.” *Proc. Natl. Acad. Sci.*, 101, 8431–8436.

- Raftery, A. E., Madigan, D., and Hoeting, J. (1997). “Bayesian Model Averaging for Linear Regression Models.” *Journal of the American Statistical Association*, 92, 1197–1208.
- Rajaratnam, B., Massam, H., and Carvalho, C. M. (2008). “Flexible covariance estimation in graphical Gaussian models.” *Ann. Statist.*. To appear.
- Schafer, J. and Strimmer, K. (2005). “An empirical Bayes approach to inferring large-scale gene association networks.” *Bioinformatics*, 21, 754–764.
- Scott, J. G. and Berger, J. O. (2006). “An exploration of aspects of Bayesian multiple testing.” *J. Stat. Plan. Inf.*, 136, 2144–2162.
- Segal, E., Shapira, M., Regev, A., Pe’er, D., Botstein, D., Koller, D., and Friedman, N. (2003). “Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data.” *Nature Genetics*, 34, 166–176.
- Shen, H. and Huang, J. Z. (2005). “Analysis of call center arrival data using singular value decomposition.” *Applied Stochastic Models in Business and Industry*, 21, 251–263.
- Speed, T. P. and Kiiveri, H. T. (1986). “Gaussian Markov distributions over finite graphs.” *Ann. Statist.*, 14, 138–150.
- Steuer, R., Kurths, J., Fiehn, O., and Weckwerth, W. (2003). “Observing and interpreting correlation in metabolomic networks.” *Bioinformatics*, 19, 1019–1026.
- Tierney, L. and Kadane, J. (1986). “Accurate Approximations for Posterior Moments and Marginal Densities.” *J. Amer. Statist. Assoc.*, 81, 82–86.
- Tusher, V. G., Tibshirani, R., and Chu, G. (2001). “Significance Analysis of Microarrays Applied to the Ionizing Radiation Response.” *Proceedings of the National Academy of Sciences*, 98, 5116–5121.
- van’t Veer, L. J., Hongyue, D., van de Vijver, M. J., He, Y. D., Hart, A. A. M., Mao, M., Peterse, H. L., van der Kooy, K., Marton, M. J., Witteveen, A. T., Schreiber, G. J., Kerckhoven, R. M., Roberts, C., Linsley, P. S., Bernards, R., and Friend, S. H. (2002). “Gene Expression Profiling Predicts Clinical Outcome of Breast Cancer.” *Nature*, 415, 530–536.
- Wille, A. and Bühlmann, P. (2006). “Low-order conditional independence graphs for inferring genetic networks.” *Statistical applications in genetics and molecular biology*, 5, article 1.
- Yeung, K., Bumgarner, R., and Raftery, A. (2005). “Bayesian Model Averaging: Development of an Improved Multi-class, Gene Selection and Classification Tool for Microarray Data.” *Bioinformatics*, 21, 2394–2402.
- Yu, J., Smith, V. A., Wang, P. P., Hartemink, A. J., and Jarvis, E. D. (2004). “Advances in Bayesian network inference for generating causal networks from observational biological data.” *Bioinformatics*, 20, 3594–3603.

Yuan, M. and Lin, Y. (2007). “Model selection and estimation in the Gaussian graphical model.” *Biometrika*, 94, 19–35.

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.