DEPARTMENT OF STATISTICS
University of Wisconsin
1300 University Ave.
Madison, WI 53706

TECHNICAL REPORT NO. 1171
July 30, 2012

Multivariate Bernoulli Distribution Models

Bin Dai[1]
Department of Statistics, University of Wisconsin, Madison WI

[1]Research supported in part by NIH Grant EY09946 and NSF Grant DMS 0604572.

MULTIVARIATE BERNOULLI DISTRIBUTION MODELS

by

Bin Dai

A dissertation submitted in partial fulfillment of

the requirements for the degree of

Doctor of Philosophy

(Statistics)

at the

UNIVERSITY OF WISCONSIN–MADISON

2012

Date of final oral examination: 07/24/12

The dissertation is approved by the following Final Oral Committee members:

Grace Wahba, IJ Schoenberg-Hilldale Professor, Statistics
Stephen Wright, Professor, Computer Sciences
Sündüz Keles, Associate Professor, Statistics
Peter Z.G. Qian, Associate Professor, Statistics
Sijian Wang, Assistant Professor, Statistics

Acknowledgments

First and most importantly, I would like to express my deepest gratitude toward my advisor Professor Grace Wahba. Her guidance and encouragement through my PhD study into various statistical machine learning methods is the key factor to the success of this dissertation. Grace is a brilliant and passionate statistician, and her insightful ideas in both statistical theories and applications inspire me. It is a great honor and privilege to have the opportunity to work closely and learn from her.

This work is also the product of collaboration with a number of researchers. In particular, I would like to thank Professor Stephen Wright from Department of Computer Science for his guidance in computation. Without him, the proposed models in this thesis would not be solved with efficient optimization techniques. In addition, I am grateful to other professors in my thesis committee. I benefit from Professor Sündüz Keles' expertise in biostatistics and her valuable ideas in the Thursday group. On the other hand, Professor Peter Qian and Sijian Wang raised questions with deep perception and helped greatly improve the thesis. I am also greatly influenced by Professor Karl Rohe and Xinwei Deng for their improving suggestion to this work.

I want to thank Xiwen Ma and Shilin Ding for their effort on our collaborative projects. The current and past fellows graduate students in the Thursday group helped me in various ways: Weiliang Shi, Héctor Corrada Bravo, Pei-fen Kuan, Kevin Eng, Xin Li, Zhigeng Geng, Tai Qin, Jing Kong, Xin Zeng, Dongjun Chung and Chen Zuo. The synergies gained from discussions during the Thursday group meetings were beneficial to my PhD research study and made my life at Madison much more enjoyable.

Finally, I would like to thank the Department of Statistics at University of Wisconsin-Madison. My solid background is built from receiving rigid training here in both theories and applications of statistics, which are not only beneficial to my research but also priceless for my career. What's more, support and understanding from my dear parents Jifang Wang and Guanying Cui, and my girlfriend Xinxin Yu all these years is paramount for my success. I own them all I have and all I am about to have.

DISCARD THIS PAGE

Contents

DISCARD THIS PAGE

List of Tables

DISCARD THIS PAGE

List of Figures

MULTIVARIATE BERNOULLI DISTRIBUTION MODELS

Bin Dai

Under the supervision of Professor Grace Wahba

At the University of Wisconsin-Madison

This thesis is devoted to the study of graphs with binary nodes and when there are covariate effects on both the nodes, edges and cliques level. The models proposed deal with data with multiple 0-1 coded outcomes and there are known predictor variables having profound influence not only on the nodes, but also on the edges and cliques.

Firstly, in Chapter 1 we consider the multivariate Bernoulli distribution as a model to estimate the structure of the binary graphs. This distribution is discussed in the framework of the exponential family, and its statistical properties regarding independence of the nodes are demonstrated. Importantly the multivariate Bernoulli logistic model is developed under generalized linear model theory by utilizing the canonical link function in order to include covariate information on the nodes, edges and cliques. Furthermore, the model is extended to the framework of smoothing spline ANOVA to enable estimation of non-linear effects of the predictor variables on the graph.

What's more, the multivariate Bernoulli LASSO model is discussed in Chapter 2 to incorporate the variable selection techniques in the inference of graph structure. The accelerated block coordinate relaxation optimization approach is applied to the problem with the ability to handle very large scale real-world data. The tuning of the smoothing parameters in the model is studied and various different

approaches are compared. Both numerical and real-world examples are examined to demonstrate the power and efficiency of the model and the algorithm.

Further, the multivariate Bernoulli logistic model is extended to the paradigm of mixed-effects model in Chapter 3. The model is more flexible to handle more complex variance-covariance structure. However, as the likelihood function involves non-analytical integral, the Laplacian approximation approach is applied. The model is implemented to analyze real-world problem.

Finally, Chapter 4 introduces two `R` packages `MVB` and `oem`. `MVB` is designed for the various multivariate Bernoulli models introduced in this thesis and all the numerical examples are implemented in this package. On the other hand, `oem` is developed to implement a new algorithm to optimize penalized least squares problems.

Grace Wahba

Chapter 1

Multivariate Bernoulli Distribution and Logistic Models

1.1 Introduction

Undirected graphical models have been proved to be useful in a variety of applications in statistical machine learning. Statisticians and computer scientists devoted resources to studies in graphs with nodes representing both continuous and discrete variables. Such models consider a graph $G = (V, E)$, whose nodes set V represents K random variables $Y_1, Y_2, \ldots, Y_K$ connected or disconnected defined by the undirected edges set E. This formulation allows pairwise relationships among the nodes to be described in terms of edges, which in statistics are defined as correlations. The graph structure can thus be determined under the independence assumptions on the random variables. Specifically, variables Y_i and Y_j are conditionally independent given all other variables if the associated nodes are not

linked by an edge. Two important types of graphical models are the Gaussian model, where the K variables are assumed to follow a joint multivariate Gaussian distribution, and the discrete Markovian model, which captures the relationships between categorical variables.

However, the assumption that only the pairwise correlations among the variables are considered may not be sufficient. When the joint distribution of the nodes is multivariate Gaussian, the graph structure can be directly inferred from the inverse of the covariance matrix of the random variables and in recent years, a large body of literature has emerged in this area for high dimensional data. Researchers mainly focus on different sparse structure of the graphs or, in other words, the covariance matrix for high-dimensional observations. For example, Meinshausen and Buhlmann (2006) proposes a consistent approach based on LASSO from Tibshirani (1996) to model the sparsity of the graph. Due to the fact that the Gaussian distribution can be uniquely determined by the means and covariance matrix, it is valid to consider only the pairwise correlations, but this may not be true for some other distributions. The multivariate Bernoulli distribution discussed in Whittaker (1990), which will be studied in Section 1.3, has a probability density function involving terms representing third and higher order moments of the random variables, which are also referred to as clique effects. To alleviate the complexity of the graph, the so-called Ising model borrowed from physics gained popularity in the machine learning literature. Wainwright and Jordan (2008) introduces several important discrete graphical models including the Ising model and Banerjee et al. (2008) discusses a framework to infer sparse graph structure with both Gaussian and binary variables. In this chapter, higher than second order interactions among

a group of binary random variables are studied in detail.

What's more, in some real applications, people are not only interested in the graph structure but also want to include predictor variables that potentially have influence on the nodes in the graph. Gao et al. (2001) considers a multivariate Bernoulli model which uses a smoothing spline ANOVA model to replace the linear predictor (McCullagh and Nelder (1989)) for main effects on the nodes, but set the second and higher order interactions between the nodes as constants. Higher order outcomes with hierarchical structure assumptions on the graph involving predictor variables are studied in Ding et al. (2011).

This chapter aims at building a unified framework of a generalized linear model for the multivariate Bernoulli distribution which includes both higher order interactions among the nodes and covariate information. The remainder is organized as follows. Section 1.2 starts from the simplest multivariate Bernoulli distribution, the so-called bivariate Bernoulli distribution, where there are only two nodes in the graph. The mathematical formulation and statistical properties of the multivariate Bernoulli distribution are addressed in Section 1.3. Section 1.4 serves to get a better understanding of the differences and similarities of the multivariate Bernoulli distribution with the Ising and the multivariate Gaussian models. Section 1.5 extends the model to include covariate information on the nodes, edges and cliques, and discusses parameter estimation, optimization and associated problems in the resulting multivariate Bernoulli logistic model.

1.2 Bivariate Bernoulli Distribution

To start from the simplest case, we extend the widely used univariate Bernoulli distribution to two dimensions in this section and the more complicated multivariate Bernoulli distribution is explored in Section 1.3. The Bernoulli random variable Y, is one with binary outcomes chosen from $\{0, 1\}$ and its probability density function is

$$f_Y(y) = p^y(1-p)^{1-y}.$$

Next, consider the bivariate Bernoulli random vector (Y_1, Y_2), which takes values from $(0, 0)$, $(0, 1)$, $(1, 0)$ and $(1, 1)$ in the Cartesian product space $\{0, 1\}^2 = \{0, 1\} \times \{0, 1\}$. Denote $p_{ij} = P(Y_1 = i, Y_2 = j)$, $i, j = 0, 1$, then its probability density function can be written as

$$\begin{aligned} P(Y = y) &= p(y_1, y_2) \\ &= p_{11}^{y_1 y_2} p_{10}^{y_1(1-y_2)} p_{01}^{(1-y_1)y_2} p_{00}^{(1-y_1)(1-y_2)} \\ &= \exp\left\{\log(p_{00}) + y_1 \log\left(\frac{p_{10}}{p_{00}}\right) + y_2 \log\left(\frac{p_{01}}{p_{00}}\right) + y_1 y_2 \log\left(\frac{p_{11}p_{00}}{p_{10}p_{01}}\right)\right\}, \end{aligned} \tag{1.1}$$

where the side condition $p_{00} + p_{10} + p_{01} + p_{11} = 1$ holds to ensure it is a valid probability density function.

To simplify the notation, define the natural parameters f's from general param-

eters as follows:

$$f^1 = \log\left(\frac{p_{10}}{p_{00}}\right), \tag{1.2}$$

$$f^2 = \log\left(\frac{p_{01}}{p_{00}}\right), \tag{1.3}$$

$$f^{12} = \log\left(\frac{p_{11}p_{00}}{p_{10}p_{01}}\right), \tag{1.4}$$

and it is not hard to verify the inverse of the above formula

$$p_{00} = \frac{1}{1+\exp(f^1)+\exp(f^2)+\exp(f^1+f^2+f^{12})}, \tag{1.5}$$

$$p_{10} = \frac{\exp(f^1)}{1+\exp(f^1)+\exp(f^2)+\exp(f^1+f^2+f^{12})}, \tag{1.6}$$

$$p_{01} = \frac{\exp(f^2)}{1+\exp(f^1)+\exp(f^2)+\exp(f^1+f^2+f^{12})}, \tag{1.7}$$

$$p_{11} = \frac{\exp(f^1+f^2+f^{12})}{1+\exp(f^1)+\exp(f^2)+\exp(f^1+f^2+f^{12})}. \tag{1.8}$$

Figure 1.1 illustrates the effects we consider in a bivariate Bernoulli graph. There are two nodes in the graph each with binary 0-1 code. The main effects f^1 and f^2 come with the nodes and the interaction f^{12} represents the inter-connectivity of the two nodes.

Here the original density function (1.1) can be viewed as a member of the exponential family, and represented in a log-linear formulation as:

$$P(Y=y) = \exp\left\{\log(p_{00}) + y_1 f^1 + y_2 f^2 + y_1 y_2 f^{12}\right\}. \tag{1.9}$$

Consider the marginal and conditional distribution of Y_1 in the random vector

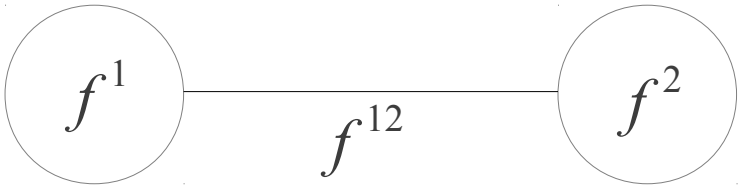

Figure 1.1: The graph example of the bivariate Bernoulli graph.

(Y_1, Y_2), we have

Proposition 1.1. *The marginal distribution of Y_1 in a bivariate Bernoulli vector (Y_1, Y_2) following density function* (1.1) *is univariate Bernoulli with density*

$$P(Y_1 = y_1) = (p_{10} + p_{11})^{y_1}(p_{00} + p_{01})^{(1-y_1)}. \tag{1.10}$$

What's more, the conditional distribution of Y_1 given Y_2 is also univariate Bernoulli with density

$$P(Y_1 = y_1|Y_2 = y_2) = \left(\frac{p(1, y_2)}{p(1, y_2) + p(0, y_2)}\right)^{y_1} \left(\frac{p(0, y_2)}{p(1, y_2) + p(0, y_2)}\right)^{1-y_1}. \tag{1.11}$$

The proposition implies that the bivariate Bernoulli distribution is similar to the bivariate Gaussian distribution, in that both the marginal and conditional distributions are still Bernoulli distributed. On the other hand, it is also important to

know under what conditions the two random variables Y_1 and Y_2 are independent.

Lemma 1.2. *The components of the bivariate Bernoulli random vector* (Y_1, Y_2) *are independent if and only if* f^{12} *in* (1.9) *and defined in* (1.4) *is zero.*

Lemma 1.2 is a special case for Theorem 1.4 in Section 1.3, and the proof is attached in **Appendix**. It is not hard to see from the log-linear formulation (1.9) that when $f^{12} = 0$, the probability density function of the bivariate Bernoulli is separable in y_1 and y_2 so the lemma holds. In addition, a simple calculation of covariance between Y_1 and Y_2 gives

$$\begin{aligned} \text{cov}(Y_1, Y_2) &= E[Y_1 - (p_{11} + p_{10})][Y_2 - (p_{11} + p_{01})] \\ &= p_{11}p_{00} - p_{01}p_{10}, \end{aligned} \tag{1.12}$$

and using (1.4), the disappearance of f^{12} indicates that the correlation between Y_1 and Y_2 is null. When dealing with the multivariate Gaussian distribution, the uncorrelated random variables are independent as well and Section 1.3 below shows uncorrelatedness and independence is also equivalent for the multivariate Bernoulli distribution.

The importance of Lemma 1.2 was explored in Whittaker (1990) where it was referred to as proposition 2.4.1. The importance of f^{12} (denoted as *u-terms*) is discussed and called *cross-product ratio* between Y_1 and Y_2. The same quantity is actually *log odds* described for the univariate case in McCullagh and Nelder (1989) and for the multivariate case in Ma (2010).

1.3 Formulation and Statistical Properties

Probability Density Function

As discussed in Section 1.2, the two dimensional Bernoulli distribution possesses good properties analogous to the Gaussian distribution. This section is to extend it to high dimensions and construct the so-called multivariate Bernoulli distribution.

Let $Y = (Y_1, Y_2, \ldots, Y_K)$ be a K-dimensional random vector of possibly correlated Bernoulli random variables (binary outcomes) and let $y = (y_1, \ldots, y_K)$ be a realization of Y. The most general form $p(y_1, \ldots, y_K)$ of the joint probability density is

$$\begin{aligned} P(Y_1 = y_1, Y_2 = y_2, \ldots, Y_K = y_K) &= p(y_1, y_2, \ldots, y_K) \\ &= p(0, 0, \ldots, 0)^{[\prod_{j=1}^{K}(1-y_j)]} \\ &\quad p(1, 0, \ldots, 0)^{[y_1 \prod_{j=2}^{K}(1-y_j)]} \\ &\quad p(0, 1, \ldots, 0)^{[(1-y_1)y_2 \prod_{j=3}^{K}(1-y_j)]} \\ &\quad \ldots p(1, 1, \ldots, 1)^{[\prod_{j=1}^{K} y_j]}, \end{aligned}$$

or in short

$$p(y) = p_{0,0,\ldots,0}^{[\prod_{j=1}^{K}(1-y_j)]} p_{1,0,\ldots,0}^{[y_1 \prod_{j=2}^{K}(1-y_j)]} p_{0,1,\ldots,0}^{[(1-y_1)y_2 \prod_{j=3}^{K}(1-y_j)]} \ldots p_{1,1,\ldots,1}^{[\prod_{j=1}^{K} y_j]}. \tag{1.13}$$

To simplify the notation, denote the quantity S to be

$$S^{j_1 j_2 \cdots j_r} = \sum_{1 \le s \le r} f^{j_s} + \sum_{1 \le s < t \le r} f^{j_s j_t} + \cdots + f^{j_1 j_2 \cdots j_r}, \tag{1.14}$$

and in the bivariate Bernoulli case $S^{12} = f^1 + f^2 + f^{12}$. To eliminate the product in the tedious exponent of (1.13), define the interaction function B

$$B^{j_1 j_2 \cdots j_r}(y) = y_{j_1} y_{j_2} \ldots y_{j_r}, \tag{1.15}$$

so correspondingly in the bivariate Bernoulli distribution for the realization (y_1, y_2) of random vector (Y_1, Y_2), the interaction function of order 2 is $B^{12}(y) = y_1 y_2$. This is the only order two interaction for the bivariate case. In general, there are $\binom{K}{2} = \frac{K(K-1)}{2}$ different second interactions among the binary components of the multivariate Bernoulli random vector of length K.

The log-linear formulation of the multivariate Bernoulli distribution induced from (1.13) is

$$\begin{aligned} l(y, \mathbf{f}) &= -\log[p(y)] \\ &= -\left[\sum_{r=1}^{K} \left(\sum_{1 \le j_1 < j_2 < \ldots < j_r \le K} f^{j_1 j_2 \cdots j_r} B^{j_1 j_2 \cdots j_r}(y) \right) - b(\mathbf{f}) \right], \end{aligned} \tag{1.16}$$

where $\mathbf{f} = (f^1, f^2, \ldots, f^{12\ldots K})^T$ is the vector of the natural parameters for the multivariate Bernoulli distribution, and the normalizing factor $b(\mathbf{f})$, or sometimes referred to as partition function in Computer Sciences literature, is defined as

$$b(\mathbf{f}) = \log \sum_{r=1}^{K} \left[1 + \left(\sum_{1 \le j_1 < j_2 < \ldots < j_r \le K} \exp[S^{j_1 j_2 \cdots j_r}] \right) \right]. \tag{1.17}$$

As a member of the exponential distribution family, the multivariate Bernoulli distribution has the fundamental 'link' between the natural and general parameters.

Lemma 1.3. *(Parameters Transformation) For the multivariate Bernoulli model, the general parameters and natural parameters have the following relationship.*

$$\exp(f^{j_1 j_2 \cdots j_r}) =$$

$$\frac{\prod p(\textit{even \# zeros among } j_1, j_2 \ldots, j_r \textit{ components and other components are all zero})}{\prod p(\textit{odd \# zeros among } j_1, j_2 \ldots, j_r \textit{ components and other components are all zero})},$$

where # refers to the number of zeros among the superscript $y_{j_1} \ldots y_{j_r}$ of f. In addition,

$$\exp(S^{j_1 j_2 \cdots j_r}) = \frac{p(j_1, j_2 \ldots, j_r \textit{ positions are one, others are zero})}{p(0, 0, \ldots, 0)} \tag{1.18}$$

and conversely the general parameters can be represented by the natural parameters

$$p(j_1, j_2 \ldots, j_r \textit{ positions are one, others are zero}) = \frac{\exp(S^{j_1 j_2 \cdots j_r})}{\exp\left(b(\mathbf{f})\right)} \tag{1.19}$$

Based on the log-linear formulation (1.16) and the fact that the multivariate Bernoulli distribution is a member of the exponential family, the interactions functions $B^{j_1 j_2 \cdots j_r}(y)$ for all combinations $j_1 j_2 \ldots j_r$ define the sufficient statistics. In addition, the log-partition function $b(\mathbf{f})$ as in (1.17) is useful to determine the expectation and variance of the sufficient statistics to be addressed in later sections.

Figure 1.2 displays a graph with three binary nodes. There are three main effects f^1, f^2 and f^3 born with the nodes and the second order interactions f^{12},

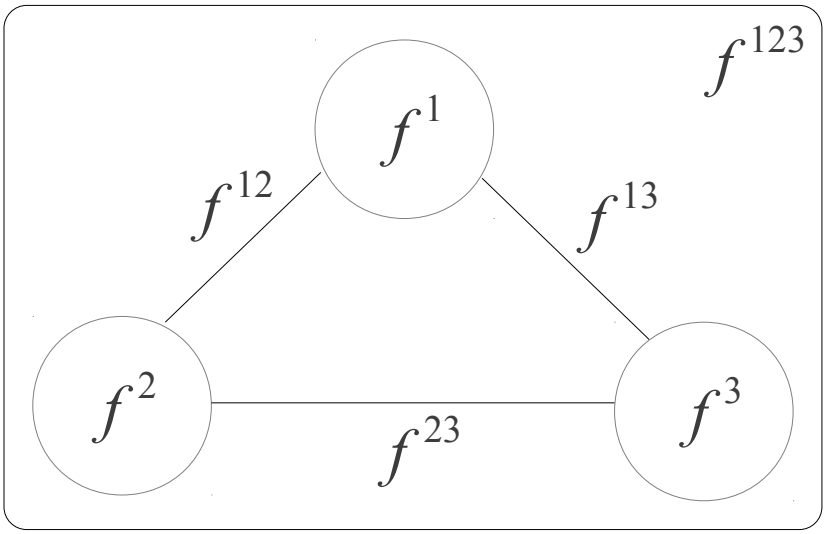

Figure 1.2: The graph example of the trivariate Bernoulli distribution.

f^{13} and f^{23} representing the connectivity of the nodes. Modern graph theory are particularly interested in these interactions such as Ising model (Ising, 1925) to be compared in the later sections. Moreover, the third order interaction f^{123} as the rectangle in Figure 1.2 studying the effect of all three nodes is also of interest in this thesis. When there are more nodes in the graph, higher order interactions can come into play in model (1.16).

Independence, Marginal and Conditional Distributions

One of the most important statistical properties for the multivariate Gaussian distribution is the equivalence of independence and uncorrelatedness. As a natural multivariate extension of the univariate Bernoulli distribution, it is of great interest to explore independence among components of the multivariate Bernoulli distribution and it is the topic for this section.

The independence of components of a random vector is determined by separability of coordinates in its probability density function but it is hard to get directly from (1.13). However, based on the relationship between the natural parameters and the outcome in the log-linear formulation (1.16), the independence theorem of the distribution can be derived as follows with proof deferred to **Appendix**.

Theorem 1.4. *(Independence of Bernoulli outcomes) For the multivariate Bernoulli distribution, the random vector $Y = (Y_1, \ldots, Y_K)$ is independent element-wise if and only if*

$$f^{j_1 j_2 \cdots j_r} = 0, \;\; \forall\, 1 \leq j_1 < j_2 < \ldots < j_r \leq K, \; r \geq 2. \tag{1.20}$$

In addition, the condition in equation (1.20) *can be equivalently written as*

$$S^{j_1 j_2 \cdots j_r} = \sum_{k=1}^{r} f^{j_k}, \;\; \forall\, r \geq 2 \tag{1.21}$$

The importance of the theorem is to link the independence of components of a random vector following the multivariate Bernoulli distribution to the natural parameters. Notice that to ensure all the single random variable to be independent

of all the others is a strong assertion and in graphical models, researchers are more interested in the independence of two groups of nodes, so we have the following theorem

Theorem 1.5. *(Independence of Groups) For random vector* $Y = (Y_1, \ldots, Y_K)$ *following the multivariate Bernoulli distribution, without loss of generality, suppose two blocks of nodes* $Y' = (Y_1, Y_2, \ldots, Y_r)$, $Y'' = (Y_{r+1}, Y_{r+2}, \ldots, Y_s)$ *with* $1 \leq r < s \leq K$, *and denote index set* $\tau_1 = \{1, 2, \ldots, r\}$ *and* $\tau_2 = \{r+1, r+2, \ldots, s\}$. *Then* Y' *and* Y'' *are independent if and only if*

$$f^{\tau} = 0, \quad \forall\, \tau \cap \tau_1 \neq \emptyset \text{ and } \tau \cap \tau_2 \neq \emptyset \tag{1.22}$$

The proof of Theorem 1.5 is also deferred to **Appendix**. The theorem delivers the message that the two groups of binary nodes in a graph are independent if all the natural parameters f's corresponding to the index sets that include indices from both groups disappear. In other words, the two groups of nodes can be perfectly separated without any edges or cliques linking them.

Furthermore, analogous to the multivariate Gaussian distribution, researchers are interested in statistical distributions of marginal and conditional distributions for the multivariate Bernoulli distribution. Likewise, the multivariate Bernoulli distribution maintains the good property that both the marginal and conditional distributions are still multivariate Bernoulli as stated in the following proposition.

Proposition 1.6. *The marginal distribution of the random vector* $(Y_1, \ldots, Y_K)$ *which follows multivariate Bernoulli distribution with density function* (1.13) *to any order is*

*still a **multivariate Bernoulli** with density*

$$P(Y_1 = y_1, Y_2 = y_2, \ldots, Y_r = y_r) = \sum_{y_{r+1}} \ldots \sum_{y_K} p(y_1, \ldots, y_K) \tag{1.23}$$

for some $r < K$.

*What's more, the conditional distribution of $(Y_1, Y_2, \ldots, Y_r)$ given the rest is also **multivariate Bernoulli** with density*

$$P(Y_1 = y_1 \ldots, Y_r = y_r | Y_{r+1} = y_{r+1}, \ldots, Y_K = y_K) = \frac{p(y_1, \ldots, y_K)}{p(y_{r+1}, \ldots, y_K)}. \tag{1.24}$$

Moment Generating Functions

The moment generating function for the multivariate Bernoulli distribution is useful when dealing with moments and proof of Theorem 1.4.

$$\begin{aligned} \psi(\mu_1, \mu_2, \ldots, \mu_K) &= E\left[\exp\left(\mu_1 Y_1 + \mu_2 Y_2 + \cdots + \mu_K Y_K\right)\right] \\ &= p_{00\ldots0} e^0 + p_{10\ldots0} e^{\mu_1} + \cdots + p_{11\ldots1} e^{\mu_1 + \mu_2 + \cdots + \mu_K} \\ &= \sum_{r=1}^{K} \sum_{j_1 \le j_2 \le \ldots \le j_r} \frac{\exp[S^{j_1 j_2 \ldots j_r}]}{\exp[b(\mathbf{f})]} \exp\left[\sum_{k=1}^{r} \mu_{j_k}\right]. \end{aligned} \tag{1.25}$$

Hence, from the formula the moment generating function is solely determined by the S functions, which are the transformation of the natural parameters f's.

Gradient and Hessian

As a member of the exponential family, the gradient and Hessian (Fisher information) are closely related to the mean and covariance of the random vector

$(Y_1, Y_2, \ldots, Y_K)$. Therefore, they are important in statistics but also crucial for model inference when the proper optimization problem is established. To examine the formulation of gradient and Hessian for the logarithm probability density function of the multivariate Bernoulli distribution (1.13), let us define some notations.

Denote $\mathcal{T}$ to be the set of all possible superscripts of the f's including the null superscript with $f^{\emptyset} = 0$, so it has 2^K elements. In other words, $\mathcal{T}$ is the power set of indices $\{1, 2, \ldots, K\}$. Let $|\cdot|$ be the cardinality of a set then $|\mathcal{T}| = 2^K$. We can define the relation subset $\subset$ for $\tau_1, \tau_2 \in \mathcal{T}$ as follows.

Definition 1.7. *For any two superscripts $\tau_1 = \{j_1, j_2, \ldots, j_r\}$ such that $\tau_1 \in \mathcal{T}$ and $\tau_2 = \{k_1, k_2, \ldots, k_s\}$ with $\tau_2 \in \mathcal{T}$ and $r \leq s$, we say that $\tau_1 \subseteq \tau_2$ if for any $j \in \tau_1$, there is a $k \in \tau_2$ such that $j = k$.*

Based on the definition, the S's in (1.14) can be reformulated as

$$S^{\tau} = \sum_{\tau_0 \subseteq \tau} f^{\tau_0}, \tag{1.26}$$

specifically, $S^{\emptyset} = 0$. Consider the gradient of the log-linear form (1.16) with respect to the f's, for any $\tau \in \mathcal{T}$,

$$\begin{aligned} \frac{\partial l(y, \mathbf{f})}{\partial f^{\tau}} &= -B^{\tau}(y) + \frac{\partial b(\mathbf{f})}{\partial f^{\tau}} \\ &= -B^{\tau}(y) + \frac{\sum_{\tau_0 \supseteq \tau} \exp[S^{\tau_0}]}{b(\mathbf{f})}. \end{aligned} \tag{1.27}$$

The derivation of the first partial derivative of b in equation (1.17) with respect

to f^τ in (1.27) is

$$\begin{aligned}
\frac{\partial b(\mathbf{f})}{\partial f^\tau} &= \frac{1}{\exp[b(\mathbf{f})]} \cdot \frac{\partial \exp[b(\mathbf{f})]}{\partial f^\tau} \\
&= \frac{1}{\exp[b(\mathbf{f})]} \cdot \frac{\partial \sum_{\tau_0 \in \mathcal{T}} \exp[S^{\tau_0}]}{\partial f^\tau} \\
&= \frac{\sum_{\tau_0 \supseteq \tau} \exp[S^{\tau_0}]}{\exp[b(\mathbf{f})]} \\
&= E[B^\tau(Y)],
\end{aligned} \tag{1.28}$$

and the result can also be derived from the moment generating function (1.25) by taking derivatives with respect to the μ's.

A simple example of (1.27) in the bivariate Bernoulli distribution (1.9) is

$$\frac{\partial l(y, \mathbf{f})}{\partial f^1} = -y_1 + \frac{\exp(f^1) + \exp(S^{12})}{b(\mathbf{f})},$$

Further, the general formula for the second order derivative of (1.16) with respect to any two natural parameters f^{τ_1} and f^{τ_2} is

$$\begin{aligned}
\frac{\partial^2 l(y, f)}{\partial f^{\tau_1} \partial f^{\tau_2}} &= \frac{\partial^2 b(\mathbf{f})}{\partial f^{\tau_1} \partial f^{\tau_2}} \\
&= \frac{\partial}{\partial f^{\tau_1}} \left(\frac{\sum_{\tau_0 \supseteq \tau_2} \exp[S^{\tau_0}]}{\exp[b(\mathbf{f})]} \right) \\
&= \frac{\sum_{\tau_0 \supseteq \tau_1,\, \tau_0 \supseteq \tau_2} \exp[S^{\tau_0}] \exp[b(\mathbf{f})] - \sum_{\tau_0 \supseteq \tau_1} \exp[S^{\tau_0}] \sum_{\tau_0 \supseteq \tau_2} \exp[S^{\tau_0}]}{\exp[2b(\mathbf{f})]} \\
&= \text{cov}\ \left(B^{\tau_1}(Y), B^{\tau_2}(Y) \right).
\end{aligned} \tag{1.29}$$

In the bivariate Bernoulli distribution,

$$\frac{\partial^2 l(y,f)}{\partial f^1 \partial f^2} = \frac{\exp[S^{12}]\exp[b(\mathbf{f})] - (\exp[f^1]+\exp[S^{12}])(\exp[f^2]+\exp[S^{12}])}{\exp[2b(\mathbf{f})]}$$

1.4 The Ising and the Multivariate Gaussian Models

As mentioned in Section 1.1, the Ising and the multivariate Gaussian distributions are two main tools to study undirected graphical models, and this section is to compare the multivariate Bernoulli model introduced in Section 1.3 with these two popular models.

The Ising Model

The Ising model, which originated from Ising (1925), becomes popular when the graph structure is of interest with nodes taking binary values. The log-linear density of the random vector $(Y_1, \ldots, Y_K)$ is

$$\log[f(Y_1,\ldots,Y_K)] = \sum_{j=1}^{K} \theta_{j,j} Y_j + \sum_{1\le j<j'\le K} \theta_{j,j'} Y_j Y_{j'} - \log[Z(\Theta)], \tag{1.30}$$

where $\Theta = (\theta_{j,j'})_{K\times K}$ is a symmetric matrix specifying the network structure, but it is not necessarily positive semi-definite. The log-partition function $Z(\Theta)$ is defined as

$$Z(\Theta) = \sum_{Y_j\in\{0,1\}, 1\le j\le K} \exp\left(\sum_{j=1}^{K} \theta_{j,j} Y_j + \sum_{1\le j<j'\le K} \theta_{j,j'} Y_j Y_{j'}\right), \tag{1.31}$$

and notice that it is not related to Y_j due to the summation over all possible values of Y_j for $j = 1, 2, \ldots, K$.

It is not hard to see that the multivariate Bernoulli is an extension of the Ising model, which assumes all $S^\tau = 0$ for any τ such that $|\tau| > 2$ and $\theta_{j,j'} = S^{jj'}$. In other words, in the Ising model, only pairwise interactions are considered. Ravikumar et al. (2010) pointed out that higher order interactions, which are referred to as clique effects in this chapter, can be converted to pairwise ones through the introduction of additional variables and thus retain the Markovian structure of the network defined in Wainwright and Jordan (2008).

The Multivariate Gaussian Model

When continuous nodes are considered in a graphical model, the multivariate Gaussian distribution is important since, similar to the Ising model, it only considers interactions up to order two. The log-linear formulation is

$$\log[f(Y_1, \ldots, Y_K)] = \left(-\frac{1}{2}(Y-\mu)^T\Sigma(Y-\mu)\right) - \log[Z(\Sigma)], \tag{1.32}$$

where $Z(\Sigma)$ is the normalizing factor which only depends on the covariance matrix Σ of the nodes in the graph.

Comparison of Different Graphical Models

The multivariate Bernoulli (1.16), Ising (1.30) and the multivariate Gaussian (1.32) are three different kinds of graphical models and they share many similarities

1. All of them are members of the exponential family.

2. Uncorrelatedness and independence are equivalent.

3. Conditional and marginal distributions maintain the same structure.

However, some differences do exist. the multivariate Bernoulli and the Ising models both serve as tools to model graph with binary nodes, and are certainly different from the multivariate Gaussian model which formulates continuous variables. In addition, the multivariate Bernoulli specifies clique effects among nodes whereas the Ising model simplifies to deal with only pairwise interactions and the multivariate Gaussian distribution essentially is uniquely determined by its mean and covariance structure, which is also based on first and second order moments. Table 1.1 illustrates the number of parameters needed to uniquely determine the distribution for these models as the number of nodes K in the graph increases.

Graph dimension	multivariate Bernoulli	Ising	multivariate Gaussian
1	1	1	2
2	3	3	5
3	7	6	9
...	...	...	...
K	$2^K - 1$	$\frac{K(K+1)}{2}$	$K + \frac{K(K+1)}{2}$

Table 1.1: The number of parameters in the multivariate Bernoulli, the Ising and the multivariate Gaussian models.

1.5 Multivariate Bernoulli Logistic Models

Generalized Linear Model

As discussed in Section 1.3, the multivariate Bernoulli distribution is a member of the exponential family and as a result, the generalized linear model theory in McCullagh and Nelder (1989) applies. The natural parameters (f's) in Lemma 1.3 can be formulated as a linear predictor in McCullagh and Nelder (1989) such that for any $\tau \in \mathcal{T}$ with $\mathcal{T}$ being the power set of $\{1, 2, \ldots, K\}$

$$f^{\tau}(x) = c_0^{\tau} + c_1^{\tau} x_1 + \cdots + c_p^{\tau} x_p, \tag{1.33}$$

where the vector $c^{\tau} = (c_0^{\tau}, \ldots, c_p^{\tau})$ for $\tau \in \mathcal{T}$ is the coefficient vector to be estimated and $x = (x_1, x_2, \ldots, x_p)$ is the observed covariate. Here p is the number of variables and there are $2^K - 1$ coefficient vectors to be estimated so in total $(p+1) \times (2^K - 1)$ unknown parameters including the constants. (1.33) is built on the canonical link where natural parameters are directly modeled as linear predictors, but other links are possible and valid as well.

When there are n samples observed from a real data set with outcomes denoted as $y(i) = (y_1(i), \ldots, y_K(i))$ and predictor variables $x(i) = (x_1(i), \ldots, x_p(i))$, the negative log likelihood for the generalized linear model of the multivariate Bernoulli distribution is

$$l(y, \mathbf{f}(x)) = \sum_{i=1}^{n} \left[- \sum_{\tau \in \mathcal{T}} f^{\tau}(x(i)) B^{\tau}(y(i)) + b(\mathbf{f}(x)) \right], \tag{1.34}$$

where, similar to (1.17) the log partition function b is

$$b(\mathbf{f}(x)) = \log\left[1 + \sum_{\tau \in \mathcal{T}} \exp[S^{\tau}(x(i))]\right].$$

When dealing with the univariate Bernoulli distribution using formula (1.34), the resulting generalized linear model corresponding to the multivariate Bernoulli model is the same for logistic regression. Thus the model is referred to as the multivariate Bernoulli logistic model in this chapter.

Gradient and Hessian

To optimize the negative log likelihood function (1.33) with respect to the coefficient vector c^{τ}, the efficient and popular iterative re-weighted least squares algorithm mentioned in McCullagh and Nelder (1989) can be implemented. Nevertheless, the gradient vector and Hessian matrix (Fisher Information) with respect to the coefficients c^{τ} are still required.

Consider any $\tau \in \mathcal{T}$, the first derivative with respect to c_j^{τ} in the negative log likelihood (1.34) of the multivariate Bernoulli logistic model, according to (1.27) and ignoring index i, is

$$\begin{aligned} \frac{\partial l(y,f)}{\partial c_j^{\tau}} &= \frac{\partial l(y,f)}{\partial f^{\tau}} \frac{\partial f^{\tau}}{\partial c_j^{\tau}} \\ &= \sum_{i=1}^{n} \left[-B^{\tau}(y) + \frac{\sum_{\tau_0 \supseteq \tau} \exp[S^{\tau_0}(x)]}{\exp[b(\mathbf{f}(x))]}\right] x_j \end{aligned} \tag{1.35}$$

Further, the second derivative for any two coefficients $c_j^{\tau_1}$ and $c_k^{\tau_2}$ is

$$\begin{aligned}
\frac{\partial^2 l(y,f)}{\partial c_j^{\tau_1} \partial c_k^{\tau_2}} &= \frac{\partial}{\partial c_j^{\tau_1}} \left(\frac{\partial l(y,f)}{\partial f^{\tau_2}} \frac{\partial f^{\tau_2}}{\partial c_k^{\tau_2}} \right) \\
&= \frac{\partial f^{\tau_1}}{\partial c_j^{\tau_1}} \frac{\partial^2 l(y,f)}{\partial f^{\tau_1} \partial f^{\tau_2}} \frac{\partial f^{\tau_2}}{\partial c_k^{\tau_2}} \\
&= \sum_{i=1}^{n} \frac{\partial^2 l(y,f)}{\partial f^{\tau_1} \partial f^{\tau_2}} x_j x_k \\
&= \frac{\sum_{\tau_0 \supseteq \tau_1,\, \tau_0 \supseteq \tau_2} \exp[S^{\tau_0}(x)]}{\exp[b(f(x))]} x_j x_k - \\
&\quad \frac{\sum_{\tau_0 \supseteq \tau_1} \exp[S^{\tau_0}(x)] \sum_{\tau_0 \supseteq \tau_2} \exp[S^{\tau_0}(x)]}{\exp[2b(f(x))]} x_j x_k
\end{aligned} \tag{1.36}$$

Parameters Estimation and Optimization

With gradient (1.35) and Hessian (1.36) at hand, the minimization of the negative log likelihood (1.34) with respect to the coefficients c^{τ} can be solved with Newton-Raphson or the Fisher's scoring algorithm (iterative re-weighted least squares) when the Hessian is replaced by the Fisher information matrix. Therefore, in every iteration, the new step for current estimate $\hat{c}^{(s)}$ is computed as

$$\triangle c = - \left(\frac{\partial^2 l(y,f)}{\partial c_j^{\tau_1} \partial c_k^{\tau_2}} \bigg|_{c=\hat{c}^{(s)}} \right)^{-1} \cdot \left(\frac{\partial l(y,f)}{\partial c_j^{\tau}} \bigg|_{c=\hat{c}^{(s)}} \right). \tag{1.37}$$

The process continues until the convergence criterion is met, which is declared when the absolute value of the coefficients change is less than tolerance or the maximum iteration number is reached.

Smoothing Spline ANOVA Model

The smoothing spline model gained popularity in non-linear statistical inference since it was proposed in Craven and Wahba (1978) for univariate predictor variables. More importantly, multiple smoothing spline models for generalized linear models enable researchers to study complex real world data sets with increasingly powerful computers as described in Wahba et al. (1995).

As a member of the exponential family, the multivariate Bernoulli distribution can be formulated under smoothing spline ANOVA framework. Gao et al. (2001) considers the smoothing spline ANOVA multivariate Bernoulli model but the interactions are restricted to be constant. However, in general, the natural parameters or linear predictors f's can be relaxed to reside in a reproducing kernel Hilbert space. That is to say, for the observed predictor vector x, we have

$$f^\tau(x) = \eta^\tau(x), \quad \text{with } \eta^\tau \in \mathcal{H}^\tau,\ \tau \in \mathcal{T}, \tag{1.38}$$

where $\mathcal{H}^\tau$ is a reproducing kernel Hilbert space and the superscript τ allows a more flexible model such that the natural parameters can come from different reproducing kernel Hilbert spaces. Further, $\mathcal{H}^\tau$ can be formulated to have several components to handle multivariate predictor variables, that is $\mathcal{H}^\tau = \oplus_{\beta=0}^{p} \mathcal{H}_\beta^\tau$ and details can be found in Gu (2002).

As a result, the η^τ is estimated from the variational problem

$$\mathcal{I}_\lambda(x, y) = \frac{1}{n} \sum_{i=1}^{n} l(y(i), \eta(x(i))) + \lambda J(\eta), \tag{1.39}$$

where η is the vector form of η^τ's. The penalty is seen to be

$$\lambda J(\eta) = \lambda \sum_{\tau \in \mathcal{T}} \theta_\tau^{-1} ||P_1^\tau \eta^\tau||^2, \tag{1.40}$$

with λ and θ_τ being the smoothing parameters. This is an over-parameterization adopted in Gu (2002), as what really matters are the ratios λ/θ_τ. The functional P_1^τ projects function η^τ in $\mathcal{H}^\tau$ onto the smoothing subspace $\mathcal{H}_1^\tau$.

By the argument of smoothing spline ANOVA model in Gu (2002), the minimizer η^τ has the expression as in Wahba (1990),

$$\eta^\tau(x) = \sum_{\nu=1}^{m} d_\nu^\tau \phi_\nu^\tau(x) + \sum_{i=1}^{n} c_i^\tau R^\tau(x_i, x), \tag{1.41}$$

where $\{\phi_\nu^\tau\}_{\nu=1}^m$ is a basis of $\mathcal{H}_0^\tau = \mathcal{H}^\tau \ominus \mathcal{H}_1^\tau$, the null space corresponding to the projection functional P_1^τ. $R^\tau(\cdot, \cdot)$ is the reproducing kernel for $\mathcal{H}_1^\tau$.

The variational problem (1.39) utilizing the smoothing spline ANOVA framework can be solved by iterative re-weighted least squares (1.37) due to the linear formulation (1.41).

Chapter 2

Multivariate Bernoulli with LASSO

2.1 Introduction

In this chapter, the LASSO variable selection technique implemented for the multivariate Bernoulli logistic model is discussed. As Moore Law realized in modern computing power, variable selection becomes one of the hot topics in modern statistical inferences. It is crucial to select only the significant variables to determine the structure of the graph for better model identification and prediction accuracy especially when large amount of data and millions of candidate predictors are available for analysis.

There are various approaches proposed to deal with univariate binary outcome problems. Logistics regression in McCullagh and Nelder (1989) built on the framework of generalized linear model and Support Vector Machines (SVM) proposed in Cortes and Vapnik (1995) are two of the most popular methods. Here the logistic regression models the probability of outcome 1 by fitting a maximum

likelihood to the data assuming the data follows a Bernoulli or binomial distribution. Thus, the prediction for new observed samples are the probability between 0 and 1 and as a result, referred to as *soft* classification approach in Wahba (2002). When facing the need of fitting non-linear trend of the covariates, regression spline model such as the Multivariate Adaptive Regression Splines (MARS) (Friedman, 1991) and the smoothing spline ANOVA model (Xiang and Wahba, 1994) as an extension of the ridge regression is widely adopted. On the other hand, the SVM and other kernel-based methods are discussed in detail in Cristianini and Shawe-Taylor (2000). The SVM is based on margin maximization between two distinct classes so the predicted outcome of a new sample is the estimated label (either positive or negative) and defined as a *hard* classification approach in Wahba (2002). Furthermore, the tree-based approaches such as CART (Breiman et al., 1984) and GUIDE (Loh, 2012) are proved to be both intuitive and accurate in prediction. The basic idea is to find the most significant predictor variables in each step and dichotomize at a level in order to minimize the prediction error or other equivalent losses. There are also other popular and very efficient classification approaches in machine learning literature as well as well-developed softwares for real world data analysis. For recent survey of machine learning approaches, refer to Bishop (2007) and Hastic et al. (2009), and free softwares like Weka (Witten et al., 2011) and SHOGUN (Sonnenburg et al., 2010).

The pioneering paper Tibshirani (1996) introduce the LASSO approach to linear models based on Gaussian distribution. Various properties of the method are demonstrated such as model selection consistency discussed in Zhao and Yu (2006), and extensions to different frameworks studied in Meinshausen and

Buhlmann (2006), Zhao and Yu (2007), Park and Casella (2008) etc. Park and Hastie (2007) develops an algorithm based on ideas from LARS (Efron et al., 2004) to extend LASSO to generalized linear models. In addition, the LASSO model applied to penalized logistic regressions are also widely explored such as in Shi et al. (2008) where the LASSO-patternsearch algorithm was introduced and is capable of handling large number of unknowns provided that it is known that at most a modest number are non-zeros. Recently, Shi et al. (2012) has extended the algorithm in Shi et al. (2008) to the scale of multi-millions of unknowns. Coordinate descent (Friedman et al., 2010) is also proven to be fast in solving large p small n problems.

The goal of this chapter is to build the multivariate Bernoulli LASSO model to do variable selection in the multivariate Bernoulli logistic model. The remainder of the chapter is organized as follows. Section 2.2 starts from the general formulation of the multivariate Bernoulli LASSO model where the target function is introduced. Section 2.3 develops the algorithm to solve the optimal solution of the problem, which is based on the Accelerated Block-Coordinate Relaxation with theories discussed in Wright (2011). Furthermore, the choice of the tuning parameter is studied in Section 2.4 where the generalized approximate cross-validation (GACV) designed for non-Gaussian observations adopted from Xiang and Wahba (1994) is applied to the multivariate Bernoulli LASSO model. Finally, Section 2.5 is devoted to both simulation examples and real data set to demonstrate the performance of the multivariate Bernoulli LASSO model.

2.2 Model Formulation

The LASSO approach can be extended to the multivariate Bernoulli logistic model in Section 1.5 since it is a special case of the generalized linear model. What we have to do is to apply the l_1 penalty to the coefficients in (1.33), and the resulting target function is

$$L_\lambda(x, y) = \frac{1}{n}\sum_{i=1}^{n} l(y(i), \mathbf{f}(x(i))) + \sum_{\tau\in\mathcal{T}} \lambda_\tau \sum_{j=1}^{p} |c_j^\tau|, \tag{2.1}$$

where $y(i)$ and $x(i)$ indicates the ith observed outcome and covariate.

To simplify the notation, let λ be the multi-dimensional tuning parameter $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_{12\ldots K})$. We also denote the l_1 norm of the coefficients to be $||\mathbf{c}||_1 = (||c^1||_1, ||c^2||_1, \ldots, ||c^{12\ldots K}||_1)$, where $||c^\tau||_1 = \sum_{j=1}^{p} |c_j^\tau|$. As a result, the vectorized form of (2.1) is

$$\mathcal{I}_\lambda(\mathbf{c}) = \frac{1}{n}\sum_{i=1}^{n} l(Y(i), \mathbf{f}(X(i))) + \lambda||\mathbf{c}||_1. \tag{2.2}$$

In real applications especially when the graph is large with many nodes, we often restrict our attention to main effects and low level interactions due to computation and interpretation difficulty. In this case, all the f functions beyond the pre-determined maximum order of interactions will be set to zero. The number of coefficients to be estimated will be significantly reduced, but the higher order S functions still need to be evaluated in full.

In this section, we aim at solving a general problem which involves a convex loss

function and an l_1 penalty applied to the unknown variables so that at the optimum, only a small set of variables are nonzero provided the tuning parameters λ is not too small. The specialized algorithm is designed to use gradient information for the smooth term $l(y, f)$ to form an estimate of the correct zero set, which is the set of components of $\{c_j^\tau\}$ that are zero at the minimizer of (2.1). Some iterations of the algorithm also attempt Newton-like enhancement to the search directly, computed using the projection of Hessian of $l(y, f)$ onto the set of nonzero components of $\{c_j^\tau\}$. This approach is similar to the two-metric gradient projection approach for bound-constraint minimization, but avoids duplication of variables and allows certain other resources saving in the implementation.

As the negative log-likelihood of the multivariate Bernoulli LASSO model, the loss function $\mathcal{I}_\lambda$ in (2.2) is convex, so $\mathbf{c}$ is optimal for (2.1) if and only if the following condition holds:

$$\bigtriangledown \mathcal{I}(\mathbf{c}) + \lambda^T v = 0, \tag{2.3}$$

where $\mathcal{I}(\mathbf{c})$ is the negative log-likelihood for the multivariate Bernoulli logistic model (1.5), and for some vector $v = (v^1, v^2, \ldots, v^{12\ldots K})$ in the sub-differential of $||\mathbf{c}||_1$ (denoted by $\partial||\mathbf{c}||_1$). In other words,

$$v_j^\tau = \begin{cases} -1 & \text{if } c_j^\tau < 0 \\ \in [-1, 1] & \text{if } c_j^\tau = 0 \\ 1 & \text{if } c_j^\tau > 0 \end{cases} \tag{2.4}$$

A measure of near-optimality is given as follows:

$$\delta(\mathbf{c}) = \min_{v \in ||\mathbf{c}||_1} || \bigtriangledown \mathcal{I}(\mathbf{c}) + \lambda^T v||, \tag{2.5}$$

and $\delta(\mathbf{c}) = 0$ if and only if $\mathbf{c}$ is optimal.

2.3 The Accelerated Block-Coordinate Relaxation

The Accelerated Block-Coordinate Relaxation algorithm studied in Wright (2011) is adopted to solve the optimization problem (2.2) in this section. We also discuss an outline of some enhancements applied to the algorithm to improve the efficiency.

The basic (first-order) step at iteration k is obtained by forming a simple model of the objective by expanding around current iterate $\mathbf{c^k}$ as follows:

$$d^k \quad = \quad arg\min_d \bigtriangledown \mathcal{I}(\mathbf{c^k})^T d + \frac{1}{2}\alpha_k d^T d + \lambda^T ||\mathbf{c^k} + d||_1, \tag{2.6}$$

where α_k is a positive scalar (whose value is discussed below) and d^k is the proposed step, which is a matrix instead of vector, which is the case for univariate logistic regression studied in Shi et al. (2008). The subproblem (2.6) is separable in the components of d and therefore trivial to solve in closed form in $O(|\mathcal{T}|p)$ operations. The solution d^k can be examined to obtain an estimate of the zero set:

$$\mathcal{Z}_k = \{j = 1, 2, \ldots, |\mathcal{T}|p \mid (\mathbf{z^k} + d^k)_j = 0\}. \tag{2.7}$$

The definition of the "nonzero set" $\mathcal{N}_k$, or referred to as active set in some literature, is the complement of the zero set estimate, that is:

$$\mathcal{N}_k = \{1, 2, \ldots, |\mathcal{T}|p\} \setminus \mathcal{Z}_k. \tag{2.8}$$

If the step d^k computed from (2.6) does not yield a decrease in the objective function $\mathcal{I}_\lambda$, then α_k is increased and re-solve (2.6) to obtain a new d^k. This process can be repeated as needed. It can be shown that, provided $\mathbf{c^k}$ does not satisfy an optimality condition, the d^k obtained from (2.6) will yield $\mathcal{I}_\lambda(\mathbf{c^k} + d^k) < \mathcal{I}_\lambda(\mathbf{c^k})$ for α_k sufficiently large.

The step is enhanced by computing the restriction of the Hessian $\bigtriangledown^2\mathcal{I}(\mathbf{c^k})$ to the set $\mathcal{N}_k$ (denoted by $\bigtriangledown^2_{\mathcal{N}_k\mathcal{N}_k}\mathcal{I}(\mathbf{c^k})$) and then computing a Newton-like step in the $\mathcal{N}_k$ components as follows:

$$(\bigtriangledown^2_{\mathcal{N}_k\mathcal{N}_k}\mathcal{I}(\mathbf{c^k}) + \delta_k I)p^k_{\mathcal{N}_k} \quad = \quad -\bigtriangledown_{\mathcal{N}_k}\mathcal{I}(\mathbf{c^k}) - \lambda^T\omega_{\mathcal{N}_k}, \tag{2.9}$$

where δ_k is a small damping parameter that goes to zero as $\mathbf{c^k}$ approaches the solution, and $\omega_{\mathcal{N}_k}$ captures the gradient of the term $||\mathbf{c}||_1$ at the nonzero components of $\mathbf{c^k} + d^k$. Specifically, $\omega_{\mathcal{N}_k}$ coincides with $\partial||\mathbf{c^k} + d^k||_1$ on the components $i \in \mathcal{N}_k$. If δ_k were set to zero, $p^k_{\mathcal{N}_k}$ would be the (exact) Newton step for the subspace defined by $\mathcal{N}_k$; the use of a damping parameter ensures that the step is well defined even when the partial Hessian $\bigtriangledown^2_{\mathcal{N}_k\mathcal{N}_k}\mathcal{I}(\mathbf{c^k})$ is singular or nearly singular, as happens with our problems. In practice, choose:

$$\delta_k = \min(\delta(\mathbf{c^k}), \text{mean diagonal of } \bigtriangledown^2_{\mathcal{N}_k\mathcal{N}_k}\mathcal{I}(\mathbf{z^k})), \tag{2.10}$$

where $\delta(\mathbf{c})$ is defined in (2.5).

Because of the special form of $\mathcal{I}(\mathbf{c})$, the Hessian is not expensive to compute once the gradient is known. However, it is dense in general, so considerable saving can be made by evaluating and factoring this matrix on only a reduced subset of the variables, as in the scheme described above.

If the partial Newton step calculated above fails to produce a decrease in the objective function $\mathcal{I}_\lambda$, a shortened step is evaluated with its length reduced by a factor γ_k, to the point where $c_j^k + \gamma_k p_j^k$ has the same sign as c_j^k for all $j \in \mathcal{N}_k$. If this modified step also fails to decrease the objective $\mathcal{I}_\lambda$, the first-order step calculated from (2.6) is taken if it decreases $\mathcal{I}_\lambda$. Otherwise, the parameter α_k is increased, leaving $\mathbf{c^k}$ unchanged, and proceed to the next iteration.

The algorithm is summarized as follows:

Algorithm

given *initial point* z^0, *initial damping* $\alpha_0 > 0$, *constant* `tol` > 0 *and* $\eta \in (0, 1)$;

for $k = 0, 1, 2, \ldots$

  if $\delta(\mathbf{c^k}) <$ `tol`

    stop *with approximate solution* $\mathbf{c^k}$;

  endif

  Evaluate e^{f^τ} *and* e^b

  Solve (2.6) *for* d^k; *(% first-order step)*

  Evaluate $\mathcal{Z}_k$ *and* $\mathcal{N}_k$

  Compute $p^k_{\mathcal{N}_k}$ *from* (2.9); *(% reduced Newton step)*

  Set $c^+_{\mathcal{N}_k} = c^k_{\mathcal{N}_k} + p^k_{\mathcal{N}_k}$ *and* $c^+_{\mathcal{Z}_k}$

  if $\mathcal{I}_\lambda(c^+) < \min(\mathcal{I}_\lambda(\mathbf{c^k} + d^k), \mathcal{I}_\lambda(\mathbf{c^k}))$ *(% Newton step successful)*

$c^{k+1} \longleftarrow c^+$

else

Choose γ_k *as the largest positive number such that*

$(\mathbf{c^k} + \gamma_k p^k)_j > 0$ *for all* j *with* $c_j^k \neq 0$*; (% damp the Newton step)*

Set $c^+_{\mathcal{N}_k} = c^k_{\mathcal{N}_k} + \gamma_k p^k_{\mathcal{N}_k}$ *and* $c^+_{\mathcal{Z}_k} = 0$*;*

if $\mathcal{I}_\lambda(c^+) < \min(\mathcal{I}_\lambda(\mathbf{c^k} + d^k), \mathcal{I}_\lambda(\mathbf{c^k}))$ *(% damped Newton step successful)*

$c^{k+1} \longleftarrow c^+$**;**

else if $\mathcal{I}_\lambda(\mathbf{c^k} + d^k) < \mathcal{I}_\lambda(\mathbf{c^k})$ *(% first-order step successful; use it if Newton steps fail)*

$c^{k+1} \longleftarrow c^k + d^k$**;**

else *(% unable to find a successful step)*

$c^{k+1} = \mathbf{c^k}$**;**

endif

endif

(% increase or decrease α *depending on success of first-order step)*

if $\mathcal{I}_\lambda(\mathbf{c^k} + d^k) < \mathcal{I}_\lambda(\mathbf{c^k})$

$\alpha_{k+1} \longleftarrow \eta\alpha_k$**;** *(% first-order step decreased* T_λ*, so decrease* α*)*

else

$\alpha_{k+1} \longleftarrow \alpha_k/\eta$**;**

endif

endfor

Some enhancements to this basic approach can lead to in significant improvements to the execution time. Note first that evaluation of the full gradient $\bigtriangledown\mathcal{I}(\mathbf{c^k})$, which is needed to compute the first-order step in (2.6) can be quite expensive.

Since in most cases the vast majority of components of $\mathbf{c^k}$ are zero, and will remain so after the next step is taken, this can be economized by selecting just a subset of components of $\bigtriangledown\mathcal{I}(\mathbf{c^k})$ to evaluate at each step, and allowing just these of the first-order step d to be nonzero. Specifically, for some chosen constant $\sigma \in (0, 1]$, we select $\sigma \times |\mathcal{T}|p$ components from the index set $\{1, 2, \ldots, |\mathcal{T}|p\}$ at random (using a different random selection at each iteration), and define the working set W_k to be the union of this set with the set of indices j for which $c_j^k \neq 0$. Then just the components of $\bigtriangledown\mathcal{I}(\mathbf{c^k})$ for the indices $j \in W_k$ are evaluated and (2.6) can be solved subject to the constraint that $d_j = 0$ for $j \notin W_k$.

Since $\delta(\mathbf{c^k})$ cannot be calculated without knowing the full gradient $\bigtriangledown\mathcal{I}(\mathbf{c^k})$, a modified version of this quantity is defined by taking the norm in (2.2) over the vector defined by W_k, and use this version to compute the damping parameter δ_k. The convergence criterion is modified by forcing the *full* gradient vector to be computed on the next iteration $k + 1$ when the threshold condition $\delta(\mathbf{c^k}) < \texttt{tol}$ is satisfied. If this condition is satisfied again at iteration $k + 1$, success is declared.

A further enhancement is that the second-order step is computed only when the number of components in $\mathcal{N}_k$ is small enough to make computation and factorization of the reduced Hessian economical. In the experiments reported here, only the first order step is computed if the number of components in $\mathcal{N}_k$ exceeds 500.

2.4 Tune the Parameters

So far, all smoothing parameters λ are considered fixed. However, the choice of the tuning parameters in equation (2.2) is crucial since in general the larger the

λ's, the smaller the number of patterns picked up by the algorithm.

The GACV (generalized approximate cross-validation) is used to approximately minimize the comparative Kullback-Leibler (CKL) distance, see Lin et al. (2000). Gao et al. (2001) extends it to the multivariate Bernoulli smoothing spline ANOVA model and Shi et al. (2008) derives a version for l_1 penalized logistic regression. Similarly, this section is devoted to derive a version of GACV for the multivariate Bernoulli distribution to be combined with LASSO penalty.

In the optimization problem (2.1), the natural parameter f matrix and augmented response matrix can be written into vectors

$$\vec{f}_\lambda(X) = (f_\lambda^1(X(1)), f_\lambda^2(X(1)), \ldots, f_\lambda^{12\ldots K}(X(n)))^T, \tag{2.11}$$

$$\vec{B}(Y) = (B(Y(1)), B(Y(2)), \ldots, B(Y(n))), \tag{2.12}$$

where $\vec{f_\lambda}$ is indexed by λ since it is the optimal linear predictor corresponding to the tuning parameter vector λ.

Therefore, the optimization problem (2.2) can be reformulated in a vectorized form as

$$I_\lambda(Y, f) = \frac{1}{n}\left[-\vec{B}(Y)^T\vec{f} + b(\vec{f})\right] + \sum_{\tau\in\mathcal{T}} \lambda||\mathbf{c}||_1. \tag{2.13}$$

This formula is very similar to the equation in Shi et al. (2008), except that the Hessian matrix is block diagonal instead of strictly diagonal due to correlation among binary outcomes (nodes).

The matrix form of the linear predictor in (1.33) in vectorized form (2.11) and

(2.12) is

$$\vec{f} = \mathcal{D}\beta,$$

then the expanded design matrix and the vectorized unknown coefficients are

$$\begin{aligned}
\mathcal{D} &= \begin{pmatrix}
X(1) & \vec{0} & \dots & \vec{0} \\
\vec{0} & X(1) & \dots & \vec{0} \\
\vdots & \vdots & \vdots & \vdots \\
\vec{0} & \vec{0} & \dots & X(1) \\
X(2) & \vec{0} & \dots & \vec{0} \\
\vdots & \vdots & \vdots & \vdots \\
\vec{0} & \vec{0} & \dots & X(n)
\end{pmatrix}, \\
\beta &= \left(c^1, c^2, \dots, c^{12\dots K}\right)^T,
\end{aligned}$$

where the matrix $\mathcal{D}$ is of dimension $n(2^K-1)\times p(2^K-1)$ and the length of unknown vector β is $p(2^K-1)$.

Similarly, the mean of the augmented response for the ith observation can be formulated as

$$\begin{aligned}
\vec{\mu}(i) &= (\mu^1(Y(i)), \mu^2(Y(i)), \dots, \mu^{1\dots K}(Y(i))) && (2.14) \\
&= E[B(Y(i))|X(i), f], && (2.15)
\end{aligned}$$

which is the gradient of the partition function in (1.17) for the multivariate Bernoulli

logistic model. This leads to the augmented CKL distance

$$CKL(\lambda) \;=\; \frac{1}{n}\sum_{i=1}^{n}\left[-\vec{\mu}(i)\vec{f}_\lambda(X(i)) + b(f_\lambda(X(i))\right]. \tag{2.16}$$

To obtain the leave-one-out Lemma for the multivariate Bernoulli LASSO model, denote $\vec{f}_{\lambda,\epsilon}$ to be the minimizer of the optimization problem (2.13) with tuning parameter λ and small perturbation ϵ added to Y, and $\vec{f}_\lambda^{[-i]}$ is the optimizer for the target function tuned by λ with the ith observation deleted.

The GACV tuning method discussed in Xiang and Wahba (1994) for binary responses is a faster version of GCV in Craven and Wahba (1978). The method involves an approximation to $\vec{f}_\lambda - \vec{f}_{\epsilon,\lambda}$ using Taylor's theorem. The influence matrix H (Wahba, 1990), (Xiang and Wahba, 1994) is used as a medium for this purpose

$$\vec{f}_\lambda - \vec{f}_{\epsilon,\lambda} = H\epsilon,$$

where $\vec{f}_{\epsilon,\lambda}$ is the estimated f when a small perturbation ϵ is added to the outcome Y.

The leave-one-out Lemma (Craven and Wahba, 1978) for the multivariate Bernoulli LASSO model is stated as follows

Lemma 2.1. *(Leave-one-out) For fixed sample i, and augmented response $\vec{Y}$, let $h_\lambda[i,\vec{Y}]$ be the minimizer of*

$$\left[-\vec{B}(Y)^T\vec{f} + b(\vec{f})\right] - l(Y(i), f(X(i))) + \lambda||\mathbf{c}||_1.$$

Then $h_\lambda[i,\vec{\mu}_\lambda^{[-i]}(i)] = f_\lambda^{[-i]}$. Here $\vec{\mu}_\lambda^{[-i]}(i) = E[B(Y)|X(i), f_\lambda^{[-i]}]$.

The detailed proof can be found in Ma (2010).

The Hessian matrix for the augmented outcome sample i is

$$W_i(f_\lambda) = \frac{\partial^2 - l(Y(i), f(X(i)))}{\partial f(X(i)) \partial f(X(i))^T} = \text{Var}(B(Y)|f_\lambda(X(i))), \tag{2.17}$$

in which the formula can be derived from (1.36).

The covariance matrix for the combined response is:

$$W(f_\lambda) = diag(W_1(f_\lambda), \ldots, W_n(f_\lambda)). \tag{2.18}$$

The above matrix is of dimension $n(2^K-1) \times n(2^K-1)$ and block diagonal with block size $(2^K - 1) \times (2^K - 1)$. The difference between LASSO and the smoothing spline ANOVA for the multivariate Bernoulli model in Gao et al. (2001) is that the Hessian matrix of the target function (2.1) is the same as the Hessian of negative log-likelihood excluding the neighborhoods of any zero coefficients, whereas in smoothing spline ANOVA model the penalty is quadratic so the kernel matrix needs to be taken into consideration (Wahba, 1990), (Ding et al., 2011).

Denote the number of nonzero elements in β to be s and $\mathcal{D}^*$ is the sub-matrix of $\mathcal{D}$ with columns corresponding to the nonzero elements in β. By first-order Taylor expansion, the H matrix is approximately

$$\begin{aligned} H &= \mathcal{D}^{*T} U^{-1} \mathcal{D}^* \\ &= \mathcal{D}^{*T} \left(\mathcal{D}^* W (\mathcal{D}^*)^T \right)^{-1} \mathcal{D}^*. \end{aligned} \tag{2.19}$$

In ordinary leave-out-one cross validation,

$$\begin{aligned} CV(\lambda) &= \frac{1}{n}\sum_{i=1}^{n}\left[-B(Y(i))^T f_\lambda(X(i)) + b(f_\lambda(X(i)) + B(Y(i))^T\left(f_\lambda(X(i)) - f_\lambda^{[-i]}(X(i))\right)\right] \\ &= OBS(\lambda) + \frac{1}{n}\sum_{i=1}^{n} B(Y(i))^T\left(f_\lambda(X(i)) - f_\lambda^{[-i]}(X(i))\right) \\ &\approx OBS(\lambda) + \frac{tr(H)}{n}\frac{\sum_{i=1}^{n} B(\vec{Y})(B(\vec{Y}) - \vec{\mu})}{tr\left[I - (W^{1/2}HW^{1/2})\right]}, \end{aligned} \tag{2.20}$$

where $\vec{\cdot}$ denotes the vector form of a general matrix as in equations (2.11) and (2.12). $f_\lambda^{[-i]}$ is the estimate of the f function when the ith observation is deleted from the dataset, the so-called leave-one out estimate. It is not hard to verify that $tr(W^{1/2}HW^{1/2}) = s$, the simplified GACV score can thus be derived

$$GACV(\lambda) = OBS(\lambda) + \frac{tr(H)}{n}\frac{\sum_{i=1}^{n} B(\vec{Y})(B(\vec{Y}) - \vec{\mu})}{n - s}. \tag{2.21}$$

Similarly, the BGACV score (Shi et al., 2008) for the multivariate Bernoulli LASSO model, which is a BIC like version (Schwarz, 1978) of GACV can be defined accordingly.

$$BGACV(\lambda) = OBS(\lambda) + \frac{tr(H)\log n}{2n}\frac{\sum_{i=1}^{n} B(\vec{Y})(B(\vec{Y}) - \vec{\mu})}{n - s}. \tag{2.22}$$

What's more, some other popular tuning scores can be derived with the help of Stein's unbiased risk estimate (SURE) discussed in Efron et al. (2004). The detailed derivation of the AIC and BIC especially the degrees of freedom in the generalized linear LASSO models can be found in Ma (2010).

$$AIC(\lambda) = OBS(\lambda) + \frac{1}{n}s, \tag{2.23}$$

$$BIC(\lambda) = OBS(\lambda) + \frac{\log(n)}{2n}s. \tag{2.24}$$

Nonetheless, the calculation of the GACV or BGACV involves evaluating the inverse of a large matrix $(\mathcal{D}^* W (\mathcal{D}^*)^T)^{-1}$, which renders the method infeasible when dealing with large scale problems. To alleviate the difficulty, Lin et al. (2000) proposes the randomized GACV to avoid inverting the matrix.

The idea is that in formula (2.21), the only quantity we need to calculate related to the inverted matrix is $tr(H)$. This can be done via a "black box" method on perturbed data $Y + \epsilon$, where the components of ϵ come from random normal distribution with zero mean and sufficiently small variance. The case for the outcomes following Gaussian distribution has been studied extensively and shown to be as good as the exact calculation for large n in Girard (1998). The randomized trace method essentially is based on the fact that for a square matrix A of dimension $n \times n$, and ϵ, the zero mean random n-vector with independent components with variance σ_ϵ^2, then

$$\frac{1}{\sigma_\epsilon^2} E\epsilon^T A\epsilon = tr(A).$$

In practice, σ_ϵ^2 is replaced by the estimate $\frac{1}{n}\sum_{i=1}^n \epsilon_i^2$. Following the argument in Xiang and Wahba (1996), the approximation is

$$\vec{f}_\lambda^{Y+\epsilon} - \vec{f}_\lambda^{Y} \approx H\epsilon,$$

where $\vec{f}_\lambda^{Y+\epsilon}$ represents the estimate of vectorized f for perturbed outcome $Y+\epsilon$ and tuning parameter λ. This suggests that $\frac{1}{\sigma_\epsilon^2}\epsilon^T(\vec{f}_\lambda^{Y+\epsilon} - \vec{f}_\lambda^{Y})$ provide a good estimate to $tr(H)$.

As a result, following the argument in Lin et al. (2000), the randomized GACV for the multivariate Bernoulli LASSO model is

$$\begin{aligned} ranGACV(\lambda) &= OBS(\lambda) + \frac{1}{n}B(\vec{Y})^T(B(\vec{Y}) - \vec{\mu}) \\ &\quad \times \left[\epsilon^T(\vec{f}_\lambda^{Y+\epsilon} - \vec{f}_\lambda^{Y})\right] / \left[\epsilon^T\epsilon - \epsilon^T W(\vec{f}_\lambda^{Y+\epsilon} - \vec{f}_\lambda^{Y})\right]. \end{aligned}$$

Notice here the length of vectorized Y is longer than the random vector ϵ since there are some augmented interaction terms in the outcome. In this case, for example, the corresponding $\hat{\epsilon}$ for $B^{12}(Y)$ can be calculated as

$$\hat{\epsilon} = (Y_1 + \epsilon_1)(Y_2 + \epsilon_2) - Y_1Y_2.$$

To reduce the variance in calculating ranGACV due to randomness of ϵ, we draw R replicates $\epsilon_1, \ldots, \epsilon_R$ and simply take the average

$$\begin{aligned} ranGACV(\lambda) &= OBS(\lambda) + \frac{1}{n}B(\vec{Y})^T(B(\vec{Y}) - \vec{\mu}) \\ &\quad \times \frac{1}{R}\sum_{r=1}^{R}\left[\epsilon_r^T(\vec{f}_\lambda^{Y+\epsilon_r} - \vec{f}_\lambda^{Y})\right] / \left[\epsilon_r^T\epsilon_r - \epsilon_r^T W(\vec{f}_\lambda^{Y+\epsilon_r} - \vec{f}_\lambda^{Y})\right] \end{aligned} \quad (2.25)$$

Without detailed specification, the tuning in this thesis for GACV and BGACV are implemented as ranGACV and ranBGACV with 20 replicates.

The optimal multiple dimensional tuning parameter λ is guaranteed only when we iterate all the possible points on the grid. However, in real applica-

tion, the derivative-free Nelder-Mead algorithm (down-hill simplex) proposed in Nelder and Mead (1965) is more suitable due to time constraint. All the numerical examples in this thesis are implemented under Nelder-Mead approach.

2.5 Numerical Examples

This section is designed to use simulation examples and real data application to illustrate the efficiency and application of the multivariate Bernoulli LASSO model.

Simulation 1

Firstly, we start from the simplest case of the multivariate Bernoulli graph, the bivariate Bernoulli, where only two binary nodes are involved as shown in Figure 1.1. The graph also includes 5 covariate independently distributed as standard normal, represented by $X_1, \ldots, X_5$. The true models for the f's functions are

$$\begin{aligned} f^1 &= 0.5 + 2X_1, \\ f^2 &= 0.5 - 2X_2, \\ f^{12} &= 1.5X_5, \end{aligned}$$

therefore the model has 5 nonzero patterns out of $3 \times 6 = 18$ candidates.

The simulations involve 100 independent data sets with each having 500 samples. The data sets are fitted by the multivariate Bernoulli LASSO model but tuned with the four different criteria discussed in Section 2.4.

Tuning	selected	true (total 5)
AIC	$10.04(3.435)$	$4.77(0.694)$
BIC	$5.89(1.626)$	$4.76(0.818)$
GACV	$5.08(0.367)$	$4.99(0.100)$
BGACV	$5.16(0.896)$	$4.97(0.171)$

Table 2.1: The results for the simulation 1, where the averages of the selected and true patterns out of 100 replicates are illustrated with standard deviations shown in parentheses.

The results are shown in Table 2.1. The first column reports the number of patterns the four methods selected with the standard deviation in these 100 replicates displayed in parentheses. Here AIC as expected selects larger models and large variance of model sizes than its corresponding B-type tuning, BIC. BIC, on the contrary favors smaller models on average but maintains similar level of accuracy in terms of picking up the true patterns. On the other hand, the final models selected by GACV and BGACV are even smaller than BIC with small size variations. At the same time, these two are capable of capturing all the true nonzero coefficients most of the time (4.99 and 4.97 out of 5, the perfect score).

Simulation 2

Secondly, we try a larger graph with more nodes. Consider the graph structure in Figure 1.2 so there are three binary nodes in the graph. Similarly, 5 independent standard normal distributed covariate are associated with some edge effects but

no clique effect f^{123}. The true model is

$$\begin{aligned} f^1 &= 0.5 + 2X_1, \\ f^2 &= 0.5 - 2X_2, \\ f^3 &= 0.5 + 2X_3, \\ f^{12} &= 1.5X_4, \\ f^{23} &= -1.5X_5, \\ f^{13} &= 0, \\ f^{123} &= 0, \end{aligned}$$

hence, there are 8 true nonzero coefficients out of $6 \times 7 = 42$ candidates. Figure 2.1 displays the true model in a graph.

Tuning	selected	true (total 8)
AIC	$11.56(4.977)$	$7.40(1.563)$
BIC	$8.13(2.616)$	$7.35(1.690)$
GACV	$8.22(3.080)$	$7.76(1.006)$
BGACV	$7.79(1.140)$	$7.68(1.053)$

Table 2.2: The results for the simulation 2, where the averages of the selected and true patterns out of 100 replicates are illustrated with standard deviations shown in parentheses.

We still fit the multivariate Bernoulli LASSO model to the 100 independently generated data sets and consider up to third order interactions among the outcomes. The results as before are shown in Table 2.2 for all four different tuning approaches. AIC still selects the models with largest sizes with large variations. BIC results to smaller models with low standard deviation but sacrifices slightly in terms of true

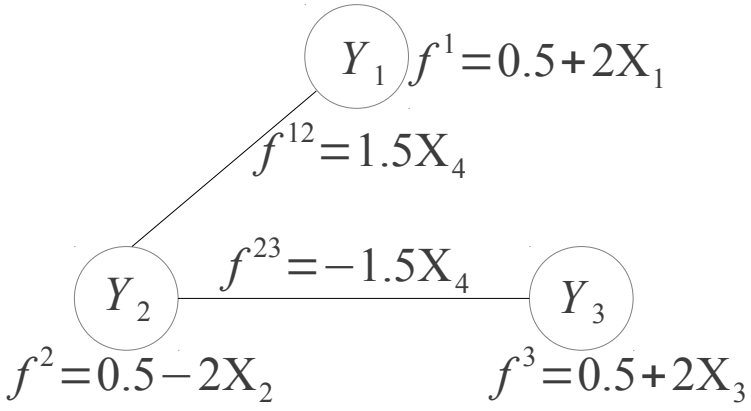

Figure 2.1: The graph for the simulation example 2.

patterns captured. GACV favors larger models than BIC on average but beats all others in picking up the true nonzero coefficients. BGACV successfully maintains the smallest models while kept most of the true patterns in the final models.

Simulation 3

The simulations 1 and 2 are simple graphs with nonzero edges, which is consistent with the Ising model compared in Section 1.4. The example examined in this simulation involves clique effect in a 3-node graph, the 3rd order interaction.

The setting is the same as in simulations 1 and 2 but the true model is

$$\begin{aligned} f^1 &= 0.5 + 2X_1, \\ f^2 &= 0.5 - 2X_2, \\ f^3 &= 0.5 + 2X_3, \\ f^{12} &= 0, \\ f^{23} &= 0, \\ f^{13} &= 0, \\ f^{123} &= 2X_5, \end{aligned}$$

as shown in Figure 2.2. Essentially, the graph only has three nodes but no edges connecting them. On the other hand, there is a clique f^{123} effect with covariate X_5 having positive influence.

Tuning	selected	true (total 7)
AIC	$11.82(5.997)$	$6.51(1.227)$
BIC	$7.18(3.000)$	$6.35(1.473)$
GACV	$7.08(1.061)$	$6.82(0.796)$
BGACV	$6.93(1.148)$	$6.68(0.909)$

Table 2.3: The results for the simulation 3, where the averages of the selected and true patterns out of 100 replicates are illustrated with standard deviations shown in parentheses.

This is a hard graph to learn since the only mutual influence among the nodes comes from a weak third order interaction. Nevertheless, the model can successfully capture the true patterns especially GACV and BGACV. They get more than 6.5 true coefficients out of 7 on average with small standard deviation and also

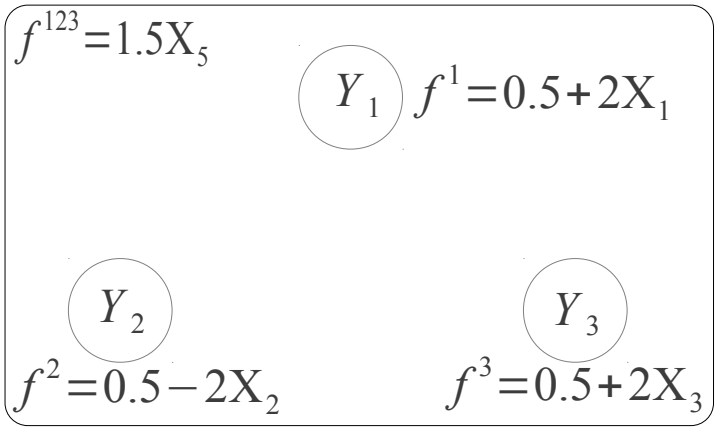

Figure 2.2: The graph for the simulation example 3.

maintain a small model sizes. In other words, they achieve high accuracy without sacrificing the specificity. On the other hand, AIC with the larger model sizes in general, get more nonzero coefficients than BIC, which has small final models, but both AIC and BIC suffer from large variation of performance which implies that they are not stable when the graph is hard to learn.

Simulation 4

Finally, we consider a more complicated graph when there are five nodes. The graph structure is shown in Figure 2.3 and the true model is

$$\begin{aligned} f^1 &= 0.5, \\ f^2 &= 0.5, \\ f^3 &= -0.5, \\ f^4 &= -0.5, \\ f^5 &= 0.5, \\ f^{12} &= -3X_1, \\ f^{14} &= 2X_3, \\ f^{24} &= 3X_2, \\ f^{34} &= -2X_4, \\ f^{123} &= 2X_5, \end{aligned}$$

and all the other f's are null.

The goal of this simulation is to examine the performance of the model to graph with both second order and third order interactions. The structure of the graph is fairly complex in the sense that the nodes Y_1, Y_2 and Y_4 with nonzero mutual edges including covariate effects. X_4 has negative influence on pair Y_3 and Y_5, but the clique (Y_1, Y_2, Y_3) is dependent on X_5 so f^{123} is nonzero. This graph cannot be analyzed using the Ising model mentioned in Section 1.4 since there are nontrivial third order clique effect.

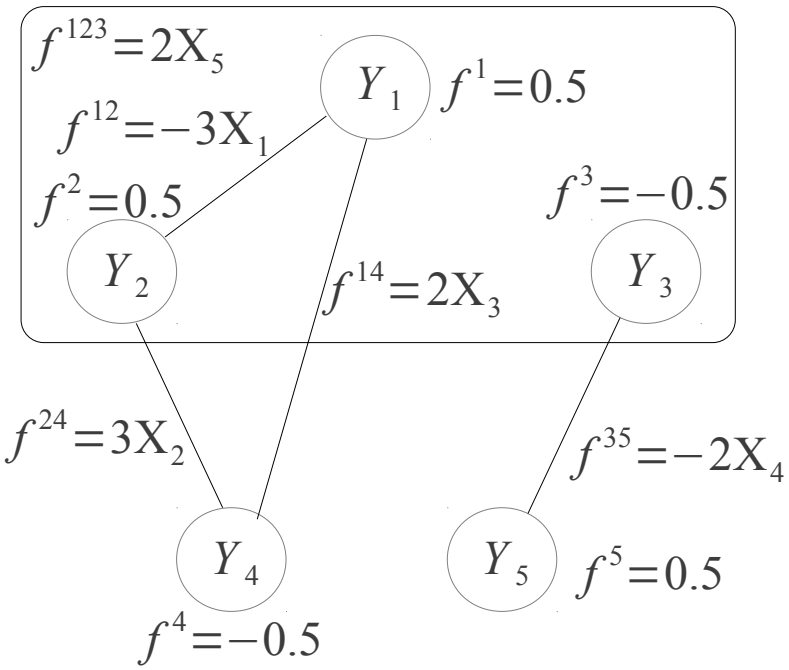

Figure 2.3: The graph for the simulation example 4.

The result of this simulation is illustrated in Table 2.4. The model is capable of capturing the nonzero coefficients on average since all four tuning criteria can pick up at least 9.60 of 10 true patterns with small standard deviation. As before BIC, GACV and BGACV keep the model sizes small and at the same time with at least 9.90 of 10.

To sum up from the simulation examples, BIC, GACV and BGACV are capable of maintaining the model sizes small and capturing the true patterns in the final model as illustrated from Table 2.1, 2.2, 2.3 and 2.4. On the other hand, AIC is capable of capturing the true patterns in the true model but it favors larger models

Tuning	selected	true (total 10)
AIC	14.28(4.582)	9.87(0.737)
BIC	11.34(1.380)	10.0(0)
GACV	12.44(3.057)	9.98(0.145)
BGACV	12.06(2.809)	9.87(0.737)

Table 2.4: The results for the simulation 4, where the averages of the selected and true patterns out of 100 replicates are illustrated with standard deviations shown in parentheses.

and as a result introducing more variation and false positive to the final model.

Real Data Analysis

Here the US Census Bureau data (US Census Bureau, 2007) is used to examine the performance of the multivariate Bernoulli LASSO model in real-world applications. The data is based on several statistics of counties level spreading all the US states. After removing counties with missing observations, the total samples in the analysis is 2573. Table 2.5 summarizes the outcomes of interest and some of their descriptive statistics. The nodes in the trivariate Bernoulli graph are coded 1 if the observed value is greater than the national median and 0 otherwise.

Interestingly, the county with the lowest population changes in the analysis is Bernard from Louisiana (Orleans, LA ranks second lowest), which suffered significantly from Hurricane Katrina in 2005. Up to now, the county is still below half of its population before the hurricane.

On the other hand, we want to include some covariate in the model to see whether these predictors have association effects on the outcomes. The predictor variables are listed in Table 2.6. All the predictor variables are normalized to have

Name	Minimum	Maximum	Median
Poverty Rate	3.1	34.8	13.0
Unemployment Rate	1.5	15.3	4.7
Population Change Rate	−76.9	66.7	2.6

Table 2.5: The outcomes (nodes in graph) to be analyzed for the US census Bureau data, all the values are in percentage.

mean 0 and variance 1 to put into the multivariate Bernoulli LASSO model tuned by all the four criteria discussed in Section 2.4.

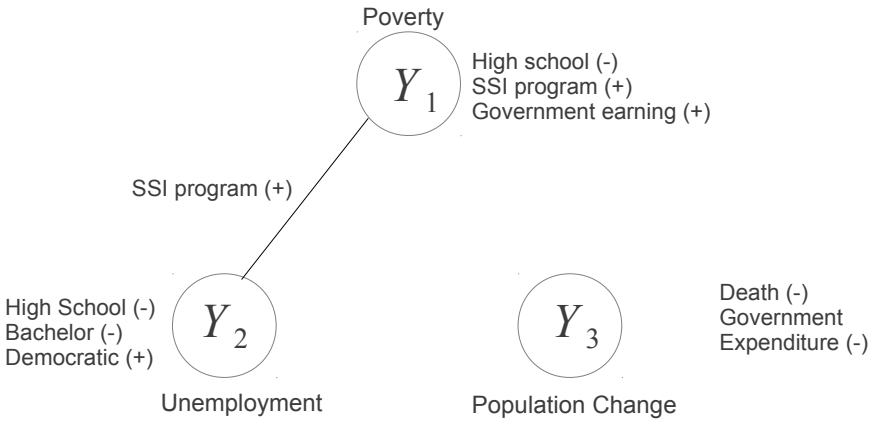

Figure 2.4: The graph structure fitted for the US Census Bureau data.

The list of coefficients estimated under the tuning criteria mentioned in Section 2.4 are shown in Tables B.1, B.2, B.3 and B.4. To better understand the graph fitted by the model, Figure 2.4 displays the result modeled by the multivariate Bernoulli LASSO and tuned by BGACV. The selected variables are noted near the corresponding nodes or edges, and the signs ($+$ or $-$) in the parentheses indicate

the direction of the influence of the covariate.

Interestingly, *Supplemental Security Income Recipients* (SSI) plays a very important role in the graph, which is a United States government program that provides stipends to low-income people who are either aged (65 or older), blind, or disabled. The edge connecting poverty and unemployment is significantly affected by supplemental security income recipients that the higher rate of recipients, the higher correlation between poverty rate and unemployment rate. On the nodes level, the poverty rate is related to high school education people, supplemental security income recipients and personal income. This indicates that the higher education level (more people with high school diploma) or the better the personal income, the lower the poverty level. The high percentage of supplemental security income recipients in the county is also an indicator of more low-income residents in the area, resulting in higher poverty rate. In addition, the high school education level and bachelor degree recipients number negatively affect the unemployment level or in other words, the education level can decrease the unemployment rate of the population. Moreover, the the more Democratic supporters in the county, the lower the employment rate. Finally, the population change is more likely due to death rate and government expenditure of the region. There are also edges and clique effect in the graph but the predictor variables are not very strong, the estimated coefficients for BGACV are listed in Table B.4.

The analysis of the results tuned by other criteria is deferred to **Appendix**.

Table 2.6: Predictor Variables used in the model.

Index	Name	Minimum	Maximum	Median
1	Retired Rate	3.79%	33.71%	14.23%
2	Under 18 Year-old Rate	18.8%	62.7%	35.1%
3	High School Education Rate	34.7%	97%	79.3%
4	Bachelor Education Rate	5.4%	60.5%	14.9%
5	Foreign Born Population Rate	0.1%	50.9%	1.9%
6	Birth Rate	4.3%	27.6%	12.4%
7	Death Rate	1.7%	23.6%	9.8%
8	Medical Program Enrolled Rate	2.72%	60.85%	16.70%
9	Social Security Program Beneficiaries Rate	5.05%	41.94%	19.79%
10	Supplemental Security Income Recipients Rate	0.13%	15.23%	2.14%
11	Personal Income Per Capita	$5,148	$89,028	$26,432
12	Retail Trade Per Capita	$0	$69,588	$8,033
13	Accommodation and Food per Capita	$0	$130,742	$826
14	Government Expenditure per Capita	$1,206	$105,868	$6,056
15	Government Earning	$2.6	$126.5	$20.1
16	Government Employment Rate	4.3%	61.2%	14.6%
17	Republican Rate	15.2%	92%	60.8%
18	Democratic Rate	7.1%	83%	38.3%

Chapter 3

Multivariate Bernoulli Mixed-Effects Models

3.1 Introduction

This Chapter considers the extension of the generalized linear mixed-effects models (GLMMs) in Pinheiro and Chao (2006) for grouped data with single level to the multivariate Bernoulli logistic model discussed in Section 1.5. The mixed effects models provide powerful and useful tools for analyzing data with repeated measurements, which often arise in different real applications such as economics and clinical trials. When a number of subjects are repeated observed under varying conditions, the simple linear or generalized linear models fail to explain the underlying process of the so-called repeated-measured data. The mixed effects models assume that there are both fixed and random effects in the model but the responses are independent conditioning on the random effects. This allows the model to

handle unbalanced repeated-measured data with flexible variance-covariance structure.

Let us consider the simplest case, the linear mixed effects model (LME). Suppose the observations come from M groups and the ith group contains n_i of them. The linear mixed effects model formulation thus can be written as

$$y(i) = X(i)\beta + Z(i)b_i + \epsilon_i, \quad \text{for } \ i = 1, \ldots, M, \tag{3.1}$$

where $y(i)$ is the response vector of length n_i for group i. $X(i)$ and $Z(i)$ are design matrices for fixed effects β of length p and random effect b_i of length q, respectively. The error vector ϵ_i follow an independent Gaussian distribution $N(0, \sigma_\epsilon^2 I_{n_i})$, and I_{n_i} is the identity matrix of dimension $n_i \times n_i$. Importantly, the random effect b_i assumed to distribute as $N(0, \Psi)$, makes the key difference here since without b_i, the equation (3.1) is equivalent to a linear regression model.

The linear mixed-effects model (3.1) can be solved via maximum likelihood estimate since b_i is an additive term with Gaussian distribution. $y(i)$ still follows a Gaussian distribution although the global structure is no longer independent. However, in real applications especially in longitudinal studies, nonlinear trend of response with respect to the covariate is crucial to prediction accuracy. The more general nonlinear mixed-effects models (NLMMs) has the form

$$y(i) = h(\Phi(i)) + \epsilon_i \quad \text{for } \ i = 1, \ldots, M, \tag{3.2}$$

where h is the known nonlinear function. Some of the widely used nonlinear functions h are listed in the appendix of Pinheiro and Bates (2000). $\Phi(i)$ is the

parameter vector defined as

$$\Phi(i) = X(i)\beta + Z(i)b_i, \quad \text{for } \; i = 1, \ldots, M,$$

where $X(i)$, $Z(i)$, β and b_i are the same as in (3.1). The biggest challenge to solve the NLMMs (3.2) is that the likelihood function involves an integral of b_i's and in general no analytical solution. There are several ways to deal with this and two most popular ones are through Monte Carlo Markov Chains and quadrature approach. In this thesis, we mainly focus on the quadrature method studied in Pinheiro and Bates (1995) for nonlinear mixed-effects models. Similarly, the generalized linear mixed-effects models also involves a non-analytical integral needed to be approximated and we defer this to later sections since it is more relevant to the multivariate Bernoulli mixed-effects models. There are already many well-developed softwares available to solve the nonlinear or generalized linear mixed-effects models. Notably the `lme4` (Bates et al., 2012) based on S4 class of `R` (R Development Core Team, 2005).

This Chapter is organized as follows. A brief review of the generalized linear mixed effects model is presented in Section 3.2, where the notation and the likelihood estimation in the context of the multivariate Bernoulli distribution are introduced. However, as the target function involves integrals without explicit form, we devoted the Section 3.3 to discussion of the Laplacian approximation to the marginal log-likelihood function, which is one of the efficient ways to solve this kind of problems.

3.2 Model Formulation

In this thesis, we only consider single-level generalized linear mixed-effects models for the multivariate Bernoulli logistic model, and the multi-level model analysis can be found in Pinheiro and Chao (2006). As mentioned in Section 3.1, the data the model is particularly designed for often involves several groups of observations, so we denote the response matrix to be $y(i)$, which has dimension $n_i \times K$ for a given group i in the multivariate Bernoulli environment. Conditioning on the random effects b_i, the rows of $y(i)$ are distributed as independent multivariate Bernoulli distribution like in the multivariate Bernoulli logistic model discussed in Section 1.5. Notice the change of notation here, the matrix $y(i)$ stands for observed outcomes from a group of subjects, for instance, people from a same pedigree (family). Specifically, $y(i,j)$ refers to the jth sample in group i.

Given b_i, the conditional density of $y(i)$ is

$$p(y(i)|b_i) = \prod_{j=1}^{n_i} \exp\left\{\left(\sum_{r=1}^{K} \sum_{1\le j_1\le\cdots\le j_r\le K} f^{j_1\cdots j_r}(i) B_{j_1\ldots j_r}(y(i,j))\right) - b(f(i))\right\} \quad (3.3)$$

In addition, the natural parameter f's in the multivariate Bernoulli logistic model (1.34) especially the linear predictor (1.33) is further modified to have the form

$$f^\tau(i) = X(i)\beta^\tau + Z(i)b_i^\tau, \quad (3.4)$$

where $X(i)$ and $Z(i)$ are the design matrix of fixed effects and mixed effects for group i, respectively. $\beta^\tau = (c_0^\tau, \ldots, c_p^\tau)$ is the coefficients vector for the fixed effects,

which is similar to the coefficients vectors $\{c_j^\tau\}$ in (1.33). Moreover, the random effects b_i^τ are indexed by τ for flexibility and it is further assumed that b^τ has the distribution $N(0, \Psi^\tau)$, where Ψ^τ's for different τ are not necessarily equal. For instance, in the bivariate Bernoulli distribution, there are only three random effects vectors b_i^1, b_i^2 and b_i^{12}. They are independent with different distributions denoted as $N(0, \Psi^1)$, $N(0, \Psi^2)$ and $N(0, \Psi^{12})$. It then follows that the joint density of $(y(i), b_i)$ is given by

$$p(y(i), b_i) = p(y(i)|b_i) \prod_{\tau \in \mathcal{T}} \frac{1}{(2\pi)^{q/2}|\Psi^\tau|^{1/2}} \exp\left[-(b_i^\tau)^T (\Psi^\tau)^{-1} b_i^\tau / 2\right], \tag{3.5}$$

where q is the number of random effects (length of b_i).

As with any mixed-effects models, the random effects are non-observable quantities, likelihood estimation must rely on the marginal density of $y(i)$, which is obtained by integrating the joint likelihood function with respect to b_i. Specifically, the target function for the multivariate Bernoulli mixed model is

$$p(y(i)|\beta) = \int_{-\infty}^{\infty} p(y(i)|b_i) \prod_{\tau \in \mathcal{T}} \frac{1}{(2\pi)^{q/2}|\Psi^\tau|^{1/2}} \exp\left[-(b_i^\tau)^T (\Psi^\tau)^{-1} b_i^\tau / 2\right] \prod_{\tau \in \mathcal{T}} db_i^\tau. \tag{3.6}$$

However, similar to NLMMs, for GLMMs integral (3.6) does not have a closed form expression and approximations are required for computationally feasible estimation. This thesis extends the Laplacian approximation methods to NLMMs and GLMMs to the multivariate Bernoulli Mixed-effects models as discussed in the next section.

3.3 Laplacian Approximation

Laplacian approximations are frequently used in Bayesian inference to estimate marginal posterior densities and predictive distributions. The use of the Laplacian approximation for single-level nonlinear mixed effects models was described by Pinheiro and Bates (1995). The technique can also be used for approximating the log-likelihood function in GLMMs studied in Pinheiro and Chao (2006) and this paper extends it to the multivariate Bernoulli logistic model as stated in the following. The marginal likelihood can be expressed as

$$\begin{aligned} p(y(i)|\beta) &= \int p(y(i), b_i|\beta) db_i \\ &= (2\pi)^{-\frac{q|\mathcal{T}|}{2}} \prod_{\tau\in\mathcal{T}} |\Psi^\tau|^{-1/2} \int \exp[g(\beta, \Psi, y(i), b_i)] \prod_{\tau\in\mathcal{T}} db_i^\tau, \quad (3.7) \\ g(\beta, \Psi, y(i), b_i) &= \sum_{j=1}^{n_i} \left(\sum_{r=1}^{K} \sum_{1\le j_1\le\cdots\le j_r\le K} f^{j_1\ldots j_r} B_{j_1\ldots j_r}(y(i,j)) - b(f) \right) \\ &\quad - \sum_{\tau\in\mathcal{T}} (b_i^\tau)^T (\Psi^\tau)^{-1} b_i^\tau/2. \end{aligned}$$

The idea behind the Laplacian and to some extent the Gaussian Hermitian approximation, is to approximate $g(\beta, \Psi, y(i), b_i)$ by a second-order Taylor expansion around the value of b_i that maximizes $g(\cdot, b_i)$. Note that

$$\begin{aligned} \frac{\partial g(\beta, \Psi, y(i), b_i)}{\partial b_i^\tau} &= Z(i)^T [B^\tau(y(i)) - \frac{\partial b(f)}{\partial f^\tau}] - (\Psi^\tau)^{-1} b_i^\tau, \quad (3.8) \\ \frac{\partial^2 g(\beta, \Psi, y(i), b_i)}{\partial (b_i^\tau)^2} &= -[Z(i)^T \frac{\partial^2 b(f)}{\partial (f^\tau)^2} Z(i) + (\Psi^\tau)^{-1}]. \quad (3.9) \end{aligned}$$

The first and second derivatives of $b(f)$ in the previous formula can be induced from (1.28) and (1.29). It follows from (3.9) that $\partial^2 g(\beta, \Psi, y(i), b_i^\tau)/\partial (b_i^\tau)^2$ is negative-

definite and, as a result, $g(\cdot, b_i^\tau)$ is strictly concave function of b_i^τ. Therefore, there exists a unique $\hat{b_i^\tau}$ corresponding to $\partial g(\cdot, b_i^\tau)/\partial b_i^\tau|_{b_i^\tau = \hat{b_i^\tau}} = 0$. Equation (3.8) provides a recursive formula to determine $\hat{b_i^\tau}$

$$(\hat{b_i^\tau})^{(k+1)} = \Psi^\tau Z(i) \left(B^\tau(y(i)) - \frac{\partial b(f)}{\partial f^\tau} \right) \tag{3.10}$$

where the iterations start at $(\hat{b_i^\tau})^{(0)}$.

The integral in (3.7) then can be approximated by Laplacian method

$$\int \exp[g(\beta, \Psi, y(i), b_i)] \prod_{\tau \in \mathcal{T}} db_i^\tau \quad \approx \quad \frac{(2\pi)^{\frac{q|\mathcal{T}|}{2}} \exp[g(\beta, \Psi, y(i), \hat{b}_i)]}{\sqrt{|\partial^2 g(\beta, \Psi, y(i), b_i^\tau)/\partial(\hat{b}_i)\partial(\hat{b}_i)^T|}}$$

Given the assumption that the b_i^τ are mutually independent, the Laplacian approximation to the single-level GLMM negative log-likelihood corresponding to group i is then given by

$$l_{\text{Lap}}(\beta, \Psi^\tau | y(i)) = \sum_{\tau \in \mathcal{T}} [\log |\Psi^\tau| + \log \left| \frac{\partial^2 g(\beta, \Psi, y(i), b_i^\tau)}{\partial (b_i^\tau)^2} \right| - 2g(\beta, \Psi^\tau, y(i), \hat{b_i^\tau})]$$

The objective negative log-likelihood to be

$$l_{\text{Lap}}(\beta, \Psi^\tau | y) = \sum_{i=1}^{M} l_{\text{Lap}}(\beta, \Psi^\tau | y(i)) \tag{3.11}$$

where M is the number of groups.

3.4 Analysis of Census Bureau Data

As discussed and examined in Section 2.5, several outcomes listed in Table 2.5 for the US Census Bureau Data (US Census Bureau, 2007) are interconnected and are affected by other predictor variables in Table 2.6. Although the multivariate Bernoulli LASSO model is capable of capturing both the significant variables and the graph structure as demonstrated in the simulations of Section 2.5, it fails to consider the special structure of the dataset especially for the US Census Bureau Data.

As mentioned before, the data set includes observational quantities from over 2000 counties in all 50 states of US, but it is also paramount to consider the mutual influences among the counties. The independence of the counties within the same state is challenged by the fact that the counties share the same government fiscal policy of state level and without doubt correlated more than out-of-state counties. Therefore, we consider a random intercept on the state level for all the f functions in the model.

Outcome	Variance
Poverty Rate	0.076
Unemployment Rate	0.076
Population Change Rate	0.083
Poverty * Unemployment	0.074
Poverty * Pop Change	0.086
Unemployment * Pop Change	0.088
p value	0.186

Table 3.1: Estimated variance of random effects for both node and edge effects in US census Bureau data.

The multivariate Bernoulli mixed-effects model is applied to the data with the

same outcomes and covariate as in Section 2.5. Since there is very weak third order clique effect in the graph, we restrict our attention only on node and edge effects. This is plausible due to difficulty of interpreting random effect on clique levels.

Table 3.1 illustrate the estimated variance of the random effects on the nodes and the edges. The variation is small in general, which indicates there are not much difference between the counties from different states. This can also be seen from the p value 0.186 shown in the table. The p value is calculated via the likelihood ratio test and based on χ^2 test with degrees of freedom 6.

Chapter 4

R Packages

4.1 Introduction

The statistical models for data analysis are important since without rigorous theory, there is no guarantee the results will have power and efficiency in real applications. Nevertheless, it is also paramount to build reliable software to fit general use of the statistical models and this chapter is designed for two `R` packages (R Development Core Team, 2005) for two different models.

This chapter is organized with two sections. Section 4.2 introduces the development of `MVB`, which stands for the multivariate Bernoulli. The package includes the necessary functions to implement the models introduced in Chapters 1, 2 and 3. In addition, Section 4.3 is devoted to numerical examples of another optimization technique called orthogonalizing expectation maximization (OEM) algorithm. The approach is designed for fitting regularized linear regression by utilizing the popular EM technique proposed by Dempster et al. (1977). The results are built on

Xiong et al. (2011).

4.2 Multivariate Bernoulli Fitting

The R package MVB, is short for *Multivariate Bernoulli*. It is developed to implement the functionality studied in this thesis. To speed up the calculation, the main process of the algorithm is written in objective-oriented C++ with the help of powerful R package Rcpp (Eddelbuettel and Francois, 2011). In addition, as the algorithms involve significant linear algebra calculations, several routines of RcppArmadillo (Armadillo Project, 2012), (Sanderson, 2010) are linked to carry out matrix manipulations.

Table 4.1 displayed the functions in the packages and their descriptions.

Name	Description
unifit	Fit generalized linear model (Gaussian and binomial)
unilps	Fit generalized linear LASSO model using LASSO pattern search
mvbfit	Fit multivariate Bernoulli model
mvblps	Fit multivariate Benroulli LASSO model
stepfit	Fit step-wise (forward or backward) multivariate Benroulli model
mvbme	Fit multivariate Bernoulli mixed effects model
plot	Plot solution path for lps object

Table 4.1: Functions in package MVB.

The documentation for the functions are also available in the package with detailed functionality and arguments. The package is published on CRAN (http://cran.r-project.org).

4.3 Orthogonalizing EM

Introduction

Recently, non-convex penalized regression problem such as smoothly clipped absolute deviation (SCAD) penalty proposed in Fan and Li (2001) and minimax concave penalty (MCP) discussed in Zhang (2010) gains popularity in statistical machine learning. However, these penalized regression problems have multiple local optima, as a result, finding a local solution to achieve the so-called oracle property is an open problem. An iterative algorithm is proposed, called the orthogonalizing EM (OEM) algorithm, to fill this gap. The development of the algorithm draws direct impetus from a missing-data problem arising in design of experiments with an orthogonal complete matrix. In each iteration, the algorithm imputes the missing data based on the current estimates of the parameters and updates a closed-form solution associated with the complete data. By introducing a procedure called active orthogonization, the algorithm is broadly applicable to problems with arbitrary regression matrices. In addition to the SCAD and MCP, the proposed algorithm works for other penalties such as LASSO and nonnegative garrote. Convergence and convergence rate of the algorithm are examined in Xiong et al. (2011). For various penalties, an OEM sequence converges to a point having grouping coherence for fully aliased regression matrices. For computing the ordinary least squares estimator with a singular regression matrix, an OEM sequence converges to the Moore-Penrose generalized inverse-based least squares estimator.

This section is organized into two subsections. First, Subsection **Model Formu-**

lation formulates the penalized least squares problem and discusses the procedure for OEM algorithm. Then Subsection **Numerical Examples** illustrates the group coherence property of the algorithm with several simulation examples and compare the efficiency of OEM and coordinates descent studied in Friedman et al. (2010) and Breheny and Huang (2011) under different scenarios.

Model Formulation

Consider the linear model,

$$Y = X\beta + \epsilon, \tag{4.1}$$

where Y is the response vector of length n and X is the design matrix of size $n \times p$. $\beta = (\beta_1, \ldots, \beta_p)'$ is the vector of regression coefficients and the distribution of the error term is $\epsilon \sim N(0, \sigma_\epsilon I_n)$. The general form for a penalized least square problem is to solve a minimization problem involving sum of squares and penalty of regression coefficients,

$$\min_{\beta} \left[||Y - X\beta||^2 + 2\sum_{j=1}^{p} P_\lambda(|\beta_j|) \right], \tag{4.2}$$

where P_λ is penalty function tuned by λ.

The choice of the P_λ function is important in statistical machine learning and there are several different ones proposed and already proved to be efficient and powerful. For instance, when $P_\lambda(|\beta_j|) = \lambda|\beta_j|$, it is the LASSO penalty discussed

in Chapter 2. Moreover, the SCAD penalty has the form

$$P'_\lambda(\theta) = \lambda I(\theta \leq \lambda) + (a\lambda - \theta)_+(\theta > \lambda)/(a-1), \text{ for } \theta > 0, \tag{4.3}$$

here $a > 2, \lambda > 0$ are the tuning parameters, and I is the indicator function.

In statistical design theory, orthogonalization is important and in most experiments a desired property. Details can be referred to statistical experiments design textbooks such as Box and Draper (2007) and Wu and Hamada (2009). Therefore, the OEM is motivated by orthogonalizing the design matrix X in (4.2) and as a result, the problem has a trivial explicit solution. The orthogonalization process involves imputing more observations, in other words, the original design matrix X is a sub-matrix of an $m \times p$ complete orthogonal matrix

$$X_c = (X^T \ \Delta^T)^T, \tag{4.4}$$

where δ is the $(m-n) \times p$ *missing matrix*. At the same time, the response vector is also needed to be imputed to

$$Y_c = (Y^T \ Y_\Delta^T)^T, \tag{4.5}$$

where Y_Δ is defined corresponding to Δ. If Y_Δ is observable, then the ordinary least squares estimator of β based on the complete data (Y_c, X_c) has a closed form solution. In light of this fact, Healy and Westmacott (1956) developed an iterative procedure to compute ordinary least squares estimator β_{OLS}. The OEM algorithm follows the same paradigm but solving the penalized least squares problem (4.2)

instead. The idea called active orthogonalization is to expand the design matrix into an orthogonal matrix and thus obtains the explicit solution of the problem.

Define

$$A = \Delta^T \Delta. \tag{4.6}$$

Let $(d_1, \ldots, d_p)$ be the diagonal elements of $X_c^T X_c$. The OEM algorithm to solve (4.2) proceeds as follows. Denote $\beta^{(0)}$ as the initial estimate of β, then for $k = 1, 2, \ldots$, impute Y_Δ as $Y_\Delta = \Delta\beta^{(k)}$, and solve

$$\beta^{(k+1)} = \text{argmin}_\beta \left[||Y_c - X_c\beta||^2 + 2\sum_{j=1}^{p} P_\lambda(|\beta_j|) \right], \tag{4.7}$$

until $\{\beta^{(k)}\}$ converges. Letting

$$u = (u_1, \ldots, u_p)^T = X^T Y + A\beta^{(k)},$$

equation (4.7) can be rewritten as

$$\beta^{(k+1)} = \text{argmin}_\beta \left[\sum_{j=1}^{p} (d_j\beta_j^2 - 2u_j\beta_j) + 2\sum_{j=1}^{p} P_\lambda(|\beta_j|) \right], \tag{4.8}$$

which is separable in components of β.

Generally, the design matrix X used in (4.2) is standardized to have mean 0 and variance 1 with $d_j \geq 1$ for all j. Various penalized least squares can be solved by this iterative procedure and Xiong et al. (2011) lists explicit solutions for several popular penalties including the LASSO (Tibshirani, 1996), the nonnegative garrote

(Breiman, 1995), the elastic-net (Zou and Hastie, 2005) and the MCP (Zhang, 2010). These penalties are implemented in the `R` package `oem`.

Numerical Examples

In this subsection, we illustrate the algorithm using simulation examples, to assess both statistical properties and computational efficiency of OEM. As a general optimization approach, OEM can be employed to both penalized and unpenalized least squares problems and we show its application to SCAD and generalized inverse by comparing it with several existing `R` packages.

Group Coherence

The optimization problem for LASSO is convex but not strictly when there are predictor variables perfectly correlated. The coordinate descent (CD) algorithm and LARS utilize a marginal approach so only one of the group of correlated variables will be included in the final solution path whereas all the others will be estimated as zero for any tuning parameter λ. However, the OEM algorithm examines all the predictor variables in every iteration, and consequently will give equal weights to all the components in the perfectly correlated group of predictor variables. Therefore, the OEM exhibits a different solution path from CD on the same data set although both algorithms land on local minimum of the target function.

Consider a data set with 4 predictor variables in which case the variables X_1 and X_2 are generated from independent standard normal distributions. In addition, the degenerated design matrix is formulated by $X_3 = -X_1$ and $X_4 = -X_2$. In other

words, the predictors consist two pairs of perfectly negative correlated random variables. The true linear relationship between the response and the predictors is

$$Y = X_3 + 2X_4 \tag{4.9}$$

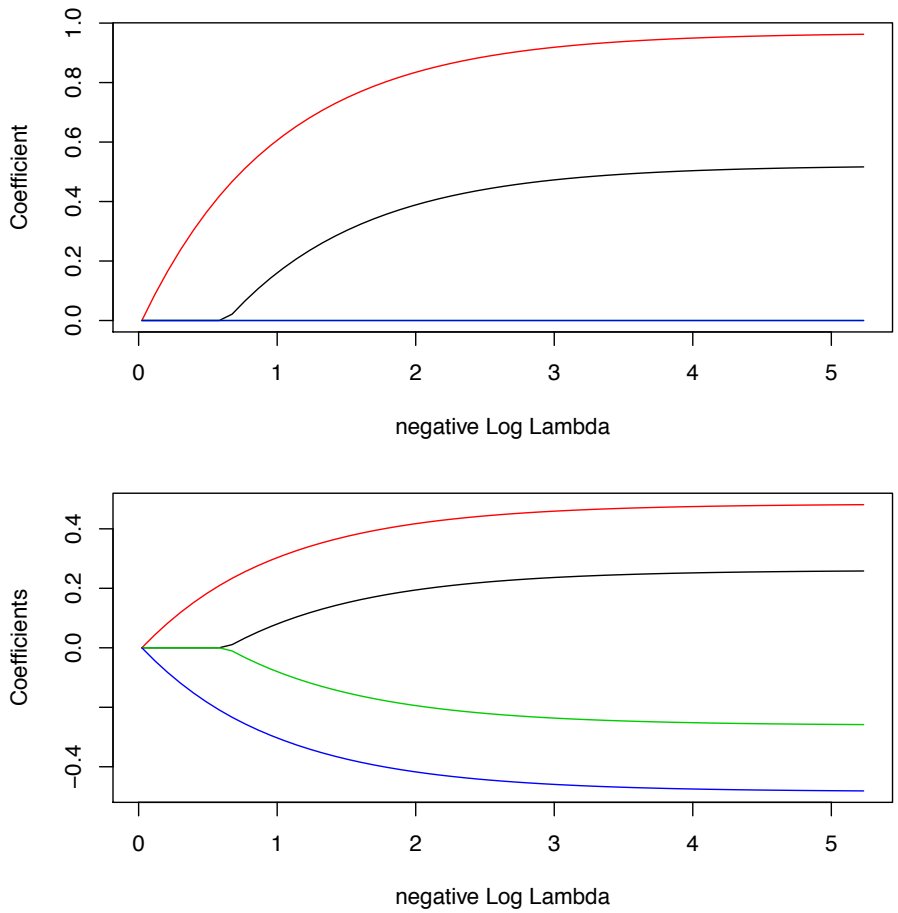

Figure 4.1: Solution paths of LASSO fitted by CD (from package `glmnet`) in the upper panel and OEM for the lower panel.

Figure 4.1 displays the solution paths for this data set using LASSO fitted from

R packages glmnet and oem on the same set of tuning parameters λ. Notice here the package lars gives the exactly same solution path as glmnet. A close scrutiny of the solutions reveals that OEM estimates the perfectly negative correlated pairs to have exactly the opposite signs but CD only has X_1 and X_2 in the model and fixes X_3 and X_4 to be zero for any λ. This effect is due to the fact that in every iteration, both CD and LARS will find the predictor with the largest improvement on the target function and if more than one coordinates can give a better residual sum of squares, only the one with the smallest index will enter the model. On the other hand, OEM considers all the predictors in every iteration so the ones with same contribution to the target will receive equal steps. One remark to the group coherence is that this property is automatically maintained for group LASSO when the perfectly correlated variables reside in the same group since they are penalized by l_2 or so-called ridge penalty instead of l_1. Nevertheless, group LASSO still suffers this not strictly convex problem when the correlated predictors appear in different groups where the penalty is still of l_1 structure.

This group coherence property of OEM over CD is also true for non-convex penalties such as SCAD, with the solution paths shown in figure 4.2

Computational Efficiency

As mentioned in this paper, OEM is suitable for various penalized least squares problem. The R package glmnet implemented in Fortran is the fastest in fitting LASSO and is superior in most scenarios to our package oem, which is written in C++ . In this section, we compare computational efficiency of OEM and the package ncvreg developer in C on SCAD penalty.

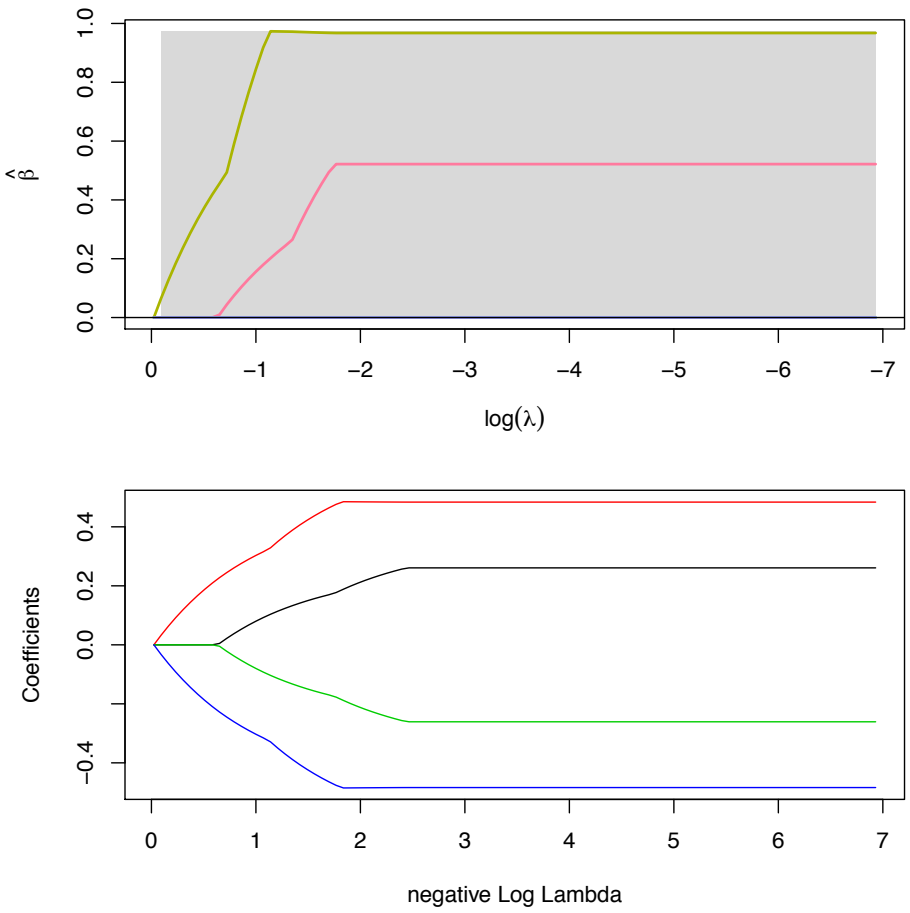

Figure 4.2: Solution paths of SCAD fitted by CD (from package `ncvreg`) in the upper panel and OEM for the lower panel.

To begin with, we consider the cases when the sample size n is significantly larger than the number of variables p. Three different covariance matrices structures were considered for the predictor variables. The first is the case where all the variables are independently generated from standard normal distribution, the second and third cases involve design matrix with correlations structure among

the covariate

$$\text{Cor}(X_i, X_j) = \rho^{|i-j|} \text{ for } i, j = 1, \ldots, p$$

We consider two special examples when $\rho = 0.2, 0.8$ in the following simulations. On the other hand, the response is generated independent of the design matrix therefore the true model is

$$Y = 0 \tag{4.10}$$

To compare the performance of OEM and CD algorithms for SCAD penalty, the data are generated 10 times and the average runtime in seconds are shown in table 4.2. Here the CD algorithm was carried out in `ncvreg` and OEM was implemented by `oem`.

p	n	OEM			CD		
		$\rho = 0$	$\rho = 0.2$	$\rho = 0.8$	$\rho = 0$	$\rho = 0.2$	$\rho = 0.8$
20	400	0.0052	0.0059	0.0240	0.0451	0.0245	0.0209
	1000	0.0061	0.0073	0.0262	0.0449	0.0516	0.0452
	2000	0.0088	0.0099	0.0277	0.0826	0.0927	0.0844
50	1000	0.0189	0.0261	0.1803	0.1398	0.1437	0.1797
	2500	0.0311	0.0380	0.1918	0.4483	0.4808	0.4613
	5000	0.0609	0.0689	0.2291	0.833	0.9233	0.8912
100	2000	0.0946	0.1193	1.0037	0.8865	0.8612	0.9964
	5000	0.1689	0.2085	1.1002	2.2004	2.4043	2.691
	10000	0.4551	0.5342	1.2832	4.8513	5.6488	7.7149

Table 4.2: Average runtime in seconds comparison between OEM and CD for SCAD when n is larger than p.

From the table, it is not hard to see that OEM has advantages when the sample

size is significantly larger than the number of variables especially for independent design. On the other hands, both algorithms require more fitting time when the correlations among the covariate increase but OEM still keeps the lead.

What's more, with recent improvement of techniques in genetic studies, there are more and more data sets with large p small n. We also compared these two aforementioned algorithms in this scenario as illustrated in table 4.3. It turns out that the coordinate descent algorithm is faster and the computational gap gets wider when the ratio of p/n increases.

p	n	OEM			CD		
		$\rho = 0$	$\rho = 0.2$	$\rho = 0.8$	$\rho = 0$	$\rho = 0.2$	$\rho = 0.8$
200	100	0.8425	0.9703	1.2412	0.1430	0.1584	0.1505
	200	6.2634	6.784	8.4708	0.4800	0.4728	0.462
	400	3.0315	3.2366	6.7653	0.8429	0.8311	1.0044
500	250	4.7629	5.1432	7.0855	1.3622	1.421	1.1643
	500	51.338	51.941	57.536	4.4924	4.4082	3.4217
	1000	31.070	32.631	54.069	10.175	9.0097	9.8956
1000	500	7.8277	8.6695	23.833	8.5252	8.1363	7.2247
	1000	741.54	978.13	1063.7	64.216	67.935	45.511
	2000	658.19	676.82	739.18	152.80	129.01	100.25
1200	100	14.313	12.049	14.722	0.9061	0.8197	0.9102
	150	20.443	15.972	18.676	1.8636	1.3811	1.3246
	240	24.885	20.313	24.714	3.6128	2.7939	2.5308

Table 4.3: Average runtime in seconds comparison between OEM and CD for SCAD for large p.

A close scrutiny of the two algorithms reveals that they take similar number of iterations but the computation of OEM required a one time computation of matrix multiplication $X^T X$ and the complexity of this process is $O(np^2)$, which dominates

the algorithm especially when p is very large. This is the main drawback of the OEM algorithm.

Generalized Inverse

OEM algorithm as a general optimization approach does not only apply to penalized least squares problem but also suitable for solving ordinary least squares problem. When the design matrix is of full rank, the QR decomposition is proved to be efficient but we concentrate on finding the solution of generalized inverse when the design matrix is degenerated. Moore-Penrose pseudo-inverse is widely used as a best fit to system of linear equations when there is no unique solution and when applied to this degenerated system, OEM converges to the Moore-Penrose pseudo-inverse.

n	p	OEM	`pinv`
	10	0.0433	0.0956
	50	0.2439	0.4098
$50,000$	200	1.4156	4.9765
	1000	5.4165	45.3270
	$5,000$	72.0630	442.3300

Table 4.4: Average runtime in seconds comparison among OEM and generalized inverse for $n > p$.

In `MATLAB`, the function `pinv` is used for computing Moore-Penrose pseudo-inverse. The approach is to solve the singular value decomposition of design matrix X. The method is thus prohibitive when both dimensions p and n of X is large. Tables 4.4 and 4.5 compare `oem` and `ginv` in solving MooreâĂŞPenrose pseudo-inverse of degenerated linear systems. The data was generated with X

following independent standard normal distribution but the response Y from another standard normal distribution which is independent of X like in 4.10. A new predictor variable calculated as the mean of all the other covariate is added to degenerate the design matrix so that there is no unique solution to the ordinary least square problem.

p	n	OEM	`pinv`
	10	0.0482	0.1153
	50	0.4203	0.4176
$50,000$	200	1.9159	5.2053
	1000	8.4626	47.7653
	$5,000$	71.8477	440.6741

Table 4.5: Average runtime in seconds comparison among OEM and generalized inverse for $p > n$.

Both OEM and `pinv` will converge to Moore-Penrose pseudo-inverse solution and their solutions differ by less than the tolerance of the algorithms. From the tables, it can be inferred that `pinv` is efficient when any dimension of the design matrix not too large, which determines the complexity of singular value decomposition. On the other hand, OEM is superior in any combinations of n and p since the eigenvalue we need in the procedure of OEM only requires power method applied to X^TX or XX^T so only the smaller of them will be computed. Thus, the complexity of OEM is dependent also on the smaller of n and p but in a more efficient way as illustrated in 4.4 and 4.5. Notice here the `R` function `ginv` can also be applied to this problem but is significantly slower than `pinv`.

DISCARD THIS PAGE

Appendix A

Proofs

Proof. of **Proposition 1.1**

With the joint density function of the random vector (Y_1, Y_2), the marginal distribution of Y_1 can be derived

$$\begin{aligned} P(Y_1 = 1) &= P(Y_1 = 1, Y_2 = 0) + P(Y_1 = 1, Y_2 = 1) \\ &= p_{10} + p_{11}. \end{aligned}$$

Similarly,

$$P(Y_1 = 0) = p_{00} + p_{11}.$$

Combining the side condition of the parameters p's,

$$P(Y_1 = 1) + P(Y_1 = 0) = p_{00} + p_{01} + p_{10} + p_{11} = 1.$$

This demonstrates that Y_1 follows the univariate Bernoulli distribution and its density function is (1.1).

Regarding the conditional distribution, notice that

$$\begin{aligned} P(Y_1 = 0|Y_2 = 0) &= \frac{P(Y_1 = 0, Y_2 = 0)}{P(Y_2 = 0)} \\ &= \frac{p_{00}}{p_{00} + p_{10}}, \end{aligned}$$

and the same process can be repeated to get

$$P(Y_1 = 1|Y_2 = 0) = \frac{p_{10}}{p_{00} + p_{10}}.$$

Hence, it is clear that with condition $Y_2 = 0$, Y_1 follows a univariate Bernoulli distribution as well. The same scenario can be examined for the condition $Y_2 = 1$. Thus, the conditional distribution of Y_1 given Y_2 is given as (1.11). □

Proof. of **Lemma 1.2**

Expand the log-linear formulation of the bivariate Bernoulli distribution (1.9) into factors

$$P(Y_1 = y_1, Y_2 = y_2) = p_{00} \exp(y_1 f^1) \exp(y_2 f^2) \exp(y_1 y_2 f^{12}). \tag{A.1}$$

It is not hard to see that when $f^{12} = 0$, the density function (A.1) is separable to two components with only y_1 and y_2 in them. Therefore, the two random variables corresponding to the formula are independent. Conversely, when Y_1 and Y_2 are independent, their density function should be separable in terms of y_1 and y_2, which implies $y_1 y_2 f^{12} = 0$ for any possible values of y_1 and y_2. The assertion

dictates that f^{12} is zero. □

Proof. of **Lemma 1.3**

Consider the log-linear formulation (1.16), the natural parameters f's are combined with products of some components of y. Let us match terms in the $f^{j_1 \cdots j_r} B^{j_1 \cdots j_r}(y)$ from log-linear formulation (1.16) with the coefficient for the corresponding product $y_{j_1} \ldots y_{j_r}$ terms in (1.13). The exponents of p's in (1.13) can be expanded to summations of different products $B^{\tau}(y)$ with $\tau \in \mathcal{T}$ and all the p's with $y_{j_1} \ldots y_{j_r}$ in the exponent have effect on $f^{j_1 \cdots j_r}$ so all the positions other than $j_1, \ldots j_r$ must be zero. Furthermore, those p's with positive $y_{j_1} \ldots y_{j_r}$ in its exponent appear in the numerator of $exp[f^{j_1 \cdots j_r}]$ and the product is positive only if there are even number of 0's in the positions $j_1, \ldots, j_r$. The same scenario applies to the p's with negative products in the exponents.

What's more, notice that $p_{00\ldots0} = b(\mathbf{f})$ and

$$\begin{aligned} \exp[S^{j_1 \cdots j_r}] &= \exp[\sum_{1 \le s \le r} f^{j_s} + \sum_{1 \le s < t \le r} f^{j_s j_t} + \cdots + f^{j_1 j_2 \cdots j_r}] \\ &= \prod_{1 \le s \le r} \exp[f^{j_s}] \prod_{1 \le s < t \le r} \exp[f^{j_s j_t}] \cdots \exp[f^{j_1 j_2 \cdots j_r}] \end{aligned} \tag{A.2}$$

and apply the formula for $\exp[f^{j_1 \cdots j_r}]$ with cancellation of terms in the numerators and the denominators. The resulting (1.18) can then be verified.

Finally, (1.19) is a trivial extension of (1.18) by exchanging the numerator and the denominator. □

Proof. of **Theorem 1.4**

Here, we take use of the moment generating function (1.25) but it is also possible to directly work on the probability density function (1.13). The mgf can

be rewritten as

$$\psi(\mu_1, \ldots, \mu_K) = \frac{1}{\exp[b(\mathbf{f})]} \sum_{r=1}^{K} \sum_{j_1 \le j_2 \le \ldots \le j_r} \exp[S^{j_1 j_2 \ldots j_r}] \prod_{k=1}^{r} \exp\left[\mu_{j_k}\right]. \tag{A.3}$$

It is not hard to see that this is a polynomial function of the unknown variables $\exp(\mu_k)$ for $k = 1, \ldots, K$. The independence of the random variables $Y_1, Y_2, \ldots, Y_K$ is equivalent to that (A.3) can be separated into components of μ_k or equivalently $\exp(\mu_k)$.

($\Rightarrow$) If the random vector Y is independent, the moment generating function should be separable and assume the formulation is

$$\psi(\mu_1, \ldots, \mu_K) = C \prod_{k=1}^{K} (\alpha_k + \beta_k \exp[\mu_k]), \tag{A.4}$$

where α_k and β_k are functions of parameters S's and C is a constant. If we expand (A.4) to polynomial function of $\exp[\mu_k]$ and determine the corresponding coefficients, (1.20) and (1.21) will be derived.

($\Leftarrow$) Suppose (1.21) hold, then we have

$$\exp[S^{j_1 j_2 \ldots j_r}] = \prod_{k=1}^{r} \exp[f^{j_k}],$$

and as a result, the moment generating function can be decomposed to product of

components of $\exp[\mu_k]$ like (A.4) with the following relations

$$\begin{aligned} C &= \frac{1}{\exp[b(\mathbf{f})]} \\ \alpha_k &= 1, \\ \beta_k &= \exp[f^k], \end{aligned}$$

□

Proof. of **Theorem 1.5**

The idea of proving the group independence of multivariate Bernoulli variables are similar to Theorem 1.4. Instead of decomposing the moment generating function to products of μ_k, we only have to separate them into groups with each only involving the dependent random variables. That is to say, the moment generating function with two separately independent nodes in the multivariate Bernoulli should have the form

$$\begin{aligned} \psi(\mu_1, \ldots, \mu_K) &= (\alpha_0 + \alpha_1 \exp[\mu_1] + \cdots + \alpha_r \exp[\mu_r]) \\ &\quad \cdot(\beta_0 + \beta_1 \exp[\mu_{r+1}] + \cdots + \beta_s \exp[\mu_K]). \end{aligned}$$

Matching the corresponding coefficients of this separable moment generating function and the natural parameters leads to the conclusion (1.22). □

Appendix B

US Census Bureau Results

Tables B.1, B.2, B.3 and B.4 display the results of US Census Bureau Data (US Census Bureau, 2007). The index shows the corresponding predictor variables in Table 2.6.

Similar to the discussion in Section 2.5, *Supplemental Security Income Recipients* plays the biggest role in both poverty and unemployment rates of the counties. The higher the supplemental security income recipient percentage in the area, the more low-income residents in the region, resulting in more poor and people without jobs.

Furthermore, it seems that the local politics have correlation with economic indicators such as poverty and unemployment rates. For instance, in models tuned by AIC, BIC and partially BGACV, the counties with more votes to Democratic party have high unemployment rate. This is hard to explain since it might be the case that the voters are disappointed with the current governing party (possibly Republicans) and move to favor other party. It is also likely that the Democrat as

the ruling party failed to promote employment. The covariate in the graph implies only association or co-occurrence instead of causality.

Table B.1: Estimated Coefficients tuned by AIC.

Index	Poverty (f^1)	Unemployed (f^2)	Pop Change (f^3)	f^{12}	f^{13}	f^{13}	f^{123}
0	**0.2489**	**−0.2660**	**−0.0312**	**0.3035**	0	**−0.0219**	0
1	**−0.1448**	**−0.2303**	**−0.2185**	0	0	0	0
2	**−0.1477**	0	**0.0370**	**−0.0349**	**−0.2487**	0	**−0.0385**
3	**−0.9159**	**0.0901**	**−0.2375**	**0.0204**	0	0	0
4	**0.3325**	**−0.7896**	**0.1469**	**0.0186**	0	0	0
5	0	0	**0.0357**	0	**0.0122**	**0.0453**	**0.1211**
6	**0.3648**	**−0.1340**	**0.0409**	**0.0193**	**0.0869**	0	**0.0145**
7	**0.1833**	**−0.0213**	**−0.8353**	0	**0.0500**	0	0
8	0	0	**−0.2361**	0	0	0	0
9	0	**0.0898**	**0.1486**	0	**0.0666**	**0.2104**	**0.0042**
10	**1.8793**	**0.2722**	**−0.1790**	**0.4743**	**0.0038**	**−0.0378**	**−0.1405**
11	**−0.6291**	**−0.3708**	0	0	**−0.3062**	**−0.0034**	0
12	**0.0714**	0	**0.1046**	**0.0673**	**0.0120**	0	**0.0223**
13	0	**0.0327**	**0.1413**	**0.0217**	0	**0.0426**	0
14	**0.0918**	**−0.0927**	**−0.1460**	**0.0279**	**−0.0636**	**−0.0986**	**−0.0626**
15	**0.1404**	0	**0.0436**	0	**−0.0164**	0	0
16	**0.2222**	0	**−0.3148**	0	**−0.0031**	0	0
17	0	**−0.2673**	**0.0210**	0	**0.0142**	**0.0364**	0
18	**0.0237**	**0.4795**	**−0.1010**	0	**−0.0594**	**−0.0888**	0

Table B.2: Estimated Coefficients tuned by BIC.

Index	Poverty (f^1)	Unemployed (f^2)	Pop Change (f^3)	f^{12}	f^{13}	f^{13}	f^{123}
0	$\mathbf{-0.0417}$	$\mathbf{-0.1115}$	$\mathbf{-0.0180}$	$\mathbf{0.0344}$	0	0	0
1	0	0	$\mathbf{-0.0658}$	0	0	0	0
2	0	0	$\mathbf{0.2853}$	0	$\mathbf{-0.0512}$	0	0
3	$\mathbf{-0.6438}$	$\mathbf{-0.0478}$	0	$\mathbf{-0.0554}$	0	0	0
4	$\mathbf{-0.0196}$	$\mathbf{-0.1416}$	$\mathbf{0.0899}$	0	0	0	0
5	0	$\mathbf{-0.0144}$	$\mathbf{0.0295}$	0	0	0	0
6	$\mathbf{0.1532}$	0	0	0	0	0	0
7	$\mathbf{0.0568}$	0	$\mathbf{-0.2321}$	0	0	0	0
8	0	$\mathbf{0.0401}$	$\mathbf{-0.1010}$	0	0	0	0
9	0	$\mathbf{0.0051}$	$\mathbf{-0.0453}$	0	0	0	0
10	$\mathbf{0.2999}$	$\mathbf{0.3584}$	$\mathbf{-0.1044}$	$\mathbf{0.1088}$	0	0	0
11	$\mathbf{-0.1088}$	$\mathbf{-0.0700}$	0	$\mathbf{-0.6700}$	0	0	0
12	0	0	$\mathbf{0.0409}$	0	0	0	0
13	0	0	0	0	0	0	0
14	0	0	$\mathbf{-0.0671}$	0	0	0	0
15	$\mathbf{0.1858}$	0	$\mathbf{-0.0316}$	0	0	0	0
16	$\mathbf{0.0110}$	0	$\mathbf{-0.0827}$	0	0	0	0
17	0	$\mathbf{-0.1168}$	$\mathbf{0.0012}$	$\mathbf{-0.3124}$	0	0	0
18	0	$\mathbf{0.0295}$	0	0	0	0	0

Table B.3: Estimated Coefficients tuned by GACV.

Index	Poverty (f^1)	Unemployed (f^2)	Pop Change (f^3)	f^{12}	f^{13}	f^{13}	f^{123}
0	**0.3148**	**−0.0440**	**−0.0292**	0	0	0	0
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	**−0.3820**	0	0	0	0	0	0
4	0	**−0.5737**	0	0	0	0	0
5	0	0	0	0	0	0	0
6	**0.0759**	0	0	0	0	0	0
7	0	0	**−0.7025**	0	0	0	0
8	0	0	**−0.1354**	0	0	0	0
9	0	0	0	0	0	0	0
10	**1.9529**	**0.4983**	**−0.0472**	0	0	0	0
11	**−0.4260**	**−0.1456**	0	0	0	0	0
12	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0
14	0	0	**−0.0717**	0	0	0	0
15	0	0	0	0	0	0	0
16	**0.1113**	0	**−0.1971**	0	0	0	0
17	0	0	**0.0378**	0	0	0	0
18	0	**0.4240**	0	0	0	0	0

Table B.4: Estimated Coefficients tuned by BGACV.

Index	Poverty (f^1)	Unemployed (f^2)	Pop Change (f^3)	f^{12}	f^{13}	f^{13}	f^{123}
0	**0.0960**	**−0.2517**	**−0.0339**	**0.2271**	0	0	0
1	0	0	**−0.0452**	0	0	0	0
2	0	0	0	0	**−0.0223**	0	0
3	**−0.7737**	**−0.1102**	0	0	0	0	0
4	0	**−0.6276**	**0.0158**	0	0	0	0
5	0	0	**0.0725**	0	0	0	0
6	**0.0557**	**−0.0697**	**0.0164**	0	**0.0112**	0	0
7	0	0	**−0.8091**	0	0	0	0
8	0	0	**−0.0384**	0	0	0	0
9	0	0	0	0	0	**0.0062**	0
10	**1.2300**	**0.0621**	**−0.0499**	**0.5489**	0	0	**−0.0131**
11	**−0.0926**	**−0.0666**	0	**−0.0259**	**−0.0679**	**−0.0050**	0
12	0	0	**0.0021**	**0.0033**	**0.0322**	0	0
13	0	0	**0.0253**	0	0	**0.0023**	0
14	0	0	**−0.2542**	0	0	**−0.0231**	0
15	**0.3059**	**0.0032**	**−0.0362**	0	**−0.0438**	0	0
16	**0.0263**	0	**−0.0806**	0	**−0.0025**	**−0.0289**	0
17	0	**−0.0686**	**0.0576**	**−0.0170**	**0.0278**	0	0
18	0	**0.3555**	**−0.0456**	**0.0208**	**−0.0312**	0	0

Bibliography

Armadillo Project (2012), `http://arma.sourceforge.net/`.

Banerjee, O., Ghaoui, L., and d'Aspremont A. (2008), "Model Selection Through Sparse Maximum Likelihood Estimation for Multivariate Gaussian or Binary Data," *Journal of Machine Learning Research*, 9, 485–516.

Bates, D., Maechler, M., and Bolker, B. (2012), "Linear mixed-effects models using S4 classes," `http://cran.r-project.org/web/packages/lme4/index.html`.

Bishop, C. (2007), *Pattern Recognition and Machine Learning*, Springer.

Box, G. and Draper, N. (2007), *Response Surfaces, Mixtures, and Ridge Analyses*, Wiley-Interscience.

Breheny, P. and Huang, J. (2011), "Coordinate Descent Algorithms for Nonconvex Penalized Regression, with Application to Biological Feature Selection," *The Annals of Applied Statistics*, 5, 232–253.

Breiman, L. (1995), "Better Subset Regression Using the Nonnegative Garrote," *Technometrics*, 37, 373–384.

Breiman, L., Friedman, J., Stones, C. J., and Olshen, R. (1984), *Classification and Regression Trees*, Chapman and Hall/CRC.

Cortes, C. and Vapnik, V. N. (1995), "Support-Vector Networks," *Machine Learning*, 20, 273–297.

Craven, P. and Wahba, G. (1978), "Smoothing noisy data with spline functions Estimating the correct degree of smoothing by the method of generalized cross-validation," *Numerische Mathematik*, 31, 377–403.

Cristianini, N. and Shawe-Taylor, J. (2000), *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press.

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*, 39, 1–38.

Ding, S., Wahba, G., and Zhu, X. (2011), "Learning Higher-Order Graph Structure with Features by Structure Penalty," in *Advances in Neural Information Processing Systems 24*, pp. 253–261.

Eddelbuettel, D. and Francois, R. (2011), "Rcpp: Seamless R and C++ Integration." *Journal of Statistical Software*, 40, 1–18, `http://www.jstatsoft.org/v40/i08/`.

Efron, B., Hastie, T., Johnston, I., and Tibshirani, R. (2004), "Least Angle Regression," *Annals of Statistics*, 32, 407–451.

Fan, J. and Li, R. (2001), "Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties," *Journal of the American Statistical Association*, 96, 1348–1359.

Friedman, J., Hastie, T., and Tibshirani, R. (2010), "Regularization Paths for Generalized Linear Models via Coordinate Descent," *Journal of Statistical Software*, 44, 1–22.

Friedman, J. H. (1991), "Multivariate Adaptive Regression Splines," *Annals of Statistics*, 19, 1–67.

Gao, F., Wahba, G., Klein, R., and Klein, B. (2001), "Smoothing Splne ANOVA for Multivariate Bernoulli Observations, With Application to Ophthalmology Data," *Journal of the American Statistical Association*, 96, 127–160.

Girard, D. (1998), "Asymptotic comparison of (partial) cross-validation, GCV and randomized GCV in nonparametric regression." *Annals of Statistics*, 26, 315–334.

Gu, C. (2002), *Smoothing Spline ANOVA Models*, Springer.

Hastic, T., Tibshirani, R., and Friedman, J. (2009), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer.

Healy, M. and Westmacott, M. (1956), "Missing Values in Experiments Analysed on Automatic Computers," *Journal of the Royal Statistical Society, Series C*.

Ising, E. (1925), "Beitrag zur Theorie des Ferromagnetismus," *Z. Phys.*, 31, 253–258.

Lin, X., Wahba, G., Xiang, D., Gao, F., Klein, R., and Klein, B. (2000), "Smoothing Spline ANOVA Models for Large Data Sets with Bernoulli Observations and the Randomized GACV," *Annals of Statistics*, 28, 1570–1600.

Loh, W.-Y. (2012), *Variable Selection for Classification and Regression in Large p, Small n Problems*, vol. 205, Springer.

Ma, X. (2010), "Penalized Regression in Reproducing Kernel Hilbert Spaces with Randomized Covariate Data," Tech. Rep. 1159, Department of Statisitcs, University of Wisconsin, Madison, WI 53706.

McCullagh, P. and Nelder, J. (1989), *Generalized Linear Models*, Chapman and Hall/CRC.

Meinshausen, N. and Buhlmann (2006), "High Dimensional Graphs and Variable Selection with the LASSO," *Annals of Statistics*, 34, 1436–1462.

Nelder, J. and Mead, R. (1965), "A Simplex Method for Function Minimization," *Computer Journal*, 7, 308–313.

Park, M. and Hastie, T. (2007), "An L1 Regularization-path Algorithm for Generalized Linear Models," *Journal of the Royal Statistical Society, Series B*, 69, 659–677.

Park, T. and Casella, G. (2008), "The Bayesian Lasso," *Journal of the American Statistical Association*, 103, 681–686.

Pinheiro, J. and Bates, D. (1995), "Approximations to the Log-likelihood Function in the Nonlinear Mixed Effects Model," *Journal of Computational and Graphical Statistics*, 4, 12–35.

Pinheiro, J. and Chao, E. (2006), "Efficient Laplacian and Adaptive Gaussian Quadrature Algorithms for Multilevel Generalized Linear Mixed Models," *Journal of Computational and Graphical Statistics*, 15, 58–81.

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

R Development Core Team (2005), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0.

Ravikumar, P., Wainwright, M., and Lafferty, J. (2010), "High-dimensional Ising Model Selection using l_1-regularized logistic regression," *Annals of Statistics*, 38, 1287–1319.

Sanderson, C. (2010), "Armadillo: An Open Source C++ Linear Algebra Library for Fast Prototyping and Computationally Intensive Experiements," Tech. rep., NICTA.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *Annals of Statistics*, 6, 461–464.

Shi, W., Wahba, G., Irizarry, R., Corrado Bravo, H., and Wright, S. (2012), "The Partitioned LASSO-Patternsearch Algorithm with Application to Gene Expression Data," *BMC Bioinformatics*, 13, 98–110, doi:10.1186/1471-2105-13-98, http://www.biomedcentral.com/1471-2105/13/98.

Shi, W., Wahba, G., Wright, S., Lee, K., Klein, R., and Klein, B. (2008), "LASSO-Patternsearch Algorithm with Application to Ophthalmology and Genomic Data," *Statistics and Its Interface*, 1, 137–153.

Sonnenburg, S., Raetsch, G., Henschel, S., Widmer, C., Behr, J., Zien, A., de Bona, F., Binder, A., Gehl, C., and Franc, V. (2010), "The SHOGUN Machine Learning Toolbox," *Journal of Machine Learning Research*, 11, 1799–1802.

Tibshirani, R. (1996), "Regression Shrinkage and Selection via the lasso," *Journal of the Royal Statistical Society, Series B*, 58, 267–288.

US Census Bureau (2007), "County and City Data Book: 2007," `http://www.census.gov/statab/www/ccdb.html`.

Wahba, G. (1990), *Spline Models for Observational Data*, SIAM: Society for Industrial and Applied Mathematics.

— (2002), "Soft and Hard Claasification by Reproducing Kernel Hilber Space Methods," *Proceedings of the National Academy of Sciences*, 102, 12332–12337.

Wahba, G., Wang, Y., Gu, C., Klein, R., and Klein, B. (1995), "Smoothing Spline ANOVA for Exponential Families, with Application to the Wisconsin Epidemiological Study of Diabetic Retinopathy," *Annals of Statistics*, 12, 1865–1895.

Wainwright, M. and Jordan, M. (2008), "Graphical models, exponential families, and variational inference," *Foundations and Trends in Machine Learning*, 1, 1–305.

Whittaker, J. (1990), *Graphical Models in Applied Mathematical Multivariate Statistics*, Wiley.

Witten, I. H., Frank, E., and Hall, M. A. (2011), *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann.

Wright, S. (2011), "Accelerated Block-Coordinate Relaxation for Regularized Optimization," Tech. rep., Department of Computer Science, University of Wisconsin, Madison, WI 53706.

Wu, J. and Hamada, M. (2009), *Experiments: Planning, Analysis, and Optimization*, Wiley.

Xiang, D. and Wahba, G. (1994), "A Generalized Approximate Cross Validation for Smoothing Splines with Non-Gaussian Data," Tech. Rep. 930, Department of Statisitcs, University of Wisconsin, Madison, WI 53706.

— (1996), "A Generalized Approximate Cross Validation for Smoothing Splines with Non-Gaussian Data," *Statistica Sinica*, 6, 675–692.

Xiong, S., Dai, B., Jin, T., and Qian, P. (2011), "Orthogonalizing Penalized Regression," Tech. rep., Department of Statistics, University of Wisconsin, Madison, WI 53706.

Zhang, C.-H. (2010), "Nearly Unbiased Variable Selection under Minimax Concave Penalty," *Annals of Statistics*, 38, 894–942.

Zhao, P. and Yu, B. (2006), "On Model Selection Consistency of Lasso," *Journal of Machine Learning Research*, 7, 2541–2563.

— (2007), "Stagewise Lasso," *Journal of Machine Learning Research*, 8, 2701–2726.

Zou, H. and Hastie, T. (2005), "Regularizatio and Variable Selection via the Elastic Net," *Journal of the Royal Statistical Society, Series B*, 67, 301–320.