STATISTICAL COMPUTATION FOR MODEL SPACE EXPLORATION IN HIGH-DIMENSIONAL PROBLEMS

by

Liang Zhang

Department of Statistical Science
Duke University

Date: _____

Approved:

_____
Dr. Mike West, Supervisor

_____
Dr. David Dunson

_____
Dr. Jerome Reiter

_____
Dr. Mark Huber

Dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Department of Statistical Science
in the Graduate School of
Duke University

2008

ABSTRACT

STATISTICAL COMPUTATION FOR MODEL SPACE EXPLORATION IN HIGH-DIMENSIONAL PROBLEMS

by

Liang Zhang

Department of Statistical Science
Duke University

Date: _____

Approved:

_____
Dr. Mike West, Supervisor

_____
Dr. David Dunson

_____
Dr. Jerome Reiter

_____
Dr. Mark Huber

An abstract of a dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Department of Statistical Science
in the Graduate School of
Duke University

2008

Abstract

The increasing dimension of data sets and resulting parameter spaces in modern statistical models raise demands for new methods of statistical computation for scalability and efficiency. Model space exploration, in particular, is an increasingly important and challenging area. This dissertation focuses on graphical and regression model space exploration arising in statistical models for high-dimensional data.

In contrast to the traditional graphical model space exploration algorithms, which focus on exploring of the graphical model of all variables, this dissertation develops and evaluates an innovative concept: local graphical model search. Local graphical model search algorithms apply to problems where we are interested in a single targeted gene, Y, among thousands of genes in the gene expression data, for example, and wish to understand the graphical structure of Y and its neighborhood. Usual (global) graphical model search methods will not be efficient and precise in such problems. To implement local graphical model search, this dissertation employs stochastic search algorithms subject to restrictions on the model space as well as develops a novel Metropolis-Hasting method referred to as targeted Metropolis-Hastings (TMH). TMH is empirically compared with the usual Metropolis-Hasting (UMH) algorithm in terms of local convergence and the convergence of the stationary "local edge" inclusion distributions. The performances of the methods developed herein are tested with simulation studies and high-dimensional cardiovascular genomics data.

Variable selection in generalized linear models with many candidate covariates, is a very challenging problem and widely developed in many applications. Because current stochastic regression model search algorithms rely on conjugacy, they are not appropriate for generalized linear models without use of approximation methods for

the marginal likelihood. This dissertation studies two possible marginal likelihood approximation methods: variational Bayes and Laplace approximation. These methods are compared in simulation studies and then applied to the problem of predicting conception using data on timing of intercourse in the menstrual cycle.

The final topic of this dissertation concerns large-scale modeling of high-dimensional data in a problem of forecasting click events with content match data in computational advertising. This challenging problem of modeling and computation generally arises in internet advertising, and the study discussed in this dissertation is part of a collaboration with Yahoo! Research. In models that reflect the hierarchy of the high-dimensional data structure, Kalman filtering and Expectation-Maximization algorithms aid in providing scalability without losing much precision in generating relevant, applied computational approaches. The studies using both simulated and real "content match" data sets demonstrate the feasibility, utility and efficacy of the developed approach.

Acknowledgements

I would like to acknowledge all the professors in Department of Statistical Science who have taught, guided, and supported me in the past four years. In particular, I would like to thank my advisor, Mike West, for his excellent guidance in my Ph.D research, great patience in teaching me Bayesian statistics, and continuous support during the development of this dissertation. I also want to thank David Dunson, who collaborates and shares many innovative ideas with me.

I would like to give thanks to Adrian Dobra, for his guidance and collaboration during my early years as a Ph.D student.

I want to say thanks to Deepak Agarwal for guiding me during my summer practical training at Yahoo! Research.

I would like to thank Mark Huber and Carlos Carvalho, for plenty of useful discussions about the work in this dissertation. I want to thank David Seo, for his useful remarks of the biological discussions in this dissertation. I am grateful for Matt Heaton and Scott, L. Schwartz, for their comments during the development of this work. Thanks also go to Zhenglei Gao, Joyee Ghosh, Chunlin Ji, Fei Liu, Jinqin Luo, Robin Mitra, Zhi Ouyang, Huiyan Sang, Hao Wang, Yuhong Wu, Peidong Yu, Lingchu Yu, Jie Hu, Junhua Liu and all my other colleagues and friends. I would never have had such a wonderful time at Duke without you.

I am extremely grateful for my wife, Ying Wang, who always offer me love, supports, and encouragements. Without her, completing this dissertation would have been impossible.

Finally, I sincerely want to say thank you to my parents, grandpa and grandma. There are no words for me to describe how grateful I am for your continuous love. I

really miss you and want to dedicate this dissertation to you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Scientific data collected by scientists and researchers is increasingly high-dimensional and highly structured. From gene expression data in genomics and bioinformatics to Internet advertising observational studies, the necessary statistical analyses involve both methodological and computational challenges. These challenges include requirements of not only inference and forecasting precision, but also scalability and computational efficiency. Bayesian analysis is ideal in complex and hierarchical modeling for high dimensional data. The development of Monte Carlo methods, especially Markov chain Monte Carlo (MCMC), provide extremely important computational advances in the exploration of complex posterior distributions for high-dimensional statistical models. Other computational advances, including parallel and cluster computing, have also given statisticians and computer scientists freedom to apply stochastic simulation methods as well as advanced search algorithms to implement increasingly realistic statistical models.

Due to the challenges from high-dimensional statistical models, model space exploration becomes an important area in statistical analysis. In many practical problems, there are usually more than one model under consideration for the observed data. Sometimes, the high dimension of the problem results in very large model space. The Bayesian approach to model space exploration proceeds by assigning prior probabilities to all models under consideration, and calculates the posterior probabilities of the models conditional on the data. The set of posterior probabilities is then used as a criteria in model selection and search. This dissertation addresses a number of computational problems in model space exploration. Two statistical modeling areas are discussed: graphical models and regression models.

Graphical models, with their origin in statistical physics (Gibbs, 1902), genetics (Wright,

1921, 1923, 1934), economics and social sciences (Wold, 1954; Blalock, 1971), and several other scientific areas, provide useful graphical and visual representations of the conditional independence relationships among a set of variables. The core idea of graphical models relies on the conditional independence of random variables (Dawid, 1979, 1980; Dawid and Lauritzen, 1993) and the Markov property (Dawid and Lauritzen, 1993; Lauritzen, 1996); hence, a factorization of the full joint distribution of a set of variables corresponds to a decomposition of a graph, with a 1-1 mapping of nodes to the variables. Chapter 2 gives a review of the basic concepts of graphical models and approaches to Bayesian undirected graphical model search and selection, including the use of MCMC sampling methods and parallel computing for parallelized stochastic model search (Hans, 2005; Hans *et al.*, 2007; Jones *et al.*, 2005).

Chapter 3 and 4 introduce and develop new concepts, methods, and computational strategies for a new approach called "local graphical model search." The core idea of local graphical model search is, when interest lies in predicting and evaluating a relatively small subset of a very large number of original variables, developing full multivariate analysis of all variables is necessary but raises challenging questions of both statistical and computational efficiencies. Chapter 3 addresses this general problem of properly understanding "local graphical model structure" in the context of needing to explore the "global model" due to uncertainty about which variables live in the "local neighborhood" of interest. Specifically, Chapter 3 discusses the simple idea of exploring graphs that include only edges incident to the target variable, or connecting two variables living in the "local neighborhood." As a further development, Chapter 4 develops innovations including a innovative MCMC approach - referred to as "Targeted Metropolis-Hasting" (TMH) methodology. This is discussed theoretically in terms of its properties and convergence rates compared with usual Metropolis-Hastings methods in graphical model search, empirically, and practically. In addition to simulation studies in both chapters, the methods are applied to local graphical modeling studies in cardiovascular genomics, in particular, a study of identifying statistical

association structure among genes related to atherosclerotic disease risk (Seo *et al.*, 2007).

Compared to graphical models, regression models are currently more widely used in many scientific fields. In regression modeling with high-dimensional covariates, variable selection, a special case of model selection, is a key issue (George, 2000; Miller, 2002). In the context of linear models, plenty of different approaches has been proposed, applied, and discussed, such as Zellner (1986), George and McCulloch (1993), George and McCulloch (1997), Clyde and George (2004), and Hans *et al.* (2007). A more generalized class of linear models, generalized linear models (GLM) (McCullagh and Nelder, 1989), has also recently involved model uncertainty and variable selection problems, such as Raftery (1996), Chen *et al.* (1999), Dellaportas and Forster (1999), Ntzoufras *et al.* (2000), and Hans *et al.* (2007). Developments in Chapter 5 are motivated by data drawn from the European Study of Daily Fecundability (ESDF), where the focus is on analyzing the relationship between intercourse timing and fecundability (Dunson *et al.*, 1999; Wilcox *et al.*, 2000; Dunson *et al.*, 2002). The problem of many potential covariates requires some form of variable selection model search. Stochastic model search algorithms (Hans, 2005; Hans *et al.*, 2007), even when running in parallel, are improved by efficient methods of computing marginal likelihoods for each candidate model. In generalized linear models, some form of marginal likelihood approximation is therefore needed. Chapter 5 compares variational Bayes methods (Jaakkola and Jordan, 2000) with Laplace approximations (DiCiccio *et al.*, 1997; Hans, 2005) for approximating marginal likelihoods in logistic regression models. Laplace approximations are often found to be surprisingly better than variational Bayes methods, especially in view of a well-known fact that variational Bayes methods tend to underestimate posterior variances (Wang and Titterington, 2005; Consonni and Marin, 2007). This comparison is illustrated in simulation studies and finally, in the application to the problem of predicting conception using data on timing of intercourse in the menstrual cycle.

Chapter 6 concerns an applied research study with goals and challenges related to those of Chapter 5. Specifically, Chapter 6 discusses inference and forecasting problems in multi-

level hierarchical models, with a focus on logistic regression models with random effects (Lindley, 1971, 1972; Lindley and Smith, 1972; Laird and Ware, 1982; Gelman and Hill, 2007) where the data set is massive and the set of random effects parameters is high-dimensional. This research is motivated by the problem referred to as "content match" in Internet advertising (Agarwal *et al.*, 2007), where millions of observations are generated every month. The goal is to rank advertisements in terms of relevance when a user visits a publisher web page. I assume historical click-feedback is a proxy of relevance and develop regression models to estimate click-through rates when an advertisement is shown on a web-page at a particular position. Chapter 6 develops approaches that use the information from the page and advertisement taxonomies to build generalized, multi-level variance component models. Due to the massive and continuously increasing data sets involved, traditional statistical inference approaches such as Pinheiro and Bates (2000), Breslow and Clayton (1993), and Rodriguez and Goldman (2001), are not applicable as the required computations (e.g. matrix inversions) do not scale. I use creative approximations and Kalman filtering methods to reduce computations to feasible implementations. Studies using both simulated and real Yahoo! data sets demonstrate the feasibility, utility, and efficacy of the approaches and methods developed in Chapter 6.

In summary, this dissertation describes and discusses approaches to model selection, inference, and forecasting problems with high-dimensional and massive data sets. Extensions and future potential research directions are presented in Chapter 7.

Chapter 2

Gaussian Graphical Models

Graphical models provide graphical and visual representations of the conditional independence properties of a vector of random variables under a specified joint distribution. When the dimension of the vector is large, graphical models can provide useful insights via the break down of the high-dimensional joint distribution into components related to various low-dimensional joint distributions of subsets of the variables. As background, this chapter introduces the basic concepts of graphical modeling and Bayesian graphical model search and selection. This is followed by discussion of graphical model search algorithms for high-dimensional data, including approaches using shotgun stochastic search algorithms (SSS) as well as Markov chain Monte Carlo (MCMC) methods.

2.1 Basic Concepts and Notations of Graphical Models

A graphical model of a probability distribution provides a graphical representation of the implied conditional independencies (Whittaker, 1990; Lauritzen, 1996; Jones *et al.*, 2005; Carvalho, 2006). To begin, some graph theory notations are defined in the following list.

- *Vertex* : A vertex in a graphical model, also known as a *node* , corresponds to a random variable (1-1 mapping) under a joint distribution of a vector of random variables. The set of vertices is denoted by V.
- *Edge* : An element of $V \times V$.
- *Graph* : A graph G consists of a set of vertices (V) and a set of edges (E), and each

edge in set E joins two vertices in V. As a result, G can be denoted as $G = (V, E)$. $|V|$ is referred to as the size of G.

- *Path* : A sequence of vertices, where each pair of consecutive vertices are connected by an edge.

- *Subgraph* : A graph $G' = (V', E')$ is a subgraph of $G = (V, E)$, which is denoted as $G' \subseteq G$, if $V' \subseteq V$ and $E' \subseteq E$.

- *Directed/undirected edge* : An edge joining vertices a and b is called an *undirected* edge if $(a, b) \in E$ as well as $(b, a) \in E$, and an arrow pointing from a to b is called a *directed* edge if $(a, b) \in E$ but $(b, a) \notin E$.

- *Directed/undirected graph* : A graph G is called an *undirected graph* if all of the edges in E are undirected, and *directed graph* if each edge is directed.

- *Parent/child* : For a directed edge (a, b), a is said to be a *parent* of b, and b a *child* of a. The set of parents of a is denoted as $pa(a)$, and the set of children of a is denoted as $ch(a)$.

- *Neighbor* : Two vertices a and b are *adjacent* or *neighbors*, if undirected edge $(a, b) \in E$. The set of a's neighbors in G is denoted as $ne(a)$.

- *Complete graph* : A graph (subgraph) is *complete* if every pair of vertices is connected by a directed or undirected edge.

- *Clique* : A complete subgraph with maximal size is called a *clique* .

2.2 Undirected Graphical Models

2.2.1 Decomposition of a Graph

From now on G is assumed to be an undirected graph. For any disjoint subgraphs A, B, and C, if C is complete, $A \cup B \cup C = G$, and any path from a vertex in A to a vertex in B goes through C, then C is a *separator* of A and B. (A, B, C) form a decomposition of G. If neither A nor B is empty, the decomposition is *proper*. Iteratively, by proper decomposition and always choosing separators with minimal sizes, one can arrive at a sequence of subgraphs that cannot be further decomposed. Each subgraph in the sequence is a *prime component* of G. If all the prime components of a graph G are complete, G is a *decomposable* graph. Otherwise G is a *non-decomposable* graph. As an example, Figure 2.1 shows the decomposition process of a non-decomposable graph (Figure 2.1 (a)). The graph is non-decomposable because one prime component $\{4, 6, 7, 8\}$ is not complete. Two middle steps of the decomposition are shown in Figure 2.1 (b) and (c), and graph (d) contains all the prime components when the decomposition is finished. The graph is decomposed into four cliques and one non-complete prime component.

Given all the prime components $P_1, P_2, ..., P_k$ of G, if for any $i = 2, ..., k$, there exists a $j < i$ such that the separator $S_i = P_i \cap (\cup_{l<i} P_l) \subset P_j$, then $(P_1, S_2, P_2, ..., S_k, P_k)$ is a *perfect ordering* which represents the graph G. In Figure 2.1, a perfect ordering for the decomposition of this graph is: $\{P_1, S_2, P_2, S_3, P_3, S_4, P_4, S_5, P_5\}$, where $P_1 = \{1, 2, 3\}$, $S_2 = \{2, 3\}$, $P_2 = \{2, 3, 4\}$, $S_3 = \{4\}$, $P_3 = \{4, 9\}$, $S_4 = \{3, 4\}$, $P_4 = \{3, 4, 5, 6\}$, $S_5 = \{4, 6\}$, and $P_5 = \{4, 6, 7, 8\}$.

Now consider p scalar random variables in the vector $\mathbf{Y} = (y_1, ..., y_p)'$ with a specified joint distribution. When $\mathbf{Y}$ is associated with a graph G, the decomposition of G results in a mathematical factorization of the full joint distribution of $(y_1, ..., y_p)'$. Before showing this, we first introduce two definitions: The conditional independence of random variables (Dawid, 1979, 1980; Dawid and Lauritzen, 1993) and the Markov property (Dawid and

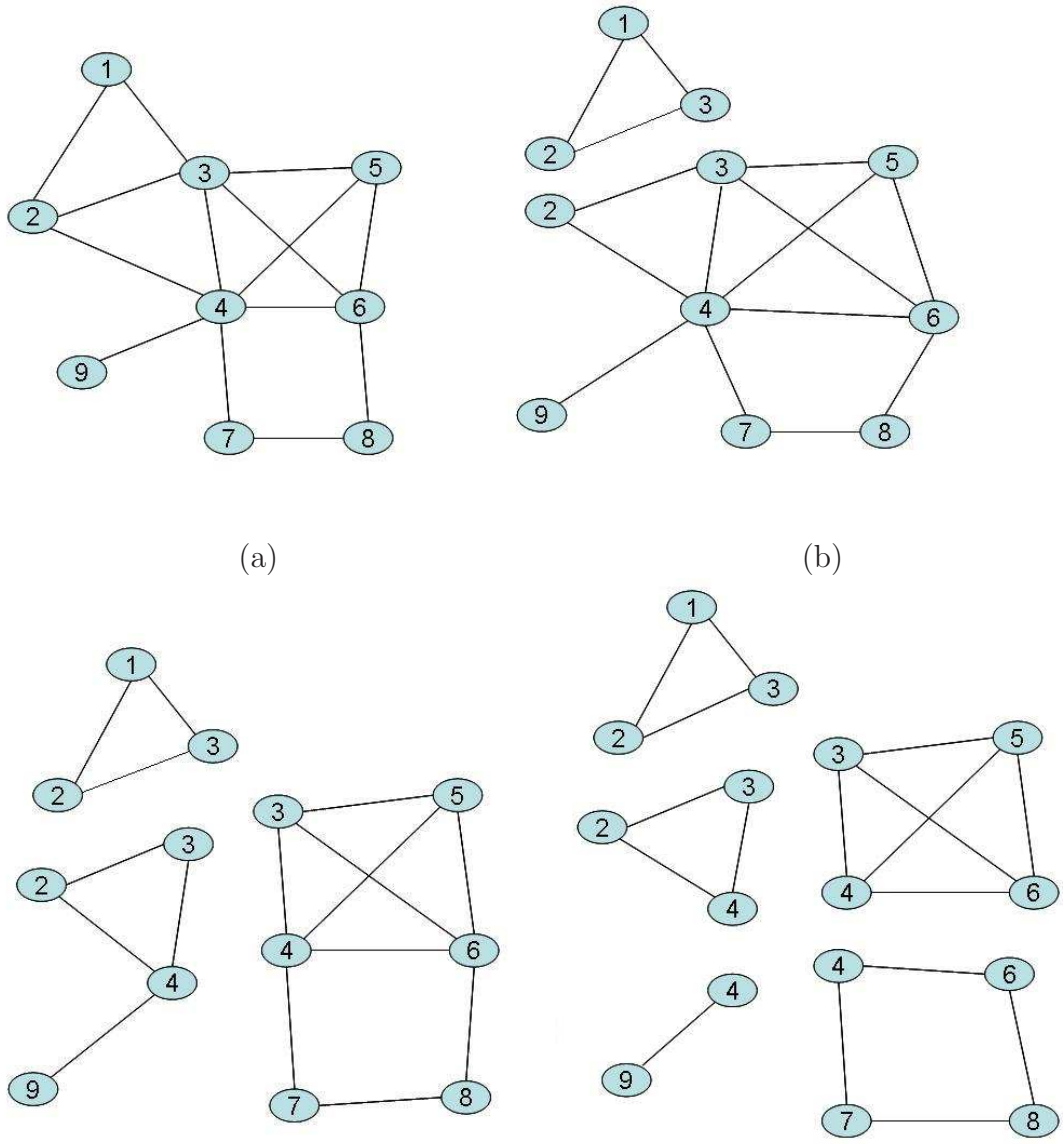


Figure 2.1: The decomposition of a non-decomposable graph.

Lauritzen, 1993; Lauritzen, 1996).

Definition 2.1. *(Dawid and Lauritzen, 1993) If X, Y, Z are random variables on a probability space $(\Omega, \mathscr{F}, P)$, and if for any measurable set A in the sample space of X, there exists a version of the conditional probability $P(X \in A|Y, Z)$ which is a function of Z alone, then X is* conditionally independent of Y given Z under P, *and can be written as $X \perp\!\!\!\perp Y|Z$.*

Consider a subset of vertices in a graph G, which is denoted as S. The set of random variables in vector $\mathbf{Y}$ that corresponds to vertices in S is denoted as $\mathbf{Y}_S$. The Markov property on undirected graphs is defined as:

Definition 2.2. *(Dawid and Lauritzen, 1993; Lauritzen, 1996)* Markov Property on undirected graphs: *A distribution P on V is called* Markov *over $G = (E, V)$ if for any triple (A, B, S) of disjoint subsets of V such that S separates A from B in G, $\mathbf{Y}_A \perp\!\!\!\perp \mathbf{Y}_B|\mathbf{Y}_S$.*

For a distribution P with the Markov property associated with a graph G, we can decompose G into a perfect ordering $(P_1, S_2, P_2, ..., S_k, P_k)$ as described. Then, iteratively (Hammersley and Clifford, 1968),

$$p(\mathbf{Y}|G) = \frac{\prod_{P_i \in P} p(\mathbf{Y}_{P_i}|G)}{\prod_{S_i \in S} p(\mathbf{Y}_{S_i}|G)}. \tag{2.1}$$

As a factorization of the full joint distribution of $\mathbf{Y} = (y_1, ..., y_p)'$, equation (2.1) plays an important role in graphical models. For example, in the Gaussian case, instead of requiring the huge number of parameters implied by a high-dimensional inverse Wishart prior on the covariance of $\boldsymbol{Y}$, we are now able to decrease the number of parameters by using sets of lower-dimensional inverse Wishart priors on the variance matrices of each of the prime components and separators, based on hyper-inverse Wishart distributions (Dawid and Lauritzen, 1993).

2.2.2 Undirected Gaussian Graphical Model Selection

Bayesian model comparison and selection involves computing the marginal posterior probabilities of the models given the observed data. For convenience, we call the models that have the highest posterior probabilities the "best" or "top" models, and use the term "score" to represent values that are proportional to the posterior probabilities after taking an exponential transformation. In the case when each model corresponds to a graph G, we are interested in posterior probabilities

$$p(G|\mathbf{Y}_{1:n}) \propto p(\mathbf{Y}_{1:n}|G)p(G), \tag{2.2}$$

where $\mathbf{Y}_{1:n}$ is p dimensional observed data with sample size equal to n, $\mathbf{Y}_{1:n} = \{\mathbf{y}_1, ...\mathbf{y}_n\}$, with $\mathbf{y}_i$ a p-dimensional observation.

From now on we take G to be decomposable. We see from equation (2.2) that, we need to specify the marginal likelihood $p(\mathbf{Y}_{1:n}|G)$ and the prior over the graph G respectively. Here, assume data $\mathbf{Y}_{1:n}$ are random samples from a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and variance matrix $\boldsymbol{\Sigma}$, i.e., $(\mathbf{Y}_{1:n}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Furthermore, without loss of generality, let $\boldsymbol{\mu} = \mathbf{0}$. Then,

$$p(\mathbf{Y}_{1:n}|G) = \int_{\boldsymbol{\Sigma}_G|G} p(\mathbf{Y}_{1:n}|\boldsymbol{\Sigma}_G)p(\boldsymbol{\Sigma}_G|G)d\boldsymbol{\Sigma}_G. \tag{2.3}$$

Following equation (2.1), $p(\mathbf{Y}_{1:n}|\boldsymbol{\Sigma}_G) = p(\mathbf{Y}_{1:n}|\boldsymbol{\Sigma}_G, G)$ has the same Markov property and factorizes as

$$p(\mathbf{Y}_{1:n}|\boldsymbol{\Sigma}_G) = \frac{\prod_{P_i \in P} p(\mathbf{Y}_{1:n,P_i}|\boldsymbol{\Sigma}_{P_i})}{\prod_{S_i \in S} p(\mathbf{Y}_{1:n,S_i}|\boldsymbol{\Sigma}_{S_i})}, \tag{2.4}$$

where $\mathbf{Y}_{1:n,P_i}$ (or $\mathbf{Y}_{1:n,S_i}$) is a submatrix of the sample matrix $\mathbf{Y}_{1:n}$ with the same n rows, and with column j of $\mathbf{Y}_{1:n}$ in $\mathbf{Y}_{1:n,P_i}$ ($\mathbf{Y}_{1:n,S_i}$) if and only if $j \in P_i$ (S_i). Also, in equation (2.4), $\boldsymbol{\Sigma}_{P_i}$ (or $\boldsymbol{\Sigma}_{S_i}$) is the component-marginal covariance matrix of $\mathbf{Y}_{1:n,P_i}$ (or $\mathbf{Y}_{1:n,S_i}$).

Wermuth (1976) showed that the *precision matrix* $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$ has the elements $\boldsymbol{\Omega}_{ij} = \boldsymbol{\Omega}_{ji} = 0$ if and only if $(i, j) \notin E(G)$. To specify the prior of the covariance matrix $\boldsymbol{\Sigma}_G$

associated with G, Dawid and Lauritzen (1993) introduced a family of Markov distributions for decomposable graphs, which are called the *hyper-inverse Wishart* (HIW) priors, denoted by $HIW(G, \delta, \Phi)$. Here, δ is the positive degree-of-freedom parameter and Φ is the positive definite location matrix. As in equation (2.1) and (2.4), the density of the HIW distribution $HIW(G, \delta, \Phi)$ also decomposes as

$$p(\mathbf{\Sigma}_G|G) = \frac{\prod_{P_i \in P} p(\mathbf{\Sigma}_{P_i}|G)}{\prod_{S_i \in S} p(\mathbf{\Sigma}_{S_i}|G)}, \tag{2.5}$$

where $\mathbf{\Sigma}_{P_i}$ has an inverse-Wishart prior $IW(\delta, \mathbf{\Phi}_{P_i})$ with the form (Giudici, 1996)

$$p(\mathbf{\Sigma}_{P_i}|G) = \frac{|\frac{\mathbf{\Phi}_{P_i}}{2}|^{\frac{\delta+|P_i|-1}{2}}}{\Gamma_{|P_i|}\left(\frac{\delta+|P_i|-1}{2}\right)} |\mathbf{\Sigma}_{P_i}|^{-\frac{\delta+2|P_i|}{2}} \exp\{-\frac{1}{2} tr(\mathbf{\Phi}_{P_i} \mathbf{\Sigma}_{P_i}^{-1})\}, \tag{2.6}$$

and where $\Gamma_k(a)$ is the multivariate gamma function

$$\Gamma_k(a) = \pi^{\frac{k(k-1)}{4}} \prod_{i=0}^{k-1} \Gamma\left(a - \frac{i}{2}\right). \tag{2.7}$$

Also, each $p(\mathbf{\Sigma}_{S_i}|G)$ has the same form as equation (2.6) with S_i substituting P_i.

Since G is decomposable, all the prime components in G are complete. The decomposability guarantees the fact that, while G determines which elements of $\mathbf{\Sigma}$ appear in the density of $\mathbf{Y}_{1:n}$ through the component densities in equation (2.5), those entries of $\mathbf{\Sigma}$ that do appear (e.g. entries in $\mathbf{\Sigma}_{P_i}$ for some i) are only constrained to form full-rank low-dimensional multivariate Gaussian distributions on the clique level. Grone *et al.* (1984) showed that the other elements in $\mathbf{\Sigma}$ are uniquely determined as functions of these free elements.

Since the hyper-inverse Wishart prior for $\mathbf{\Sigma}$ – $HIW(G, \delta, \mathbf{\Phi})$ is conjugate, the posterior is $HIW(G, \delta^*, \mathbf{\Phi^*})$, where $\delta^* = \delta + n$ and $\mathbf{\Phi^*} = \mathbf{\Phi} + \sum_{i=1}^{n} \boldsymbol{y}_i \boldsymbol{y}_i'$. Further, the marginal likelihood

of $\mathbf{Y}_{1:n}$ given the graph can be written as a function of the inverse Wishart normalizing constants

$$p(\boldsymbol{Y_{1:n}}|G) = (2\pi)^{-np/2} \frac{h(G, \delta, \boldsymbol{\Phi})}{h(G, \delta^*, \boldsymbol{\Phi^*})}, \tag{2.8}$$

where

$$h(G, \delta, \boldsymbol{\Phi}) = \frac{\prod_{P_i \in P} |\frac{\boldsymbol{\Phi}_{P_i}}{2}|^{\frac{\delta+|P_i|-1}{2}} \Gamma_{|P_i|}\left(\frac{\delta+|P_i|-1}{2}\right)^{-1}}{\prod_{S_i \in S} |\frac{\boldsymbol{\Phi}_{S_i}}{2}|^{\frac{\delta+|S_i|-1}{2}} \Gamma_{|S_i|}\left(\frac{\delta+|S_i|-1}{2}\right)^{-1}}. \tag{2.9}$$

For the choice of $\boldsymbol{\Phi}$, Jones *et al.* (2005) set $\boldsymbol{\Phi} = \tau \boldsymbol{I}$, where they also discussed the choices of τ and δ. In subsequent examples, we normally follow their settings and instructions for $\boldsymbol{\Phi}$ and δ.

When G is non-decomposable, the entries of $\boldsymbol{\Sigma}$ in any non-complete prime component P_i will have local constraints because of the missing edges. To solve that, Roverato (2002) generalized the inverse-Wishart prior on the cliques to define a suitable density for the non-complete prime components. Although this generalized HIW prior is still conjugate, the non-complete prime components that come into the product in the form of $h(G, \delta, \boldsymbol{\Phi})$ and $h(G, \delta^*, \boldsymbol{\Phi^*})$ do not have closed forms. Atay-Kayis and Massam (2005); Roverato (2002); Dellaportas *et al.* (2003) proposed different Monte Carlo methods for estimating the inverse Wishart normalizing constants in equation (2.8). Jones *et al.* (2005) discussed the choices of these methods, and adopted the method presented by Atay-Kayis and Massam (2005) to form their unrestricted (not restricted to decomposable) graphical model search.

The prior of the graph G is usually set to control the sparsity of the graph, where "sparsity" means the number of edges in the graph. A uniform prior over all the decomposable graphs, where each graph in the model space has the same prior probability, leads to the "medium" number of edges a priori to be around $p(p-1)/4$ (Jones *et al.*, 2005). For a very high-dimensional data, this uniform prior may be not useful because the inclusion of the spurious edges is not penalized enough. Wong *et al.* (2003) developed a prior where the accumulated prior probability for all the graphs with the same number of edges is the

same. In this dissertation I use the same prior following Jones *et al.* (2005), which is called a Bernoulli prior, having the form as

$$p(G) = \beta^{|E|}(1-\beta)^{\binom{p}{2}-|E|}. \tag{2.10}$$

For each edge in a decomposable graph, the prior probability for this edge to be included in the graph is around β. If it is not restricted to decomposable graphs, the prior probability is exactly β. It is therefore easy for people to set up a prior to reflect their prior knowledge of the sparsity of the graph, by choosing a proper value of β. Since the choice of β usually requires rationale, Chapter 7 discusses an alternative prior with more freedom, where each edge between any pair of nodes i and j in G can have its own prior inclusion probability β_{ij}, with a hyper prior of β_{ij}.

2.3 Graphical Model Search Algorithms

When the graphical model of the observed data ($\mathbf{Y}_{1:n}$) is unknown (which is usually the case), and the graphical space is large (e.g., if the dimension of $\mathbf{Y}_{1:n}$ is 1000, then there are $2^{\binom{1000}{2}}$ possible models in the undirected graphical model space), an efficient search algorithm is required to search for graphs in high probability regions of this huge space. A number of approaches have been proposed for both undirected and directed graphical models (Castelo and Roverato, 2006; Dobra and West, 2004). In this and subsequent chapters we focus on MCMC algorithms (Madigan and York, 1995; Dellaportas and Forster, 1999; Giudici and Castelo, 2003; Armstrong *et al.*, 2005) and shotgun stochastic search algorithms (SSS) (Jones *et al.*, 2005; Hans, 2005; Hans *et al.*, 2007) for undirected graphical models.

A typical MCMC method is the add-delete Metropolis-Hastings sampler (Metropolis *et al.*, 1953; Hastings, 1970), where for each iteration the proposal randomly decides to add or delete an edge first, and then randomly samples one proper edge to be added to or deleted from the current graph. As an algorithm which aims to sample the graphs in the model

space, MCMC is usually not efficient enough in "large p" problems. Hans (2005) and Hans *et al.* (2007) proposed shotgun stochastic search algorithm (SSS) as an alternative way to search for top regressors in both Gaussian and generalized linear regression models, and Jones *et al.* (2005) applied the same algorithm to graphical model search. This algorithm is described as follows:

Step 1: Start from a graph G.

Step 2: Running in parallel, visit all the neighbor graphs of G (graphs that differ by one edge), compute their posterior probabilities given the observed data $\mathbf{Y}_{1:n}$, and record them if they are in the top K model list among all graphs so far visited.

Step 3: Randomly select one new graph to update the current graph G from all the neighbor graphs of G. The probability of choosing one specific model G' is proportional to $p(G'|\boldsymbol{Y}_{1:n})^{\alpha}$, where α is an annealing parameter.

Step 4: Go back to Step 2 and iterate.

Different from the Metropolis-Hastings (MH) sampler, SSS is not an MCMC sampling method. While the MH sampler aims to sample the distribution over the graphical model space, SSS aims to search for graphs in high probability regions of the model space. At each iteration, SSS visits all the neighbors of the current graph, and record the ones with high posterior probabilities. In contrast, the Metropolis-Hastings sampler proposes and visits only 1 graph each iteration, and randomly decides to accept or reject the move by the acceptance ratio. The idea of SSS to visit far more graphs per iteration than MH, brings better efficiency in finding "top" graphs but also computational burdens. However, this computational problem is solved by the inherent parallelizability of SSS. By running in parallel, SSS can visit different many graphs per iterate and swiftly identify regions of models with high posterior probabilities. Jones *et al.* (2005) showed that, to find the same top graph, SSS typically substantially dominates MCMC both in terms of numbers of graphs visited and running time.

Chapter 3

Local Graphical Model Search

The literature in the graphical model field up to now usually focuses on the graphical structures of all the nodes (variables) included in the data, which I refer to as *global graphical models* in this dissertation. For example, Jones *et al.* (2005) explored search over global undirected Gaussian graphical models using MCMC and SSS methods. However, as opposed to the global search approaches developed in literature, we are now interested in learning the local graphical structure around a specific target variable of interest. I refer to this context as *local graphical model search*. This focuses on the graphical model structure of one target variable (Y) and its neighborhood, with less emphasis on the conditional independence structure elsewhere in the graph. Some questions of interest are then: which variables are neighbors of variable Y in the graphical model; and, if we call the graphical model structure of Y and $ne(Y)$ a *local graphical model*, what are the "top" local graphical models found in the search?

Here I list several definitions of concepts that are used throughout this dissertation.

- *Global graphical model*: A graphical model that contains all the variables in the data.
- *Local edge*: An edge that is incident to the targeted variable Y, or connects two nodes from $ne(Y)$.
- *Non-local edge*: An edge that is not a local edge.
- *Local graphical model*: The subgraph of a global graphical model, with $\{Y\} \cup ne(Y)$ as the set of vertices, and the local edges.

The first section of this chapter illustrates the motivations to develop local graphical model search approaches, and the differences from variable selection in regression models.

Next, I explain the general ideas as well as some problematic issues. This is followed by a simple local graphical model search approach based on shotgun stochastic search (Hans, 2005; Hans *et al.*, 2007). Finally, I study application in analysis of a cardiovascular gene expression data set.

A related set of ideas and approaches I refer to as *"Targeted Metropolis-Hasting"* are proposed and discussed in Chapter 4. Chapter 4 also includes several examples of real data analyses using the novel Targeted Metropolis-Hastings method.

3.1 Motivations

3.1.1 The Problems with Global Graphical Model Search

There are many reasons for developing local graphical model search methods, as an alternative to directly using global graphical model search approaches. Generally speaking, global graphical model search approaches are often unable to explore model uncertainty about specific, local regions of interest to a sufficient extent. In many scientific problems, the data may be very high-dimensional, but we are primarily interested in only a few variables and others that might associate with them. In these cases, global graphical model search lacks computational efficiency, because mostly the search algorithms add or delete edges that do not interest us.

To be specific, assume $\boldsymbol{X}$ is the set of all the other variables except Y, $\boldsymbol{X_N}$ is the set of Y's neighbors, and $\boldsymbol{X_{-N}}$ is the set of other variables, so data $\boldsymbol{Y}_{1:n} = \{Y, \boldsymbol{X}\} = \{Y, \boldsymbol{X}, \boldsymbol{X_{-N}}\}$. Then

$$p(Y, \boldsymbol{X}|G) = p(Y, \boldsymbol{X_N}|G)p(\boldsymbol{X_{-N}}|\boldsymbol{X_N}, G). \tag{3.1}$$

Given the sparsity prior of G introduced in equation (2.10),

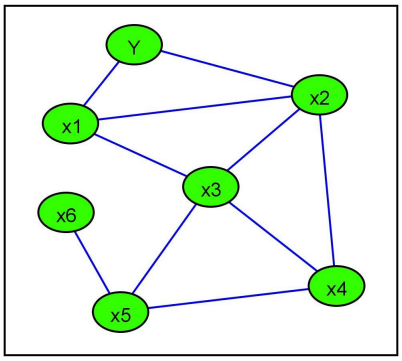
$$p(G|Y, \boldsymbol{X}) \propto p(Y, \boldsymbol{X_N}|G)p(\boldsymbol{X_{-N}}|\boldsymbol{X_N}, G)p(G). \tag{3.2}$$

In many scientific problems related to graphical models, while $|\boldsymbol{X}|$ is very large, $|\boldsymbol{X_N}|$ in Equation (3.1) is small because of the sparsity assumption of graphical models. In such situations, the term $p(\boldsymbol{X_{-N}}|\boldsymbol{X_N}, G)$ will dominate $p(Y, \boldsymbol{X_N}|G)$ in both equation (3.1) and (3.2). In equation (3.1), the domination can happen because $|\boldsymbol{X_N}|$ is much smaller than $|\boldsymbol{X_{-N}}|$. Example 3.1 illustrates a case where the marginal likelihood of a global graphical model with only true local edges can be smaller than the marginal likelihood of a model with only true non-local edges. Since the dimension of a local graph is often small in a high-dimensional problem, the influence the local edges can have to the marginal likelihood is also usually small. The non-local edges represented by term $p(\boldsymbol{X_{-N}}|\boldsymbol{X_N}, G)$, play a much more important role in the marginal likelihood of G.

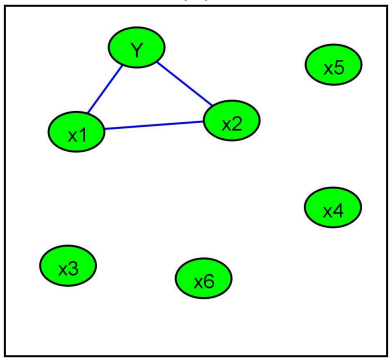
Example 3.1. *Assume Figure 3.1 (a) is the true graph for some simulated data, with 7-dimensional multivariate Gaussian distribution. For this data, there are two different graphical models (b) and (c); (b) contains the true local edges* $\{(Y, x_1), (Y, x_2), (x_1, x_2)\}$*, with the rest of the graph null. (c) contains the true non-local edges, but without any local edges. The global graphical model (b) may have a lower marginal likelihood than (c), because* $p(\boldsymbol{X_{-N}}|\boldsymbol{X_N}, G)$ *dominates equation (3.1).*

Through equation (3.2), it can be seen that the sparsity prior of G also contributes to the domination. In a very high-dimensional problem, although the marginal likelihood already involves penalties on the number of edges in a graph, the sparsity prior of G is usually desirable to further avoid the inclusion of spurious edges in the model. Since the sparsity prior of G introduced in equation (3.2) penalizes the number of edges in the graph, the "top" global graphs, found by global graphical model search, may tend to include many significant non-local edges instead of some less significant local edges. Hence, those local edges are omitted by global graphical model search.

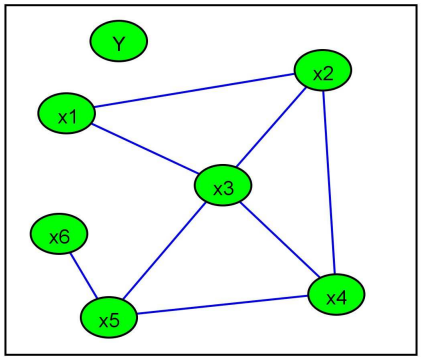
To show that the global graphical model search methods may fail to find the "top" local graphical models as discussed above, I give a simulation example.



(a)

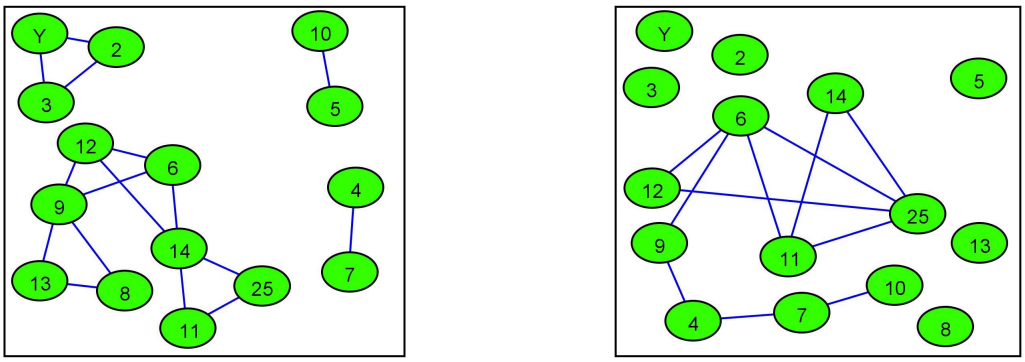


(b)



(c)

Figure 3.1: The graphs for Example 3.1. Graph (a) is the true graph. Even when (b) contains the true local edges, the global graphical model (b) may have a lower posterior probability than (c), because here (c) has the true non-local edges.



(a) (b)

Figure 3.2: Global graphical model search results using a simulated data. Set graph (a) to be a part of the "true" graph with 100 nodes, and (b) is a part of the highest probability graph found by global graphical model search (SSS).

Example 3.2. *Data with* $p = 100$ *variables and* $n = 100$ *samples are simulated based on a known "true" graph. In Figure 3.2, graph (a) shows a part of the "true" graph. I set the target variable* Y *to have weak correlation with two nodes "2" and "3"* ($Corr(Y, 2) = 0.10$, $Corr(Y, 3) = 0.13$)*. By setting the sparsity prior parameter* $\beta = 0.02$*, the global graphical model search (SSS) is not able to identify those two edges in the "top" 50 graphs. In Figure 3.2, graph (b) shows a part of the graph with the highest posterior probability.*

Example 3.2 shows that practically global graphical model search may fail to find local edges because of the domination of the non-local parts of the graph. Another problem with global graphical model search is that, after the search finishes, it is usually summarized by reporting the "top" global graphical models found during the search. However, most of the "top" global models may contain the same local graphical model, in the sense that both of $ne(Y)$ and the local edges are the same. For example, in Example 3.2, in all the "top" 50 global graphical models, Y is disconnected with any node. It is then very difficult to assess

the local model uncertainties from the "top" global models and their posterior probabilities.

3.1.2 Comparison with Regression Model Search

There are a lot of similarities between local graphical model search and regression model search. For example, they both care about one target (response) variable Y, and they both try to identify variables that are neighbors of Y or predict Y. However, there are key differences, which motivate further studies into local graphical model search methods.

Let G_i be a local graphical model, with a set of neighbors of Y and the corresponding local edges. Let R_i denote a regression model, with a set of covariates $\boldsymbol{X}_{\boldsymbol{R_i}}$. For regression model search

$$p(R_i|Y, \boldsymbol{X}) \propto p(Y, \boldsymbol{X}|R_i)p(R_i) = p(Y|\boldsymbol{X}_{R_i}, R_i)p(\boldsymbol{X}|R_i)p(R_i). \tag{3.3}$$

In regression models, people usually treat the design matrix $\boldsymbol{X}$ as given and assume $p(\boldsymbol{X}|R_i) = p(\boldsymbol{X}|R_j)$, where R_i and R_j are both regression models with different sets of predictors of Y. Therefore,

$$p(R_i|Y, \boldsymbol{X}) \propto p(Y|\boldsymbol{X}_{R_i}, R_i)p(R_i). \tag{3.4}$$

The Bernoulli prior of R_i can be defined as

$$p(R_i) \propto \beta^{|R_i|}(1-\beta)^{p-1-|R_i|}. \tag{3.5}$$

In contrast, for the local graphical model search

$$p(G_i|Y, \boldsymbol{X}) \propto p(Y, \boldsymbol{X}|G_i)p(G_i) = p(Y|\boldsymbol{X}_{N_i}, G_i)p(\boldsymbol{X}|G_i)p(G_i). \tag{3.6}$$

Since $\boldsymbol{X}$ are now assumed to have a multivariate Gaussian distribution, and two different local graphical models G_i and G_j contain different local edges, $p(\boldsymbol{X}|G_i) \neq p(\boldsymbol{X}|G_j)$. Therefore, the expression for $p(G_i|Y, \boldsymbol{X})$ cannot be simplified as is in the regression model search. Moreover, if we are also using the Bernoulli prior here,

$$p(G_i) \propto \beta^{|E(G_i)|}(1-\beta)^{p(p-1)/2-|E(G_i)|}, \tag{3.7}$$

then the priors also introduce differences.

In summary, the regression models focus on predicting Y, while not showing any inner structure of the joint distribution of the predictor variables connected to Y. In contrast, local graphical model search not only finds Y's neighbors, but also cares about the local graphical structure around Y, i.e., the interdependencies among the sets of neighbors of Y.

3.2 Some Problems

In this section I discuss some problematic issues with local graphical model search. I start by defining the posterior probability of a local graphical model given the data. The natural definition is

$$p(G_L|\boldsymbol{Y}_{1:n}) = \sum_{G_i \supseteq G_L} p(G_i|\boldsymbol{Y}_{1:n}), \tag{3.8}$$

where G_L is one local graphical model, and G_i's are all the global graphical models which contain the same local graphical structure G_L (denoted as $G_i \supseteq G_L$).

This definition encounters problems when two different G_L's contain different numbers of nodes. For example, if G_{L1} includes three nodes, (Y, x_1, x_2), and G_{L2} includes four nodes, (Y, x_1, x_2, x_3), then the number of G_{i1}'s ($G_{L1} \subseteq G_{i1}$) in the summation in equation (3.8) will be different from that of G_{i2}'s ($G_{L2} \subseteq G_{i2}$). Since $p(G_{ij}|\boldsymbol{Y}_{1:n})$'s are comparable, and the numbers of $p(G_{ij}|\boldsymbol{Y}_{1:n})$ (j=1 or 2) which come to the summation are different, $p(G_{L1}|\boldsymbol{Y}_{1:n})$ and $p(G_{L2}|\boldsymbol{Y}_{1:n})$ become incomparable.

It is not clear how to resolve this basic problem. Redefinition of $p(G_L|\boldsymbol{Y}_{1:n})$ might be a possible approach, but I prefer to sidestep the problem by doing the search in a global graphical model framework, and then appropriately refocusing on local models. These details will be explained shortly. Also, although such a definition of $p(G_L|\boldsymbol{Y}_{1:n})$ will lead to this comparability problem, it can be used in the convergence analysis of "Target Metropolis-Hastings" methods, as introduced in Chapter 4.

A naive view of the local model search problem would aim to add or delete local edges to search for "top" local graphical models without reference to the non-local component of the model. However, Example 3.3 shows that this does not work because there is no way to know the local updates of $p(G|\boldsymbol{Y}_{1:n})$ after adding or deleting one local edge in the graph without knowing the non-local edges.

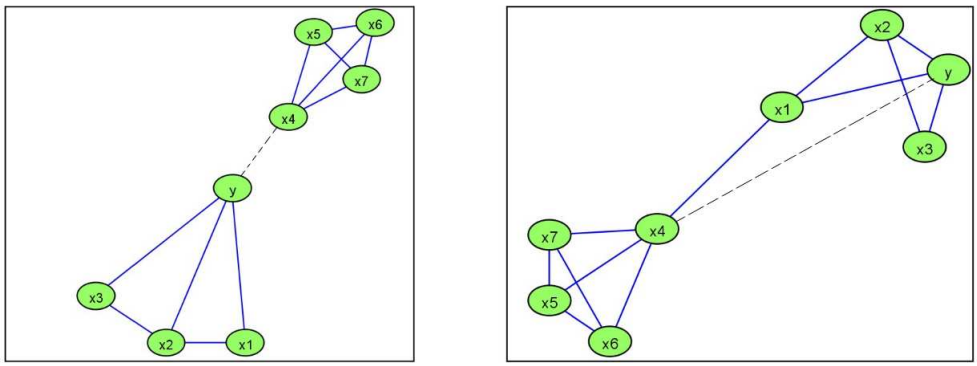
Example 3.3. *Suppose we are in the midst of the search (Figure 3.3), and the current local graph consists of four nodes: Y, x_1, x_2 and x_3. All the other non-local edges, such as (x_4, x_7), are of unknown status. Now if the proposal is to connect x_4 with Y, then there are many situations to enumerate and consider to obtain the likelihood ratio $\frac{p(\boldsymbol{X},Y|G')}{p(\boldsymbol{X},Y|G)}$. Figure 3.3 gives us two of those situations: (a) is one case in which x_4 is actually not incident to any of Y's neighbors, whereas (b) has x_4 and x_1 connected. The likelihood ratios for those two situations are as follow:*

If $(x_1, x_4) \in E(G)$: $\frac{p(\boldsymbol{X},Y|G')}{p(\boldsymbol{X},Y|G)} = \frac{p(x_4,Y)}{p(x_4)p(Y)}$,

and if $(x_1, x_4) \notin E(G)$: $\frac{p(\boldsymbol{X},Y|G')}{p(\boldsymbol{X},Y|G)} = \frac{p(x_4,x_1,y)p(x_1)}{p(x_4,x_1)p(x_1,Y)}$.

In Example 3.3, if it is unknown during the search whether $(x_1, x_4) \in E(G)$, then the likelihood ratio can not be computed if we wish to make a proposal to connect x_4 to Y. In fact, we need to know whether the edges (x_4, x_1), (x_4, x_2) and (x_4, x_3) exist, to compute this likelihood ratio. However, if we wish to set up proposals about those edges, we need more information of the neighborhood of x_1, x_2 and x_3.

More generally, suppose in the current local graph, some variable v is proposed to connect with the target variable Y during the search. Then, the likelihood ratio $\rho = \frac{p(\boldsymbol{X},Y|G')}{p(\boldsymbol{X},Y|G)}$ relies on the edges between the neighbors of Y and v. If some or all of the edges between those nodes are of unknown status, then we can not evaluate the ratio ρ. Therefore, it is impossible to implement local graphical model search without any knowledge, assumption, or search effort on the non-local parts of the graph.



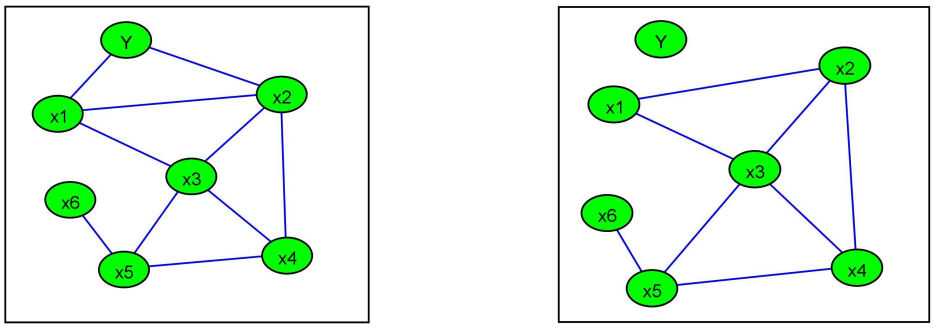
(a) (b)

Figure 3.3: Figure (a) is an example when variable x_4 is proposed to connect Y at some iteration during global graphical model search. Figure (b) contains the same edges with (a) but x_4 is incident to x_1 in (b).

3.3 SSS in Local Graphical Model Search

Example 3.3 shows a basic fact: the appropriate local graphical model search requires knowledge, assumptions or search efforts on the non-local parts of the graph. Assume that we know a priori the true graph of $\boldsymbol{X}$: we could then run SSS to add or delete the neighbors of Y, while keeping the known edges unchanged. This is very similar to the SSS used in regression models (Hans, 2005; Hans *et al.*, 2007).

Example 3.1. *(continued) I have shown the global graphical model search may have problems in this example if we want to find the "top" local models. Now I assume that the graph (b) in Figure 3.1 (which is the same as the graph (b) in Figure 3.4) has the true graphical model of* $\boldsymbol{X}$*. In this case, we can run SSS directly to search for the neighbors of* Y*. We do not need to change any edges that are not incident at* Y *because we know them beforehand. Figure 3.4 shows that there are now no longer any "domination" problems illustrated in Section 3.1.*



(a) (b)

Figure 3.4: When using a fair Bernoulli prior or a sparse Bernoulli prior that only penalizes the number of Y's neighbors, there is no "domination" problem now in this example. If graph (a) has the true local graphical structure, and (b) does not, then (a) has a higher posterior than (b) given the data with a large enough sample size.

However, it is not always realistic to assume that we know the true graph of $\boldsymbol{X}$; the dimension of $\boldsymbol{X}$ can be large, and in any case there is rarely a notion of a real "true" graph. One suggestion (an idea developed in discussion with Dobra, A.) is to consider the simple assumption that *all the non-local edges are null* during the search. Define a graph G' to be a "local neighbor" of G if G' is one of the graphs attained by adding a local edge, deleting an edge between two nodes in $ne(Y)$, or deleting all the edges incident to one node in $ne(Y)$ in G. The search algorithm using SSS to explore such graphs is as follows:

Step 1: Start from a null graph.

Step 2: Randomly select one new graph to update the current graph G from all the local neighbors of G. The probability of choosing one specific model G' is proportional to $p(G'|\boldsymbol{Y}_{1:n})^{\alpha}$, where α is an annealing parameter.

Step 3: Go back to Step 2 and iterate.

Recall that we defined *local edges* to be either edges incident to the target variable

Y, or edges between the nodes in $ne(Y)$. In fact, the "local edges" could also be defined differently. For example, if we are not only interested in the local graphical structure of Y and its neighbors, but also curious about the neighbors of nodes in $ne(Y)$, then the definition of the "local edges" could be extended to the edges that are incident to Y or $ne(Y)$, or the edges that join the neighbors of nodes in $ne(Y)$. The details of this more extensive local neighborhood idea is discussed further in section 7.1.1.

3.4 Cardiovascular Genomics Data Analysis

Seo *et al.* (2007) presented a case study of mice and human cardiovascular genomics, using DNA microarray gene expression data sets. The study focused on atherosclerosis – a disease that "hardens" arterial blood vessels and causes atheromatous plaques, and eventually stenosis (narrowing) of the artery, and possibly plaque ruptures. The original data set had 22215 gene probsets and 211 samples, including 89 mice samples, and 122 human samples. The mice experiments were well designed, with mice cross-classified by four risk factors: the genetic factor related to ApoE (Apolipoprotein E), and factors age, gender and dietary fat content (each at two levels). The 22215 genes probesets were first reduced to 7381 probesets on the mice array that have homologues on the Affymetrix HU95av2 human array used in Seo *et al.* (2004). Then these 7381 probesets are further reduced to 4287 probsets by removing those showing little or no variation above noise levels in the mice data set.

Considering all levels of interactions, the mice samples were classified into 16 design groups. For each group, (e.g., the "extreme disease" group with the ApoE knockout, older, Western diet mice that have advanced atherosclerosis), they built a multivariate ANOVA model, with coefficients subject to sparsity priors, as in Lucas *et al.* (2006). The evaluation of genes showing high posterior probabilities on regression effects related to the design factors enabled identification of atherosclerotic risk related genes. A weighted linear combination of some of those genes (metagene) was then projected onto the corresponding human gene

expression data to obtain the so-called "projected risk signature" on each human sample. The list of 19 gene probsets that generate the risk signature is in Appendix A, by weight.

On the whole human gene expression data, Seo *et al.* (2007) also fitted a Bayesian latent factor model (Carvalho *et al.*, 2008) to obtain 13 estimated factors representing key aspects of variation in expression patterns across the human data. I consider two analysis contexts. First, a small data analysis with the target variable being the projected risk signature of the "extreme disease" group across the human samples, and the 13 factors from BFRM as the $\boldsymbol{X}$ variables. Second, a much larger problem in which the target variable is still the risk signature, but now X represents all the 4287 gene probsets.

3.4.1 Projected risk signature and 13 factors from BFRM

In this subsection, the projected risk signature of the "extreme disease" group is set as the target variable Y, and the 13 factors by BFRM are set as $\boldsymbol{X}$. I compare the SSS algorithm in local graphical model search with SSS in regression model search. First, regression model search is discussed. The data is assumed to have a multivariate normal distribution, and the prior probability of each predictor being included in the model is 1/13, i.e., on average, there is only one predictor a priori in the regression model. By running SSS for 5,000 iterations, I obtain the top 5 models listed in Table 3.1.

Although I use a very sparse prior, the resulting "top" models are still not that sparse. To prove it is not over-fitting, I reran with prior variable inclusion probability of 0.001, the top models still have 5 to 6 predictors. This shows that the projected risk signature is very closely related to the factors.

I then ran the local graphical model search by SSS within the decomposable graphical model space as described by Section 3.3, and using the assumption that all the non-local edges are null. For comparability with the regression search, I also set the prior probability of edge inclusion at 1/13. I also set $\delta = 2.0$, $\tau = 1.0$. Figures 3.5 to 3.7 display the "top"

Rank	Score	Model
1	−144.44	1 2 3 4 7 9 13
2	−146.36	1 2 3 4 7 9 11 13
3	−147.53	1 2 3 4 5 7 9 13
4	−148.00	1 2 3 4 7 8 9 13
5	−148.55	1 2 3 4 6 7 9 13

Table 3.1: 13 factors data: The top 5 models selected by SSS in the regression models.

3 local graphical models obtained by running for 2,000 iterations.

It is not surprising that the neighbors of Y selected in the "top" models in the local graphical model search are almost the same as the predictors of Y chosen in the regression models. However, as discussed in Section 3.1.2, the local graphical model search can find the conditional independence structure of Y's neighbors/predictors, while the regression models cannot do that.

Seo *et al.* (2007) mentioned and discovered that factors 3 and 4 appear to relate to atherosclerotic risk, by investigating and identifying risk-related genes heavily loaded in the factors, and by studying the variation of those factors across the samples. In contrast, in my analysis, both the top regression models and local graphical models found indicate that the risk signature is highly directly related to at least factors 1, 2, 3, 4, 9, and 13. I also confirm the relevance of factors 3 and 4, while elaborating on potential interconnections via the local graphs with highest posterior probabilities.

3.4.2 Projected risk signature and the human gene probsets

To test the performance of the SSS in local graphical model search for large data sets, I now set $\boldsymbol{X}$ to be the human gene probsets (122×4287), while keeping Y as be the projected

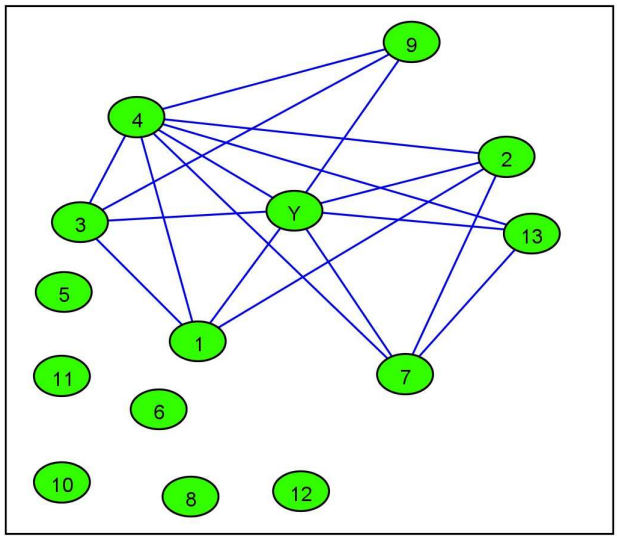


Figure 3.5: 13 factors data: The top 1 decomposable model of the local graphical model search using SSS. The score is −3961.84.

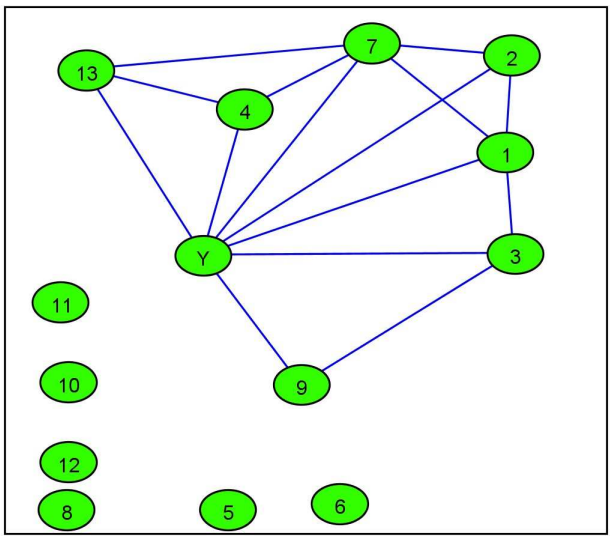


Figure 3.6: 13 factors data: The top 2 decomposable model of the local graphical model search using SSS. The score is −3962.34.

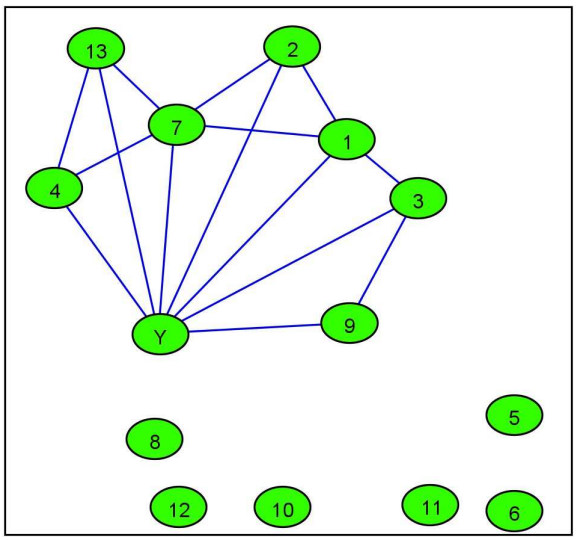


Figure 3.7: 13 factors data: The top 3 decomposable model of the local graphical model search using SSS. The score is -3962.36.

risk signature. Following Jones *et al.* (2005), I set the sparsity prior probability of edge inclusion as $2/4287 = 4.67e-4$, i.e., a priori the expected number of edges in the graph is close to 4288. I also set $\delta = 2.0$, $\tau = 1.0$.

By running local graphical model search by SSS for enough iterations, I noticed that inevitably, the top models contain thousands of neighbor nodes of Y. This is far from satisfactory, as the signature is generated by only 19 genes listed in Appendix A. However, it is in fact not surprising: Once a variable is selected to be a neighbor of Y, and connected to some other neighbors of Y which are strongly related to this variable, deleting it from the neighborhood of Y may become very difficult. This is because doing so may also involve deleting all the significant edges between this variable and the other neighbors of Y. A graph with many significant edges between the neighbors of Y can have a very high posterior probability, although many neighbors selected actually very weakly relate to Y.

As a result, it is natural to set up a constraint on the maximum number of neighbors Y can have in the model to keep the local graph to a reasonable size. One possible constraint,

which I use here, is constraining the maximum number of Y's neighbors to be 20, according to the number of gene probesets that generates this risk signature (19 of them). By this setting it is possible to learn more reasonable top local graphical models.

I ran SSS for 1 million iterations, and recorded the top 50 local models. All the top models contain the same set of neighbors of Y, but with different graphical structure. Across these 50 models, the neighbors of Y selected are listed in Appendix B. Note that only 7 of the 19 genes generating the target variable signature are selected in $ne(Y)$. For one, tumor necrosis factor-alpha-dependent vascular adhesion molecule 1 (VCAM-1), the third gene probset in the 19 genes list, is not in $ne(Y)$. Dansky *et al.* (2001) pointed out that VCAM-1 plays a pivotal role in the initiation of atherosclerosis and adhesion of monocytes to arterial endothelium. Seo *et al.* (2007) also noticed higher level VCAM-1 among old mice with ApoE and those on the western diet. Here, although VCAM-1 does have a high correlation with the risk signature and is involved in top models for the first hundreds of iterations, it is then replaced by other gene probsets because they seem to have stronger *conditional* association with Y in the context of strong associations with the other neighbors of Y selected. This shows a key property of the local graphical model search: the target variable is no longer the only factor in the model selection; the search considers the whole graphical structure of the target variable and its neighbors. More biological insights are discussed in Section 3.5.

Figures 3.8, 3.9 and 3.10 give the top 3 local graphical models found by SSS in 1 million iterations. They all have the same set of neighbors of the projected risk signature, but with some differences in local edges. Note that with such a high dimensional problem, any "top model" must be properly recognized as local modes in the posterior over graphs.

To show the efficiency of the local graphical model search, I also tried global graphical model search (SSS algorithm) for this data. Unfortunately, due to the high dimension of the data, for each iteration of SSS, around 9 million graphs have to be evaluated in parallel, which means for only 10 iterations it takes more than an hour to finish using a distributed,

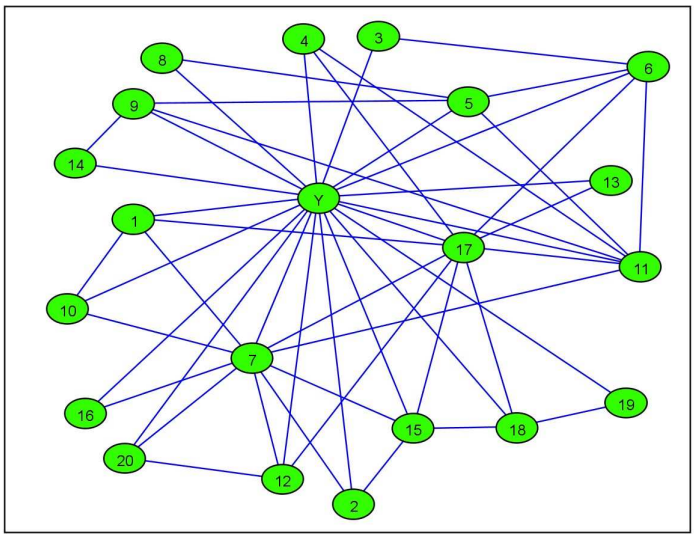


Figure 3.8: Human gene probsets data: The top 1 decomposable model of the local graphical model search using SSS. The score is -752751.02. This model is found at iteration 678444. The descriptions of variables are in Appendix B.

cluster analysis on 50 cpu cores. In contrast, local graphical model search using SSS takes only 4 seconds to finish 10 iterations with 50 cpu cores.

I also implemented regression model search via SSS to identify top regression models with predictors of Y. The average prior number of predictors in the model is set to be 20, and SSS is run for 100,000 iterations. The 16 predictors included in the top 10 models are listed in Appendix C, and Table 3.2 shows the top 10 models as well as the scores. It is very interesting to see that 11 of the 16 predictors in Table 3.2 are included and mostly highly ranked in the 19 genes list that generate the projected risk signature, and 10 of them are in the 20 genes list that are selected as neighbors of the signature across the top 50 local graphical models. Specifically, the 9 highest ranked genes except CD68 (6th ranked) from the risk signature are included at least once in the top 10 models (osteopontin and VCAM1 are included in all of the top models), while only osteopontin (ranked 1st), CD53 glycoprotein (ranked 5th), HLA-DRB1 (ranked 8th) and HLA-DQB1 (ranked 9th)

Rank	Score	Model
1	38.88	1 3 4 6 9 12
2	38.73	1 2 3 4 6 9 12
3	37.45	1 3 9 11 12
4	36.86	1 3 6 9 12
5	35.99	1 3 6 9 11 12
6	35.79	1 3 6 8 9 12
7	35.65	1 3 11 12 14
8	35.61	1 2 3 11 12 15
9	35.35	1 3 5 10 13
10	35.04	1 3 5 7 13 16

Table 3.2: Human gene probsets data: The top 10 models selected by SSS in the regression models.

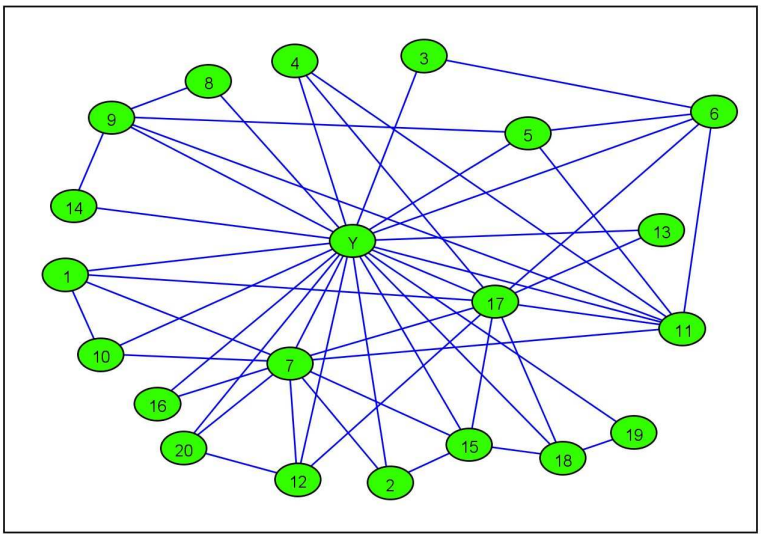


Figure 3.9: Human gene probsets data: The top 2 decomposable model of the local graphical model search using SSS. The score is -752753.61. This model is found at iteration 678444. The descriptions of variables are in Appendix B.

are included in Y's neighbors in the top local graphical models. As I discussed before, due to the high collinearity among the genes, all these selected neighbor genes may not predict Y as well as the 2nd - 4th ranked genes (e.g. VCAM-1). However, those highly ranked genes are not selected in the top local graphical models since they do not interact much with the other neighbors of Y so as to contribute substantially to the scores of the local graphs. Also, there are three genes identified that are included in both the top 10 regression models and the top local graphical models but that are not among the 19 genes in the risk signature; again, this arises due to collinearity among the genes. Although the risk signature is not generated by those two genes, the high collinearity among the genes and the randomness involved in the Gaussian model lead the two genes to predict Y and interact with the other variables in $ne(Y)$ better than the genes that are not selected.

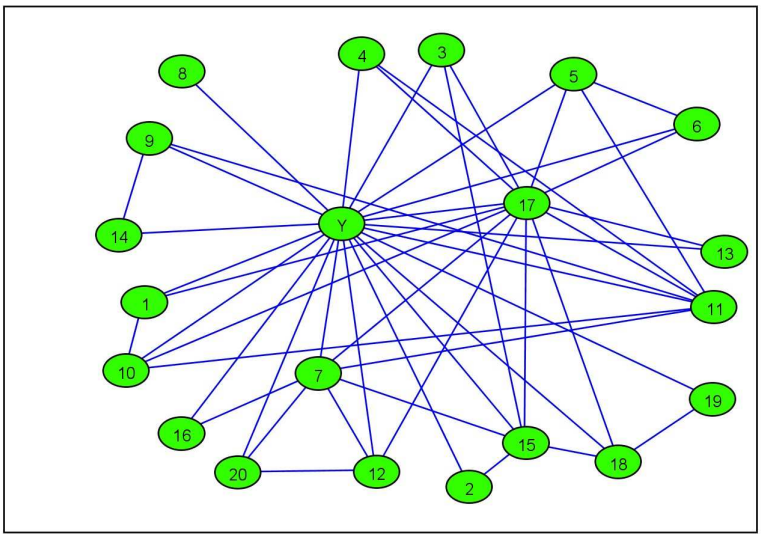


Figure 3.10: Human gene probsets data: The top 3 decomposable model of the local graphical model search using SSS. The score is -752753.78. This model is found at iteration 812874. The descriptions of variables are in Appendix B.

3.5 Biological Discussion

I refer to the literature in cardiovascular genomics for biological insights related to the top local graphical models found in Section 3.4.2. By SSS and assuming that all the non-local edges are null, I have found 20 genes that are candidate neighbors of the projected risk signature. Compared to regression modeling, the top local graphical models also give local graphical structures of the neighbors and Y. In this section, I discuss several interesting genes that are neighbors of Y across the set of 50 top graphs.

Osteopontin (minpontin in the mice genome), highest ranked in the gene list of the projected risk signature and also selected in $ne(Y)$ in the "top" 50 local graphical models, plays a major role in cardiovascular diseases and particularly in atherosclerosis. Giachelli *et al.* (1995) indicated that osteopontin, in both mice and human vascular diseases, contributes to the mediating processes of cellular adhesion and migration, and via its integrin-type re-

ceptors, contributes to vascular remodeling in plaque formation and development. Seo *et al.* (2007) noticed significantly increased expression levels of osteopontin in aged mice groups and Western diet mice groups separately, and more so for the group with both risk factors. It is well known that osteopontin is closely associated with atherosclerosis.

Human leukocyte adhesion protein (LFA-1/Mac-1) is formed by the integrin beta chain beta 2 (ITGB2, also known as CD18) combined with the alpha L/M chain. CD18 is a subunit of intercellular adhesion molecule-1 (ICAM-1). Nageh *et al.* (1997) tested the role of leukocyte and endothelial cell adhesion molecules (CAMs) in atherosclerosis in mice, and found that pharmaceutical reduction of the expression or function of the CAMs such as ICAM-1, P-selectin and CD18 may protect against atherosclerosis in mice. Kitagawa *et al.* (2002) supports these findings, where their studies on ApoE-knockout mice suggested that inhibition of ICAM-1 can delay the progression of atherosclerosis. A summarized discussion of the role of ICAM-1 and VCAM-1 in atherosclerosis can be found in Ballantyne and Entman (2002).

Human complement component 2 (C2) is an integral part of the classical pathway of the complement system. The complement system is part of the larger immune system, and it is a major way for the body to respond to infection. It is known that the components of the complement system are associated with atherosclerosis. By studying patients with C2 deficiency in Sweden, Jönsson *et al.* (2005) found that hereditary deficiency of C2 is related to frequent occurrence of invasive infection, atherosclerosis, and rheumatic disease.

Homo sapiens lysosomal acid lipase (LIPA lipase A, lysosomal acid, cholesterol esterase (Wolman disease)), also known as LAL, has been known to be involved in catalyzing the hydrolysis of cholesteryl esters and triglycerides, and this activity is associated with atherosclerosis. Zschenker *et al.* (2006) showed that a deficiency of LAL causes an accumulation of lipids in the cells and pre-mature atherosclerosis. They also discussed the influence of over-expressed LAL in atherosclerosis. Similarly, Seedorf *et al.* (1995) found that a novel variant LAL is associated with Cholesterol ester storage disease (CESD), which is a disease related

to atherosclerosis.

There are four major histocompatibility complex (MHC) Class II genes selected as neighbors of Y in "top" local graphical models. The MHC plays a critical role in the immune system, by controlling the immune response through recognition of "self" and "invader." HLA-DQB1 (major histocompatibility complex, class II, DQ beta 1), HLA-DRB1 (major histocompatibility complex, class II, DR beta 1), and HLA-DMB (major histocompatibility complex, class II, DM beta), belong to HLA class II beta chain paralogues. Another neighbor gene, HLA-DMA (major histocompatibility complex, class II, DM alpha), belongs to HLA class II alpha chain paralogues. HLA-DMA and HLA-DMB belong to the same class II molecule, where DMA is an alpha chain, and DMB is a beta chain, both anchored in the membrane. This connection is also reflected by the top local graphical models: HLA-DMA (labeled 7th in Appendix B) and HLA-DMB (labeled 11th) are connected and form a clique with Y and LAPTM5 (lysosomal associated multispanning membrane protein 5). HLA-DRB1 alleles, especially DRB1*0404 allele, are believed to be implicated in the development of endothelial dysfunction, which is associated with atherosclerosis by studying patients with rheumatoid arthritis or severe aortoiliac occlusive disease (Gonzalez-Gay *et al.*, 2004; Mas *et al.*, 2005).

Finally, human Ia-associated invariant gamma-chain gene (CD74) is a membrane protein that works as an MHC class II chaperone. Although it is not directly associated to atherosclerosis, it belongs to MIF-induced monocyte adhesion, where MIF (macrophage migration inhibitory factor (glycosylation-inhibiting factor)) is identified as an important regulator of atherosclerosis with exceptional chemokine-like functions (Schober *et al.*, 2008; Bernhagen *et al.*, 1997). The relationship of CD74 and atherosclerotic vascular disease was also shown by Seo *et al.* (2007), where CD74 was selected in most highly significant gene sets for ApoE. Age group (23 genes in total).

3.6 Computational Aspects

The shotgun stochastic search algorithm for local graphical model search, introduced in this chapter, has been implemented in parallel by C++ language with Message Passing Interface (MPI). MPI contains message passing libraries that are widely used as a standard in distributed memory, message passing, and parallel computing programs. The introduction of MPI can be found in *http://www-unix.mcs.anl.gov/mpi/* .

This code of local graphical model search by SSS can be downloaded from my home page *http://www.stat.duke.edu/∼lz9*, and it can be successfully executed in the Duke Shared Cluster Resource (DSCR), where some of the machines are provided by Computational Science, Engineering and Medicine (CSEM) at Duke University. More information about the DSCR can be found at *http://www.csem.duke.edu/pmwiki/pmwiki.php/Dscr/HomePage.*

Chapter 4

Targeted Metropolis-Hastings Methods

In this chapter, I introduce Targeted Metropolis-Hastings methods (TMH) for local graphical model search. TMH is based on the usual Metropolis-Hastings methods (UMH, introduced in Section 2.3). The motivation for developing TMH is to accelerate the local model sampling process when the global model space is enormous, while avoiding the assumption that all the non-local edges are null. We begin this chapter by describing TMH. This is followed by several simulation studies that compare TMH to UMH. In addition, theoretical explanations of performance are given. Finally, TMH and UMH are compared in local graphical model search on several real data sets.

4.1 Introduction

Example 3.3 highlighted the fact that local graphical model search needs information about the non-local parts of the graph. In Section 3.3 this issue is avoided using a simple method that assumes that all the non-local edges are null before the search starts. To avoid making this BIG assumption, we can recognize the need to appropriately focus on the non-local edges during the search, but to somehow limit the time and effort spent in non-local regions. The question is now how to do this.

Assume the graph is sparse and has 1000 nodes. During the standard UMH sampling under the 50:50 edge inclusion versus deletion proposal, UMH proposes on average only one local edge to add or delete for every 500 iterations. Hence, if we run UMH for 1 million iterations, at best the chain can visit only 2000 local models. However, we have at least hundreds of thousands local models under consideration in this example. Therefore, it is natural to think about modifying UMH to have more proposal freedom: the probability

of proposing a local edge can be changed by users for different purposes. This concept is named "Targeted Metropolis Hasting" (TMH).

The specific implementation of TMH studied here begins with any initial state, e.g., the null model or full model, and at each iteration it proposes to add or delete 1 randomly chosen local edge with probability λ ($\lambda > 0.5$), or 1 randomly chosen non-local edge with probability $1 - \lambda$. When λ is high, the "targeted" proposal favors local graphs, but the Markov chain is still irreducible and aperiodic on the global space.

At a specific iteration of the MH, assume the current graph is G, and the MH proposal proposes a certain move to change the graph from G to G'. The acceptance probability of the proposal $\rho(G, G')$ at this iteration is

$$\rho(G, G') = \min\left\{\frac{p(G'|\boldsymbol{Y}_{1:n})}{p(G|\boldsymbol{Y}_{1:n})}\frac{q(G|G')}{q(G'|G)}, 1\right\}, \tag{4.1}$$

where $q(G'|G)$ ($q(G|G')$) is the probability of the graph G' (G) to be proposed given the current graph G (G'). For UMH, the computation of of $q(G'|G)$ and $q(G|G')$ is straight forward. However, for TMH given a specific λ, $q(G'|G)$ and $q(G|G')$ can be very different in various situations (see Appendix D). $T(G, G')$, the transition probability from graph G to G' is

$$T(G, G') = \rho(G, G')q(G'|G). \tag{4.2}$$

To study the convergence to the stationary distribution of a certain Markov chain, the *total variation distance* as a measure is typically used.

Definition 4.1. *Letting $\pi_n(G)$ denote the probability of visiting graph G at step n of the Markov chain, starting from the initial distribution π_0, and letting π denote the stationary distribution, the total variation distance at step n in the finite and discrete graph space is defined as*

$$\| \pi_n - \pi \|_{TV} = \frac{1}{2}\sum_{G} |\pi_n(G) - \pi(G)|. \tag{4.3}$$

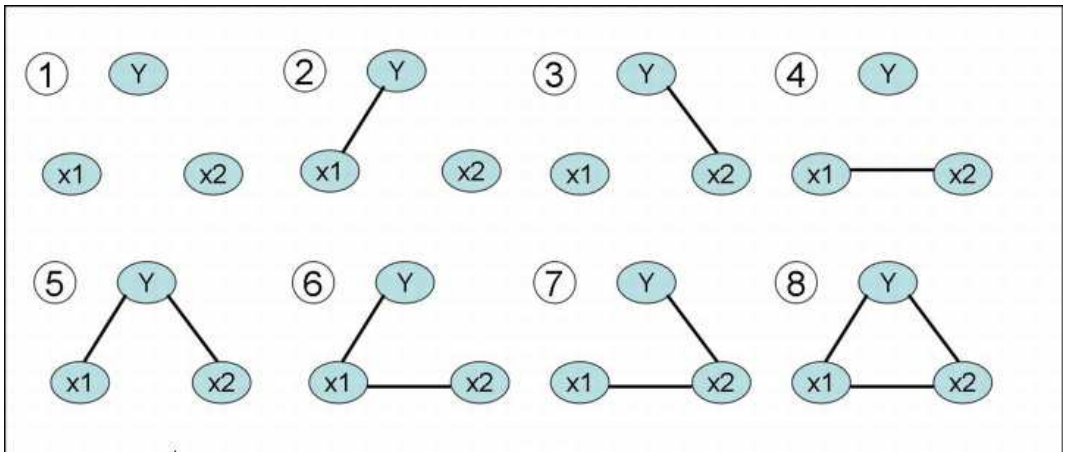


Figure 4.1: The 8 states for a 3-node graphical model example.

If the transition matrix T for each step of the chain is fixed, $\pi_n = \pi_0 T^n$. The time a Markov chain takes to converge to the stationary distribution is called the *mixing time*. The mixing time is obtained from the total variation distance.

Definition 4.2. *For $\epsilon > 0$, the mixing time $\tau(\epsilon)$ is defined as*

$$\tau(\epsilon) = min\{n : \| \pi_{n'} - \pi \|_{TV} \leq \epsilon, \forall n' \geq n\}. \tag{4.4}$$

The mixing time can be bounded and associated to the transition matrix by the following theorem:

Theorem 4.1. *(Jerrum* et al.*, 1986; Randall, 2006) Let $\pi_* = \min_G \pi(G)$. For all $\epsilon > 0$ the mixing time $\tau(\epsilon)$ satisfies*

$$\frac{|\lambda_1|}{2(1-|\lambda_1|)} \log\left(\frac{1}{2\epsilon}\right) \leq \tau(\epsilon) \leq \frac{1}{1-|\lambda_1|} \log\left(\frac{1}{\pi_* \epsilon}\right),$$

where λ_1 here denotes the eigenvalue of the transition matrix with the second largest absolute value (the largest eigenvalue is 1).

To illustrate TMH, I give a 3-node example, with nodes indexed Y, x_1 and x_2. The model space hence contains $2^3 = 8$ global models. Figure 4.1 shows in detail all the states

of the graph. In this example, we let all the states have equal posterior probabilities (1/8). Denote ρ_{ij} to be the acceptance probability from graph i to j, and T_{ij} to be the transition probability from graph i to j. Then, the TMH transition matrix $T = \{T_{ij}\}$ becomes

$$T = \begin{pmatrix} 0 & \lambda/2 & \lambda/2 & 1-\lambda & 0 & 0 & 0 & 0 \\ \lambda/2 & \lambda/2-1/4 & 0 & 0 & 1/4 & 1-\lambda & 0 & 0 \\ \lambda/2 & 0 & \lambda/2-1/4 & 0 & 1/4 & 0 & 1-\lambda & 0 \\ 1-\lambda & 0 & 0 & 0 & 0 & \lambda/2 & \lambda/2 & 0 \\ 0 & 1/4 & 1/4 & 0 & 1/6 & 0 & 0 & 1/3 \\ 0 & 1-\lambda & 0 & \lambda/2 & 0 & \lambda/2-1/3 & 0 & 1/3 \\ 0 & 0 & 1-\lambda & \lambda/2 & 0 & 0 & \lambda/2-1/3 & 1/3 \\ 0 & 0 & 0 & 0 & 1/3 & 1/3 & 1/3 & 0 \end{pmatrix}.$$

T_{ij} is calculated like the following examples:

$$\rho_{25} = min\left\{1, \frac{\pi(2|5)}{\pi(5|2)}\right\} = min\left\{1, \frac{1/4}{\lambda/2}\right\} = min\left\{1, \frac{1}{2\lambda}\right\} = \frac{1}{2\lambda}$$

$$\Rightarrow T_{25} = \pi(5|2)\rho_{25} = \frac{1}{4},$$

and

$$\rho_{52} = min\left\{1, \frac{\pi(5|2)}{\pi(2|5)}\right\} = min\left\{1, \frac{\lambda/2}{1/4}\right\} = min\left\{1, 2\lambda\right\} = 1$$

$$\Rightarrow T_{52} = \pi(2|5)\rho_{52} = \frac{1}{4}.$$

For this transition matrix T, when $\lambda = 0.9$, $\lambda_1 = -0.8307$. When $\lambda = 0.99$, $\lambda_1 = 0.8893$. When $\lambda = 0.999$, $\lambda_1 = 0.9059$. By theorem 4.1, I plot both the lower and upper bounds of the mixing time for the three different λ's in Figure 4.2.

Figure 4.2 shows that the TMH method slows down the global mixing rate when λ is high, although the global mixing time actually is not monotonically increasing in λ (λ_1 is

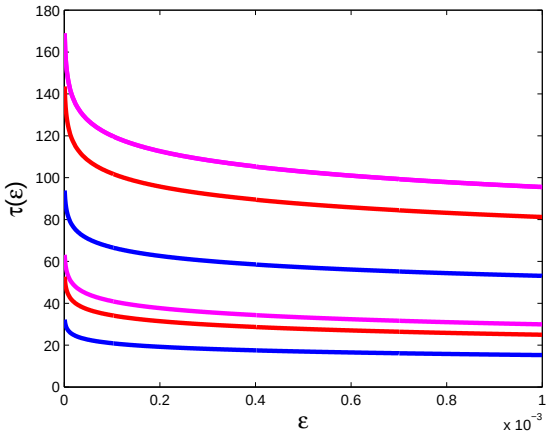
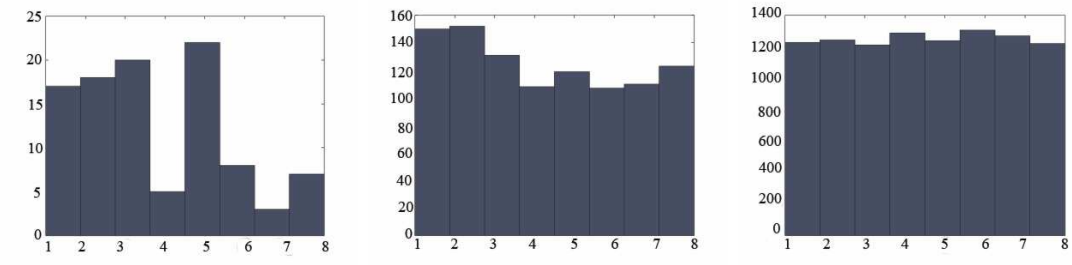


Figure 4.2: The lower and upper bounds of the mixing time for the three different λ's. The blue lines correspond to $\lambda = 0.9$, the red lines correspond to $\lambda = 0.99$, and the purple lines correspond to $\lambda = 0.999$.

sometimes positive and sometimes negative). Also, in this 3-node example, some simulation work can be done to show that TMH converges. Figure 4.3 shows the frequencies of visiting each state by the TMH methods when running for different iterations with $\lambda = 0.99$.

Therefore, since TMH is a more "targeted" method, it is not surprising that TMH slows down the global mixing rate compared to UMH. However, the performance of TMH in sampling local models is the primary concern here, since this dissertation is focused on local models. Also, it is important to propose a way to summarize the sampling results from either TMH or UMH for communication. With this example, I show some preliminary simulation results in Table 4.1.

Since every state in the model space has the same probability, the stationary probabilities of the edges (Y, x_1), (Y, x_2), and (x_1, x_2) are all 0.5. Therefore, in Table 4.1, the probability of choosing edge (x_1, x_2) converges slower than that of choosing edges (Y, x_1) or (Y, x_2). This is because (x_1, x_2) is mostly a non-local edge, while both (Y, x_1) and (Y, x_2) are always local edges. Also, the marginal joint probability of (Y, x_1) and (Y, x_2)



(a) (b) (c)

Figure 4.3: The frequencies of visiting each of the 8 states by TMH after (a) 100, (b) 1000, and (c) 10000 iterations when the number of burn-in iterations equals 10, and $\lambda = 0.99$.

Edges	100 iterations	1000 iterations	10000 iterations
(Y, x_1)	(0.36,0.63)	(0.462,0.539)	(0.488,0.513)
(Y, x_2)	(0.37,0.63)	(0.459,0.539)	(0.488,0.513)
(x_1, x_2)	(0.135,0.85)	(0.381,0.614)	(0.461,0.540)
Acc Rate	(0.80,0.94)	(0.85,0.90)	(0.87,0.88)

Table 4.1: The 95% credible interval for the percentage of time that the edges (Y, x_1), (Y, x_2), and (x_1, x_2) are in the graph respectively, after 100, 1000, and 10000 iterations. The credible interval of the acceptance rates for different iterations are also shown in the table. Here $\lambda = 0.99$

converges faster than that of choosing the other two pairs of edges, $((Y, x_1), (x_1, x_2))$ and $((Y, x_2), (x_1, x_2))$ for essentially the same reason.

4.2 Local Convergence in Targeted Metropolis-Hastings Methods

In this section, I discuss the concept of the local convergence, which is used to compare the performance of TMH and UMH in local graphical model search.

First, we call a set of global graphical models $S_l = \{G|G_l \subseteq G\}$ a *local model state (group)*, where G_l is a local graphical model, and all G's in S_l are the graphs that contain the same local subgraph G_l. Then,

$$p(S_l) = \sum_{G \subseteq G_l} p(G). \tag{4.5}$$

As mentioned before in Section 3.2, since the S_l's have different sizes when the dimensions of G_l's are different, the $p(S_l)$'s become incomparable. Here, however, the $p(S_l)$'s can be used for local convergence instead of being used as scores of local model states during the search. Note that both TMH and UMH sample the local model states. As a result, for every step of the Markov chain, we can compare the total variation distances between the current distribution of the local model states and the stationary distribution. Mathematically, if we start the Markov chain with an initial distribution of the local model states π_0 (row vector), and the transition matrix on the local model space at step i is T_i, then after n steps, the distribution becomes $\pi_n = \pi_0 \prod_{i=1}^{n} T_i$. Note that the transition matrix on the local model space keeps changing whether or not it is using TMH or UMH, because for the model spaces that contains multiple global graphical models, the proposal probability from one model state to another relies on the current distribution of the global models in the global model state space.

Assume that the stationary distribution of visiting local states is π, then after n steps of the Markov Chain, the total variation distance becomes:

$$\| \pi_n - \pi \|_{TV} = \frac{1}{2} \sum_l |\pi_n(S_l) - \pi(S_l)|, \tag{4.6}$$

where l is the index of the local model states.

We can compare the performance of TMH and UMH in local convergence by comparing the decreasing curves of the total variation distances versus the steps. Two experiments for this comparison are implemented: one is using a 10-node graph, and the other is using a 100-node graph.

4.2.1 A 10-node Graph Experiment

This section describes the experiment with a 10-node graph. To limit the total number of the local model states, I assume that the target variable Y can have at most 2 neighbors, and every graphical model that satisfies such an assumption has a uniform posterior probability. In this case, there are $\binom{p-1}{2} * 2 + (p-1) + 1 = p^2 - 2p + 2 = 82$ local model states. The corresponding stationary probabilities of the local model states in which Y has no neighbors or just 1 neighbor are about 0.0217, and the stationary probabilities of the local model states in which Y has 2 neighbors is about 0.0109. By running both TMH and UMH in this example, I compare their performance in convergence to the stationary distribution of the local model states.

Recall that the transition matrix on the local model space keeps changing; it is very difficult to write it down analytically. As a result, at each transition step n, I estimate π_n by simulating 100,000 independent chains in parallel, all starting from the null graph. Clearly, this involves a massive computational effort at each step n, but is an easy and effective way to estimate the transition probabilities.

Figures 4.4-4.10 show the comparison of TMH and UMH in terms of total variation

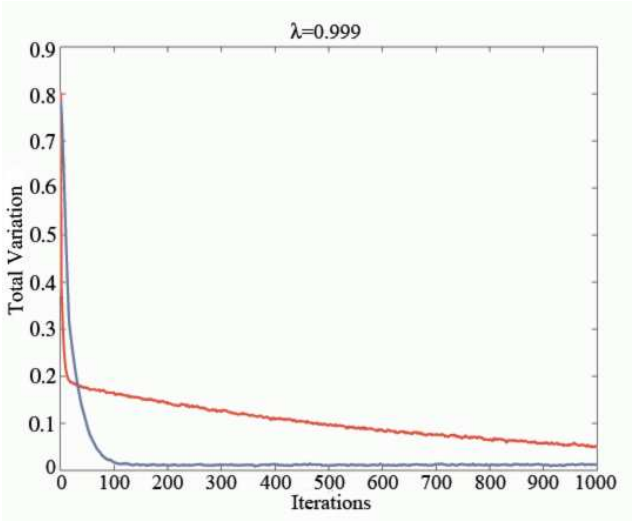


Figure 4.4: The total variation distances versus the steps of the Markov Chain (p=10). Two different sampling methods are used: TMH (λ=0.999 red curve) and UMH (blue curve).

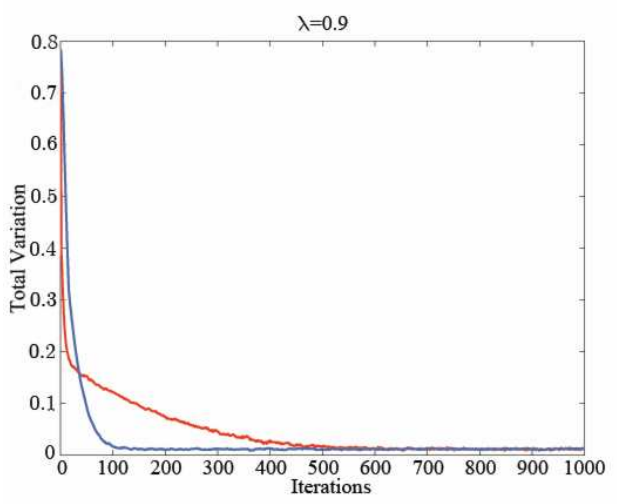


Figure 4.5: The total variation distances versus the steps of the Markov Chain (p=10). Two different sampling methods are used: TMH (λ=0.9 red curve) and UMH (blue curve).

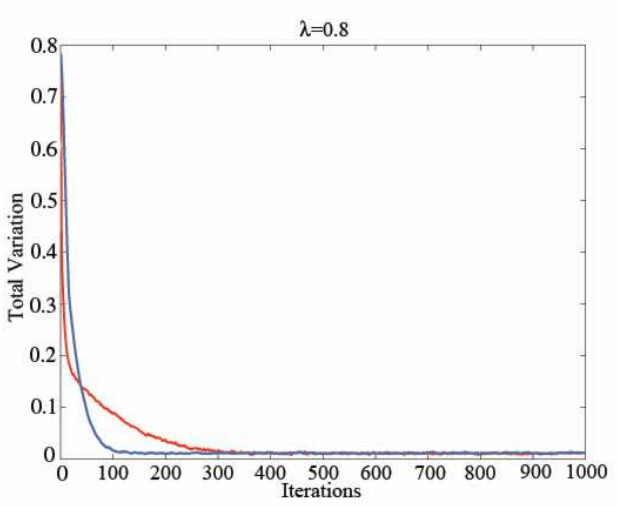


Figure 4.6: The total variation distances versus the steps of the Markov Chain (p=10). Two different sampling methods are used: TMH (λ=0.8 red curve) and UMH (blue curve).

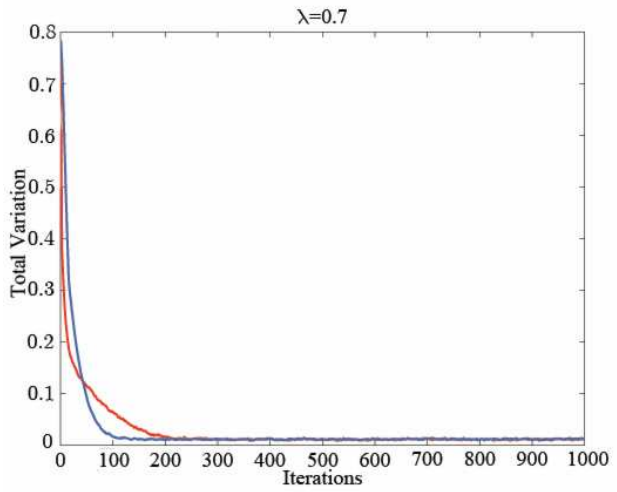


Figure 4.7: The total variation distances versus the steps of the Markov Chain (p=10). Two different sampling methods are used: TMH (λ=0.7 red curve) and UMH (blue curve).

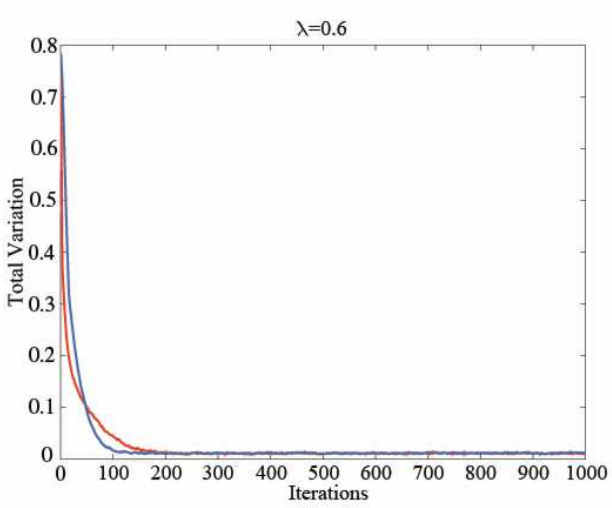


Figure 4.8: The total variation distances versus the steps of the Markov Chain (p=10). Two different sampling methods are used: TMH (λ=0.6 red curve) and UMH (blue curve).

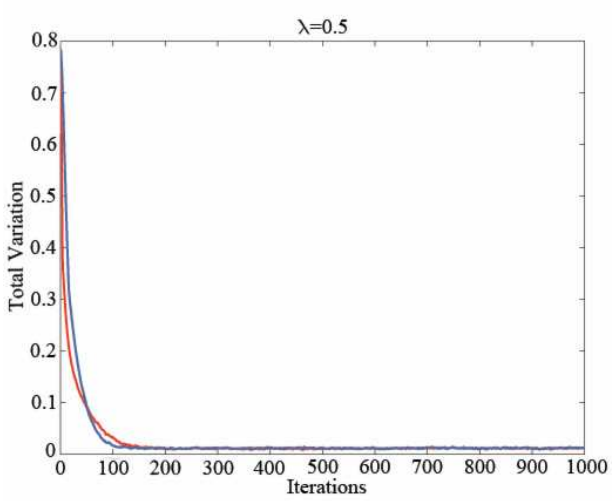


Figure 4.9: The total variation distances versus the steps of the Markov Chain (p=10). Two different sampling methods are used: TMH (λ=0.5 red curve) and UMH (blue curve).

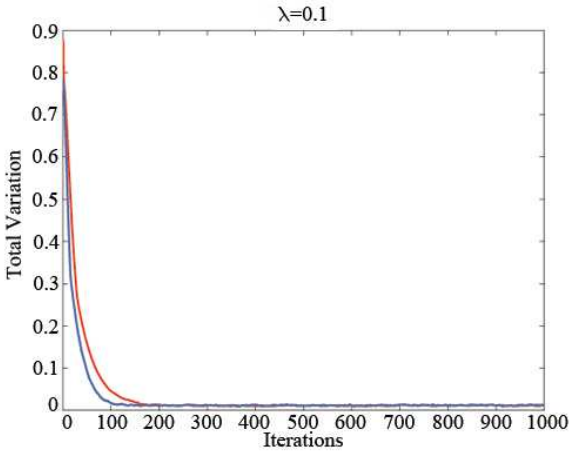


Figure 4.10: The total variation distances versus the steps of the Markov Chain (p=10). Two different sampling methods are used: TMH (λ=0.1 red curve) and UMH (blue curve).

distances from the stationary distribution in the local model space. When λ is big enough (e.g. greater than 0.5), TMH converges faster than UMH in the first tens of iterations. Afterwards, UMH starts to dominate. The closer λ is to 1, the faster TMH converges in the first several iterations, and the slower afterwards. When λ is close to the average local move proposal probability of UMH, the two curves become almost the same. When λ decreases further, TMH is worse than UMH from the beginning to the end.

4.2.2 A 100-node Graph Experiment

To further explore the local convergence performances of TMH and UMH, another simulation experiment was run with a graph of 100 nodes. To limit the number of local model states, the target variable Y was constrained to have at most 2 neighbors from the nodes set $\{x_1, ..., x_9\}$. Also, every global graphical model that satisfies such a constraint has the same posterior probability.

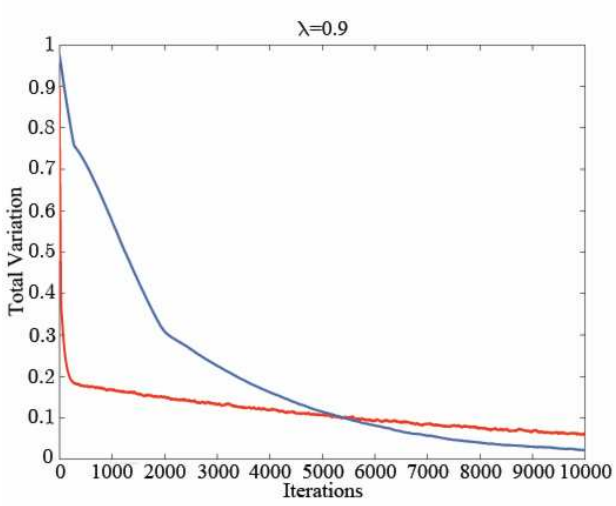


Figure 4.11: The total variation distances versus the steps of the Markov Chain (p=100). Two different sampling methods are used: TMH (λ=0.9 blue curve) and UMH (red curve).

As a result, there are 82 local model states, with stationary probabilities of 0.0217 if Y has 0 or 1 neighbors, and of 0.0109 if Y has 2 neighbors. By simulating 100,000 independent chains, all starting from the null graph, I obtained estimates of the total variation curves of UMH and TMH for different values of λ.

Figures 4.11-4.15 show the simulation results for λ = 0.9, 0.8, 0.7, 0.6, and 0.1. The curves look quite similar to those in the 10-node experiment, except that now TMH can dominate UMH for the first thousands of iterations. Intuitively, I expect in a 1000-node simulation experiment with similar settings that TMH will dominate UMH in the first hundreds of thousands of iterations.

On a log scale, the total variation distance curves become close to linear after about 2000 iterations. The mathematical explanations are in the following Section 4.3. If considering the two curves as linear, TMH has a greater absolute value of the intercept than UMH, but UMH has a sharper slope, which explains the initial dominance of TMH, and why it is

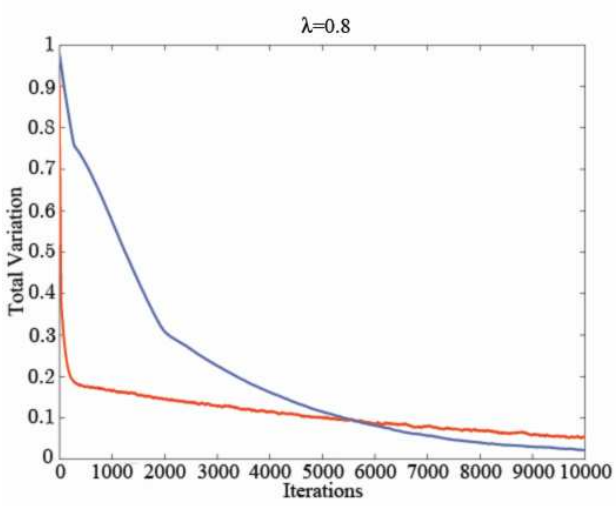


Figure 4.12: The total variation distances versus the steps of the Markov Chain (p=100). Two different sampling methods are used: TMH (λ=0.8 blue curve) and UMH (red curve).

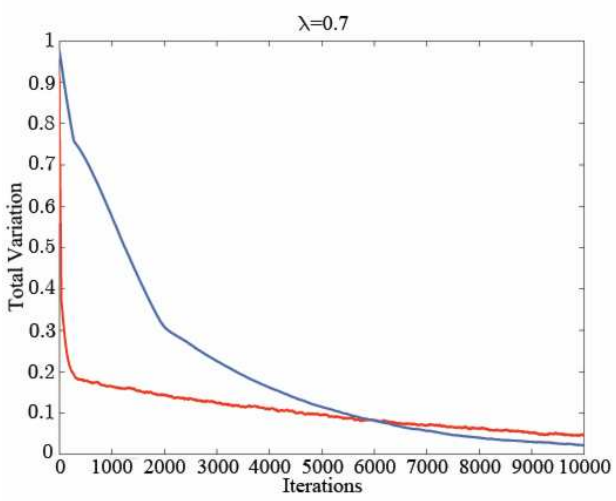


Figure 4.13: The total variation distances versus the steps of the Markov Chain (p=100). Two different sampling methods are used: TMH (λ=0.7 blue curve) and UMH (red curve).

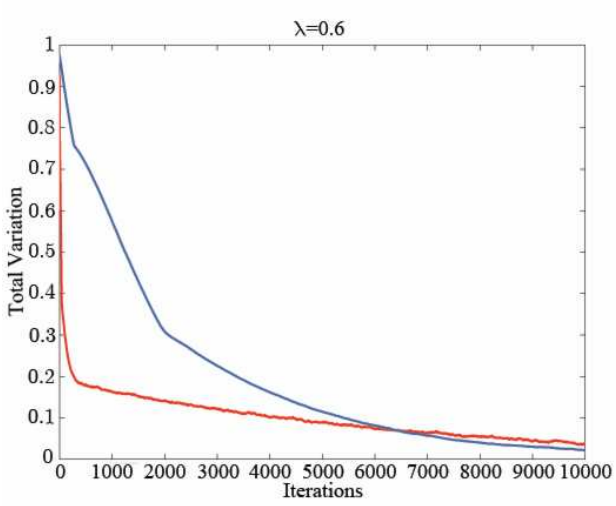


Figure 4.14: The total variation distances versus the steps of the Markov Chain (p=100). Two different sampling methods are used: TMH (λ=0.6 blue curve) and UMH (red curve).

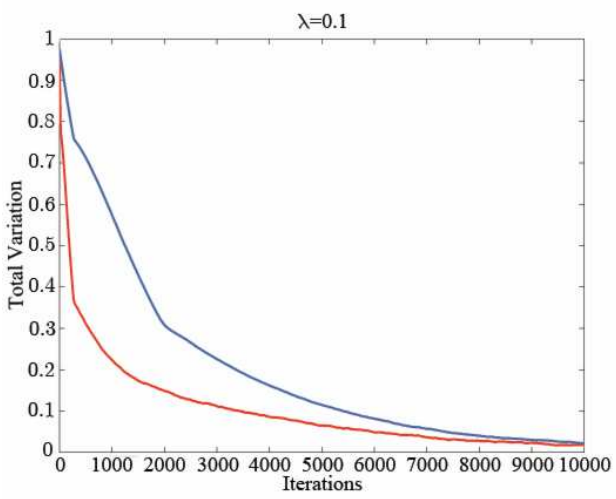


Figure 4.15: The total variation distances versus the steps of the Markov Chain (p=100). Two different sampling methods are used: TMH (λ=0.1 blue curve) and UMH (red curve).

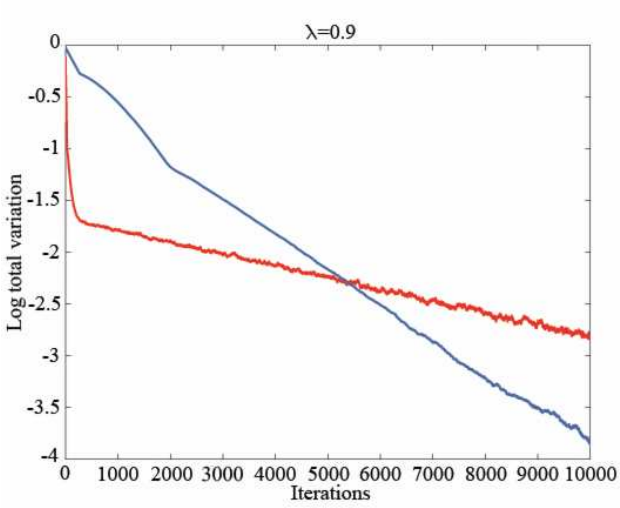


Figure 4.16: The logarithm of the total variation distances versus the steps of the Markov Chain (p=100). Two different sampling methods are used: TMH (λ=0.9 red curve) and UMH (blue curve).

dominated by UMH afterwards. The corresponding plots after the logarithm transformation of the total variation are in Figures 4.16-4.20.

Given the fact that TMH converges faster initially than UMH, but not afterwards, it is natural to think of starting with TMH for the first hundreds of iterations (to get the intercept advantage of TMH), and then switching to UMH (suddenly or gradually). Ideally, this idea can combine the advantages of both methods, i.e., the intercept of TMH and the slope of UMH after the logarithm transformation. Figure 4.21 shows the simulation results of this idea. Unfortunately, this idea does not work as well as my intuition suggests. This is because the transition matrix on the local model space actually keeps changing, regardless of whether we are using TMH or UMH. In contrast to the usual Markov chain, the transition matrix on the local model space depends on the current distribution of the global models, at each MC iterate. For an instance, in the 3-node example analyzed in Section 4.1, one can divide the 8 global states listed in Figure 4.1 into 5 local model states: $\{1, 4\}$, $\{2, 6\}$,

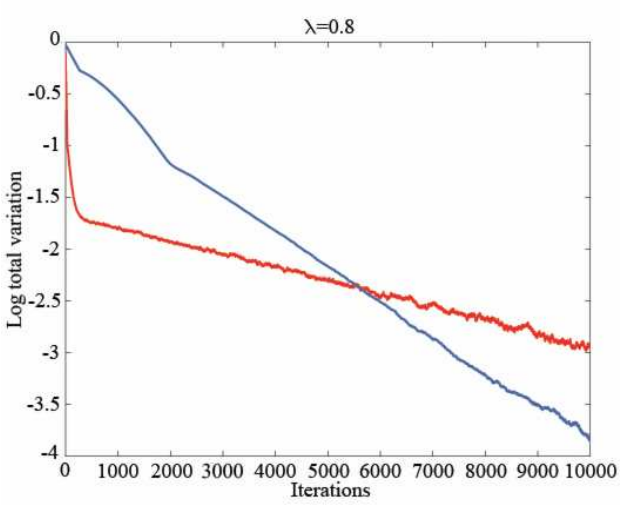


Figure 4.17: The logarithm of the total variation distances versus the steps of the Markov Chain (p=100). Two different sampling methods are used: TMH (λ=0.8 red curve) and UMH blue curve).

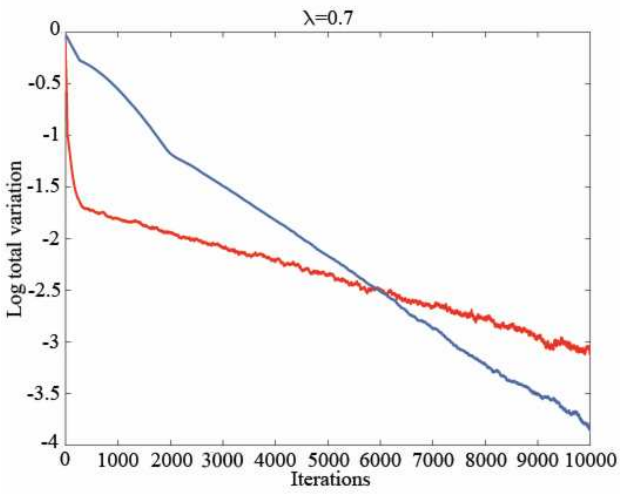


Figure 4.18: The logarithm of the total variation distances versus the steps of the Markov Chain (p=100). Two different sampling methods are used: TMH (λ=0.7 red curve) and UMH (blue curve).

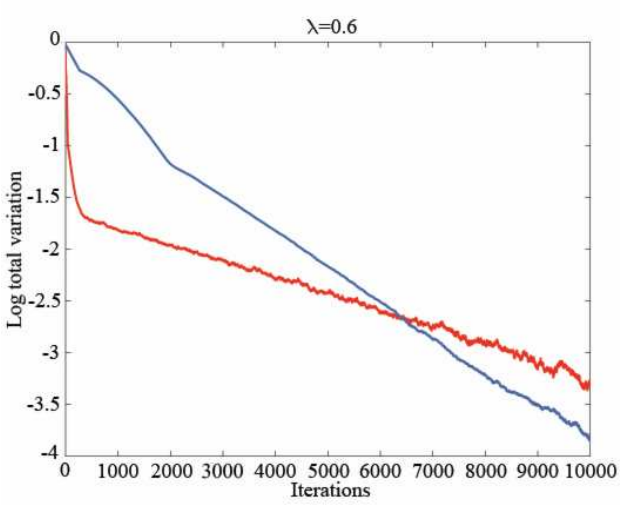


Figure 4.19: The logarithm of the total variation distances versus the steps of the Markov Chain (p=100). Two different sampling methods are used: TMH (λ=0.6 red curve) and UMH (blue curve).

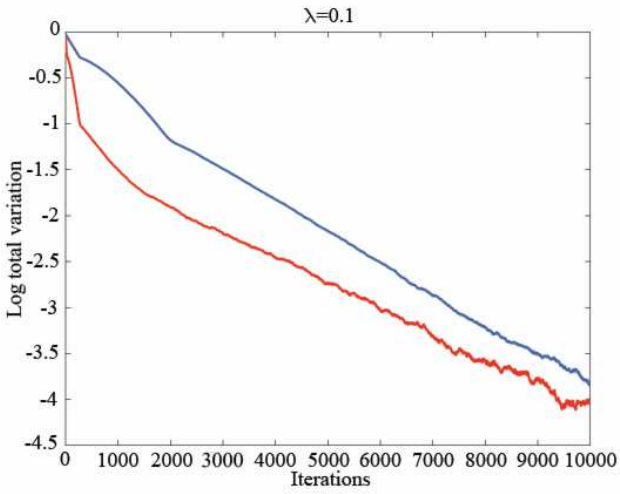


Figure 4.20: The logarithm of the total variation distances versus the steps of the Markov Chain (p=100). Two different sampling methods are used: TMH (λ=0.1 red curve) and UMH blue curve).

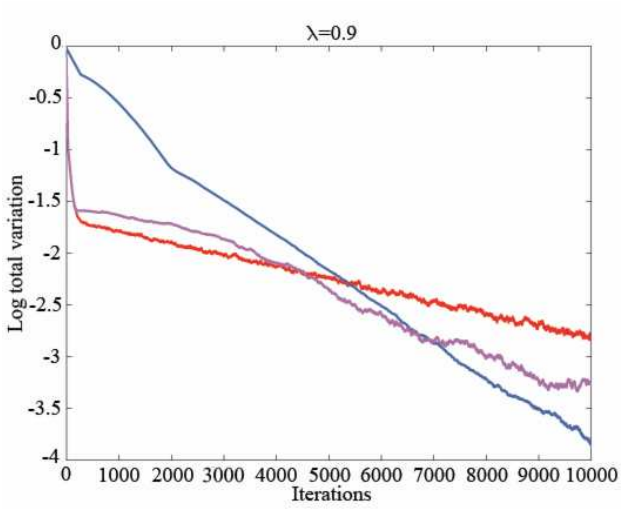


Figure 4.21: The logarithm of the total variation distances versus the steps of the Markov Chain (p=100). Three different sampling methods are used: TMH (λ=0.9 red curve), UMH (blue curve), and TMH→UMH (λ=0.9, switch point: iteration 200, Purple curve)

$\{3,7\}$, $\{5\}$, and $\{8\}$. At a specific step k of the chain, the transition probability from the local model states $\{1,4\}$ to $\{2,6\}$ obviously relies on the current distribution of the global model 1, 2, 4, and 6. That is the key reason why switching from TMH to UMH is not uniformly successful. Intuitively the current distribution of the global models obtained by TMH does not quite "fit" the UMH method, and needs time to readjust.

4.2.3 Theoretical Explanations

In this section I give a theoretical explanation of the experiments in Section 4.2 using the spectral decomposition. Recall that for a local model state S_l we define $p(S_l) = \sum_{G \in S_l} p(G)$, and the corresponding stationary distribution is

$$\pi(S_l) = \sum_{G \in S_l} \pi(G). \tag{4.7}$$

Denote the initial distribution on the global model space by π_0, and the transition matrix on the global model space by T. At step k, the spectral decomposition of π_k is

$$\pi_k = (T^k)^T \pi_0 = \pi + \sum_{i=2}^{n} \lambda_i^k l_i r_i^T \pi_0, \tag{4.8}$$

where l_i and r_i are respectively the left eigenvector and the right eigenvector of T, and the λ_i's are the eigenvalues of T. For any function f on the global model space, we have

$$f^T \pi_k = f^T \pi + \sum_{i=2}^{n} \lambda_i^k (f^T l_i)(r_i^T \pi_0). \tag{4.9}$$

Here

$$\pi_k(S_j) = \sum_{G \in S_j} \pi_k(G) = f_j^T \pi_k = f_j^T \pi + \sum_{i=2}^{n} \lambda_i^k (f_j^T l_i)(r_i^T \pi_0), \tag{4.10}$$

where f_j is a vector with 1's and 0's. In fact, $f_j^T \pi = \pi(S_j)$. Define $f_j^T l_i = \beta_{ij}$ so that

$$\pi_k(S_j) = \pi(S_j) + \sum_{i=2}^{n} \lambda_i^k \beta_{ij} (r_i^T \pi_0). \tag{4.11}$$

Then, the total variation distance between π_k and the stationary distribution π is

$$\| \pi_k - \pi \|_{TV} = \frac{1}{2} \sum_{j=1}^{m} | \sum_{i=2}^{n} \lambda_i^k \beta_{ij} (r_i^T \pi_0)|, \tag{4.12}$$

Which, when k is very large, implies

$$\| \pi_k - \pi \|_{TV} \approx \frac{|\lambda_2|^k}{2} \sum_{j=1}^{m} |\beta_{2j} (r_2^T \pi_0)|. \tag{4.13}$$

Equation (4.13) explains why the log total variation curve is close to linear in iterations after many steps of the chain. Obviously, compared to TMH, UMH has a greater value of $|\lambda_2|$ in those simulation examples, but TMH has a greater intercept of the "line", which is determined by the term $\sum_{j=1}^{m} |\beta_{2j} (r_2^T \pi_0)|$.

Both the experiments implemented and theoretical explanations indicate that TMH has a faster local convergence rate than UMH initially in the sampling process, where the time of the TMH dominance increases as the dimension of the problem increases. In a very high-dimensional problem, TMH obviously is far more efficient than UMH because convergence becomes rather difficult for both methods and in the limited running time TMH usually maintains dominance. Another point to note is that the simulation results may be different if the graphs no longer have uniform probabilities, since the transition matrix T also changes under different stationary distributions.

4.3 Local Edge Inclusion Probabilities

4.3.1 Motivation and Definitions

Given a set of graphs sampled by either TMH or UMH, it is necessary to decide how to summarize the results for communication. For example, given a gene expression data set, biologists may wonder what the local graph around some targeted gene looks like; showing the "top" graphical models is one possibility. However, as discussed in Section 3.1.1, the posterior probabilities of the global graphical models can be dominated by the non-local parts of the graphs. Also, following Jones *et al.* (2005), Metropolis-Hastings methods are not well designed to find "top" graphs. As a result, other summaries are of interest.

It is natural to think about aggregating the global graph samples into the local graph visiting frequencies by equation (4.1), and then reporting the most frequently visited local graphs. However, as mentioned in both Section 3.2 and 4.2, for different dimension of G_l's, the $p(S_l)$'s are incomparable and hence so are the visiting frequencies of the local graphs. One might suggest discounting the visiting frequency of one local graph by the total number of possible global graphs containing this local graph, but this seems artificial and lacks theoretical support.

Another way to summarize the samples of graphs is to compute the edge inclusion probabilities. In the variable selection literature, posterior variable inclusion probabilities are key posterior summaries. Formally, with the data $\boldsymbol{Y}_{1:n}$, and a set of models $\boldsymbol{\Gamma}$, the posterior inclusion probability for variable i is defined as

$$p(i) = \sum_{\gamma \in \boldsymbol{\Gamma}} p(\gamma | \boldsymbol{Y}_{1:n}) I(\gamma_i = 1), \tag{4.14}$$

where $I(\gamma_i = 1)$ is an indicator function that is equal to 1 when variable i is in the model γ, and 0 otherwise. In linear regression, Barbieri and Berger (2004) defined the median probability model to be the model consisting of those variables whose posterior inclusion

probability is at least 0.5. They also proved that, for variable section in normal linear models, the optimal predictive model is often not the model with the highest posterior probability, but the median probability model.

Similar to the variable inclusion probability, we can define the edge (i, j) inclusion probability in global graphical models as

$$p(i,j) = \sum_{G} p(G|\mathbf{Y}_{1:n}) I((i,j) \in E(G)). \tag{4.15}$$

For local graphical model search, we do not care about the edge inclusion probability of the non-local edges. Therefore, the *local edge inclusion probability* is defined as follows:

If $i = Y$ or $j = Y$,

$$p_L(i,j) = \sum_{G} p(G|\mathbf{Y}_{1:n}) I((i,j) \in E(G)), \tag{4.16}$$

otherwise

$$p_L(i,j) = \frac{\sum_{G} p(G|\mathbf{Y}_{1:n}) I((i,j) \in E(G)) I((i,Y) \in E(G)) I((j,Y) \in E(G))}{\sum_{G} p(G|\mathbf{Y}_{1:n}) I((i,Y) \in E(G)) I((j,Y) \in E(G))}. \tag{4.17}$$

We can similarly define the *mean probability local graphical model* to be the local graphical model consisting of those local edges whose posterior inclusion probability is at least 0.5. Formally, if we denote the mean probability local graphical model as G_M, any edge $(i, j) \in E(G_M)$ if and only if $p_L(i, j) > 0.5$.

Summaries of the TMH/UMH sampling results by the local edge inclusion probabilities are clearly intuitively attractive, although also clearly not sufficient. For example, if x_1 and x_2 are highly correlated, the local edge inclusion probabilities $p_L(Y, x_1)$ and $p_L(Y, x_2)$ may be both 0.49, and the pairwise local edge inclusion probability $p_L((Y, x_1), (Y, x_2))$ may be 0. The mean probability local graphical model does not include either edge, when in fact we should include one of them. Therefore, it is also important for the users to look at the

pairwise, or even multi-way local edge inclusion probabilities from the sample (Chipman *et al.*, 2001; Clyde and George, 2004).

4.3.2 Simulation Study

Convergence properties of the local edge inclusion probabilities may be different from the local convergence studied in section 4.2. Equation (4.16) consists of two terms: the posterior distribution of a graph given the data, and an indicator of whether an edge is in the graph. Intuitively, UMH is better than TMH in simulating samples from the posterior distribution of the graph space, but TMH can accelerate this sampling process by picking representative graphs while losing some precision globally.

At iteration n, we can collect all the samples drawn from the first to the nth iteration, and compute the estimated edge inclusion probabilities from those samples. Then, by knowing the true edge inclusion probabilities in simulated examples, the mean squared error (MSE) may be computed to compare the convergence performance of TMH and UMH.

A simulation study is constructed similar to that in Section 4.2. First, assume that there are 10 nodes in the graph, including the target variable Y. The graphs are a priori known to have equal probability. Hence for any edge (i, j), the true value of $p_L(i, j) = 0.5$. Figure 4.22 shows the experiment with $p = 10$ and $\lambda = 0.9$. The x-axis is the number of iterations, and the y-axis is MSE. I run both TMH and UMH for 10000 iterations from a null graph, and for each iteration collect all the samples drawn, then compute the edge inclusion probabilities as well as MSE. The above experiment is repeated for 100 times the average is taken. As a result, for either TMH and UMH, a smooth MSE decreasing curve is obtained. In Figure 4.22, TMH converges faster than UMH in the sense of edge inclusion probabilities. Figures 4.23 and 4.24 give a "zoom-in" view of Figure 4.22 to show the curves of TMH and UMH in the intervals of iterations (5000, 10000) and (90000, 100000).

Similarly, I also look at how these results change when λ is changed. Figure 4.25 is

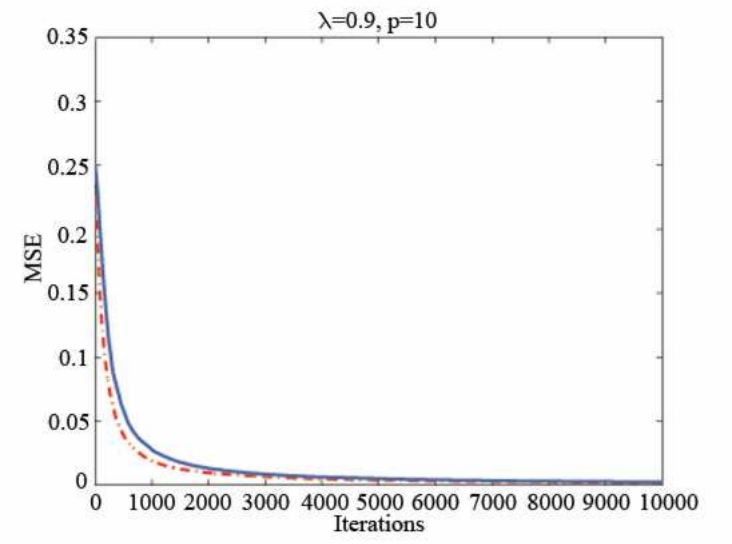


Figure 4.22: The MSE of the edge inclusion probabilities converging to the truth by TMH and UMH (p=10). TMH (λ=0.9 red dashed curve), UMH (blue curve).

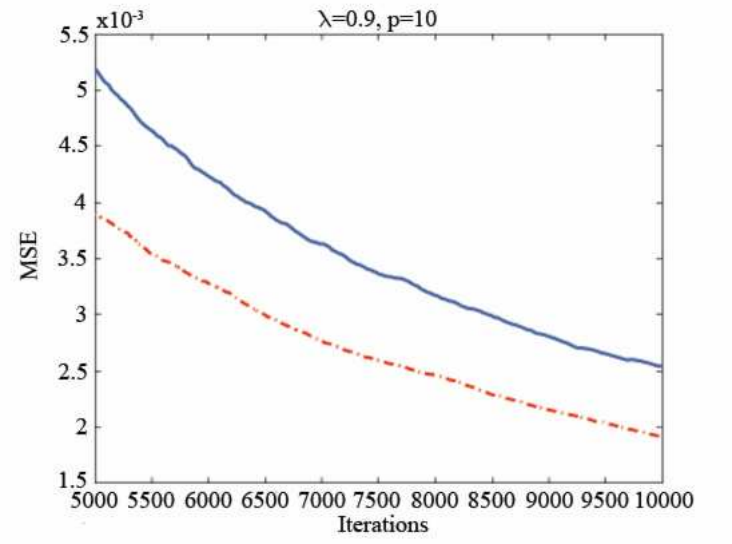


Figure 4.23: The "zoom-in" plot of Figure 4.22 in the interval of iterations (5000, 10000).

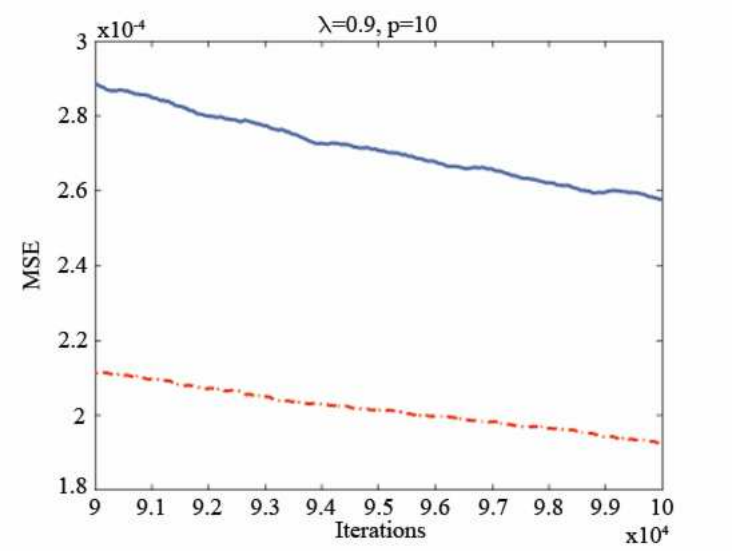


Figure 4.24: The "zoom-in" plot of Figure 4.22 in the interval of iterations (90000, 100000).

drawn by setting $\lambda = 0.6$. Clearly, The TMH curve is closer to the UMH curve in this plot compared to the case when $\lambda = 0.9$. This is mainly because the graphs are uniformly distributed, and TMH with larger λ has an advantage of proposing local edges. Again, if the graphs did not have equal probabilities, TMH would not have such a clear advantage all the time, because it is only considered as a fast approximation of UMH.

It is also necessary to look at the convergence of the pairwise edge inclusion probabilities. Figure 4.26 shows the result from the experiment where $p = 10$, $\lambda = 0.9$, and considers all the pair of edges that are both incident to the target variable Y (so in total $9 \times 8/2 = 36$ pairs). The conclusion obtained is similar to the one from the edge inclusion probabilities.

Another experiment with $p = 100$ and $\lambda = 0.9$, the same as the settings in the 100-node experiment in Section 4.2, tests the performance of the two methods when also assuming the graphs are uniformly distributed. Figure 4.27 indicates that TMH now converges much faster than UMH in the sense of edge inclusion probabilities. This plot supports the view

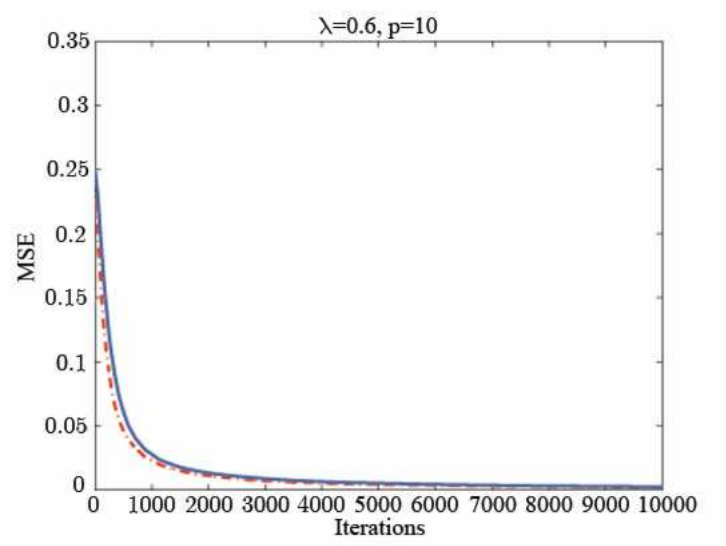


Figure 4.25: The MSE of the edge inclusion probabilities converging to the truth by TMH and UMH (p=10). TMH (λ=0.6 red dashed curve), UMH (blue curve).

that when there is a large number of nodes in the graph, TMH should be preferred to UMH.

4.4 Simulated Data from Jones *et al.* (2005)

To test the local model sampling performances by TMH and UMH, in this section I use a 15-node data set from Jones *et al.* (2005) that was inspired by patterns of daily currency exchange fluctuations against the U. S. dollar. The underlying true graph is presented in Figure 4.28. The data set contains 250 observations. By global graphical model search (SSS), and setting $\delta = 3$, $\tau = 0.0004$, and sparsity parameter $\beta = 0.1429$, Jones *et al.* (2005) gave the decomposable global graph with the highest posterior probability as shown in Figure 4.29.

For local graphical model search, I take node 7 as the target variable Y. Figures 4.30 and

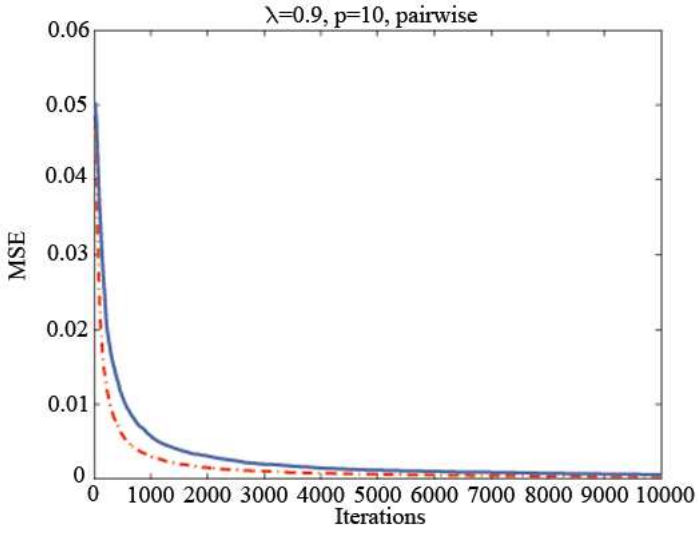


Figure 4.26: The MSE of the pairwise edge inclusion probabilities converging to the truth by TMH and UMH (p=10). TMH (λ=0.9 red dashed curve), UMH (blue curve).

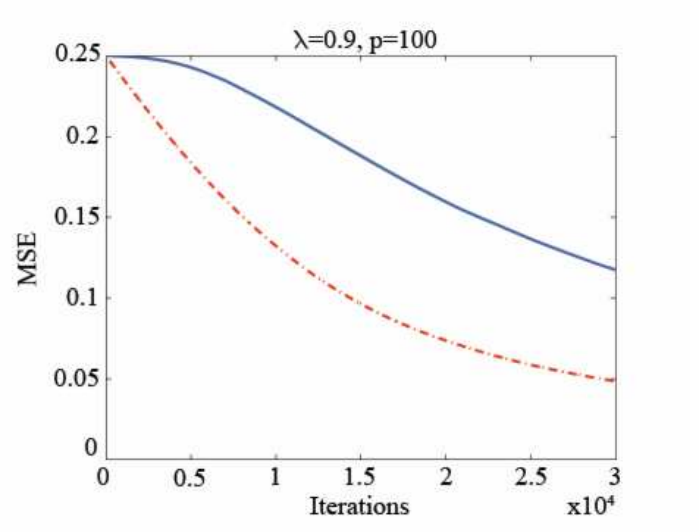


Figure 4.27: The MSE of the edge inclusion probabilities converging to the truth by TMH and UMH (p=100). TMH (λ=0.9 red dashed curve), UMH (blue curve).

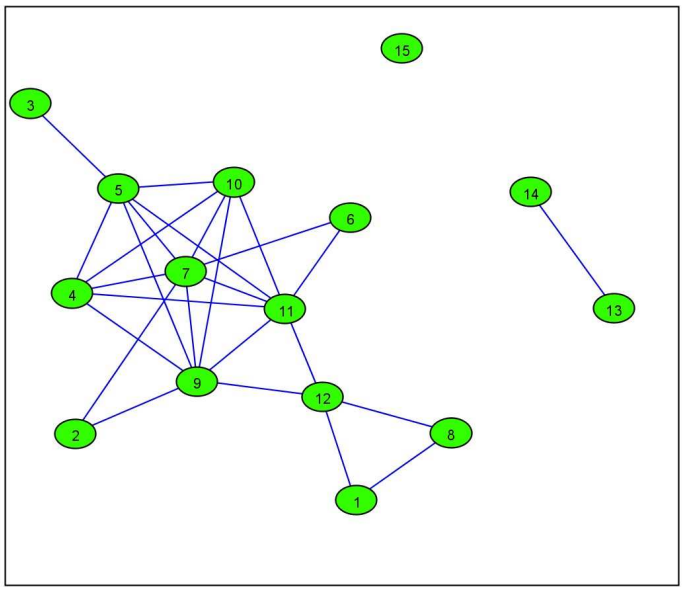


Figure 4.28: Simulation example from Jones *et al.* (2005). The true underlying decomposable graph on 15 nodes simulated example.

4.31 show the local edges whose edge inclusion probabilities are bigger than 0.5, respectively obtained by TMH ($\lambda = 0.9$) and UMH in 100K iterations plus 10K burn-in. Note that δ, τ, and the sparsity prior remains the same as Jones *et al.* (2005). It is not surprising that both TMH and UMH obtain almost the same local graphical structure around Y as the top global model, although the edge connecting node 6 and node 7 is only included with probability around 0.6. Because of the small size of this data set, it is very easy for both TMH and UMH to find the local graphical structure.

4.5 Cardiovascular Genomics Data Analysis

Following the cardiovascular genomic data analysis in Section 3.4, in this section I use the same data to test the performances of TMH and UMH. Again I set $\delta = 2.0$, $\tau = 1.0$. First, I analyze the 13 factors data with the target variable projected risk signature. This is followed

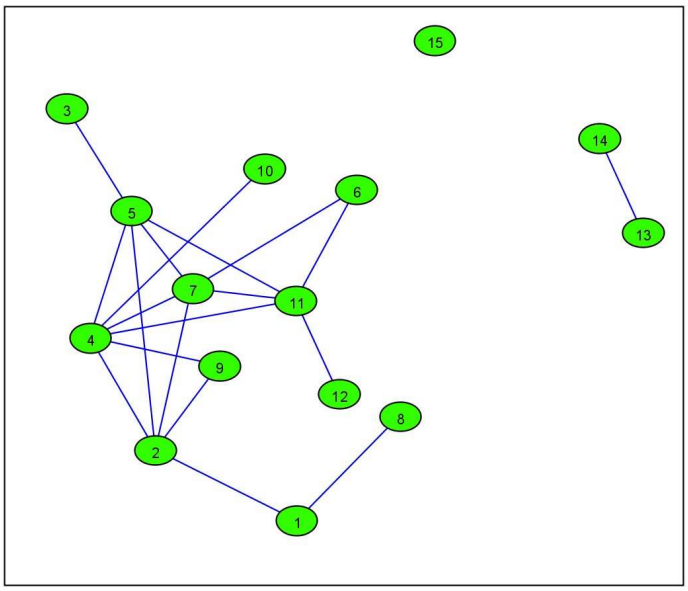


Figure 4.29: Simulation example from Jones *et al.* (2005). The top decomposable global graph on 15 nodes simulated example.

by a discussion of TMH and UMH in an analysis of the full, large cardiovascular genomics data set with the same targeted variable Y. I show that TMH is still not converging fast enough to deal with very high-dimensional data sets.

4.5.1 Projected risk signature and 13 factors from BFRM

I use the same prior of graph as Section 3.4, i.e., the prior probability of every edge being connected is 1/13. Intuitively, compared to the top models obtained in Section 3.4 I expect the mean probability local models found by TMH or UMH to be sparser because they allow non-local edges to exist. For TMH, λ is set to be 0.9, and I run TMH and UMH for 100,000 iterations plus 10,000 burn-in iterations. The local edges with inclusion probabilities bigger than 0.1 are shown in Figure 4.32 and 4.33. Note that both TMH and UMH select factors 1, 3, 4, 7 and 9 in the mean probability models, while the other three factors selected by

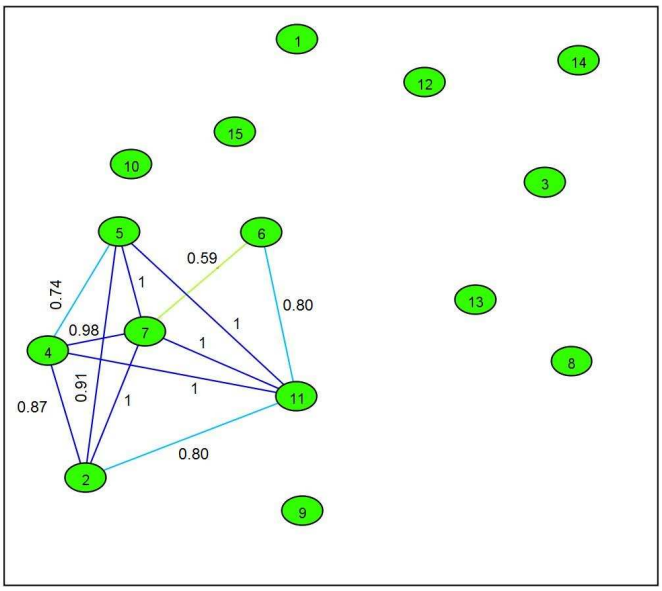


Figure 4.30: The local edges found by TMH on 15 nodes simulated example ($\lambda = 0.9$, edge inclusion probabilities bigger than 0.5).

SSS (factor 2, 6, 13) have less significant inclusion probabilities.

4.5.2 Projected risk signature and the human gene probesets

Similar to Section 3.4.2, I also applied TMH and UMH to the 122×4287 human gene data set with the projected risk signature as the target variable. Unfortunately, both TMH and UMH fail to find a local edge with inclusion probability greater than 0.1 after 1 million iterations, mainly because of slow convergence and high collinearity among the genes. This slow convergence property is due to the intrinsically slow convergence of Metropolis-Hastings methods (Jones *et al.*, 2005) in high-dimensional problems. For UMH, the non-convergence is not surprising: This is the reason that TMH is motivated and developed. For TMH, there is one main problem of the targeted proposal. In such a high-dimensional problem, the TMH adding edge proposal with a high λ tends to mainly focus on proposing to add an

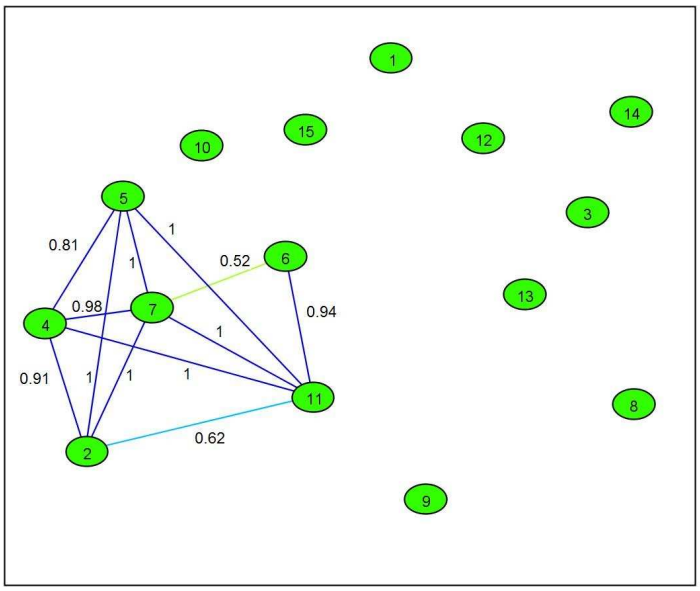


Figure 4.31: The local edges found by UMH on 15 nodes simulated example (edge inclusion probabilities bigger than 0.5).

edge incident to Y, and spend very little time on proposing edges connecting the current neighbors of Y. For example, assume that $ne(Y)$ consists of 7 neighbors at a certain iteration, the probability of proposing an edge connecting two nodes in $ne(Y)$ is only about $7 * 6/2/((4287 - 7) + 7 * 6/2) = 0.0049$. As a result, during the TMH sampling, the graph usually contains very few local edges that connect neighbors of Y. This may be one important reason of the slow convergence.

To address this slow convergence problem of both UMH and TMH, one idea is to start the Markov Chain with a "better" local graph instead of the null graph. One way to realize this is to run regression modeling SSS for a small number of iterations to find out the top set of predictors, and use them to form an initial graph as a starting point of UMH (TMH). For this human gene probsets data, we have the top regression model found by SSS in 100,000 iterations in Section 3.4, consisting of predictors $\{1, 3, 4, 6, 9, 12\}$ in Appendix C. Two corresponding initial graphs has been tested, one is a graph with only 6 edges

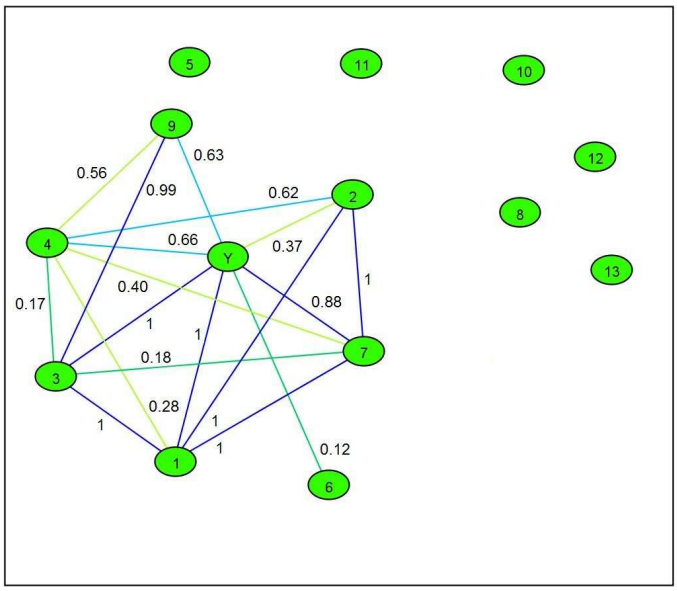


Figure 4.32: The local edges found by TMH on 13 factors data ($\lambda = 0.9$, edge inclusion probabilities bigger than 0.1)

$\{(Y,1),(Y,3),(Y,4),(Y,6),(Y,9),(Y,12)\}$, and the other one is a graph composed by a full subgraph of the vertices set $\{Y,1,3,4,6,9,12\}$ and a null subgraph of all the other vertices. Unfortunately, both initial graphs do not change the non-convergence of TMH, as all the 6 pre-selected neighbors of Y are removed from the graph in hundreds iterations of TMH.

4.6 Summary Comments

In this chapter I have introduced a new type of Metropolis-Hastings proposal: Targeted Metropolis-Hastings for local graphical model search. I compared TMH with UMH in local convergence rates, and found out empirically that initially TMH converges to the local stationary distribution much faster than UMH (the bigger λ is, the faster TMH will converge initially), while afterwards, UMH dominates TMH in the convergence rate. To summarize the sampling graphs, I suggested to look at the local edge inclusion probabilities, and select

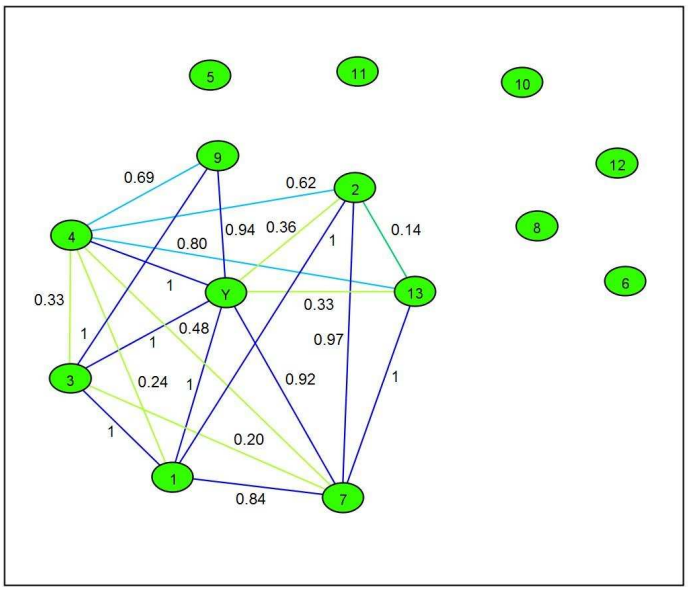


Figure 4.33: The local edges found by UMH on 13 factors data (edge inclusion probabilities bigger than 0.1).

the ones with significant inclusion probabilities. In simulation studies, I found that TMH converges faster than UMH in the sense of MSE of the edge inclusion probabilities. Finally, I applied both TMH and UMH methods to three real data sets, where I found that both methods work well when p is small (e.g. $p = 15$), but converge very slowly when p is large (e.g. $p = 4287$). The further acceleration of TMH convergence is an open question.

Chapter 5

Variational Bayes Model Selection: An Improvement Over Laplace?

Increasingly, statisticians are faced with the problem of identifying interesting subsets of predictors from among a large number of candidates. Existing methods for variable selection, such as MCMC sampling, tend to explore the model space too slowly in large dimensions. Shotgun stochastic search (SSS) algorithms (Hans, 2005; Hans *et al.*, 2007) have been proposed as an efficient alternative. As current SSS algorithms rely on conjugacy, they are not appropriate for generalized linear models without use of approximation methods. This chapter compares the frequently used Laplace approximation with two alternatives based on Variational Bayes methods. The comparison is illustrated using several simulated data examples and an application to the problem of predicting conception using data on timing of intercourse in the menstrual cycle. This application also illustrates the problem of selection of interactions.

5.1 Introduction

As the collection of massive amounts of information becomes more routine, there is a critical need for more efficient methods for identifying promising subsets of variables from among the very many candidates one is typically faced with. This problem occurs not only in genomics and bioinformatics studies, where it has received the most focus, but also in epidemiologic studies. For example, the application motivating this chapter focuses on using data on the timing of intercourse in the menstrual cycle to predict conception. The data consist of daily records of intercourse across the menstrual cycle and an indicator of conception status for

women enrolled in a European study (Dunson *et al.* (2002)). Although the fertile interval of the menstrual cycle is only 5-6 days for most women (Dunson *et al.* (1999)), the timing of the fertile days is highly uncertain (Wilcox *et al.* (2000)). Hence, there are many days in the menstrual cycle during which intercourse can potentially result in a pregnancy. In addition, potential interactions lead to a very high-dimensional set of candidate models.

One widely used approach for accommodating uncertainty in subset selection in linear regression models is the stochastic search variable selection (SSVS) algorithm originally proposed in George and McCulloch (1993), with numerous modifications later considered (George and McCulloch (1997); Ishwaran and Rao (2005); Casella and Moreno (2006)). SSVS algorithms rely on using a mixture prior for the regression coefficients, with one component concentration at 0, allowing a predictor to effectively drop out of the model. Gibbs sampling is then used to sample from the conditional posterior distributions of the coefficients, resulting in stochastic changes to the variables included in the model across the MCMC iterates. Such algorithms are very effective in modest sized models, but tend to explore the model space too slowly as the number of candidate variables increases.

Motivated by this problem, Hans (2005) and Hans *et al.* (2007) proposed a new regression model search algorithm named Shotgun Stochastic Search (SSS), with Jones *et al.* (2005) applying this approach to graphical models (Section 2.3). Compared to the variety of alternative methods available (reviewed by Dellaportas *et al.* (2002)), Hans *et al.* (2007) argue that SSS is more efficient at rapidly identifying the models with the highest posterior probabilities in large model spaces. It is interesting to extend the SSS beyond linear regression to broader classes of regression models, such as generalized linear models (GLMs). Hans (2005) and Hans *et al.* (2007) proposed to use Laplace approximation for marginal likelihood approximations in GLM, and used SSS to search in the model space. Ntzoufras *et al.* (2003) proposed a reversible jump MCMC algorithm for posterior computation in GLMs when there is uncertainty in the predictors to be included. In recent work, Wang and George (2007) proposed an adaptive Bayesian criterion for variable selection in GLMs,

relying on an integrated Laplace approximation to allow rapid computation.

My initial goal is to consider applications of SSS algorithms to GLM variable selection in massive dimensions, with a particular emphasis on logistic regression models motivated by the fertility application. For subject i ($i = 1, \ldots, n$), let y_i denote the binary response variable and let $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})'$ denote a $p \times 1$ vector of candidate predictors, with γ_j a 0/1 indicator that the jth predictor is included in the model and $\boldsymbol{x}_{\gamma,i} = \{x_{ij} : \gamma_j = 1\}$ denoting the subset of included predictors. Then, the logistic regression model can be expressed as

$$\text{logit}\{\Pr(y_i = 1 \,|\, \boldsymbol{x}_i, \boldsymbol{\gamma})\} = \boldsymbol{x}'_{\gamma,i}\boldsymbol{\beta}_\gamma, \tag{5.1}$$

where $\text{logit}(x) = ln(\frac{x}{1-x})$, and $\boldsymbol{\beta}_\gamma$ is a vector of the unknown regression coefficients in the model indexed by $\boldsymbol{\gamma}$.

Letting $\pi(\boldsymbol{\gamma})$ denote the prior probability of model $\boldsymbol{\gamma}$, the posterior model probability is

$$p(\boldsymbol{\gamma} \,|\, \boldsymbol{Y}, \boldsymbol{X}) = \frac{p(\boldsymbol{\gamma})\, p(\boldsymbol{Y} \,|\, \boldsymbol{X}, \boldsymbol{\gamma})}{p(\boldsymbol{Y} \,|\, \boldsymbol{X}, \gamma)},$$

where the marginal likelihood of the data under model $\boldsymbol{\gamma}$ is

$$p(\boldsymbol{Y}|\boldsymbol{X}, \gamma) = \int p(\boldsymbol{Y}|\boldsymbol{X}_{\boldsymbol{\gamma}}, \boldsymbol{\beta}_{\boldsymbol{\gamma}}) p(\boldsymbol{\beta}_{\boldsymbol{\gamma}}) d\boldsymbol{\beta}_{\boldsymbol{\gamma}}, \tag{5.2}$$

which is not available analytically in most cases. In logistic regression, the marginal likelihood is expressed as the integral of the Bernoulli likelihood under the logistic model over the prior distribution for the coefficients, which does not have a closed form.

To complete a Bayesian specification of the model uncertainty problem, explicit choices are required for the prior probability of model $\boldsymbol{\gamma}$ and for the prior distribution on the coefficients within that model $\boldsymbol{\beta}_{\boldsymbol{\gamma}}$, for all $\boldsymbol{\gamma} \in \Gamma$, with Γ the list of models corresponding to all possible subsets of variables. The standard choice of prior for $\boldsymbol{\gamma}$ corresponds to

$$p(\boldsymbol{\gamma}) = \prod_{j=1}^{p} \phi^{1(\gamma_j = 1)} (1 - \phi)^{1(\gamma_j = 0)}, \tag{5.3}$$

where $1(\cdot)$ is a 0/1 indicator function, and ϕ is potentially given a beta hyperprior to allow the data to inform more strongly about model size.

In conducting a high-dimensional model search, it is necessary to rapidly estimate or approximate the marginal likelihood for massive numbers of models. There are many simulation-based methods that can be used for estimating the marginal likelihood and/or Bayes factors for comparing a small number of competing models (Gelfand and Smith (1990); Gelfand and Dey (1994); Verdinelli and Wasserman (1995); Chib (1995); DiCiccio *et al.* (1997); Gelman and Meng (1998); Han and Carlin (2001); Chib and Jeliazkov (2001)). However, most approaches require substantial numbers of samples within each model to produce an accurate estimate, so are impractical in conducting a model search.

Hence, it is necessary to focus on marginal likelihood approximations that can be calculated very quickly. The Laplace approximation provides a convenient and widely used approach, which often performs well (Tierney and Kadane (1986)). DiCiccio *et al.* (1997) provide the details of the Laplace approximation to the marginal likelihood. Raftery (1996) use the Laplace approximation for Bayesian model selection in GLMs, while Hans (2005); Hans *et al.* (2007) combined this approach with SSS. There are also other alternatives for estimating the marginal likelihood in logistic regression. Jaakkola and Jordan (2000) proposed a variational Bayes (VB) approach to approximate the posterior of $\boldsymbol{\beta}$ by a variational transformation of the logistic function. Their paper suggests two different VB methods.

It is widely believed in the machine learning community that the VB approach provides an improvement over the Laplace approximation. Srebro and Jaakkola (2003) pointed out that for the Taylor expansion, the iterative improvement of the approximation is not always monotonic, resulting in no guarantee of convergence. They also claimed that the VB method is more robust and the convergence is guaranteed. However, Wang and Titterington (2005); Consonni and Marin (2007) proved that "the covariance matrices from the variational Bayes approximations are usually 'too small' compared with those for the maximum likelihood estimator". My initial motivation was to apply the VB method in the high dimensional

variable selection setting to improve upon Laplace-based methods, though I found that Laplace consistently outperformed VB in logistic regression.

I first introduce the Laplace and VB methods, and describe their use in applying the shotgun stochastic search algorithm to high-dimensional variable selection. This is followed by two simulation studies; the first compares accuracy of the marginal likelihood approximations, and the second assesses predictive performance using VB or Laplace model averaging. Finally, I implement the SSS algorithm with VB and Laplace for the fertility data set.

5.2 Marginal Likelihood Approximations

5.2.1 Laplace Method

In this subsection, I provide a brief review of the Laplace approximation to the marginal likelihood in logistic regression. Re-expressing marginal likelihood (5.2) in the form

$$\int \log\Big[\exp\big\{p(\boldsymbol{Y} \mid \boldsymbol{X}_{\boldsymbol{\gamma}}, \boldsymbol{\beta}_{\boldsymbol{\gamma}})\, p(\boldsymbol{\beta}_{\boldsymbol{\gamma}})\, d\boldsymbol{\beta}_{\boldsymbol{\gamma}}\big\}\Big],$$

expanding the exponential function using Taylor series, and keeping the items up to the second order, we have the estimator

$$\hat{p}(\boldsymbol{Y}|\boldsymbol{X}_{\boldsymbol{\gamma}}) = (2\pi)^{(k_{\boldsymbol{\gamma}}+1)/2}|\hat{\Sigma}_{\boldsymbol{\gamma}}|^{1/2}\, p(\boldsymbol{Y}|\boldsymbol{X}_{\boldsymbol{\gamma}}, \hat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}})p(\hat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}), \tag{5.4}$$

where $k_{\boldsymbol{\gamma}}$ is the size of model $\boldsymbol{\gamma}$, $\hat{\Sigma}_{\boldsymbol{\gamma}}^{-1}$ is the approximate posterior covariance matrix

$$\hat{\boldsymbol{\Sigma}_{\boldsymbol{\gamma}}}^{-1} = -\frac{\partial^2}{\partial\hat{\beta}_i \partial\hat{\beta}_j}\Big[log\big\{p(\boldsymbol{Y}|\boldsymbol{X}_{\boldsymbol{\gamma}}, \hat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}})p(\hat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}})\big\}\Big], \tag{5.5}$$

and $\hat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}}$ is the posterior mode under model $\boldsymbol{\gamma}$

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{\gamma}} = argmax_{\boldsymbol{\beta}_{\boldsymbol{\gamma}}}\big\{p(\boldsymbol{Y} \mid \boldsymbol{X}_{\boldsymbol{\gamma}}, \boldsymbol{\beta}_{\boldsymbol{\gamma}})p(\boldsymbol{\beta}_{\boldsymbol{\gamma}})\big\}. \tag{5.6}$$

To find the posterior mode, $\boldsymbol{\beta}_{\boldsymbol{\gamma}}$, for a logistic regression, a simple Newton-Raphson algorithm can be used, with Raftery (1996) suggesting a one-step approximation that allows use of maximum likelihood estimates. Here, I instead iterate the algorithm until convergence. If the prior of $\boldsymbol{\beta}_{\boldsymbol{\gamma}}$ is $N(\mathbf{0}, \tau I_{k_{\boldsymbol{\gamma}}+1})$, the algorithm is based on the following updating equation:

$$\boldsymbol{\beta}_{\boldsymbol{\gamma}}^{(t+1)} = \boldsymbol{\beta}_{\boldsymbol{\gamma}}^{(t)} - G(\boldsymbol{\beta}_{\boldsymbol{\gamma}}^{(t)})^{-1} g(\boldsymbol{\beta}_{\boldsymbol{\gamma}}^{(t)}), \tag{5.7}$$

where

$$\begin{aligned} G(\boldsymbol{\beta}_{\boldsymbol{\gamma}}) &= \frac{1}{\tau} I_{k+1} - \sum_{i=1}^{n} \boldsymbol{x}_{\boldsymbol{\gamma},i} \boldsymbol{x}'_{\boldsymbol{\gamma},i} \phi_{\boldsymbol{\gamma},i}(1 - \phi_{\boldsymbol{\gamma},i}), \\ g(\boldsymbol{\beta}_{\boldsymbol{\gamma}}) &= -\frac{\boldsymbol{\beta}_{\boldsymbol{\gamma}}}{\tau} + \sum_{i=1}^{n} (y_i - \phi_{\boldsymbol{\gamma},i}) \boldsymbol{x}_{\boldsymbol{\gamma},i}, \\ \phi_{\boldsymbol{\gamma},\boldsymbol{i}} &= \frac{1}{1 + \exp(-\boldsymbol{x}'_{\boldsymbol{\gamma},i} \boldsymbol{\beta}_{\boldsymbol{\gamma}})}. \end{aligned} \tag{5.8}$$

Iterative updating of $\boldsymbol{\beta}_{\boldsymbol{\gamma}}$ tends to converge within a few iterations, so that the approximation to the marginal likelihood under model $\boldsymbol{\gamma}$, $\hat{p}(\boldsymbol{Y}|\boldsymbol{X}_{\boldsymbol{\gamma}})$ can be obtained very quickly.

5.2.2 Variational Bayes Approximations

In this section, I describe two VB approaches to approximate the logistic regression marginal likelihood. The first approach was suggested by Jaakkola and Jordan (2000), while the second approach is also briefly introduced in their paper, though they present it as less appealing than the first approach.

Approach I

By Bayes' Theorem,

$$p(\boldsymbol{Y}|\boldsymbol{X}_{\boldsymbol{\gamma}}) = \frac{p(\boldsymbol{Y}|\boldsymbol{X}_{\boldsymbol{\gamma}}, \boldsymbol{\beta}^*_{\boldsymbol{\gamma}}) p(\boldsymbol{\beta}^*_{\boldsymbol{\gamma}})}{p(\boldsymbol{\beta}^*_{\boldsymbol{\gamma}}|\boldsymbol{Y}, \boldsymbol{X}_{\boldsymbol{\gamma}})}, \tag{5.9}$$

where $\boldsymbol{\beta}^*_{\boldsymbol{\gamma}}$ is any vector in $\Re^{k_{\boldsymbol{\gamma}}}$. Expression (5.9) is commonly used in estimating marginal likelihoods via Monte Carlo sampling (e.g. Chen (2005); Chib and Jeliazkov (2001)). Following Jaakkola and Jordan (2000), for any single observation y_i, we have

$$p(y_i|\boldsymbol{x}_{\boldsymbol{\gamma},i}, \boldsymbol{\beta}_{\boldsymbol{\gamma}}) \geq p(y_i|\boldsymbol{x}_{\boldsymbol{\gamma},i}, \boldsymbol{\beta}_{\boldsymbol{\gamma}}, \xi) = g(\xi) \exp\{(X_i - \xi)/2 + \lambda(\xi)(X_i^2 - \xi^2)\}, \tag{5.10}$$

where $g(\xi) = (1 + e^{-\xi})^{-1}$, $\lambda(\xi) = [1/2 - g(\xi)]/(2\xi)$, and $X_i = (2y_i - 1)\boldsymbol{x}'_{\boldsymbol{\gamma},i}\boldsymbol{\beta}_{\boldsymbol{\gamma}}$.

The inequality in (5.10) holds for any ξ. Because $p(y_i|\boldsymbol{x}_{\boldsymbol{\gamma},i}, \boldsymbol{\beta}_{\boldsymbol{\gamma}}, \xi)$ has a quadratic form, when the prior of $\boldsymbol{\beta}_{\boldsymbol{\gamma}}$ is Gaussian, the posterior $p(\boldsymbol{\beta}_{\boldsymbol{\gamma}}|\boldsymbol{x}_{\boldsymbol{\gamma},i}, y_i, \boldsymbol{x}_{\boldsymbol{\gamma},i}, \xi)$ is also Gaussian. Potentially, one can choose ξ so that $p(\boldsymbol{\beta}_{\boldsymbol{\gamma}}|\boldsymbol{x}_{\boldsymbol{\gamma},i}, y_i, \boldsymbol{x}_{\boldsymbol{\gamma},i}, \xi)$ provides a good approximation to $p(\boldsymbol{\beta}_{\boldsymbol{\gamma}}|\boldsymbol{x}_{\boldsymbol{\gamma},i}, y_i, \boldsymbol{x}_{\boldsymbol{\gamma},i})$. From (5.10),

$$\int p(y_i|\boldsymbol{x}_{\boldsymbol{\gamma},i}, \boldsymbol{\beta}_{\boldsymbol{\gamma}}, \xi) p(\boldsymbol{\beta}_{\boldsymbol{\gamma}}) d\boldsymbol{\beta}_{\boldsymbol{\gamma}} \geq \int p(y_i|\boldsymbol{x}_{\boldsymbol{\gamma},i}, \boldsymbol{\beta}_{\boldsymbol{\gamma}}, \xi) p(\boldsymbol{\beta}_{\boldsymbol{\gamma}}) d\boldsymbol{\beta}_{\boldsymbol{\gamma}}. \tag{5.11}$$

Hence, the best possible approximation to the posterior utilizing the bound in (5.10) is achieved by choosing the value of ξ that maximizes the right hand side of this inequality. This maximization can proceed via the Jaakkola and Jordan (2000) EM algorithm. Letting $N(\boldsymbol{\mu}_{\boldsymbol{\gamma},0}, \boldsymbol{\Sigma}_{\boldsymbol{\gamma},0})$ denote the prior for $\beta_{\boldsymbol{\gamma}}$, initializing $i = 0$, and choosing an arbitrary positive starting point for ξ, the algorithm iterations through the following steps for $i = 1, \ldots, n$:

1. Apply the following updating equations:

$$\boldsymbol{\Sigma}^{-1}_{\boldsymbol{\gamma},i} = \boldsymbol{\Sigma}^{-1}_{\boldsymbol{\gamma},i-1} + 2|\lambda(\xi)|\boldsymbol{x}_{\boldsymbol{\gamma},i}\boldsymbol{x}'_{\boldsymbol{\gamma},i} \tag{5.12}$$

$$\boldsymbol{\mu}_{\boldsymbol{\gamma},i} = \boldsymbol{\Sigma}_{\boldsymbol{\gamma},i}\{\boldsymbol{\Sigma}^{-1}_{\boldsymbol{\gamma},i-1}\boldsymbol{\mu}_{\boldsymbol{\gamma}} + (y_i - 1/2)\boldsymbol{x}_{\boldsymbol{\gamma},i}\} \tag{5.13}$$

2. Update ξ by:

$$\xi^2 = \boldsymbol{x}'_{\boldsymbol{\gamma},i}\boldsymbol{\Sigma}_{\boldsymbol{\gamma},i}\boldsymbol{x}_{\boldsymbol{\gamma},i} + (\boldsymbol{x}'_{\boldsymbol{\gamma},i}\boldsymbol{\mu}_{\boldsymbol{\gamma},i})^2 \tag{5.14}$$

   Go back to Step 1 and repeated until convergence (Jaakkola and Jordan (2000) claim 6-7 iterations is sufficient).

3. Let $i = i + 1$ and go to step 1 until all subjects are added.

The estimated posterior $p(\boldsymbol{\beta}_{\boldsymbol{\gamma}}|\boldsymbol{Y}, \boldsymbol{X}_{\boldsymbol{\gamma}}) \stackrel{d}{=} N(\boldsymbol{\beta}_{\boldsymbol{\gamma}}; \boldsymbol{\mu}_{\boldsymbol{\gamma},n}, \boldsymbol{\Sigma}_{\boldsymbol{\gamma},n})$. In using this VB approximation to the posterior to obtain an approximation to the marginal likelihood via (5.9), I find substantial sensitivity to the value of $\boldsymbol{\beta}^*_{\boldsymbol{\gamma}}$. Such sensitivity has been noted in previous use of (5.9) in approximating marginal likelihoods, and I follow common practice in using the posterior mean of $\boldsymbol{\beta}$. This is convenient, as the posterior mean conveniently corresponds to $\boldsymbol{\mu}_{\boldsymbol{\gamma},n}$, the value obtained at the final iteration of the above EM algorithm. I refer to the resulting estimator of the marginal likelihood as the VB1 estimator.

Approach II

By the variational transformation (5.10), we can also obtain

$$p(\boldsymbol{Y}|\boldsymbol{X}_{\boldsymbol{\gamma}}) \geq p(\boldsymbol{Y}|\boldsymbol{X}_{\boldsymbol{\gamma}}, \boldsymbol{\xi}) = \int \prod_{i=1}^{n} p(y_i|\boldsymbol{x}_{\boldsymbol{\gamma},i}, \boldsymbol{\beta}_{\boldsymbol{\gamma}}, \xi_i) p(\boldsymbol{\beta}_{\boldsymbol{\gamma}}) d\boldsymbol{\beta}_{\boldsymbol{\gamma}}. \tag{5.15}$$

The right hand side of the inequality (5.15) provides an alternative estimator of the marginal likelihood, with the vector $\boldsymbol{\xi}$ chosen to maximize the lower bound. As in Section 5.2.2 Approach I, the EM algorithm can be used to estimate the optimal value of $\boldsymbol{\xi}$. Initializing $\boldsymbol{\xi}$, the algorithm iterates as follows until convergence:

1. Update the estimated posterior covariance and mean as

$$\begin{aligned} \boldsymbol{\Sigma}_{\boldsymbol{\gamma}}^{-1} &= \boldsymbol{\Sigma}_{\boldsymbol{\gamma},0}^{-1} + 2\sum_{i=1}^{n} |\lambda(\xi_i)| \boldsymbol{x}_{\boldsymbol{\gamma},i} \boldsymbol{x}'_{\boldsymbol{\gamma},i} \\ \boldsymbol{\mu}_{\boldsymbol{\gamma}} &= \boldsymbol{\Sigma}_{\boldsymbol{\gamma}} \big\{ \boldsymbol{\Sigma}_{\boldsymbol{\gamma},0}^{-1} \boldsymbol{\mu}_{\boldsymbol{\gamma},0} + \sum_{i=1}^{n} (y_i - 1/2) \boldsymbol{x}_{\boldsymbol{\gamma},i} \big\} \end{aligned}$$

2. Update the variational parameters $\boldsymbol{\xi}$ by:

$$\xi_i^2 = \boldsymbol{x}_{\boldsymbol{\gamma},i}^T \boldsymbol{\Sigma}_{\boldsymbol{\gamma}} \boldsymbol{x}_{\boldsymbol{\gamma},i} + (\boldsymbol{x}'_{\boldsymbol{\gamma},i} \boldsymbol{\mu}_{\boldsymbol{\gamma}})^2, \quad i = 1, \ldots, n, \tag{5.16}$$

After convergence, the approximation to the marginal likelihood is then

$$p(\boldsymbol{Y}|\boldsymbol{X}_{\boldsymbol{\gamma}}, \boldsymbol{\xi}) = \prod_{i=1}^{n} g(\xi_i) \exp\left\{ -\frac{1}{2}\sum_{i=1}^{n} \xi_i - \sum_{i=1}^{n} \lambda(\xi_i)\xi_i^2 \right\} \frac{\left|\boldsymbol{\Sigma}_{\boldsymbol{\gamma}}\right|^{1/2}}{\left|\boldsymbol{\Sigma}_{\boldsymbol{\gamma},0}\right|^{1/2}}$$
$$\exp\left\{ \frac{\boldsymbol{\mu}_{\boldsymbol{\gamma}}'\boldsymbol{\Sigma}_{\boldsymbol{\gamma}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\gamma}} - \boldsymbol{\mu}_{\boldsymbol{\gamma},0}'\boldsymbol{\Sigma}_{\boldsymbol{\gamma},0}^{-1}\boldsymbol{\mu}_{\boldsymbol{\gamma},0}}{2} \right\}. \tag{5.17}$$

I refer to this estimator as VB2. Jaakkola and Jordan (2000) proposed this approach as an alternative to VB1, but believed that VB1 is cleaner because optimizing the variational parameters sequentially instead of jointly is usually better. However, I find that VB2 has the advantage of producing a much more stable estimator of the marginal likelihood than VB1, which has a disturbing degree of sensitivity to $\boldsymbol{\beta}_{\boldsymbol{\gamma}}^{*}$. VB2 does take longer to compute, particularly in cases involving large number of subjects, which implies a high-dimensional $\boldsymbol{\xi}$.

5.3 Shotgun Stochastic Search for Regression Modeling

Hans (2005) and Hans *et al.* (2007) proposed the Shotgun Stochastic Search (SSS) algorithm as an alternative to SSVS for searching for high posterior probability models in cases in which the model space is defined by all possible subsets of a high-dimensional vector of predictors. Let $\boldsymbol{\gamma}' \in \eta(\boldsymbol{\gamma})$ denote the subset of Γ corresponding to those models in a neighborhood of $\boldsymbol{\gamma}$, defined to consist of all those model obtained by adding or deleting a single predictor or substituting the predictor for another from among the candidates.

Similar to SSS algorithm for graphical model search (Section 2.3), the SSS algorithm for regression modeling searches the model space by iterating through the following steps a large number of times after choosing an initial model $\boldsymbol{\gamma}$:

1. Proceeding in parallel, calculate scores for all models in $\eta(\boldsymbol{\gamma})$, the neighborhood of the current model $\boldsymbol{\gamma}$, with these scores defined to be proportional to $p^*(\boldsymbol{\gamma} \mid \boldsymbol{Y}, \boldsymbol{X}_{\boldsymbol{\gamma}}) = p(\boldsymbol{\gamma})\hat{p}(\boldsymbol{Y} \mid \boldsymbol{X}_{\boldsymbol{\gamma}})$, where $\hat{p}(\boldsymbol{Y} \mid \boldsymbol{X}_{\boldsymbol{\gamma}})$ is an estimate of the marginal likelihood under model $\boldsymbol{\gamma}$. Note that one cannot calculate the posterior model probability $p(\boldsymbol{\gamma} \mid \boldsymbol{Y}, \boldsymbol{X}_{\boldsymbol{\gamma}})$ as the normalizing constant involves summing over all possible subsets.

2. Randomly select one new model $\boldsymbol{\gamma}'$ from $\eta(\boldsymbol{\gamma})$ by sampling with probabilities proportional to $p^*(\boldsymbol{\gamma} \mid \boldsymbol{Y}, \boldsymbol{X}_{\boldsymbol{\gamma}})^{\alpha}$, where $\alpha \in [0, 1]$ is an annealing parameter.

By using annealing, with the annealing parameter tuned based on the problem, one limits the tendency of SSVS algorithms to remain for long intervals in local regions of the model space. Empirically, the SSS algorithm has proven very efficient at finding the top models compared to MCMC methods.

5.4 Simulation Analysis

Unfortunately, it is very difficult to compare the three methods described in Section 5.2 (Laplace, VB1, VB2) theoretically. If the marginal likelihoods were analytically tractable, then one would not need these approximation methods, and it has proven difficult to theoretically justify the tightness of the VB lower bounds, as this is entirely problem dependent. Hence, I rely on simulations to assess relative performance. In assessing accuracy of the marginal likelihood approximations, one challenge is that we lack knowledge of the exact marginal likelihoods even for simulated data. To address this problem, I follow the approach of implementing importance sampling for a very large number of samples, with the resulting estimated marginal likelihood used as the gold standard. I also tried a number of recently proposed Monte Carlo methods for estimating the marginal likelihood, but found that alternative approaches often did not converge to the same estimate even when using a 100,000s of samples. In contrast, one could easily judge convergence of importance sampling, and collect sufficient numbers of samples to produce a highly-accurate estimate.

In Section 5.4.1, I assess the relative performance of Laplace, VB1 and VB2 in estimating marginal likelihood for logistic regression models in different simulated data sets. In Section 5.4.2, I then compare predictive performance of Bayesian model averaging using posterior model probabilities estimated under the three different approaches.

5.4.1 Accuracy of Marginal Likelihood Estimation

Let $g(\boldsymbol{\beta}_{\boldsymbol{\gamma}})$ correspond to the multivariate t distribution, with low degrees of freedom ($\nu = 3$) and with the mean and variance chosen as the VB2 approximated posterior mean and variance. Then, rewrite $p(\boldsymbol{Y}_{\boldsymbol{\gamma}} \mid \boldsymbol{X}_{\boldsymbol{\gamma}})$ in (5.2) as

$$p(\boldsymbol{Y} \mid \boldsymbol{X}_{\boldsymbol{\gamma}}) = \int \frac{p(\boldsymbol{Y}|\boldsymbol{X}_{\boldsymbol{\gamma}}, \boldsymbol{\beta}_{\boldsymbol{\gamma}})p(\boldsymbol{\beta}_{\boldsymbol{\gamma}})}{g(\boldsymbol{\beta}_{\boldsymbol{\gamma}})} g(\boldsymbol{\beta}_{\boldsymbol{\gamma}}) d\boldsymbol{\beta}_{\boldsymbol{\gamma}}. \tag{5.18}$$

By simulating samples of $\boldsymbol{\beta}_{\boldsymbol{\gamma}}$ from $g(\boldsymbol{\beta}_{\boldsymbol{\gamma}})$, the marginal likelihood can be estimated by

$$p(\boldsymbol{Y}|\boldsymbol{X}_{\boldsymbol{\gamma}}) \approx \sum_{i=1}^{N} \frac{p(\boldsymbol{Y}|\boldsymbol{X}_{\boldsymbol{\gamma}}, \boldsymbol{\beta}_{\boldsymbol{\gamma}}^{(i)})p(\boldsymbol{\beta}_{\boldsymbol{\gamma}}^{(i)})}{g(\boldsymbol{\beta}_{\boldsymbol{\gamma}}^{(i)})}, \tag{5.19}$$

where $\boldsymbol{\beta}_{\boldsymbol{\gamma}}^{(i)}$ is the ith sample from $g(\boldsymbol{\beta})$. I use $N = 100,000$ samples, since I find this empirically to give a highly accurate estimate of the marginal likelihood.

In assessing accuracy of Laplace, VB1 and VB2 approximations, I simulated data sets under three different cases in which there were 6 candidate predictors, with the size of the true model equal to 1, 3, or 5 predictors. For each case, I simulated 100 data sets having $n = 50$ samples per data set, with the coefficients for the predictors that were included sampled independently from $N(0, 4^2)$. For one simulated data set for each model size, I used importance sampling (IS), Laplace, VB1 and VB2 to estimate the marginal likelihoods under each of the $2^6 = 64$ possible models, and I then sorted the models by the marginal likelihood estimates based on IS.

Figure 5.1 plots the IS, Laplace, VB1 and VB2 estimated log marginal likelihoods under the 64 sorted models in the model size 1 simulations, while Figures 5.2 and 5.3 presents

the corresponding results for the size 3 and size 5 simulations, respectively. Because the IS estimate was considered indistinguishable from the true marginal likelihood, all three plots indicate that the Laplace approximation is very close to the true marginal likelihood. Interestingly, the VB methods tend to be highly accurate for models with relatively low marginal likelihoods, but tend to substantially underestimate the marginal likelihood for good models. For example, when there is only one predictor, the 32 models that do not include the true predictor have a low marginal likelihood, and all the three approximation methods have very similar estimates to the importance sampling results. On the other hand, for the other 32 models that include the true predictor, the VB1 and VB2 estimates are both poor.

To avoid randomness from one data set, I have also tested the marginal likelihood estimation performance by different methods on all 100 data sets. The results are very similar to what I have shown in Figure 5.1, 5.2, and 5.3. As an example, Figure 5.4 shows the difference between the highly accurate IS estimate and the three fast approximate estimates averaging over the 100 simulated data sets in the model size 3 case. For each data set, I sort all the possible models by their estimated marginal likelihood by importance sampling, and compute the difference between importance sampling and the other three methods. Then for each rank of the model, I take the average of the differences. We can see from the plot that although Laplace approximation is underestimating a little bit the marginal likelihood, its estimation line is almost parallel to the estimation line made by importance sampling, which makes the model selection by Laplace approximation more robust. For the two variational methods, as I have addressed before, their estimation precision both become worse as the model becomes better, while VB2 is better than VB1.

5.4.2 Prediction Performance by Bayesian Model Averaging

When the model space is large, there are typically many models with similar posterior probabilities, so that it is not ideal to base prediction on a single selected model, and model

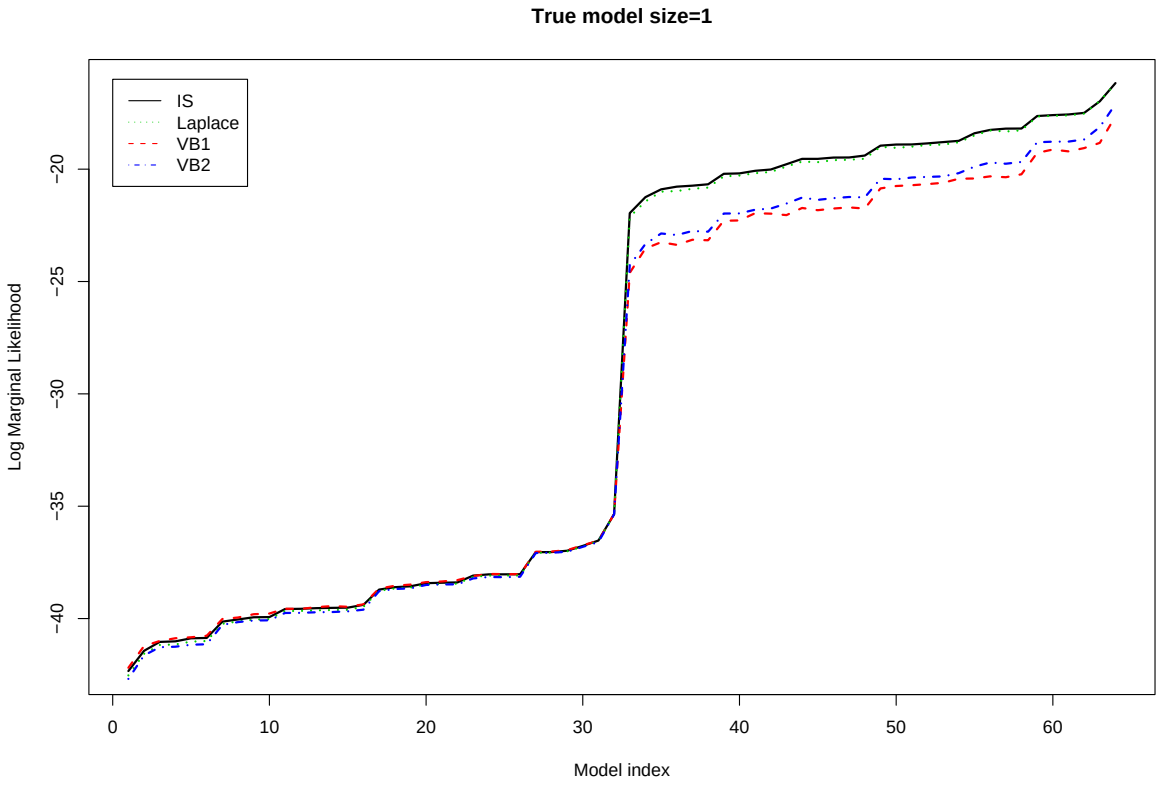


Figure 5.1: The estimated log-marginal likelihoods under importance sampling (IS), Laplace and the two variational Bayes methods (VB1, VB2) in a simulation case in which the true model size is 1. The models are ordered to be increasing in the IS estimated marginal likelihood.

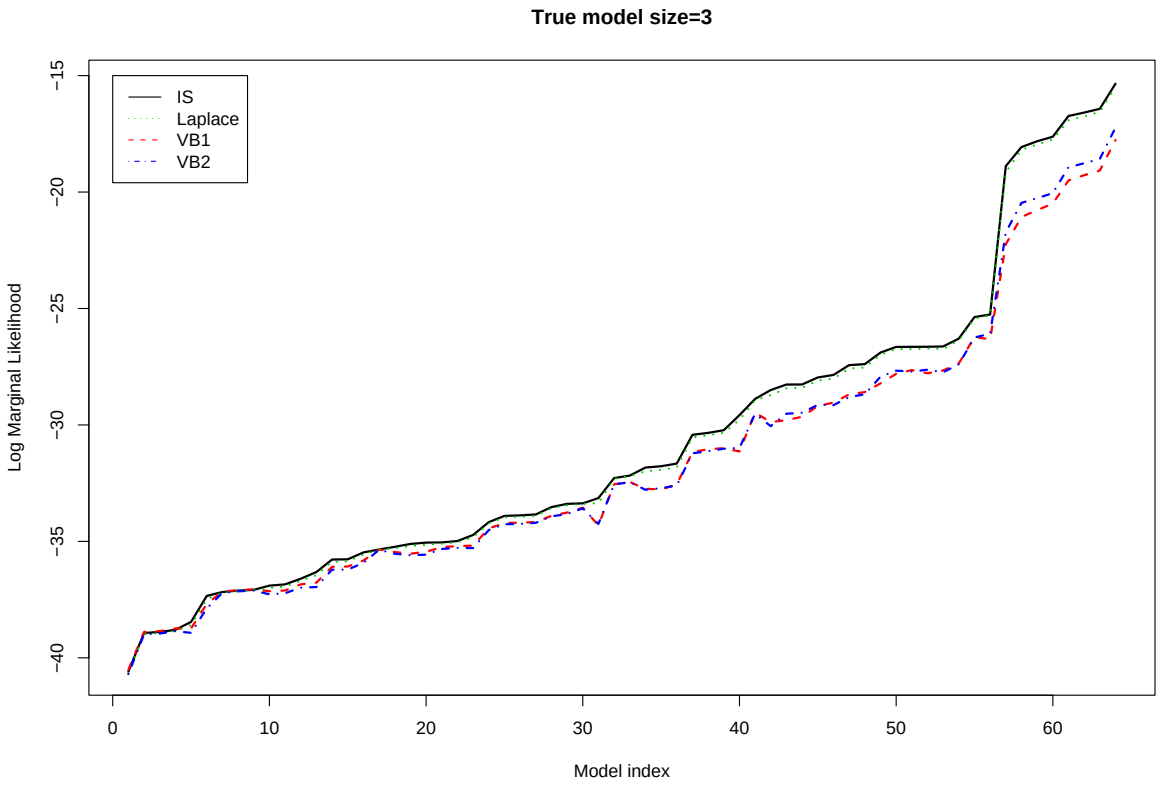


Figure 5.2: The estimated log-marginal likelihoods under IS, Laplace, VB1 and VB2 when the true model size is 3.

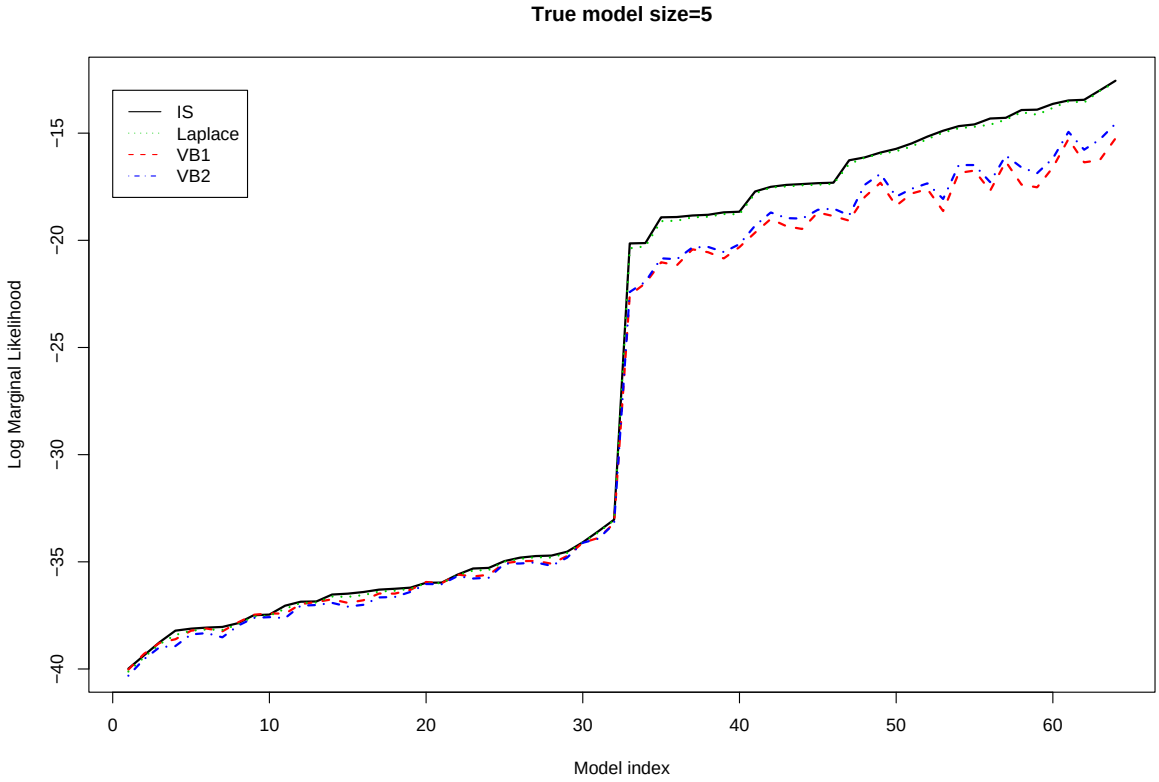


Figure 5.3: The estimated log-marginal likelihoods under IS, Laplace, VB1 and VB2 when the true model size is 5.

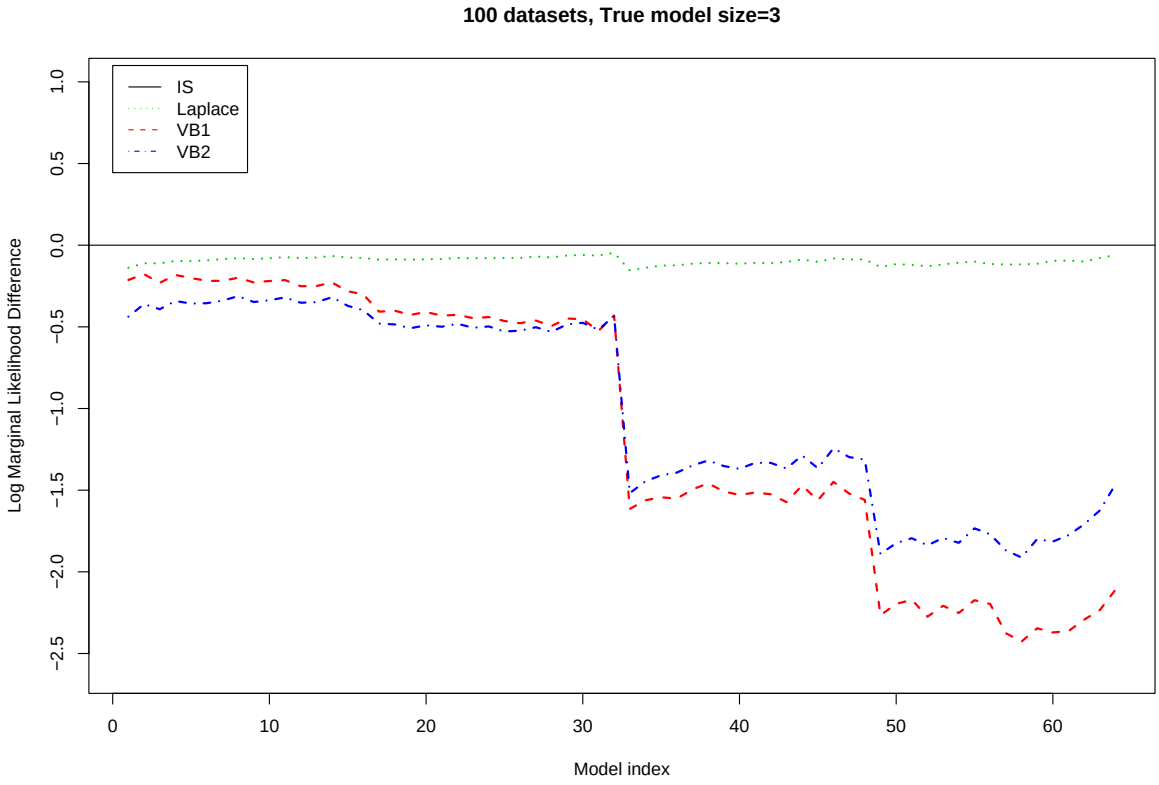


Figure 5.4: The average difference between the log-marginal likelihood estimated by IS and the estimates under Laplace, VB1 and VB2 for the 100 simulated data sets when the true model size is 3.

averaging is recommended. In particular, given predictors $\boldsymbol{X}^*$ for a new set of subjects, the model-averaged predictive distribution of $\boldsymbol{y}^*$ is

$$p(\boldsymbol{y}^*|\boldsymbol{X}^*, \boldsymbol{Y}, \boldsymbol{X}) = \sum_{\boldsymbol{\gamma}\in\Gamma} p(\boldsymbol{y}^*|\boldsymbol{X}^*_{\boldsymbol{\gamma}})p(\boldsymbol{\gamma}|\boldsymbol{Y}, \boldsymbol{X}). \tag{5.20}$$

In binary data, when $p(y_{n+1}|\boldsymbol{x}_{n+1}, \boldsymbol{Y}, \boldsymbol{X}) > 0.5$, subject $n+1$ is predicted to have $y_{n+1} = 1$ and otherwise the subject is predicted to have $y_{n+1} = 0$.

To test the predictive performance of the three methods, I simulate a data set with 200 potential predictors and 500 samples, with the size of the true model to be 9. After using SSS to search for the top 50 models for each marginal likelihood estimation method, I implement model averaging to do out-of-sample predictions for an additional 2000 samples. The results show that the Laplace approximation has a misclassification rate of 307/2000, while both VB1 and VB2 have a misclassification rate of 305/2000. In contrast, conducting prediction based on maximization of the true logistic regression model resulted in a rate of 308/2000. Hence, all three methods did as well as a frequentist analysis under the true model.

The average of the estimated posterior probabilities of $y_i = 1$ ($y_i = 0$) for subjects in the test sample with $y_i = 1$ ($y_i = 0$) was 0.728 (0.803) under the Laplace approximation and 0.709 (0.781) for both VB1 and VB2. Furthermore, the sum of the square of the difference between the true probabilities of $y_i = 1$ and the estimated probabilities by Laplace approximation, VB1, and VB2 are 6.837, 11.21, and 11.23, respectively. Hence, there is some gain in predictive performance for the Laplace approximation relative to the VB approaches in a high dimensional logistic regression setting.

5.5 Daily Fecundability Data Analysis

5.5.1 Description of Data and Scientific Problem

I now apply SSS with the three different approximation methods to build a predictive model for the probability of conception in a menstrual cycle based on daily records of intercourse timing. Data were drawn from the European Study of Daily Fecundability (ESDF), which followed women using the sympto-thermal method of natural family planning, collecting daily data on intercourse and basal body temperature. I focus on intercourse data in a 19 day window indexed relative to the last day of hypothermia, which is a commonly used marker of the ovulation day that can be obtained based on the basal body temperature charts. The 19 day window started 12 days prior to the marker of ovulation and ended 6 days after, which means that ovulation corresponds to day 13. Intercourse data consisted of a 0/1 indicator of intercourse for each day in each menstrual cycle under study.

Scarpa and Dunson (2007) used Bayesian variable selection combined with a decision-theoretic analysis to identify optimal rules for timing intercourse to achieve conception. Their analysis focused on simple rules based on the timing in the cycle, allowing for an interaction with timing and the effect of the mucus score on the probability of conception. However, given the use of typical SSVS methods in their analysis, it was not computationally possible to consider models that allow for interactions. In particular, although it is typically assumed that sperm introduced on different days commingle and then compete independently in attempting to fertilize the ovum, it is quite possible biologically that the independent competing risks assumption is not fully accurate. However, in allowing interactions between the effects of intercourse on different days, one obtains an enormous number of possible regression models.

The European data base is particular suited to exploring different models, because it is very large compared to typical prospective studies of fecundability and collected quite detailed records. Data were available for 2832 cycles from 660 different couples. Previous

analyses of day-specific conception probabilities have focused on simple, biologically-based competing risk models, which require conception probabilities to be zero if no intercourse is reported within the potentially fertile interval of the cycle (refer, for example, to Dunson and Stanford (2005)). Here, I instead focus on a logistic regression model and avoid the assumption that no reported intercourse implies zero conception probabilities, motivated by the fact that intercourse will be unreported some of the time. For example, most studies of this type have at least a few "immaculate conceptions" in which conception occurs in cycles with no reported intercourse.

5.5.2 High-dimensional Logistic Regression

I build a standard logistic regression model to study the relationship between intercourse and women's conception time. The response variable Y is set as the binary conception variable, and $\boldsymbol{X}$ consists of the 19-day intercourse variable, and all the second-order interactions between them. As a result, in total there are 190 potential predictors of Y in my logistic regression model. The prior of $\boldsymbol{\beta}_{\boldsymbol{\gamma}}$ is $N(\mathbf{0}, \boldsymbol{I}_{\boldsymbol{\gamma}})$, which corresponded to a ridge regression shrinkage prior that expressed my view that the coefficients for the included predictors should have a low probability of falling outside of the interval within ± 2 of 0. To set the sparsity prior for model selection, I assume apriori that on average there are 10 predictors in the model ($\phi = 10/190$), and the annealing parameter for SSS is 1. For all the 2832 observations, I run SSS in parallel for 10,000 iterations using 50 CPU cores, and record the top models by the three marginal likelihood approximation methods. The top 5 models found by each method are listed in Table 5.1. We can also notice in the table that Laplace approximation is the fastest of the three approximation methods.

Dunson *et al.* (1999) reported that the probability of conception is near zero unless intercourse occurs in a six day fertile interval ending on the day of ovulation. This result is consistent with days 8-13 being included in the model for the conception probability. To my knowledge, all previous analyses of day-specific conception probabilities have assumed

Method	Top 5 Models	Score	Running Time (s)
Laplace	$-2.70 + 0.66D_9 + 1.27D_{10} + 1.42D_{11} + 1.00D_{12}$ $-0.80D_{(9,10)} - 0.78D_{(10,11)} - 1.11D_{(11,12)}$	-1147.41	2140
	$-2.67 + 0.41D_9 + 1.02D_{10} + 1.47D_{11} + 1.04D_{12}$ $+0.59D_{(6,8)} - 0.84D_{(10,11)} - 1.21D_{(11,12)}$	-1147.59	
	$-2.71 + 0.66D_9 + 1.24D_{10} + 1.39D_{11} + 1.00D_{12}$ $+0.58D_{(8,13)} - 0.81D_{(9,10)} - 0.79D_{(10,11)} - 1.12D_{(11,12)}$	-1147.65	
	$-2.72 + 0.66D_9 + 1.25D_{10} + 1.40D_{11} + 0.99D_{12}$ $+0.65D_{(3,14)} - 0.82D_{(9,10)} - 0.78D_{(10,11)} - 1.10D_{(11,12)}$	-1147.75	
	$-2.73 + 0.89D_9 + 1.23D_{10} + 1.37D_{11} + 1.19D_{12}$ $-0.75D_{(9,10)} - 0.66D_{(9,12)} - 0.75D_{(10,11)} - 1.05D_{(11,12)}$	-1147.97	
VB1	$-2.33 + 0.88D_{10} + 1.25D_{11} + 0.93D_{12}$ $+0.61D_{(3,9)} - 0.71D_{(10,11)} - 1.08D_{(11,12)}$	-1153.32	9304
	$-2.40 + 0.52D_9 + 1.03D_{10} + 1.18D_{11} + 0.87D_{12}$ $-0.69D_{(9,10)} - 0.63D_{(10,11)} - 0.99D_{(11,12)}$	-1153.55	
	$-2.35 + 0.86D_{10} + 1.22D_{11} + 0.92D_{12} + 0.59D_{(3,9)}$ $+0.53D_{(8,13)} - 0.72D_{(10,11)} - 1.08D_{(11,12)}$	-1153.79	
	$-2.36 + 0.54D_9 + 0.87D_{10} + 1.02D_{11} + 0.89D_{12}$ $-0.77D_{(9,10)} - 1.07D_{(11,12)}$	-1153.82	
	$-2.43 + 0.74D_9 + 1.01D_{10} + 1.13D_{11} + 1.04D_{12}$ $-0.63D_{(9,10)} - 0.65D_{(9,12)} - 0.60D_{(10,11)} - 0.93D_{(11,12)}$	-1153.86	
VB2	$-2.70 + 0.66D_9 + 1.27D_{10} + 1.42D_{11} + 1.00D_{12}$ $-0.81D_{(9,10)} - 0.79D_{(10,11)} - 1.11D_{(11,12)}$	-1148.84	72020
	$-2.73 + 0.66D_9 + 1.22D_{10} + 1.42D_{11} + 1.01D_{12}$ $+0.57D_{(6,8)} - 0.79D_{(9,10)} - 0.74D_{(10,11)} - 1.16D_{(11,12)}$	-1149.17	
	$-2.69 + 0.67D_9 + 1.03D_{10} + 1.41D_{11} + 1.23D_{12}$ $-0.71D_{(9,12)} - 0.83D_{(10,11)} - 1.09D_{(11,12)}$	-1149.19	
	$-2.60 + 1.06D_{10} + 1.51D_{11} + 1.08D_{12} + 0.70D_{(3,9)}$ $-0.88D_{(10,11)} - 1.22D_{(11,12)}$	-1149.20	
	$-2.67 + 0.70D_9 + 1.00D_{10} + 1.20D_{11} + 1.03D_{12}$ $+0.62D_{(6,8)} - 0.89D_{(9,10)} - 1.25D_{(11,12)}$	-1149.28	

Table 5.1: The top 5 models of the daily fecundability data obtained by SSS and the three marginal likelihood approximation methods.

Variables	Laplace	VB1	VB2
9	0.83	0.53	0.69
10	1.00	1.00	1.00
11	1.00	1.00	1.00
12	1.00	1.00	1.00
13	0.18	0.12	0.15
(3 9)	0.12	0.21	0.15
(3 14)	0.27	0.22	0.15
(6 8)	0.28	0.22	0.30
(6 14)	0.06	0.05	0.13
(8 13)	0.22	0.15	0.18
(9 10)	0.71	0.41	0.47
(9 12)	0.28	0.27	0.33
(10 11)	0.78	0.62	0.77
(10 13)	0.14	0.11	0.09
(11 12)	1.00	1.00	1.00

Table 5.2: Marginal inclusion probabilities of the key variables, obtained by the scores of the top 50 models.

independent competing risks, which does not allow interactions between intercourse acts occurring on different days. Interestingly, my results suggest that a narrower four day fertile interval ending one day prior to the estimate day of ovulation is appropriate (days 9, 10, 11 and 12), but with interactions of intercourse occurring between days (9,10), (10,11), and (11,12). As shown in Table 5.1, these interactions are negative so that the log odds of conception increases substantially more with the first intercourse act occurring on a highly fecund day, and subsequent acts on high fertility days have less of an impact. The size of the model does depend somewhat on the chosen sparsity prior, and if I choose a prior favoring larger model sizes I tend to include days 8 and 13 along with several additional interaction terms. Such a model may not be sparse enough for accurate predictions. Note also from Table 5.2 that the three interactions all have high marginal inclusion probabilities. Note that several of the top models also select other interactions, such as (6,8) and (8,13), with the main effects not included in the model. This is biologically reasonable, since one may require more than one intercourse act to obtain a sizeable conception probability if intercourse only occurs on the edge of the fertile interval.

To assess the model selection performance of the three methods, I use cross-validation to compare the three methods. First, 10% of the data, which mean 283 observations, are randomly picked out to form the test data. For the other 2549 observations, I try different sample size of the training data, i.e., randomly select 1416, 1699, 1982, 2266 and 2549 observations (50% - 90% of the total 2832 cycles) to form 5 individual training data sets. To compare different approximation methods for different sizes of the training data sets, I first use SSS to do the model selection, and then use Bayesian model averaging on the top 50 models to obtain predictive probabilities of conception for the test data. The other settings such as priors are the same as the analysis for the full data.

In this cross-validation analysis, it is not suitable to use 0.5 threshold to make prediction on Y based on the predictive probabilities, because the percentage of 1's in the population is so low (about 0.153), otherwise it may give a prediction set with all 0's. Instead, I compare

Method (True Value)	50%	60%	70%	80%	90%
Laplace (Y=1)	0.1987	0.2047	0.2091	0.2108	0.2025
VB1 (Y=1)	0.2023	0.2101	0.2141	0.2168	0.2105
VB2 (Y=1)	0.1977	0.2047	0.2083	0.2104	0.2021
Laplace (Y=0)	0.1347	0.1348	0.1399	0.1415	0.1451
VB1 (Y=0)	0.1465	0.1459	0.1520	0.1541	0.1577
VB2 (Y=0)	0.1337	0.1346	0.1397	0.1413	0.1448

Table 5.3: The mean predictive probabilities of conception for true values of response that are equal to 1 or 0 in the test data respectively. Different training sample sizes are tried (50% - 90%).

the mean predictive probabilities of conception given the true value of Y in the test data for different sizes of the training data, and different approximation methods (Table 5.3).

From Table 5.3, the VB approximations have similar performance to Laplace in terms of prediction. This is because VB1 and VB2 order the models almost correctly, but just under-estimate the marginal likelihoods for the better models, flattening out the posterior probabilities across the better models. This may have a modest impact on predictive performance that may not show up compellingly in the cross validation exercise, but clearly in the previous simulation analysis. Laplace is closer to VB2 than VB1 here, which is also shown by marginal likelihood approximations in simulation studies. Furthermore, it is quite intriguing that Laplace and VB2 usually gives lower mean predictive probabilities of conception than VB1, and the predictive probabilities using training data with sizes from 50% to 80% keep increasing in each row. Note that the low prediction accuracy shown in Table 5.3 is not at all surprising due to the very high degree of unexplained heterogeneity that is characteristic of fecundability data.

5.6 Conclusion

This chapter's work was originally motivated by the goal of using variational Bayes approximations to improve methods for high-dimensional model selection and averaging. The VB approaches have been increasingly widely used in machine learning applications, and have conceptual appeal in resulting from maximization of a formal lower bound on the marginal likelihood. Although the tightness of the lower bound is in general quite difficult to assess theoretically, the good performance of VB procedures in various predictive settings has been reassuring. However, to my knowledge, the performance of VB relative to traditional Laplace methods of estimating the marginal likelihood with the goal of model selection has not been assessed.

In the setting of logistic regression model selection, this chapter uses simulations to compare the accuracy of Laplace and two types of VB approximations (VB1, VB2). I find that Laplace is highly accurate, while VB1 and VB2 have a disturbing tendency to under-estimate the marginal likelihood for high posterior probability models. When the goal is model selection or accurate estimate of posterior model probabilities, this type of under-estimation is particularly troubling, since one will under-estimate the probabilities for good models and over-estimate the probabilities for bad models. On the positive side, the VB approaches do tend to rank the models appropriately; it is only the scores that are misestimated. Perhaps for this reason, I have observed only a slight decrease in predictive performance for the VB approaches relative to Laplace in settings with few important predictors and a high-dimensional set of candidates.

The performance of the VB approach is critically dependent on the accuracy of the product factorization of the joint posterior. This gives us a clue as to the reason for my results. In particular, it is my expectation in the variable selection setting that the product factorization provides a good approximation for bad models, since bad models correspond to exclusion of important predictors and inclusion of predictors that actually have no impact. However, I expect that the approximation breaks down when several important predictors

are included and the coefficients for these predictors are correlated *a posteriori*, as one would typically expect in regression models. Because the implementation of VB methods is so tied to the product factorization, it is difficult to entirely eliminate this problem. However, one strategy is to attempt a factorization under a parameterization chosen to reduce posterior dependence.

Chapter 6

Kalman Filtering for Multi-level Hierarchical Models

Multi-level hierarchical models provide an attractive framework for estimation of the density of a response variable aggregated at multiple resolutions. These models provide improved bias-variance tradeoff by smoothing high-variance estimates at finer resolutions using data at coarser resolutions. Although such models are popular in statistics, model fitting presents a formidable challenge, especially for data sets aggregated into large hierarchies. I address this by a novel model fitting algorithm based on a multi-scale Kalman filter that is both scalable and easy to implement. The method is illustrated through simulation studies and analysis of real world data sets in health care and online advertising.

6.1 Introduction

In many real-world applications, one is interested in estimating the probability density of a response variable that is naturally grouped at multiple resolutions through a hierarchy. One key goal is to make inference at the finest resolution, i.e., the leaf nodes, a large fraction of which have small sample sizes and hence would lead to high-variance estimates if analyzed separately. Bayesian modeling that links parameters across nodes and naturally induces shrinkage estimation, exploits the natural grouping of data and is a natural method of choice. Such models are sometimes referred to as multi-level hierarchical models in the statistics literature (Gelman and Hill, 2007). However, existing methods to fit these models are not scalable to large hierarchies. In this chapter, I provide an algorithm based on a multi-scale Kalman filter that is simple to implement and scalable.

According to Stein (1956) and James and Stein (1961), the central idea of shrinkage is to smooth unreliable estimates by "borrowing strength" across estimates. In a Bayesian framework, this is accomplished by assuming node-specific parameters follow a prior across nodes, defining them as random effects. When data is organized as a hierarchy, it is natural to assume a multi-level prior wherein parameters of children nodes are assumed to be drawn from a distribution centered around the parameter of the parent. This bottom-up, recursive modeling assumption provides a posterior whose estimates at the finest resolution are smoothed using data on the lineage path of the node in the hierarchy (Lindley, 1971, 1972; Lindley and Smith, 1972). The fundamental assumption is that the hierarchy, determined from domain knowledge, provides a natural clustering to account for latent processes generating the data which, when incorporated into the model, can improve predictions. Since the pioneering work of Laird and Ware (1982), these models have been widely and successfully used in applications in biology, epidemiology, the social sciences and elsewhere (see, for example, Gelman and Hill (2007)). In machine learning and data mining, these models have been used sporadically but are increasingly receiving more attention (see, for example, Dudik *et al.* (2007); Agarwal *et al.* (2007) and references therein).

Although multi-level hierarchical (MLH hereafter) models are intuitive, parameter estimation presents a formidable challenge, especially for applications such as online advertising and web search where data is often grouped into large hierarchies constructed and carefully tweaked by humans. For Gaussian responses, the computational bottleneck is the inversion of a dense covariance matrix whose order depends on the number of nodes, and this is expensive for large problems. For non-Gaussian responses (e.g binary data), the non-quadratic nature of the log-likelihood adds an additional step of approximating an integral whose dimension depends on the number of nodes in the hierarchy. This is a very active area of research in statistics with several approximation approaches being proposed such as Pinheiro and Bates (2000), Breslow and Clayton (1993), and Rodriguez and Goldman (2001). For matrix inversion, techniques based on sparse Cholesky factorization of the covariance

matrix have been recently proposed in Pinheiro and Bates (2000). For non-Gaussian models, solutions to the integral problem based on high order Taylor series approximations have been shown to provide accurate results (Raudenbush *et al.*, 2000). However, these techniques involve algebra that is often non-intuitive and difficult to implement. A more natural computational scheme that exploits the structure of the model is based on Gibbs sampling (Gelfand and Smith, 1990); this is, however, often slow to converge, especially in large problems. In this chapter, I provide a scalable fitting procedure based on multi-scale Kalman filtering that directly exploits the hierarchical structure of the problem. Other than scalability, my fitting procedure is easy to understand and implement.

In Section 6.2, I describe and illustrate the Gaussian MLH and the new fitting algorithm; the model is then applied to a simulated data set. In Section 6.3, I introduce the non-Gaussian MLH and describe approximations and bias correction techniques along with the analysis of a real epidemiological data set. In Section 6.4, I illustrate the technique further in analysis of an online advertising application. I conclude with a discussion in Section 6.5.

6.2 MLH for Gaussian Responses

Assume we have a hierarchy T consisting of L levels (root is level 0), and for which $m_j, j = 0, \cdots, L$, denotes the number of nodes at level j. Denote the set of nodes of level j in the hierarchy T as T_j. For node r in T, denote the parent of r as $pa(r)$, and the ith child of the node r as $c_i(r)$. If some node r' is a descendent of r, we say $r' \prec r$. Since the hierarchy has L levels, T_L denotes the set of leaves in the hierarchy. Let $y_{ir}, i = 1, \cdots, n_r$ denote the i^{th} observation at *leaf* node r, and $\boldsymbol{x}_{ir}$ denote the p-dimensional vector of known covariates (features) associated with y_{ir}. Note that throughout this chapter, for simplicity, all the observations are assumed to be at the leaf nodes (level L). A more general case where each node in the hierarchy contains observations can be easily obtained using my algorithm.

Notation	Interpretation
T_j	Level j of the hierarchy T
m_j	The number of nodes at level j in T
q	The total number of nodes in T
$pa(r)$	The parent node of node r in T
$c_i(r)$	The ith child node of node r in T
n_r	The number of observations at leaf node r
y_{ir}	The ith observation (response) at *leaf* node r
$\boldsymbol{Y}$	$\{y_{ir}, i = 1, \cdots, n_r, r \in T\}$
$\boldsymbol{x}_{ir}$	The ith observation (p-dimensional covariates) at *leaf* node r
$\boldsymbol{X}$	$\{\boldsymbol{x}_{ir}, i = 1, \cdots, n_r, r \in T\}$
$\boldsymbol{\beta}$	The regression parameter vector associated with $\boldsymbol{X}$
ϕ_r^j	The random effect parameter at node r at level j
$\boldsymbol{\phi}$	$\{\phi_r^j, r \in T, j = 1, \cdots, L\}$
V	The residual variance of y_{ir}, if y_{ir} has a Gaussian model
γ_j	The variance of ϕ_r^j for all the nodes at level j
$\boldsymbol{\gamma}$	$\{\gamma_1, \cdots, \gamma_L\}$
$\phi_{r\|r}^j$	The mean of $\phi_r^j \|\{y_{ir'}, i = 1, \cdots, n_{r'}, \forall r' \prec r\}$
$\sigma_{r\|r}^j$	The variance of $\phi_r^j \|\{y_{ir'}, i = 1, \cdots, n_{r'}, \forall r' \prec r\}$
$\hat{\phi_r^j}$	The mean of $\phi_r^j \|\{y_{ir'}, i = 1, \cdots, n_{r'}, \forall r' \in T_L\}$
σ_r^j	The variance of $\phi_r^j \|\{y_{ir'}, i = 1, \cdots, n_{r'}, \forall r' \in T_L\}$

Table 6.1: A list of the key notations.

Consider the Gaussian MLH defined by,

$$y_{ir}|\phi_r^L \sim N(\boldsymbol{x}_{ir}^{'}\boldsymbol{\beta} + \phi_r^L, V), \tag{6.1}$$

where $\boldsymbol{\beta}$ is a regression parameter vector, and ϕ_r^j, parameter at node r and level j, is a random effect with joint distribution defined implicitly by the set of hierarchical conditional distributions $p(\phi_r^j|\phi_{pa(r)}^{j-1})$ across the levels of the hierarchy, where $\phi_0^0 = 0$, and $pa(r)$ denotes the parent of node r at level $j-1$. The form of $p(\phi_r^j|\phi_{pa(r)}^{j-1})$ is assumed to be,

$$\phi_r^j|\phi_{pa(r)}^{j-1} \sim N(\phi_{pa(r)}^{j-1}, \gamma_j); j = 1, \cdots, L, \tag{6.2}$$

where $\boldsymbol{\gamma} = (\gamma_1, \cdots, \gamma_L)$ is a vector of level-specific variance components that control the amount of smoothing. This specification of MLH is referred to as the *centered* parametrization and is often used in a fully Bayesian framework (Sahu and Gelfand, 1999).

An equivalent formulation of MLH in the literature is obtained by attaching independent random variables $b_r^j \sim N(0, \gamma_j)$ to the nodes and replacing ϕ_r^L in (6.1) by the sum of the b_r^j parameters along the lineage path from root to leaf node in the hierarchy. We denote this compactly as $\boldsymbol{z}_r'\boldsymbol{b}$, where $\boldsymbol{b}$ is a vector of b_r^j for all the nodes in the hierarchy, and $\boldsymbol{z}_r$ is a vector of 0/1's turned on for nodes in the path of node r. More compactly, let $\boldsymbol{y} = \{y_{ir}, i = 1, \cdots, n_r, r \in T\}$, and $\boldsymbol{X}$ as well as $\boldsymbol{Z}$ be the corresponding matrix of vectors $\boldsymbol{x}_{ir}$ and $\boldsymbol{z}_r$ for $i = 1, \cdots n_r$ and $r \in T$, then $\boldsymbol{y} \sim N(\boldsymbol{X}^{'}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{b}, V\boldsymbol{I})$ with $\boldsymbol{b} \sim N(\boldsymbol{0}, \Omega(\boldsymbol{\gamma}))$. The problem is to estimate the unknown parameters $(\boldsymbol{\beta}_{p\times 1}, \boldsymbol{b}_{q\times 1}, \boldsymbol{\gamma}_{L\times 1})$ where $q = \sum_{j=1}^{L} m_j$. The likelihood $L(\boldsymbol{\beta}, \boldsymbol{\Omega}, V|\boldsymbol{y})$ can be written as

$$L(\boldsymbol{\beta}, \boldsymbol{\Omega}, V|\boldsymbol{y}) = \int p(\boldsymbol{y}|\boldsymbol{\beta}, \boldsymbol{b}, V)p(\boldsymbol{b}|\boldsymbol{\Omega})d\boldsymbol{b}. \tag{6.3}$$

For Gaussian responses, equation (6.3) can be integrated out as a closed form. The main computational bottleneck in obtaining the MLE is computing the Cholesky factor of a $q \times q$ matrix $(\boldsymbol{Z}^{'}\boldsymbol{Z} + \boldsymbol{\Omega}^{-1})$, which is computationally expensive for large values of

q (Pinheiro and Bates, 2000). Existing state-of-the-art methods are based on the sparse Cholesky factorization. I provide a more direct way to solve this problem using a Kalman filter whose complexity is *linear* in q and *cubic* in the fan-outs at the nodes of the tree. I also note that my methods apply if the random effects are vectors and enter into equation (6.2) as a linear combination of some covariate vector. In this paper, I illustrate through a scalar.

6.2.1 Model Fitting

Throughout, I work with the centered parametrization introduced above. The heart of the fitting algorithm is computing the conditional posterior distribution of $\boldsymbol{\phi} = \{\phi_r^j, r \in T, j = 1, \cdots, L\}$ given $(\boldsymbol{\beta}, V, \boldsymbol{\gamma})$ and $p(\phi_r^j | \phi_{pa(r)}^{j-1})$ in equation (6.2) as the prior. Since the parameters V and $\boldsymbol{\gamma}$ are unknown, one can put hyper-priors on those parameters and use a fully Bayesian approach with MCMC sampling methods to obtain posterior samples. However, the problems of interest in this chapter have massive scales, and the hierarchy often contains a number of nodes. For this reason, the MCMC sampling methods are computationally very expensive, and can hardly be used with large scale data sets. By treating $\boldsymbol{\phi}$ as missing data, we may choose to obtain the maximum likelihood estimates of the unknown parameters V and $\boldsymbol{\gamma}$ using the EM algorithm (Dempster *et al.*, 1977), which is much faster without losing substantial precision. My algorithm can also be applied to a Bayesian framework; e.g., with a vague prior on V ($\pi(V) \propto 1/V$) and a mild quadratic prior on γ_j ($\pi(\gamma_j|V) \propto V/(V+\gamma_j)^2$). In that case, the EM algorithm finds the posterior modes of V and $\boldsymbol{\gamma}$ despite that, for massive data sets, the influence of the priors are quite minor and the posterior modes are approximately equal to the maximum likelihood estimates. Note that the model fitting algorithm using EM described in the following section, has to be run for multiple iterations until convergence.

I now describe the Kalman filter, which is used in the inner loop of the EM algorithm, for computing the conditional posterior of $\boldsymbol{\phi}$. As in temporal state space models, the Kalman

filter consists of two steps - *Filtering*: where one propagates information from leaves to the root and *Smoothing*: where information is propagated from root all the way down to the leaves.

Filtering:

Denote the current estimates of the parameters $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ and V as $\hat{\boldsymbol{\beta}}$, $\hat{\boldsymbol{\gamma}}$, and $\hat{V}$. Define $e_{ir} = y_{ir} - \boldsymbol{x}_{ir}^{'}\hat{\boldsymbol{\beta}}$ to be the residuals, and also, $Var(\phi_r^j) = \Sigma_j = \sum_{i=1}^{j} \hat{\gamma}_i, r \in T_j$. Denote the conditional posterior distribution $\phi_r^L|\{y_{ir}, i = 1, \cdots, n_r\} \sim N(\phi_{r|r}^L, \sigma_{r|r}^L)$, for each leaf node r. The first step is to update $\phi_{r|r}^L$ and $\sigma_{r|r}^L$ for all ϕ_r^L parameters at the leaf nodes using the standard Bayesian update formula for Gaussian models given by,

$$\phi_{r|r}^L = \frac{\Sigma_L \sum_{i=1}^{n_r} e_{ir}}{\hat{V} + n_r \Sigma_L}, \tag{6.4}$$

$$\sigma_{r|r}^L = \frac{\Sigma_L \hat{V}}{\hat{V} + n_r \Sigma_L}. \tag{6.5}$$

Next, the posteriors $\phi_r^j|\{y_{ir'}, i = 1, \cdots, n_{r'}, \forall r' \prec r\} \sim N(\phi_{r|r}^j, \sigma_{r|r}^j)$, are recursively updated from $j = L - 1$ to $j = 1$, by shrinking the parent towards each child and then combining the information. To provide intuition about the step, it is useful to invert the state equation (6.2) and express the distribution of $\phi_{pa(r)}^{j-1}$ conditional on ϕ_r^j. Note that,

$$\phi_{pa(r)}^{j-1} = E(\phi_{pa(r)}^{j-1}|\phi_r^j) + (\phi_{pa(r)}^{j-1} - E(\phi_{pa(r)}^{j-1}|\phi_r^j)). \tag{6.6}$$

Simple algebra gives the conditional expectation and variance of $\phi_{pa(r)}^{j-1}|\phi_r^j$. Hence,

$$\phi_{pa(r)}^{j-1} = B_j \phi_r^j + \psi_r^j, \tag{6.7}$$

where $B_j = \sum_{i=1}^{j-1} \hat{\gamma}_i / \sum_{i=1}^{j} \hat{\gamma}_i$, is the correlation between any two siblings at level j and $\psi_r^j \sim N(0, B_j \hat{\gamma}_j)$. The update steps are given below.

First, a new prior is obtained for the parent node based on the current estimate of each child by plugging-in the current estimates of a child into equation (6.7). For the ith child of node r,

$$\phi^{j-1}_{r|c_i(r)} = B_j \phi^{j}_{c_i(r)|c_i(r)}, \tag{6.8}$$

$$\sigma^{j-1}_{r|c_i(r)} = B_j^2 \sigma^{j}_{c_i(r)|c_i(r)} + B_j \hat{\gamma}_j. \tag{6.9}$$

Here I assume that r is at level $j-1$, and $c_i(r)$ is at level j. Next, we combine information obtained about the parent from all its children.

$$\phi^{j-1}_{r|r} = \sigma^{j-1}_{r|r} \sum_{i=1}^{k_r} (\phi^{j-1}_{r|c_i(r)} / \sigma^{j-1}_{r|c_i(r)}), \tag{6.10}$$

$$1/\sigma^{j-1}_{r|r} = \Sigma_j^{-1} + \sum_{i=1}^{k_r} ((1/\sigma^{j-1}_{r|c_i(r)}) - \Sigma_j^{-1}), \tag{6.11}$$

where k_r is the number of children of node r at level $j-1$.

Smoothing:

In the smoothing step, parents propagate information recursively from root to the leaves to provide us with the posterior of each ϕ^j_r based on the entire data. Denoting the posterior mean and variance of ϕ^j_r given all the observations by $\hat{\phi^j_r}$ and σ^j_r respectively, the update equations are given below.

For level 1 nodes, set $\hat{\phi^1_r} = \phi^1_{r|r}$, and $\sigma^1_r = \sigma^1_{r|r}$.

For node r at other levels,

$$\hat{\phi^j_r} = \phi^j_{r|r} + \sigma^j_{r|r} B_j (\hat{\phi^{j-1}_{pa(r)}} - \phi^{j-1}_{pa(r)|r}) / \sigma^j_{pa(r)|r}, \tag{6.12}$$

$$\sigma^j_r = \sigma^j_{r|r} + \sigma^j_{r|r} B_j^2 (\sigma^{j-1}_{pa(r)} - \sigma^{j-1}_{pa(r)|r}) / \sigma^{j^2}_{pa(r)|r}, \tag{6.13}$$

and set,

$$\sigma^{j,j-1}_{r,pa(r)} = \sigma^j_{r|r} B_j \sigma^{j-1}_{pa(r)} / \sigma^{j-1}_{pa(r)|r}. \tag{6.14}$$

For a more detailed derivation of these expressions, see Huang and Cressie (2000). The algorithm outlined above is called the multi-scale Kalman filter (Chou *et al.*, 1994). The computational complexity of the algorithm is linear in the number of nodes in the hierarchy and for each parent node, I perform an operation which is cubic in the number of children. Hence, for most real life hierarchies, the complexity is essentially linear in the number of nodes.

Expectation Maximization:

To estimate all parameters simultaneously, I take recourse to the EM algorithm (Dempster *et al.*, 1977) which assumes the $\boldsymbol{\phi}$ parameters to be the missing latent variables. The expectation step consists of computing the expected value of complete log-likelihood with respect to the conditional distribution of missing data $\boldsymbol{\phi}$ obtained using the multi-scale Kalman filter algorithm. The maximization step obtains revised estimates of other parameters by maximizing the expected complete log-likelihood. The expressions are,

$$\hat{V} = \sum_{r \in T_L} \frac{\sum_{i=1}^{n_r} (e_{ir} - \hat{\phi}_r^L)^2 + n_r \sigma_r^L}{\sum_{r \in T_L} n_r}. \tag{6.15}$$

For $j = 1, \cdots, L$,

$$\hat{\gamma_j} = \frac{\sum_{r \in T_j} (\sigma_r^j + \sigma_{pa(r)}^{j-1} - 2\sigma_{r,pa(r)}^{j,j-1} + (\hat{\phi_r}^j - \hat{\phi_{pa(r)}}^{j-1})^2)}{|m_j|}. \tag{6.16}$$

Updating $\hat{\boldsymbol{\beta}}$:

I use the posterior mean of $\boldsymbol{\phi}$ obtained from the Kalman filtering step, to compute the ordinary least square estimate of $\boldsymbol{\beta}$ in equation (6.17). $\boldsymbol{\beta}$ can also be considered to have a

non-informative uniform prior on the real space (although non-proper) and equation (6.17) represents the posterior mean of $\boldsymbol{\beta}$. Specifically,

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'(\boldsymbol{Y} - \hat{\boldsymbol{\phi}}^L), \tag{6.17}$$

where $\hat{\boldsymbol{\phi}}^L$ is the vector of $\hat{\phi_r^L}$ corresponding to each observation y_{ir} at different leaf node r.

6.2.2 Simulation Performance

I provide a check of my model fitting algorithm through a simulation study where the structure of the hierarchy is derived from a real data set in Rodriguez and Goldman (1995, 2001). I describe the actual epidemiological data set (which will be analyzed in Section 6.3) and the simulated one I created to test my fitting procedure on the Gaussian model. The data focus on 2449 Guatemalan children who belong to 1558 families who, in turn, live in 161 communities. The response variable of interest is binary with a positive label assigned to a child if he/she received a full set of immunizations. The actual data contains 15 covariates capturing individual, family, and community level characteristics as shown in Table 6.3. For my simulation study, I consider only three covariates, with the coefficient vector $\boldsymbol{\beta}$ set with entries all equal to 1. I simulated Gaussian response as follows: $y_{ir}|\boldsymbol{b} \sim N(\boldsymbol{x}_{ir}^{'}\boldsymbol{\beta} + b_r^1 + b_r^2, 10)$ where $b_r^1 \sim N(0, 4)$, and $b_r^2 \sim N(0, 1)$. I simulated 100 data sets and compared the estimates from the Kalman filter to the one obtained from standard routine *lme4* in the statistical software ***R***. Each point in Figure 6.1, 6.2, and 6.3 represents one simulated data set, the x-axis represents the parameter estimates from my model fitting algorithm by Kalman filtering, and the y-axis represents the estimates from "lmer." Each line in the plots is a 45° line that passes through the origin. The more similar the parameter estimates for one data set by the two methods are, the closer the corresponding point in each plot is to the line. Results in Figure 6.1, 6.2, and 6.3 clearly show my method provides estimates that are close to the one obtained from *lme4* for the Gaussian model. The EM method converged rapidly and required at most 30 iterations.

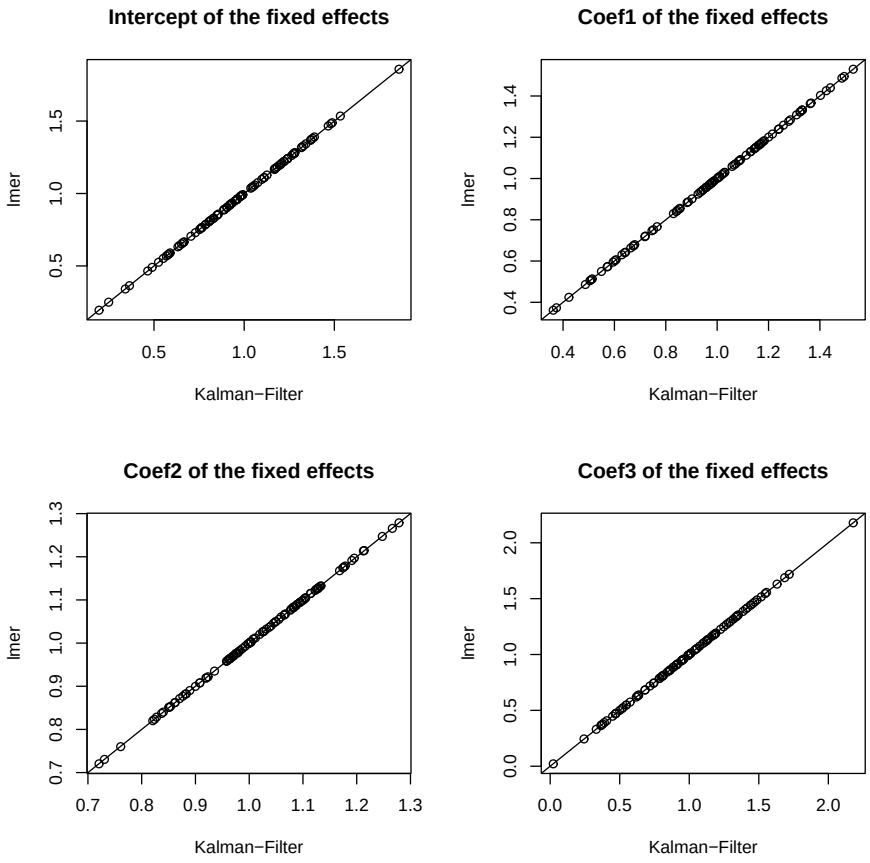


Figure 6.1: Comparison of $\boldsymbol{\beta}$ estimates in Gaussian MLH model using Kalman filter and "lmer" for 100 simulated data sets.

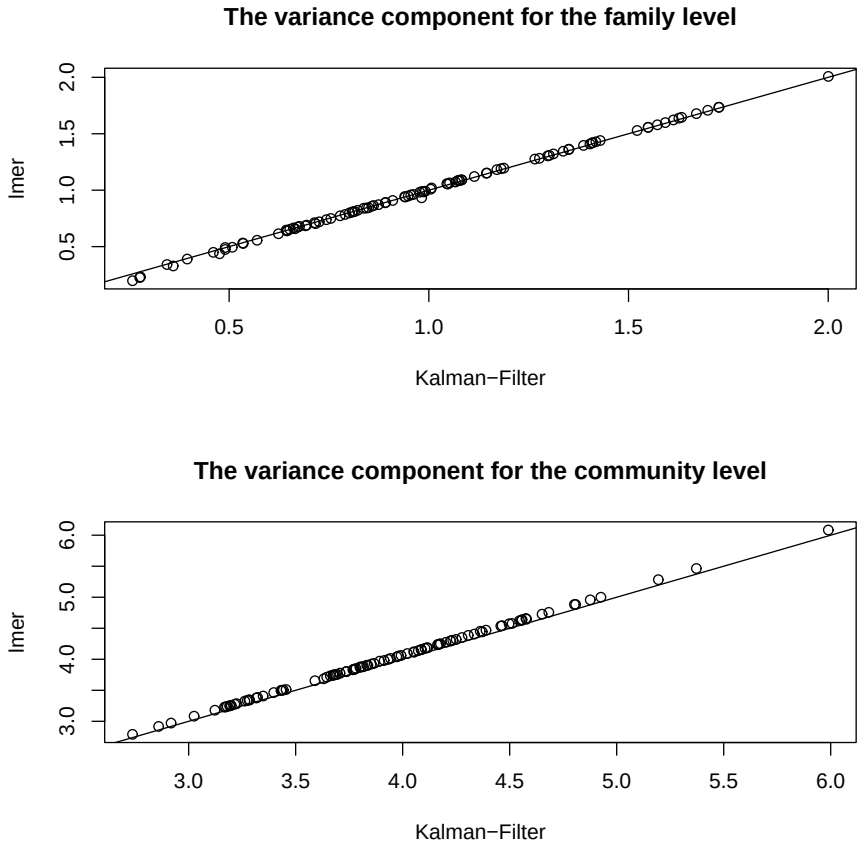


Figure 6.2: Comparison of the variance components in Gaussian MLH model using Kalman filter and "lmer" for 100 simulated data sets.

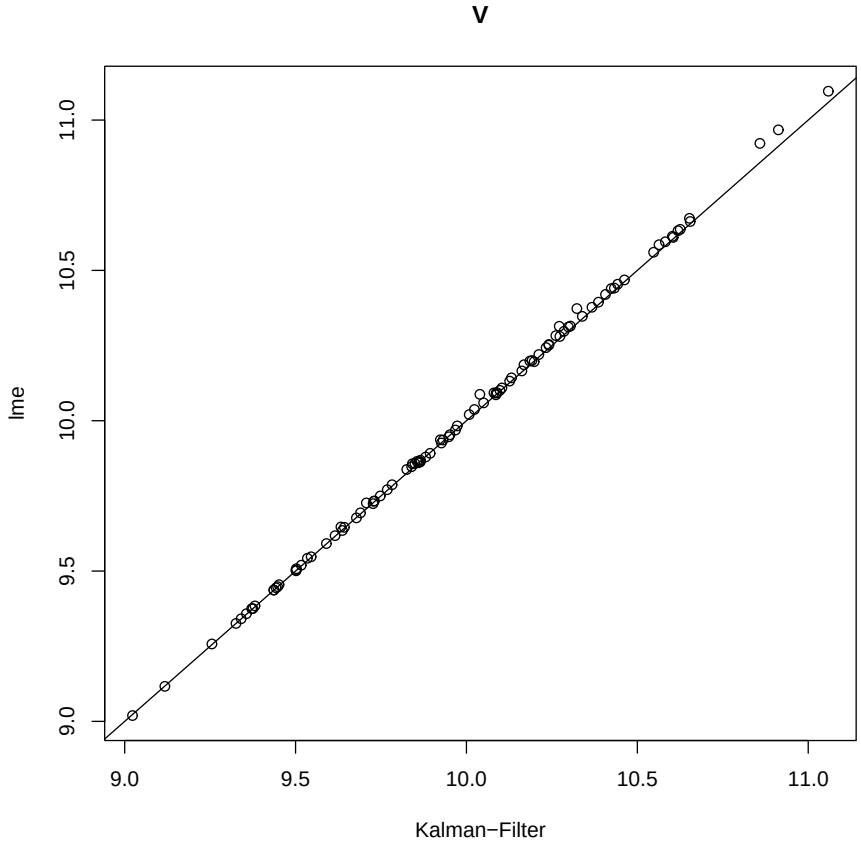


Figure 6.3: Comparison of V estimate in Gaussian MLH model using Kalman filter and "lmer" for 100 simulated data sets.

6.3 MLH for Non-Gaussian Responses

When the assumption of a Gaussian distribution for the response variable is not appropriate (e.g., in the epidemiological data the response variable is binary), generalized linear models (McCullagh and Nelder, 1989) provides a natural modeling framework to build MLH. For the sake of illustration, I only consider the case of binary response in this chapter but other distributions like Poisson, Gamma can be easily handled.

Let $y_{ir} \sim Bernoulli(p_{ir})$, i.e. $P(y_{ir}) = p_{ir}^{y_{ir}}(1-p_{ir})^{1-y_{ir}}$. Let $\theta_{ir} = log\frac{p_{ir}}{1-p_{ir}}$ be the log-odds. The MLH logistic regression is defined as:

$$\theta_{ir} = \boldsymbol{x}_{ir}^{'}\boldsymbol{\beta} + \phi_r^L, \tag{6.18}$$

with the same multi-level prior as described in equation (6.2). The normal multi-level prior on the $\boldsymbol{\phi}$ parameters with a logistic likelihood does not provide a closed form Gaussian conditional posterior for the $\boldsymbol{\phi}$ parameters as in the case of a Gaussian likelihood, this makes the computation more difficult. Several approximations have been suggested in the literature. These include quadratic approximations of the complete data likelihood (joint likelihood of $\boldsymbol{y}$ and $\boldsymbol{\phi}$) to enforce a Gaussian like computation. The notable ones in this category are the marginal quasi-likelihood (MQL-1, MQL-2) methods of Goldstein (1991), penalized quasi-likelihood (PQL-1) method of Breslow and Clayton (1993), an enhancement of PQL called PQL-2 method of Goldstein (1991). As in the Gaussian case, inversion of a large $q \times q$ matrix is computationally expensive for such approximations. Rodriguez and Goldman (2001) compared the performance of MQL-1, MQL-2, PQL-1, and PQL-2 with more accurate but slow procedures based on adaptive quadrature and Gibbs sampling. They demonstrated that methods based on simple quadratic approximations may lead to bias in the estimates. A bootstrap procedure to correct for the bias was also described in Kuk (1995). I directly refer to Table 6.2 presented in Rodriguez and Goldman (2001) that shows estimates for the binary MLH model of Guatemalan data by naive logistic regression model without random effects (column "logit" in the table), MQL, PQL, and bootstrap

corrections to PQL-1 estimates (PQL-B). They also compared the results with maximum likelihood estimates via Gauss-Hermite quadrature, as well as Bayesian estimates obtained via Gibbs Sampling using non-informative priors.

I note that the main objective of all the above mentioned approximations is statistical inference for $\boldsymbol{\beta}$, while my focus is providing a scalable methodology to estimate the predictive distribution of a response variable grouped at multiple resolutions via a hierarchy. I take recourse to Taylor series and variational Bayes (Jaakkola and Jordan, 2000) approximations coupled with the Kalman filtering algorithm to achieve scalability. Surprisingly, variational Bayes performs poorly providing estimates that are badly biased. For Taylor series approximation, the estimates are much better but the bias is still present; estimating variance components through a grid search via cross-validation works well in my empirical evaluation. For problems involving a hierarchy with large number of variance components, cross-validation may become computationally expensive; the bootstrap correction procedure may be attractive in such scenarios. I describe my approximation and the bias correction procedures from Kuk (1995) below.

6.3.1 Approximation Methods

For logistic regression, the log-likelihood of the complete data ($\boldsymbol{y}$ and $\boldsymbol{\phi}$) is not quadratic. Let $\eta_{ir} = \boldsymbol{x}_{ir}\hat{\boldsymbol{\beta}} + \hat{\phi}_r^L$, where $\hat{\boldsymbol{\beta}}$, $\hat{\phi}_r^L$ are current estimates of the parameters in our algorithm. I do a quadratic approximation of the log-likelihood through a second order Taylor expansion around η_{ir} as in Srebro and Jaakkola (2003). This enables us to do the calculations as in the Gaussian case with the response y_{ir} being replaced by Z_{ir} where

$$Z_{ir} = \eta_{ir} + \frac{2y_{ir} - 1}{g((2y_{ir} - 1)\eta_{ir})}, \tag{6.19}$$

and $g(x) = 1/(1 + \exp(-x))$. Approximately,

$$Z_{ir} \sim N(\boldsymbol{x}'_{ir}\boldsymbol{\beta} + \phi_r^L, \frac{1}{g(\eta_{ir})g(-\eta_{ir})}). \tag{6.20}$$

Effects	Logit	MQL-1	MQL-2	PQL-1	PQL-2	PQL-B	Maximum likelihood	Gibbs
Fixed effects								
Individual								
Child age $\geq$ 2 years	0.95	0.93	1.11	0.98	1.44	1.80	1.72	1.84
Mother age $\geq$ 25 years	-0.08	-0.08	-0.10	-0.09	-0.16	-0.19	-0.21	-0.26
Birth order 2-3	-0.08	-0.09	-0.11	-0.10	-0.19	-0.15	-0.26	-0.29
Birth order 4-6	0.09	0.13	0.15	0.13	0.17	0.27	0.18	0.21
Birth order $\geq$ 7	0.15	0.19	0.23	0.20	0.33	0.39	0.43	0.50
Family								
Indigenous, no Spanish	0.28	-0.04	-0.05	-0.05	-0.13	-0.06	-0.18	-0.22
Indigenous Spanish	0.22	0.01	0.01	0.00	-0.05	0.03	-0.08	-0.11
Mother's education primary	0.25	0.21	0.25	0.22	0.34	0.42	0.43	0.48
Mother's education secondary or better	0.30	0.22	0.27	0.23	0.34	0.46	0.42	0.46
Husband's education primary	0.29	0.28	0.34	0.30	0.44	0.57	0.54	0.59
Husband's education secondary or better	0.21	0.25	0.31	0.27	0.41	0.47	0.51	0.55
Husband's education missing	0.03	0.02	0.02	0.02	0.01	0.07	-0.01	0.00
Mother ever worked	0.25	0.19	0.24	0.20	0.31	0.37	0.39	0.42
Community								
Rural	-0.50	-0.47	-0.57	-0.50	-0.73	-0.93	-0.89	-0.96
Proportion indigenous, 1981	-0.78	-0.64	-0.78	-0.67	-0.95	-1.21	-1.15	-1.22
Random effects								
Standard deviations $\boldsymbol{\gamma}$								
Family	—	0.63	0.72	0.73	1.75	2.69	2.32	2.60
Community	—	0.53	0.55	0.56	0.84	1.06	1.02	1.13

Table 6.2: Estimates for the binary MLH model of complete immunization, all the results from Rodriguez and Goldman (2001).

Now denote $e_{ir} = Z_{ir} - \boldsymbol{x}'_{ir}\hat{\boldsymbol{\beta}}$, and the approximated variance of Z_{ir} as V_{ir}. Analogous to equation (6.4) and (6.5), the resulting filtering step for the leaf nodes becomes:

$$\phi^L_{r|r} = \sigma^L_{r|r} \sum_{i=1}^{n_r} \frac{e_{ir}}{V_{ir}}, \tag{6.21}$$

$$\sigma^L_{r|r} = (\frac{1}{\Sigma_L} + \sum_{i=1}^{n_r} \frac{1}{V_{ir}})^{-1}. \tag{6.22}$$

The step for estimating $\boldsymbol{\beta}$ becomes:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{W}\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{W}(\boldsymbol{Z} - \hat{\boldsymbol{\phi}^L}), \tag{6.23}$$

where $\boldsymbol{W} = diag(\frac{1}{V_{ir}})$. All the other computational steps remain the same as in the Gaussian case.

For logistic regression, one could also use the variational Bayes (VB hereafter) method described in Jaakkola and Jordan (2000). The VB approximation provides a lower bound to the predictive log-likelihood that approaches the actual one monotonically with each iteration. In general, VB tends to provide a more robust algorithm compared to quadratic approximations. In our context, the only detail that changes is the definition of Z_{ir} in equations (6.19) and (6.20). In fact, for VB the Z_{ir} are defined as follows.

$$Z_{ir} = \frac{(2y_{ir} - 1)\eta_{ir}}{tanh(\eta_{ir}/2)}, \tag{6.24}$$

then approximately,

$$Z_{ir} \sim N(\boldsymbol{x}'_{ir}\boldsymbol{\beta} + \phi^L_r, \frac{2tanh(\eta_{ir}/2)}{\eta_{ir}}). \tag{6.25}$$

Surprisingly, the VB method badly underestimates the γ's compared to the Taylor approximation. On closer scrutiny I found the VB to always provide smaller estimates of the posterior variance $\sigma_{r|r}$ relative to Taylor, this propagates all the way in the entire hierarchy and across iterations. Similar observations on VB have been made recently by Wang and

Titterington (2005); Consonni and Marin (2007) in the context of a different model, and also discussed in Chapter 5.

6.3.2 Bias correction

Table 6.3 shows estimates of parameters obtained from our approximation method in the column titled KF. As evident, the estimates are very close to PQL-1 estimates in Table 6.2, thus biased compared to the ones obtained using the slow Gibbs sampler. I corrected the bias in the estimates by modifying the procedure outlined in Kuk (1995); described in Algorithm 1. In the table, $\boldsymbol{\theta}$ denotes the unknown parameters $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$. The initial estimates $\tilde{\boldsymbol{\theta}}$ are the ones obtained from my EM algorithm using Taylor series approximation. At each iteration, I simulate M new data sets and get new estimates of the parameters by re-running my algorithm, this provides a bias correction and the procedure is repeated several times. As evident, this is an expensive procedure but is easily parallelizable. In general, a value of $M = 50$ with about $100 - 200$ iterations worked well for us. For my online advertising application (described in Section 6.4) where one needs to process large volumes of data that may not even fit into memory, the Map-Reduce (Dean and Ghemawat, 2004) paradigm provides an attractive computational framework to implement the parallel computation. The bias corrected estimates are reported under KF-B in Table 6.3. The estimates after bootstrap correction are closer to the estimates obtained from Gibbs sampling.

It is customary to estimate parameters like the $\boldsymbol{\gamma}$ parameters using a tuning hold-out set. For instance, such a strategy was adopted in Dudik *et al.* (2007) in the context of training a hierarchical MaxEnt model. To test the performance of such a strategy, I created a two-dimensional grid for $(\sqrt{\gamma_1}, \sqrt{\gamma_2})$ for the epidemiological Guatemalan data set ranging in $[.1, 3] \times [.1, 3]$ and computed the log-likelihood on a 10% randomly sampled hold-out data; the surface plot is shown in Figure 6.4. For each point on the two-dimensional grid, I estimated the other parameters $\boldsymbol{\phi}$ and $\boldsymbol{\beta}$, using our EM algorithm that does not update the value of $\boldsymbol{\gamma}$. The estimates at the optimal value of $\boldsymbol{\gamma}$ are shown in Table 6.3 under KF-C. The

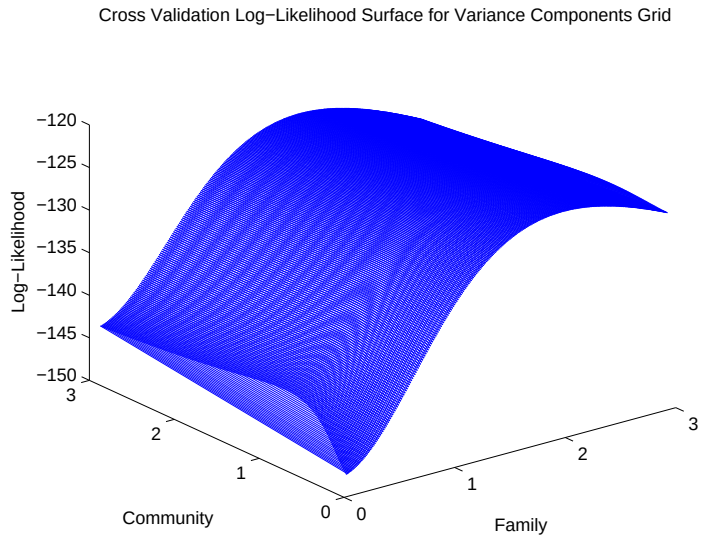


Figure 6.4: Predictive log-likelihood as function of family and community level variance components (standard deviation) for Guatemalan data set.

Algorithm 1 The bootstrap procedure

Obtain $\tilde{\boldsymbol{\theta}}$ as an initial estimate of $\boldsymbol{\theta}$. Bias $\boldsymbol{b}^{(0)} = 0$.

for $i = 1$ **to** N **do**

$\hat{\boldsymbol{\theta}} = \tilde{\boldsymbol{\theta}} - \boldsymbol{b}^{(i)}$.

for $j = 1$ **to** M **do**

Use $\hat{\boldsymbol{\theta}}$ to simulate new data j, by simulating $\boldsymbol{\phi}$ and the corresponding $\boldsymbol{Y}$.

For data j, obtain an new estimate of $\boldsymbol{\theta}$ as $\tilde{\boldsymbol{\theta}}^{(j)}$.

end for

$\boldsymbol{b}^{(i+1)} = \frac{1}{M} \sum_{j=1}^{M} \tilde{\boldsymbol{\theta}}^{(j)} - \hat{\boldsymbol{\theta}}$.

end for

estimates are better than KF but worse than KF-B. Hence, KF-B is recommended when computing resources are available (especially multiple processors) and running time is not a big constraint; when runtime is a big issue I recommend simple grid search using a small number of points around the initial estimate. For problems with large number of tweaking parameters, efficient grid search may be as expensive as a bootstrap correction.

6.4 Content Match Data Analysis

Web advertising supports a large swath of today's Internet ecosystem and has become a billion dollar business. Some of these advertisements are textual and some are graphical. *Contextual advertising* or *Content Match* (CM) refers to the placement of commercial textual advertisements within the content of a generic web page, while *Sponsored Search* (SS) advertising consists in placing advertisements on result pages from a web search engine, with advertisements driven by the originating query. In contextual advertising usually there is a commercial intermediary, called an *ad-network*, in charge of optimizing the advertisement selection with the twin goal of increasing revenue (shared between publisher and

Effects	KF	KF-B	KF-C	Gibbs
Fixed effects				
Individual				
Child age $\geq$ 2 years	0.99	1.77	1.18	1.84
Mother age $\geq$ 25 years	-0.09	-0.16	-0.10	-0.26
Birth order 2-3	-0.10	-0.18	-0.25	-0.29
Birth order 4-6	0.13	0.25	0.10	0.21
Birth order $\geq$ 7	0.20	0.36	0.21	0.50
Family				
Indigenous, no Spanish	-0.05	-0.11	0.02	-0.22
Indigenous Spanish	0.00	0.01	0.02	-0.11
Mother's education primary	0.22	0.44	0.32	0.48
Mother's education secondary or better	0.23	0.44	0.27	0.46
Husband's education primary	0.30	0.53	0.39	0.59
Husband's education secondary or better	0.27	0.48	0.35	0.55
Husband's education missing	0.02	0.04	-0.08	0.00
Mother ever worked	0.21	0.35	0.24	0.42
Community				
Rural	-0.50	-0.91	-0.62	-0.96
Proportion indigenous, 1981	-0.67	-1.23	-0.89	-1.22
Random effects				
Standard deviations $\boldsymbol{\gamma}$				
Family	0.74	2.40	1.92	2.60
Community	0.56	1.05	0.81	1.13

Table 6.3: Estimates for the binary MLH model of complete immunization (Kalman Filtering results)

ad-network) and improving user experience. Typically the ad-network and the publisher are paid only when the user *clicks* on an advertisement.

In this section I analyze data generated by a content match system where every showing of an advertisement on a web page (called an *impression*) constitutes an event. The goal is to place suitable advertisements on a given page to maximize long-term revenue to the ad-network. Since clicks generate revenue, it is tempting to build a predictive model for click-rates based on features derived from pages and advertisements. However, an algorithm that optimizes click rates alone may not produce relevant advertisements on pages unless the serving scheme is subject to constraints (e.g. one cannot serve washing machine advertisements on golf pages). Devising a serving scheme purely based on clicks may be detrimental in the long-run (e.g. a small fraction of bad quality but enticing advertisements may produce large number of clicks in the short run and may skew away the click model from relevance; this may hurt the system in the long run). I build a model that combines semantic information about (page,advertisement) pairs with click feedback as follows.

Pages and advertisements are classified into two separate large seven-level content hierarchies that are constructed based on editorial inputs. For each page or advertisement, there are usually more than one possible classifications. I work with the top path, but note that multiple membership can be easily handled in my case by slight modification of the likelihood. Instead of using the two original hierarchies of page and advertisement, I formed a new hierarchy (a pyramid) by taking the cross product of the two hierarchies. This is used to estimate smooth click-rates of (page,advertisement) pairs.

6.4.1 Training and Test Data

Although the page and advertisement hierarchies consist of 7 levels, classification is often done at coarser levels by the classifier. In fact, the average level at which the classification took place was 3.8. To train my model, I only consider the top 3 levels of the original hierarchy. Pages and advertisements that are classified at coarser levels are randomly as-

signed to the children nodes. Overall, the pyramid has 441, 25751 and 241292 nodes for the top 3 levels. Since the original data is incredibly massive, a sampling scheme from the original data is required. To train my model, I include all "slates" (set of advertisements shown together on a page) that had at least one click. Among the large fraction of slates with no clicks, I randomly sample .1% to be included in the sample. Thus, non-clicks that accompany clicks get much higher weight. A sampling scheme better than random (e.g. heavily downsample non-clicked slates for non-clickers) can be constructed but beyond the scope of this chapter. The training data were collected by confining to a specific subset of data which is sufficient to illustrate my methodology but in no way representative of the actual publisher traffic received by the ad-network under consideration. The training data I collected after sampling as described above over 23 days consisted of approximately 11M binary observations with approximately 1.9M clicks. The test set consisted of 1 day's worth of data with approximately .5M observations. I randomly split the test data into 20 equal sized partitions to report my results. The covariates include the position at which an advertisement is shown; ranking advertisements on pages after adjusting for positional effects is important as the positional effects introduce strong bias in the estimates (a bad advertisement shown at a lucrative position often has better click-throughs than a good advertisement shown at an unimportant position). In the training data a large fraction of leaf nodes in the pyramid (approx 95%) have zero clicks, this provides a good motivation to fit the binary MLH on this data to get smoother estimates at leaf nodes by using information at coarser resolutions.

6.4.2 Results

Since almost 95% of nodes at level 3 in the pyramid have 0 clicks, the naive regression model using leaf nodes as covariates does not converge (Silvapulle, 1981). I also note that constructing a metric that will quantify the impact of an algorithm when it is used to actually serve advertisements is a non-trivial problem; the best way is to run the algorithm

on live traffic. For the purposes of this chapter, I will report the average test log-likelihood as a goodness measure of a model. I compare the following models: a) The optimal model which predicts constant probability for all examples, b) 3 level MLH but without positional effects, c) top 2 level MLH to illustrate the gains of using information at a finer resolution, and d) 3 level MLH with positional effects to illustrate the generality of the approach; one can incorporate both additional features and the hierarchy into a single model. Figure 6.5 shows the distribution of average test likelihood on the partitions. As expected, all variations of MLH are better than the constant model. The MLH model which uses only 2 levels is inferior to the 3 level MLH while the general model that uses both covariates and hierarchy is the best.

6.5 Conclusion

In applications where data is aggregated at multiple resolutions with sparsity at finer resolutions, multi-level hierarchical models provide an attractive class to reduce variance by smoothing estimates at finer resolutions using data at coarser resolutions. However, the smoothing provides a better bias-variance tradeoff only when the hierarchy provides a natural clustering for the response variable and captures some latent characteristics of the process; often true in practice. I proposed a novel algorithm to fit these models based on a multi-scale Kalman filter that is both scalable and easy to implement. For the non-Gaussian case, the estimates are biased but performance can be improved by using a bootstrap correction or estimation through a tuning set.

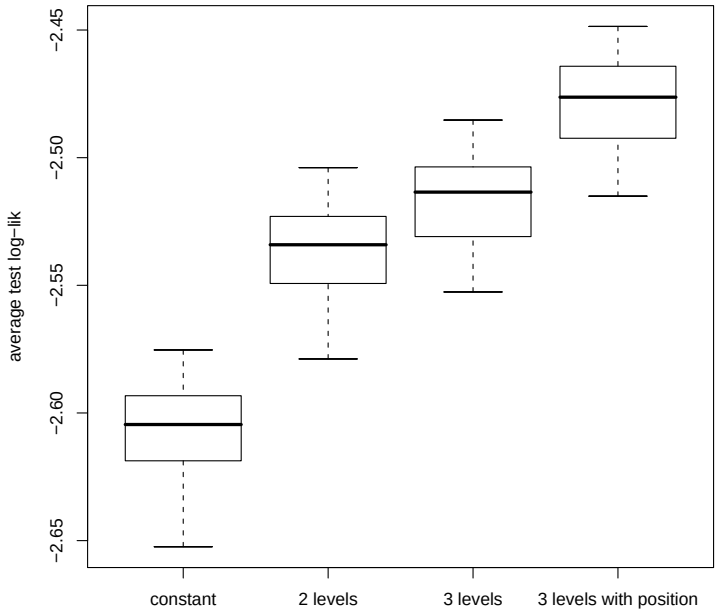


Figure 6.5: Average test log-likelihood on different variations of MLH computed on 20 equal sized splits of test data.

Chapter 7

Final Comments and Extensions

7.1 Extensions of Local Graphical Model Search

7.1.1 Extensive local neighborhood

Chapter 3 and 4 described the local graphical model search problem and the corresponding solutions. A local graphical model in previous chapters is defined to include edges that are either incident to the target variable or connecting two neighbors of the target variable. In Chapter 3, I assumed all the non-local edges remain null during the SSS and applied such an algorithm to cardiovascular genomics data. However, one may also wish to know the graphical structure of the neighborhood of $ne(Y)$. For example, in the cardiovascular genomics data analysis in Section 3.4, after having found glycoprotein (variable 3 in Appendix B) as one node in $ne(Y)$, one may also wish to know the local graphical model structure around glycoprotein. In this case, the algorithm using SSS, introduced and discussed in Chapter 3, has to be changed to a more refined way. If a node a is connected to at least one nodes in $ne(Y)$ in the graph, and $a \neq Y$, $a \notin ne(Y)$, we say a is a lv2 neighbor of Y. During the refined search, assume all the edges remain null, except which are incident to Y or $ne(Y)$ or connect two lv2 neighbors of Y. Note that these edges, which are set to be null in this chapter, can also be fixed by results from the preliminary studies. The corresponding new definition of the "local neighbor" of G for the SSS algorithm with the null non-local edge assumption becomes:

Definition 7.1. *A graphical model G' is defined as one local neighbor of G if any of the following is satisfied:*

1. G' is one of the graphs attained by adding one edge incident to Y or one node in $ne(Y)$ in G.

2. G' is one of the graphs attained by adding one edge connecting two nodes, each in $ne(Y)$ or to be one of the lv2 neighbors of Y in G.

3. G' is one of the graphs attained by deleting one edge incident to Y or one node in $ne(Y)$, and cleaning all the resulting non-local edges in G.

4. G' is one of the graphs attained by deleting one edge connecting two nodes, each in $ne(Y)$ or to be one of the lv2 neighbors of Y in G.

By this new definition of "local neighbor", SSS algorithm can be applied to find the top local graphical model with lv2 neighbors of Y.

7.1.2 Modified Targeted Metropolis-Hastings Method

Chapter 4 developed a new kind of Metropolis-Hastings method using a targeted proposal, which focuses on proposing to add or delete local edges during the sampling process. However, in section 4.5, I also noted that the TMH adding edge proposal is dominated by adding edges incident to Y, when p is large, which may result in the slow convergence of TMH. Therefore, one further extension is to slightly modify the TMH proposal so that it is more "targeted."

1. Choose to add or delete one edge with probability 1/2.

2. Choose to add or delete one local edge with probability λ, and non-local edge with probability $1 - \lambda$.

3. If adding/deleting a local edge, choose to add/delete an edge incident to Y with probability δ, and an edge connecting Y's neighbors with probability $1 - \delta$.

By adding an extra parameter δ into the proposal, one can control the proportion of time the proposal adds or deletes an edge incident to the target variable Y, which is now independent of p. When p is large, the proposal can now spend reasonable time on the

local edges connecting nodes in $ne(Y)$. Note that the MH acceptance probability forms in various cases are more complicated than the original TMH acceptance probabilities in Appendix D.

7.2 Hierarchical Priors over the Undirected Graphical Models

By Bayes' Theorem, the posterior probabilities of the graphical models are determined by two factors: the likelihood of the graphical models conditional on the data and the prior model probability. Mathematically, if one undirected graph is denoted by G and the data by $\boldsymbol{Y}_{1:n}$, we have $p(G|\boldsymbol{Y}_{1:n}) \propto p(\boldsymbol{Y}_{1:n}|G)p(G)$.

A potential hierarchical prior over undirected graphs is introduced in this section. In Chapters 3 and 4, Bernoulli priors were used for $p(G)$. The core idea of Bernoulli priors is to assume that every edge between any pair of nodes have the same prior probability to be included in the graph. Denote this prior probability as β, then

$$p(G) = \beta^{|E(G)|}(1-\beta)^{\frac{p(p-1)}{2}-|E(G)|}, \tag{7.1}$$

Bernoulli priors are quite convenient, and widely used in literature such as Jones *et al.* (2005). The value of β controls the sparsity of the "top" graphs found in the graphical model search. However, the choice of β actually requires rationale, because by decreasing or increasing β, one can obtain different "top" graphs with different sparsity. Although people usually have prior information of the sparsity of the graph, it is ideal if β can be treated as an unknown parameter with a hyper prior distribution. In this section I use the idea from Lucas *et al.* (2006), where they defined a class of hierarchical sparsity priors in Bayesian factor models and applied the priors in gene expression genomics.

As discussed above, instead of assuming that each edge between any pair of nodes i and j has the same prior probability β to be included in G, I allow them to have their

own probabilities of inclusion, which now are denoted as β_{ij} parameters. Note that sparsity plays an important role in graphical model selection and search, hence $|E(G)|$ should be small relative to p, the dimension of G. The natural hyper prior of β_{ij} can be

$$\beta_{ij} \sim (1-\rho_{ij})\delta_0(\beta_{ij}) + \rho_{ij} Be(\beta_{ij}|sr, s(1-r)), \tag{7.2}$$

where s and r are known parameters, ρ_{ij} is the probability that β_{ij} is not zero, and has its own hyper prior distribution. To simplify the prior, I assume that all the ρ_{ij} parameters are both unknown but equal, and denoted by ρ. Then, the prior of β_{ij} can be expressed as

$$\beta_{ij} \sim (1-\rho)\delta_0(\beta_{ij}) + \rho Be(\beta_{ij}|sr, s(1-r)). \tag{7.3}$$

I also assume that ρ has a beta prior distribution $\rho \sim Be(\rho|av, a(1-v))$ with a and v known. The joint distribution of G, $\boldsymbol{\beta}$, and ρ given the data $\boldsymbol{Y}_{1:n}$ is then

$$p(G, \boldsymbol{\beta}, \rho|\boldsymbol{Y}_{1:n}) \propto p(\boldsymbol{Y}_{1:n}|G)p(G|\boldsymbol{\beta})p(\boldsymbol{\beta}|\rho)p(\rho). \tag{7.4}$$

Integrating out $\boldsymbol{\beta}$, equation (7.4) becomes

$$p(G, \rho|\boldsymbol{Y}_{1:n}) \propto p(\boldsymbol{Y}_{1:n}|G)p(\rho)\prod_{i,j}\int_0^1 \beta_{ij}^{e_{ij}}(1-\beta_{ij})^{1-e_{ij}}p(\beta_{ij})d\beta_{ij}, \tag{7.5}$$

where $e_{ij} = 1$ if there is an edge between nodes i and j in G, and $e_{ij} = 0$ otherwise.

Note that when $e_{ij} = 1$, the integral on each β_{ij} in equation (7.5) is just $E[\beta_{ij}|\rho]$, and when $e_{ij} = 0$, it becomes $1 - E[\beta_{ij}|\rho]$. By the point-mass mixture prior of β_{ij}, we have $E[\beta_{ij}|\rho] = \rho r$. Therefore, equation (7.5) becomes

$$p(G, \rho|\boldsymbol{Y}_{1:n}) \propto p(\boldsymbol{Y}_{1:n}|G)p(\rho)(\rho r)^{|E(G)|}(1-\rho r)^{\frac{p(p-1)}{2}-|E(G)|}. \tag{7.6}$$

By Denoting $K = \frac{p(p-1)}{2}$, and integrating out ρ from $p(G, \rho|\boldsymbol{Y}_{1:n})$, we can then obtain the marginal posterior probability of the graph G given the data $\boldsymbol{Y}_{1:n}$.

$$
\begin{aligned}
p(G|\boldsymbol{Y}_{1:n}) &\propto p(\boldsymbol{Y}_{1:n}|G)\int_0^1(\rho r)^{|E(G)|}(1-\rho r)^{K-|E(G)|}p(\rho)d\rho \\
&= p(\boldsymbol{Y}_{1:n}|G)\int_0^1(\rho r)^{|E(G)|}\sum_{k=0}^{K-|E(G)|}\binom{K-|E(G)|}{k}(-1)^{K-|E(G)|-k}(\rho r)^{K-|E(G)|-k}p(\rho)d\rho \\
&= p(\boldsymbol{Y}_{1:n}|G)\sum_{k=0}^{K-|E(G)|}\binom{K-|E(G)|}{k}(-1)^{K-|E(G)|-k}r^{K-k}E[\rho^{K-k}] \\
&\propto p(\boldsymbol{Y}_{1:n}|G)\sum_{k=0}^{K-|E(G)|}\binom{K-|E(G)|}{k}(-1)^{K-|E(G)|-k}r^{-k}\frac{\Gamma(av+K-k)}{\Gamma(a+K-k)}.
\end{aligned}
$$

Therefore, the marginal prior of the graph G is

$$
p(G) \propto \sum_{k=0}^{K-|E(G)|}\binom{K-|E(G)|}{k}(-1)^{K-|E(G)|-k}r^{-k}\frac{\Gamma(av+K-k)}{\Gamma(a+K-k)}. \tag{7.7}
$$

The prior introduced in equation (7.7) allows each edge to have its own prior probability to be included in the graph. Also, the marginal posterior $p(\rho|\boldsymbol{Y}_{1:n})$ implies the posterior distribution of the probability of any β_{ij} being non-zero given the data. If considering ρ as the sparsity parameter, ρ implies how sparse on average the graphical models should be given the data. Obviously,

$$
p(\rho|\boldsymbol{Y}_{1:n}) = \sum_G p(G,\rho|\boldsymbol{Y}_{1:n}), \tag{7.8}
$$

which is not easy to compute because of the huge number of possible graphs if p is very large. However, this can make us theoretically understand how this prior works within the whole framework.

Appendix A

The Gene Probsets Generating the Projected Risk Signature

- 1. J04765 /FEATURE= /DEFINITION=HUMOSTRO Human osteopontin mRNA, complete cds
- 2. Cluster Incl. AB006780: Homo sapiens mRNA for galectin-3, complete cds /cds=(53,805) /gb=AB006780 /gi=2385451 /ug=Hs.621 /len=943
- 3. M30257 /FEATURE= /DEFINITION=HUMCAM1V Human vascular cell adhesion molecule 1 mRNA, complete cds
- 4. Cluster Incl. AA868382: ak41e04.s1 Homo sapiens cDNA, 3 end/ clone=IMAGE-1408542/clone_ end=3 /gb=AA868382 /gi=2963827 /ug=Hs.198253 /len=936
- 5. Cluster Incl. M37033: Human CD53 glycoprotein mRNA, complete cds /cds=(93,752) /gb=M37033 /gi=180142 /ug=Hs.82212 /len=1480
- 6. Cluster Incl. S57235: CD68=110kda transmembrane glycoprotein [human, promonocyte cell line U937, mRNA, 1722 nt] /cds=(15,1079) /gb=S57235 /gi=298664 /ug=Hs.226237 /len=1689
- 7. Cluster Incl. X03084: Human mRNA for C1q B-chain of complement system /cds=(0,687) /gb=X03084 /gi=29537 /ug=Hs.8986 /len=935
- 8. Cluster Incl. M32578: Human MHC class II HLA-DR beta-1 mRNA (DR2.3), 5end /cds=(61,861) /gb=M32578 /gi=188305 /ug=Hs.181366 /len=1216
- 9. Cluster Incl. M81141: Human MHC class II HLA-DQ-beta mRNA (DR7 DQw2), complete cds /cds=(35,820) /gb=M81141 /gi=188202 /ug=Hs.73933 /len=1171

- 10. Cluster Incl. M76665: Human 11-beta-hydroxysteroid dehydrogenase (HSD11) gene /cds=(94,972) /gb=M76665 /gi=179474 /ug=Hs.37012 /len=1375
- 11. M30257 /FEATURE= /DEFINITION=HUMCAM1V Human vascular cell adhesion molecule 1 mRNA, complete cds
- 12. Cluster Incl. J02923: Human 65-kilodalton phosphoprotein (p65) mRNA, complete cds /cds=(74,1957) /gb=J02923 /gi=189501 /ug=Hs.76506 /len=3175
- 13. Cluster Incl. X00442: Human mRNA for haptoglobin alpha(2FS)-beta precursor /cds=(26,1246) /gb=X00442 /gi=31749 /ug=Hs.75990 /len=1412
- 14. Cluster Incl. U15085: Human HLA-DMB mRNA, complete cds /cds=(233,1024) /gb=U15085 /gi=557701 /ug=Hs.1162 /len=1362
- 15. Cluster Incl. U51240: Human lysosomal-associated multitransmembrane protein (LAPTm5) mRNA, complete cds /cds=(75,863) /gb=U51240 /gi=1255239 /ug=Hs.79356 /len=2232
- 16. Cluster Incl. M60315: Human transforming growth factor-beta (tgf-beta) mRNA, complete cds /cds=(159,1700) /gb=M60315 /gi=339561 /ug=Hs.238991 /len=2923
- 17. X59065 /FEATURE=exon /DEFINITION=HSFGFEX3 H.sapiens FGF gene, exon 3
- 18. M21121 /FEATURE= /DEFINITION=HUMTCSM Human T cell-specific protein (RANTES) mRNA, complete cds
- 19. Cluster Incl. W22541: 69B4 Homo sapiens cDNA /clone=(not-directional) /gb=W22541 /gi=1299374 /ug=Hs.20930 /len=809

Appendix B

The Neighbors of the Projected Risk Signature in Top 50 Models

- 1. Cluster Incl. M37033: Human CD53 glycoprotein mRNA, complete cds /cds=(93,752) /gb=M37033 /gi=180142 /ug=Hs.82212 /len=1480 (*No. 5 in generating signature list*)
- 2. Cluster Incl. M63835: Human IgG Fc receptor I gene /cds=(155,1279) /gb=M63835 /gi=180278 /ug=Hs.77424 /len=1437
- 3. Cluster Incl. J02923: Human 65-kilodalton phosphoprotein (p65) mRNA, complete cds /cds=(74,1957) /gb=J02923 /gi=189501 /ug=Hs.76506 /len=3175 (*No. 12 in generating signature list*)
- 4. Cluster Incl. M81141: Human MHC class II HLA-DQ-beta mRNA (DR7 DQw2), complete cds /cds=(35,820) /gb=M81141 /gi=188202 /ug=Hs.73933 /len=1171 (*No. 9 in generating signature list*)
- 5. X03663 /FEATURE=cds /DEFINITION=HSCFMS Human mRNA for c-fms proto-oncogene /NOTE=replacement of probe set 1864_at
- 6. Cluster Incl. M15395: Human leukocyte adhesion protein (LFA-1/Mac-1/p150,95 family) beta subunit mRNA /cds=(72,2381) /gb=M15395 /gi=186933 /ug=Hs.83968 /len=2776
- 7. Cluster Incl. X62744: Human RING6 mRNA for HLA class II alpha chain-like product /cds=(45,830) /gb=X62744 /gi=36062 /ug=Hs.77522 /len=1079

- 8. X87212 /FEATURE=cds /DEFINITION=HSCATHCGE H.sapiens mRNA for cathepsin C
- 9. Cluster Incl. L09708: Human complement component 2 (C2) gene allele b /cds=(271,2529) /gb=L09708 /gi=2804581 /ug=Hs.2253 /len=2844
- 10. Cluster Incl. X76488: H.sapiens mRNA for lysosomal acid lipase /cds=(145,1344) /gb=X76488 /gi=434305 /ug=Hs.85226 /len=2599
- 11. Cluster Incl. U15085: Human HLA-DMB mRNA, complete cds /cds=(233,1024) /gb=U15085 /gi=557701 /ug=Hs.1162 /len=1362 (*No. 14 in generating signature list*)
- 12. Cluster Incl. M32578: Human MHC class II HLA-DR beta-1 mRNA (DR2.3), 5end /cds=(61,861) /gb=M32578 /gi=188305 /ug=Hs.181366 /len=1216 (*No. 8 in generating signature list*)
- 13. J04765 /FEATURE= /DEFINITION=HUMOSTRO Human osteopontin mRNA, complete cds (*No. 1 in generating signature list*)
- 14. Cluster Incl. J02876: Human placental folate binding protein mRNA, complete cds /cds=(262,1029) /gb=J02876 /gi=182413 /ug=Hs.24194 /len=1211
- 15. Cluster Incl. N90862: zb11b06.s1 Homo sapiens cDNA, 3 end /clone=IMAGE-301715 /clone_end=3 /gb=N90862 /gi=1444189 /ug=Hs.172684 /len=605
- 16. Cluster Incl. X16832: Human mRNA for cathepsin H (EC 3.4.22.16) /cds=(34,1041) /gb=X16832 /gi=29709 /ug=Hs.76476 /len=1399
- 17. Cluster Incl. U51240: Human lysosomal-associated multitransmembrane protein (LAPTm5) mRNA, complete cds /cds=(75,863) /gb=U51240 /gi=1255239 /ug=Hs.79356 /len=2232 (*No. 15 in generating signature list*)

- 18. Cluster Incl. J03909: Human gamma-interferon-inducible protein (IP-30) mRNA, complete cds /cds=(40,951) /gb=J03909 /gi=186264 /ug=Hs.14623 /len=1032
- 19. Cluster Incl. M21186: Human neutrophil cytochrome b light chain p22 phagocyte b-cytochrome mRNA, complete cds /cds=(28,615) /gb=M21186 /gi=189105 /ug=Hs.68877 /len=687
- 20. Cluster Incl. M13560: Human Ia-associated invariant gamma-chain gene /cds=(795,1493) /gb=M13560 /gi=184518 /ug=Hs.84298 /len=2080

Appendix C

The Predictors of the Projected Risk Signature in Top 10 Models

- 1. J04765 /FEATURE=/DEFINITION=HUMOSTRO Human osteopontin mRNA, complete cds (*No. 1 in generating signature list, also selected as Y's neighbor*)
- 2. Cluster Incl. AB006780: Homo sapiens mRNA for galectin-3, complete cds /cds=(53,805) /gb=AB006780 /gi=2385451 /ug=Hs.621 /len=943 (*No. 2 in generating signature list*)
- 3. M30257 /FEATURE= /DEFINITION=HUMCAM1V Human vascular cell adhesion molecule 1 mRNA, complete cds (*No. 3 in generating signature list*)
- 4. Cluster Incl. AA868382: ak41e04.s1 Homo sapiens cDNA, 3 end /clone=IMAGE-1408542 /clone_end=3 /gb=AA868382 /gi=2963827 /ug=Hs.198253 /len=936 (*No. 4 in generating signature list*)
- 5. Cluster Incl. M37033: Human CD53 glycoprotein mRNA, complete cds /cds=(93,752) /gb=M37033 /gi=180142 /ug=Hs.82212 /len=1480 (*No. 5 in generating signature list, also selected as Y's neighbor*)
- 6. Cluster Incl. X03084: Human mRNA for C1q B-chain of complement system /cds=(0,687) /gb=X03084 /gi=29537 /ug=Hs.8986 /len=935 (*No. 7 in generating signature list*)
- 7. Cluster Incl. M32578:Human MHC class II HLA-DR beta-1 mRNA (DR2.3), 5end /cds=(61,861) /gb=M32578 /gi=188305 /ug=Hs.181366 /len=12 (*No. 8 in generating signature list, also selected as Y's neighbor*)

- 8. Cluster Incl. M81141: Human MHC class II HLA-DQ-beta mRNA (DR7 DQw2), complete cds /cds=(35,820) /gb=M81141 /gi=188202 /ug=Hs.73933 /len=1171 (*No. 9 in generating signature list, also selected as Y's neighbor*)

- 9. Cluster Incl. J02923: Human 65-kilodalton phosphoprotein (p65) mRNA, complete cds /cds=(74,1957) /gb=J02923 /gi=189501 /ug=Hs.76506 /len=3175 (*No. 12 in generating signature list, also selected as Y's neighbor*)

- 10. Cluster Incl. U15085: Human HLA-DMB mRNA, complete cds /cds=(233,1024) /gb=U15085 /gi=557701 /ug=Hs.1162 /len=1362 (*No. 14 in generating signature list, also selected as Y's neighbor*)

- 11. Cluster Incl. U51240: Human lysosomal-associated multitransmembrane protein (LAPTm5) mRNA, complete cds /cds=(75,863) /gb=U51240 /gi=1255239 /ug=Hs.79356 /len=2232 (*No. 15 in generating signature list, also selected as Y's neighbor*)

- 12. Cluster Incl. X62744: Human RING6 mRNA for HLA class II alpha chain-like product /cds=(45,830) /gb=X62744 /gi=36062 /ug=Hs.77522 /len=1079 (*Selected as Y's neighbor*)

- 13. Cluster Incl. N90862: zb11b06.s1 Homo sapiens cDNA, 3 end /clone=IMAGE-301715 /clone_end=3 /gb=N90862 /gi=1444189 /ug=Hs.172684 /len=605 (*Selected as Y's neighbor*)

- 14. Cluster Incl. M21186: Human neutrophil cytochrome b light chain p22 phagocyte b-cytochrome mRNA, complete cds /cds=(28,615) /gb=M21186 /gi=189105 /ug=Hs.68877 /len=687 (*Selected as Y's neighbor*)

- 15. M16592 /FEATURE=mRNA /DEFINITION=HUMHCKB Human hemopoietic cell protein-tyrosine kinase (HCK) gene, complete cds, clone HK24 Cluster Incl.

- 16. J04131:Human gamma-glutamyl transpeptidase (GGT) protein mRNA, complete cds /cds=UNKNOWN /gb=J04131 /gi=183137 /ug=Hs.135 /len=2535

Appendix D

Acceptance Probabilities in the Targeted Metropolis-Hastings Methods

In Metropolis-Hasting Methods, the acceptance probability $\rho(G, G')$ is defined by

$$\rho(G, G') = \min\{\frac{p(G'|\boldsymbol{Y}_{1:n})}{p(G|\boldsymbol{Y}_{1:n})}\frac{q(G|G')}{q(G'|G)}, 1\}, \tag{D.1}$$

where $q(G'|G)$ ($q(G|G')$) is the probability of the graph G' (G) to be proposed given the current graph G (G').

This appendix discusses the conditional densities $q(G|G')$ and $q(G'|G)$ for different cases in Targeted Metropolis-Hasting methods. In every iteration, I set the probability to add or delete an edge to be 0.5, if either proposal is possible. Also, assume there are currently k variables in $ne(Y)$, with $p-k-1$ variables out of $ne(Y)$. Among those k variables and Y, assume there are l local edges, with the total number of edges in the current graph G being e.

First of all, we discuss $q(G|G')$ as well as $q(G'|G)$ when adding one edge between nodes i and j. Without losing much precision, I assume that this adding proposal does not violate the decomposability of the graph. All the different cases are:

1. It is a local edge (either i or j is target variable Y, or both i and j are currently in $ne(Y)$).

   (1) Both i and j are currently Y's neighbors ($l \neq 0$ and $e < \frac{p(p-1)}{2} - 1$).

$$\frac{q(G|G')}{q(G'|G)} = \frac{\frac{(k+1)k}{2} - l + p - k - 1}{l+1}.$$

(2) Either i or j is target variable Y ($l \neq 0$, $e < \frac{p(p-1)}{2} - 1$, and $k \neq p-2$). Set $Y = i$ for convenience. When we add j to Y, all of the edges connecting j and the other neighbors of Y will become local edges. Denote the number of those edges as l_j.

$$\frac{q(G|G')}{q(G'|G)} = \frac{\frac{(k+1)k}{2} - l + p - k - 1}{l + l_j + 1}.$$

(3) Either i or j is target variable Y, and $k = p - 2$.

$$\frac{q(G|G')}{q(G'|G)} = \frac{\frac{(k+1)k}{2} - l + p - k - 1}{\lambda(l + l_j + 1)}.$$

(4) When $l = 0$, there is no way to delete a local edge. Therefore, in this case,

$$\frac{q(G|G')}{q(G'|G)} = \frac{\frac{(k+1)k}{2} - l + p - k - 1}{2(l + 1)}.$$

(5) When $e = \frac{p(p-1)}{2} - 1$, the graph is full but one edge.

$$\frac{q(G|G')}{q(G'|G)} = \frac{2(\frac{(k+1)k}{2} - l + p - k - 1)}{(l + 1)},$$

or:

$$\frac{q(G|G')}{q(G'|G)} = \frac{2(\frac{(k+1)k}{2} - l + p - k - 1)}{(l + l_j + 1)},$$

or:

$$\frac{q(G|G')}{q(G'|G)} = \frac{2(\frac{(k+1)k}{2} - l + p - k - 1)}{\lambda(l + l_j + 1)}.$$

2. It is not a local Edge. (When $k = p - 1$, this cannot happen!)

(1) When $0 < e - l < \frac{(p-k-1)(p-k-2)}{2} + (p - k - 1)k - 1$,

$$\frac{q(G|G')}{q(G'|G)} = \frac{\frac{(p-k-1)(p-k-2)}{2} + (p - k - 1)k - e + l}{e - l + 1}.$$

(2) When $e - l = 0$,

$$\frac{q(G|G')}{q(G'|G)} = \frac{\frac{(p-k-1)(p-k-2)}{2} + (p-k-1)k - e + l}{2(e-l+1)}.$$

(3) When $e - l = \frac{(p-k-1)(p-k-2)}{2} + (p-k-1)k - 1$,

$$\frac{q(G|G')}{q(G'|G)} = \frac{(p-k-1)(p-k-2) + 2(p-k-1)k - 2e + 2l}{(e-l+1)}.$$

Now we discuss the case of deleting one edge between nodes i and j. I also assume that this deleting proposal does not violate the decomposability of the graph.

1. It is a local edge.

   (1) Both i and j are currently in $ne(Y)$ ($l \neq 1$ and $e < \frac{p(p-1)}{2}$).

   $$\frac{q(G|G')}{q(G'|G)} = \frac{l}{\frac{(k+1)k}{2} + p - k - l}.$$

   (2) Either i or j is target variable Y ($l \neq 1$, $e < \frac{p(p-1)}{2}$, and $k \neq p-1$). Set $Y = i$ for convenience. When we delete the edge (Y, j), all of the edges connecting j and the other neighbors of Y will become non-local edges. Denote the number of those edges as l_j.

   $$\frac{q(G|G')}{q(G'|G)} = \frac{l}{\frac{k(k-1)}{2} + p - k - l + 1 + l_j}.$$

   (3) Either i or j is target variable Y, and $k = p - 1$.

   $$\frac{q(G|G')}{q(G'|G)} = \frac{\lambda l}{\frac{k(k-1)}{2} + p - k - l + 1 + l_j}.$$

   (4) When $l = 1$, the proposal deletes the last edge of the graph (Y, i).

   $$\frac{q(G|G')}{q(G'|G)} = \frac{2l}{\frac{k(k-1)}{2} + p - k - l + 1 + l_j}.$$

(5) When $e = \frac{p(p-1)}{2}$, the proposal deletes an edge from a full graph.

$$\frac{q(G|G')}{q(G'|G)} = \frac{l}{(k+1)k + 2p - 2k - 2l},$$

or:

$$\frac{q(G|G')}{q(G'|G)} = \frac{l}{k(k-1) + 2p - 2k - 2l + 2 + 2l_j},$$

or:

$$\frac{q(G|G')}{q(G'|G)} = \frac{\lambda l}{k(k-1) + 2p - 2k - 2l + 2 + 2l_j}.$$

2. It is not a local edge.

(1) When $1 < e - l < \frac{(p-k-1)(p-k-2)}{2} + (p-k-1)k$,

$$\frac{q(G|G')}{q(G'|G)} = \frac{e-l}{\frac{(p-k-1)(p-k-2)}{2} + k(p-k-1) - e + l + 1}.$$

(2) When $e - l = 1$,

$$\frac{q(G|G')}{q(G'|G)} = \frac{2(e-l)}{\frac{(p-k-1)(p-k-2)}{2} + k(p-k-1) - e + l + 1}.$$

(3) When $e - l = \frac{(p-k-1)(p-k-2)}{2} + (p-k-1)k$,

$$\frac{q(G|G')}{q(G'|G)} = \frac{e-l}{(p-k-1)(p-k-2) + 2k(p-k-1) - 2e + 2l + 2}.$$

Bibliography

Agarwal, D., Broder, A., Chakrabarti, D., Diklic, D., Josifovski, V., and Sayyadian, M. (2007). Estimating rates of rare events at multiple resolutions. In P. Berkhin, R. Caruana, and S. Gaffney, eds., *Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining*, 16–25.

Armstrong, H., Carter, C. K., Wong, K., and Kohn, R. (2005). Bayesian covariance matrix estimation using a mixture of decomposable graphical models. *Unpublished manuscript, available at SSRN: http://ssrn.com/abstract=966635* .

Atay-Kayis, A. and Massam, H. (2005). The marginal likelihood for decomposable and non-decomposable graphical Gaussian models. *Biometrika* **92**, 317–335.

Ballantyne, C. M. and Entman, M. L. (2002). Soluble adhesion molecules and the search for biomarkers for atherosclerosis. *Circulation* **106**, 766–767.

Barbieri, M. M. and Berger, J. O. (2004). Optimal predictive model selection. *The Annals of Statistics* **32**, 3, 870–897.

Bernhagen, J., Krohn, R., Lue, H., Gregory, J., Zernecke, A., Koenen, R., Dewor, M., Georgiev, I., Schober, A., Leng, L., Kooistra, T., Fingerle-Rowson, G., Ghezzi, P., Kleemann, R., McColl, S., Bucala, R., Hickey, M., and Weber, C. (1997). MIF is a noncognate ligand of CXC chemokine receptors in inflammatory and atherogenic cell recruitment. *Nature Medicine* **13**, 587–596.

Blalock, H. M., J. e. (1971). *Causal Models in The Social Sciences*. Aldine-Atheston, Chicago.

Breslow, N. E. and Clayton, D. G. (1993). Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association* **88**, 421, 9–25.

Carvalho, C. M. (2006). *Structure and Sparsity in High-Dimensional Multivariate Analysis*. PhD. Thesis, Institute of Statistics and Decision Sciences, Duke University, http://isds.duke.edu/people/theses/carlos.html.

Carvalho, C. M., Lucas, J., Wang, Q., Nevins, J., and West, M. (2008). High-dimensional sparse factor modelling: Applications in gene expression genomics. *Journal of the American Statistical Association* To appear.

Casella, G. and Moreno, E. (2006). Objective Bayesian variable selection. *Journal of the American Statistical Association* **101**, 157–167.

Castelo, R. and Roverato, A. (2006). A robust procedure for Gaussian graphical model search from microarray data with p larger than n. *The Journal of Machine Learning Research* **7**, 2621–2650.

Chen, M. (2005). Computing marginal likelihoods from a single MCMC output. *Statistica Neerlandica* **59**, 16–29.

Chen, M. H., Ibrahim, J. G., and Yiannoutsos, C. (1999). Prior elicitation, variable selection and Bayesian computation for logistic regression models. *Journal of the Royal Statistical Society. Series B (Methodological)* **61**, 223–242.

Chib, S. (1995). Marginal likelihood from the Gibbs output. *Journal of the American Statistical Association* **90**, 1313–1321.

Chib, S. and Jeliazkov, I. (2001). Marginal likelihood from the Metropolis-Hastings output. *Journal of the American Statistical Association* **96**, 270–281.

Chipman, H., George, E. I., and McCulloch, R. E. (2001). The practical implementation of Bayesian model selection (with discussion). In P. Lahiri, ed., *Model Selection*, vol. 38, 66–134. IMS, Beachwood, OH.

Chou, K. C., Willsky, A. S., and Nikoukhah, R. (1994). Multiscale systems, Kalman filters, and Ricatti equations. *IEEE Transactions on Automatic Control* **39**, 479–492.

Clyde, M. and George, E. I. (2004). Model uncertainty. *Statistical Science* **19**, 81–94.

Consonni, G. and Marin, J. (2007). Mean field variational Bayesian inference for latent variable models. *Computational Statistics and Data Analysis* **52**, 790–798.

Dansky, H., Barlow, C., Lominska, C., Sikes, J., Kao, C., Weinsaft, J., Cybulsky, M., and Smith, J. (2001). Adhesion of monocytes to arterial endothelium and initiation of atherosclerosis are critically dependent on vascular cell adhesion molecure-1 gene dosage. *Arteriosclerosis, Thrombosis, and Vascular Biology* **21**, 1662–1667.

Dawid, A. P. (1979). Conditional independence in statistical theory (with discussion). *Journal of the Royal Statistical Society. Series B (Methodological)* **41**, 1–31.

Dawid, A. P. (1980). Conditional independence for statistical operations. *The Annals of Statistics* **8**, 598–617.

Dawid, A. P. and Lauritzen, S. L. (1993). Hyper-Markov laws in the statistical analysis of decomposable graphical models. *The Annals of Statistics* **3**, 1272–1317.

Dean, J. and Ghemawat, S. (2004). Mapreduce: Simplified data processing on large clusters. In *Sixth Symposium on Operating System Design and Implementation, http://labs.google.com/papers/mapreduce.html*.

Dellaportas, P., Forster, J., and Ntzoufras, I. (2002). On Bayesian model and variable selection using MCMC. *Statistics and Computing* **12**, 27–36.

Dellaportas, P. and Forster, J. J. (1999). Markov chain Monte Carlo model determination for hierarchical and graphical log-linear models. *Biometrika* **86**, 615–633.

Dellaportas, P., Giudici, P., and Roberts, G. (2003). Bayesian inference for non-decomposable graphical Gaussian models. *Sankhya, Series A,* **65**, 43–55.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)* **39**, 1–38.

DiCiccio, T. J., Kass, R. E., and Wasserman, L. (1997). Computing Bayes factors by combining simulation and asymptotic approximations. *Journal of the American Statistical Association* **92**, 903–915.

Dobra, A. and West, M. (2004). Bayesian covariance selection. *Duke Statistics Discussion Papers* **23**.

Dudik, M., Blei, D. M., and Schapire, R. E. (2007). Hierarchical maximum entropy density estimation. In Z. Ghahramani, ed., *Proceedings of the Twenty-Fourth International Conference (ICML 2007)*, 249–256.

Dunson, D., Baird, D., Wilcox, A., and Weinberg, C. (1999). Day-specific probabilities of clinical pregnancy based on two studies with imperfect measures of ovulation. *Human Reproduction* **14**, 1835–1839.

Dunson, D., Colombo, B., and Baird, D. (2002). Changes with age in the level and duration of fertility in the menstrual cycle. *Human Reproduction* **17**, 1399–1403.

Dunson, D. and Stanford, J. (2005). Bayesian inferences on predictors of conception probabilities. *Biometrics* **61**, 126–133.

Gelfand, A. and Dey, D. (1994). Bayesian model choice: asymptotics and exact calculations. *Journal of the Royal Statistical Society. Series B (Methodological)* **56**, 3, 501–514.

Gelfand, A. E. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association* **85**, 410, 398–409.

Gelman, A. and Hill, J. (2007). *Data Analysis using Regression and Multi-Level/Hierarchical Models*. Cambridge University Press.

Gelman, A. E. and Meng, X. L. (1998). Simulating normalized constants: From importance sampling to bridge sampling to path sampling. *Statistical Science* **13**, 163–185.

George, E. and McCulloch, R. (1993). Variable selection via Gibbs sampling. *Journal of American Statistical Association* **88**, 881–889.

George, E. and McCulloch, R. (1997). Approaches for Bayesian variable selection. *Statistica Sinica* **7**, 339–373.

George, E. I. (2000). The variable selection problem. *Journal of the American Statistical Association* **95**, 1304–1308.

Giachelli, C., Liaw, L., Murry, C., S.M., S., and Almeida, M. (1995). Osteopontin expression in cardiovascular diseases. *Annals of the New York Academy of Sciences* **21**, 109–126.

Gibbs, W. (1902). *Elementary Principles of Statistical Mechanics*. Yale University Press, NewHaven, Connecticut.

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

Giudici, P. and Castelo, R. (2003). Improving Markov chain Monte Carlo model search for data mining. *Machine Learning* **50**, 127–158.

Goldstein, H. (1991). Nonlinear multilevel models, with an application to discrete response data. *Biometrika* **78**, 45–51.

Gonzalez-Gay, M., Gonzalez-Juanatey, C., and Ollier, W. (2004). Endothelial dysfunction in rheumatoid arthritis: influence of HLA-DRB1 alleles. *Autoimmunity reviews* **3**, 301–304.

Grone, R., Johnson, C. R., Sà, E. M., and Wolkowice, H. (1984). Positive definite completions of partial Hermitian matricies. *Linear algebra and its applications* **58**, 109–124.

Hammersley, J. M. and Clifford, P. E. (1968). Markov fields on finite graphs and lattices. Preprint, UC. Berkeley.

Han, C. and Carlin, B. (2001). Markov chain Monte Carlo methods for computing Bayes factors: A comparative review. *Journal of the American Statistical Association* **96**, 1122–1132.

Hans, C. (2005). *Regression Model Search and Uncertainty with Many Predictors*. PhD. Thesis, Duke University.

Hans, C., Dobra, A., and West, M. (2007). Shotgun stochastic search for "large p" regression. *Journal of the American Statistical Association* **102**, 507–516.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57**, 97–109.

Huang, H. and Cressie, N. (2000). Fast spatial prediction of global processes from satellite data. *SIAM Journal on Scientific Computing* **21**, 4, 1551–1566.

Ishwaran, H. and Rao, J. (2005). Spike and slab variable selection: Frequentist and Bayesian strategies. *Annals of Statistics* **33**, 730–773.

Jaakkola, T. S. and Jordan, M. I. (2000). Bayesian parameter estimation via variational methods. *Statistics and Computing* **10**, 25–37.

James, W. and Stein, C. (1961). Estimation with quadratic loss. In J. Neyman, ed., *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, 361–379. University of California Press.

Jerrum, M. R., Valiant, L. G., and Vazirani, V. V. (1986). Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science* **43**, 169–188.

Jones, B., Carvalho, C., Dobra, A., Hans, C., Carter, C., and West, M. (2005). Experiments in stochastic computation for high-dimensional graphical models. *Statistical Science* **20**, 388–400.

Jönsson, G., Truedsson, L., Sturfelt, G., Oxelius, V., Braconier, J., and Sjöholm, A. (2005). Hereditary C2 deficiency in Sweden: frequent occurrence of invasive infection, atherosclerosis, and rheumatic disease. *Medicine (Baltimore)* **84**, 23–34.

Kitagawa, K., Matsumoto, M., Sasaki, T., Hashimoto, H., Kuwabara, K., Ohtsuki, T., and Hori, M. (2002). Involvement of ICAM-1 in the progression of atherosclerosis in APOE-knockout mice. *Atherosclerosis* **160**, 305–310.

Kuk, A. Y. C. (1995). Asymptotically unbiased estimation in generalized linear models with random effects. *Journal of the Royal Statistical Society, Series B (Methodological),* **57**, 395–407.

Laird, N. M. and Ware, J. H. (1982). Random-effects models for longitudinal data. *Biometrics* **38**, 4, 963–974.

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

Lindley, D. V. (1971). The estimation of many parameters. In V. P. Godambe and D. A. Sprott, eds., *Foundations of Statistical Inference*, 435–466. Toronto: Holt, Rinehart, and Winston.

Lindley, D. V. (1972). *Bayesian Statistics: A Review*. SIAM, Philadelphia.

Lindley, D. V. and Smith, A. F. M. (1972). Bayes estimates for the linear model. *Journal of the Royal Statistical Society. Series B (Methodological)* **34**, 1–41.

Lucas, J., Carvalho, C., Wang, Q., Bild, A., Nevins, J. R., and West, M. (2006). Sparse statistical modelling in gene expression genomics. In K.-A. Do, P. Müller, and M. Vannucci, eds., *Bayesian Inference for Gene Expression and Proteomics*, 155–176. Cambridge University Press.

Madigan, D. and York, J. (1995). Bayesian graphical models for discrete data. *International Statistical Review* **63**, 215–232.

Mas, A., Blanco, E., Moñux, G., Urcelay, E., Serrano, F., de la Concha, E., and Martínez, A. (2005). DRB1-TNF-alpha-TNF-beta haplotype is strongly associated with severe aortoiliac occlusive disease, a clinical form of atherosclerosis. *Human immunology* **66**, 1062–1067.

McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models (2nd ed.).* Chapman and Hall, London.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *J. Chemical Physics* **21**, 1087–1092.

Miller, A. J. (2002). *Subset Selection in Regression, 2nd ed.* Chapman and Hall, London.

Nageh, M., Sandberg, E., Marotti, K., Lin, A., Melchior, E., Bullard, D., and Beaudet, A. (1997). Deficiency of inflammatory cell adhesion molecules protects against atherosclerosis in mice. *Arteriosclerosis, Thrombosis, and Vascular Biology* **17**, 1517–1520.

Ntzoufras, I., Dellaportas, P., and Forster, J. (2003). Bayesian variable and link determination for generalised linear models. *Journal of Statistical Planning and Inference* **111**, 165–180.

Ntzoufras, I., Forster, J. J., and Dellaportas, P. (2000). Stochastic search variable selection for log-linear models. *Journal of Statistical Computation and Simulation* **68**, 23–37.

Pinheiro, J. C. and Bates, D. M. (2000). *Mixed-Effects Models in S and S-PLUS.* Springer-Verlag, New York.

Raftery, A. (1996). Approximate Bayes factors and accounting for model uncertainty in generalised linear models. *Biometrika* **83**, 251–266.

Randall, D. (2006). Rapidly mixing markov chains with applications in computer science and physics. *Computing in Science and Engineering* **8**, 30–41.

Raudenbush, S. W., Yang, M. L., and Yosef, M. (2000). Maximum likelihood for generalized linear models with nested random effects via high-order, multivariate Laplace approximation. *Journal of Computational and Graphical Statistics* **9**, 1, 141–157.

Rodriguez, G. and Goldman, N. (1995). An assessment of estimation procedures for multilevel models with binary responses. *Journal of Royal Statistical Society, Series A,* **158**, 73–89.

Rodriguez, G. and Goldman, N. (2001). Improved estimation procedures for multilevel models with binary response: A case-study. *Journal of the Royal Statistical Society, Series A,* **164**, 2, 339–355.

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.

Sahu, S. K. and Gelfand, A. E. (1999). Identifiability, improper priors, and Gibbs sampling for generalized linear models. *Journal of the American Statistical Association* **94**, 445, 247–254.

Scarpa, B. and Dunson, D. (2007). Bayesian methods for searching for optimal rules for timing of intercourse to achieve pregnancy. *Statistics in Medicine* **26**, 1920–1936.

Schober, A., Bernhagen, J., and Weber, C. (2008). Chemokine-like functions of MIF in atherosclerosis. *Journal of Molecular Medicine* To appear.

Seedorf, U., Wiebusch, H., Muntoni, S., Christensen, N., Skovby, F., Nickel, V., Roskos, M., Funke, H., Ose, L., and Assmann, G. (1995). A novel variant of lysosomal acid lipase (Leu335→ Pro) associated with acid lipase deficiency and cholesterol ester storage disease. *Arteriosclerosis, Thrombosis, and Vascular Biology* **15**, 773–778.

Seo, D., Wang, T., Dressman, H., Herderick, E., Iversen, E., Dong, C., Vata, K., Milano, C., Nevins, C., Pittman, J., West, M., and Goldschmidt-Clermont, P. (2004). Gene expression phenotypes of atherosclerosis. *Arteriosclerosis, Thrombosis and Vascular Biology* **24**, 1922–1927.

Seo, D. M., Goldschmidt-Clermont, P. J., and West, M. (2007). Of mice and men: Sparse statistical modelling in cardiovascular genomics. *The Annals of Applied Statistics* **1**, 152–178.

Silvapulle, M. J. (1981). On the existence of maximum likelihood estimates for the binomial response models. *Journal of the Royal Statistical Society, Series B (Methodological),* **43**, 310–313.

Srebro, N. and Jaakkola, T. (2003). Weighted low-rank approximations. In T. Fawcett and N. Mishra, eds., *Proceedings of the Twentieth International Conference on Machine Learning (ICML)*, 720–727. The AAAI Press, Menlo Park, California.

Stein, C. (1956). Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. In J. Neyman, ed., *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, 197–206. University of California Press.

Tierney, L. and Kadane, J. (1986). Accurate approximations for posterior moments and marginal likelihoods. *Journal of the American Statistical Association* **81**, 82–86.

Verdinelli, I. and Wasserman, L. (1995). Computing Bayes factors using a generalization of the Savage-Dickey density ratio. *Journal of the American Statistical Association* **90**, 430.

Wang, B. and Titterington, D. M. (2005). Inadequacy of interval estimates corresponding to variational Bayesian approximations. In R. Cowell and Z. Ghahramani, eds., *10th International workshop on Artificial Intelligence and Statistics*, 373–380. The Society for Artificial Intelligence and Statistics.

Wang, X. and George, E. (2007). Adaptive Bayesian criteria in variable selection for generalized linear models. *Statistica Sinica* **17**, 667–690.

Wermuth, N. (1976). Model search among multiplicative models. *Biometrics* **32**, 253–263.

Whittaker, J. (1990). *Graphical Models in Applied Multivariate Statistics*. John Wiley and Sons, Chichester, United Kingdom.

Wilcox, A., Dunson, D., and Baird, D. (2000). The timing of the "fertile window" in the menstrual cycle: day specific estimates from a prospective study. *British Medical Journal* **321**, 1259–1262.

Wold, H. D. A. (1954). Causality and econometrics. *Econometrica* **22**, 162–177.

Wong, F., Carter, C., and Kohn, R. (2003). Efficient estimation of covariance selection models. *Biometrika* **90**, 809–830.

Wright, S. (1921). Correlation and causation. *Journal of Agricultural Research* **20**, 557–585.

Wright, S. (1923). The theory of path coefficients: a reply to Niles' criticism. *Genetics* **8**, 239–255.

Wright, S. (1934). The method of path coefficients. *Annals of Mathematical Statistics* **5**, 161–215.

Zellner, A. (1986). On assessing prior distributions and Bayesian regression analysis with g-prior distributions. In P. Goel and A. Zellner, eds., *Bayesian Inference and Decision Techniques: Essays in Honors of Bruno de Finetti*, 223–243. Amsterdam: North Holland.

Zschenker, O., Illies, T., and Ameis, D. (2006). Overexpression of lysosomal acid lipase and other proteins in atherosclerosis. *Journal of Biochemistry* **140**, 23–38.

Biography

Liang Zhang was born in 1983 in Wuhu, Anhui, P. R. CHINA. He went to the Special Class for the Gifted Young of the University of Science and Technology of China to study mathematics in 2000, and received his bachelor's degree in science in 2004. Afterwards, Liang came to Department of Statistical Science at Duke University to be a Ph.D student, and a Bayesian. He became a Ph.D candidate in 2006. Liang's research interests are graphical model search methods, Bayesian covariance selection, marginal likelihood approximations in generalized linear models, and scalable statistical inference and forecasting with massive data.