2019

Knowledge-Guided Bayesian Support Vector Machine Methods For High-Dimensional Data

Wenli Sun
University of Pennsylvania

Knowledge-Guided Bayesian Support Vector Machine Methods For High-Dimensional Data

Abstract

Support vector machines (SVM) is a popular classification method for analysis of high dimensional data such as genomics data. Recently, new SVM methods have been developed to achieve variable selection through either frequentist regularization or Bayesian shrinkage. The Bayesian framework provides a probabilistic interpretation for SVM and allows direct uncertainty quantification. In this dissertation, we develop four knowledge-guided SVM methods for the analysis of high dimensional data.

In Chapter 1, I first review the theory of SVM and existing methods for incorporating the prior knowledge, represented bby graphs into SVM. Second, I review the terminology on variable selection and limitations of the existing methods for SVM variable selection. Last, I introduce some Bayesian variable selection techniques as well as Markov chain

Monte Carlo (MCMC) algorithms .

In Chapter 2, we develop a new Bayesian SVM method that enables variable selection guided by structural information among predictors, e.g, biological pathways among genes. This method uses a spike and slab prior for feature selection combined with an Ising prior for incorporating structural information. The performance of the proposed method is evaluated in comparison with existing SVM methods in terms of prediction and feature selection in extensive simulations. Furthermore, the proposed method is illustrated in analysis of genomic data from a cancer study, demonstrating its advantage in generating biologically meaningful results and identifying potentially important features.

The model developed in Chapter 2 might suffer from the issue of phase transition \citep{li2010bayesian} when the number of variables becomes extremely large. In Chapter 3, we propose another Bayesian SVM method that assigns an adaptive structured shrinkage prior to the coefficients and the graph information is incorporated via the hyper-priors imposed on the precision matrix of the log-transformed shrinkage parameters. This method is shown to outperform the method in Chapter 2 in both simulations and real data analysis..

In Chapter 4, to relax the linearity assumption in chapter 2 and 3, we develop a novel knowledge-guided Bayesian non-linear SVM. The proposed method uses a diagonal matrix with ones representing feature selected and zeros representing feature unselected, and combines with the Ising prior to perform feature selection. The performance of our method is evaluated and compared with several penalized linear SVM and the standard kernel SVM method in terms of prediction and feature selection in extensive simulation settings. Also, analyses of genomic data from a cancer study show that our method yields a more accurate prediction model for patient survival and reveals biologically more meaningful results than the existing methods.

In Chapter 5, we extend the work of Chapter 4 and use a joint model to identify the relevant features and learn the structural information among them simultaneously. This model does not require that the structural information among the predictors is known, which is more powerful when the prior knowledge about pathways is limited or inaccurate. We demonstrate that our method outperforms the method developed in Chapter 4 when the prior knowledge is partially true or inaccurate in simulations and illustrate our proposed model with an application to a gliobastoma data set.

In Chapter 6, we propose some future works including extending our methods to more general types of outcomes such as categorical or continuous variables.

KNOWLEDGE-GUIDED BAYESIAN SUPPORT VECTOR MACHINE METHODS FOR HIGH-DIMENSIONAL DATA

Wenli Sun

A DISSERTATION

in

Epidemiology and Biostatistics

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2019

Supervisor of Dissertation

_____

Qi Long, Professor of Biostatistics, University of Pennsylvania

Graduate Group Chairperson

_____

Nandita Mitra, Professor of Biostatistics, University of Pennsylvania

Dissertation Committee

Justine Shults, Professor of Biostatistics, University of Pennsylvania

Jason A. Roy, Professor of Biostatistics, Rutgers University

Kristin A. Linn, Assistant Professor of Biostatistics, University of Pennsylvania

Yize Zhao, Assistant Professor of Biostatistics, Yale University

KNOWLEDGE-GUIDED BAYESIAN SUPPORT VECTOR MACHINE METHODS FOR HIGH-DIMENSIONAL DATA

ACKNOWLEDGEMENT

It is a pleasure to convey my gratitude to those who are important to the successful realization of this dissertation.

In the first place, I would like to show my deepest gratitude to my advisors, Dr. Qi Long for his excellent guidance, thoughtful supervision, and unflinching encouragement, which have triggered and nourished my intellectual maturity that I will benefit from for a long time. His sincere intuition and passion in research have greatly inspired me to develop and enrich my own research philosophy. This dissertation would not have been possible without his considerate advice and generous support. I am grateful to my advisor in every possible way.

I would like to show my gratitude to Dr. Changgee Chang, who greatly helped and supported me in the laboratory so that my projects can be completed smoothly.

I gratefully acknowledge my advisory committee members: Dr. Justine Shults, Dr. Jason Roy, Dr. Kristin Linn and Dr. Yize Zhao for their valuable advice and comments on my dissertation. They kindly granted me their precious time for discussions and gave many constructive suggestions.

I am also indebted to many professors in our department for helping and advising me in a number of ways during my graduate studies. I am very grateful to our department for giving me the excellent opportunity to study and work with many great people. Many thanks go to our administrative associates, Cathy and Marissa. I am also indebted to many of my colleagues in our department. Their kindness and friendship will make my past four years one of the most wonderful time in my life.

My parents Yuheng Sun and Chunyan liu and my parents-in-law Lei Xin and Jie Yang deserve my special appreciation for being supportive during my graduate studies.

Lastly I am extremely grateful to my husband Yi Xin and my sons Daniel and Tyson for their constant love, support, and understanding.

ABSTRACT

KNOWLEDGE-GUIDED BAYESIAN SUPPORT VECTOR MACHINE METHODS FOR HIGH-DIMENSIONAL DATA

Wenli Sun

Qi Long

Support vector machines (SVM) is a popular classification method for analysis of high dimensional data such as genomics data. Recently, new SVM methods have been developed to achieve variable selection through either frequentist regularization or Bayesian shrinkage. The Bayesian framework provides a probabilistic interpretation for SVM and allows direct uncertainty quantification. In this dissertation, we develop four knowledge-guided SVM methods for the analysis of high dimensional data.

In Chapter 1, I first review the theory of SVM and existing methods for incorporating the prior knowledge, represented bby graphs into SVM. Second, I review the terminology on variable selection and limitations of the existing methods for SVM variable selection. Last, I introduce some Bayesian variable selection techniques as well as Markov chain Monte Carlo (MCMC) algorithms .

In Chapter 2, we develop a new Bayesian SVM method that enables variable selection guided by structural information among predictors, e.g, biological pathways among genes. This method uses a spike and slab prior for feature selection combined with an Ising prior for incorporating structural information. The performance of the proposed method is evaluated in comparison with existing SVM methods in terms of prediction and feature selection in extensive simulations. Furthermore, the proposed method is illustrated in analysis of genomic data from a cancer study, demonstrating its advantage in generating biologically meaningful results and identifying potentially important features.

The model developed in Chapter 2 might suffer from the issue of phase transition (Li and Zhang, 2010) when the number of variables becomes extremely large. In Chapter 3, we propose another Bayesian SVM method that assigns an adaptive structured shrinkage prior to the coefficients and the graph information is incorporated via the hyper-priors imposed on the precision matrix of the

log-transformed shrinkage parameters. This method is shown to outperform the method in Chapter 2 in both simulations and real data analysis..

In Chapter 4, to relax the linearity assumption in chapter 2 and 3, we develop a novel knowledge-guided Bayesian non-linear SVM. The proposed method uses a diagonal matrix with ones representing feature selected and zeros representing feature unselected, and combines with the Ising prior to perform feature selection. The performance of our method is evaluated and compared with several penalized linear SVM and the standard kernel SVM method in terms of prediction and feature selection in extensive simulation settings. Also, analyses of genomic data from a cancer study show that our method yields a more accurate prediction model for patient survival and reveals biologically more meaningful results than the existing methods.

In Chapter 5, we extend the work of Chapter 4 and use a joint model to identify the relevant features and learn the structural information among them simultaneously. This model does not require that the structural information among the predictors is known, which is more powerful when the prior knowledge about pathways is limited or inaccurate. We demonstrate that our method outperforms the method developed in Chapter 4 when the prior knowledge is partially true or inaccurate in simulations and illustrate our proposed model with an application to a gliobastoma data set.

In Chapter 6, we propose some future works including extending our methods to more general types of outcomes such as categorical or continuous variables.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

CHAPTER 1

INTRODUCTION

1.1. Support Vector Machines (SVMs)

1.1.1. Linear Classification

Support Vector Machine (SVM) originally proposed by Vapnik and Vapnik (1998), is a powerful tool for classification problems in the machine learning field. It has achieved success in various tasks such as image classification, pattern recognition and forecasting (Nayak, Naik, and Behera, 2015; Salcedo-Sanz et al., 2014). The basic idea of SVM for classification is to find a linear hyperplane that separate two classes of data points with the largest minimal separating distance or margin. Suppose there are n samples in the training set of data. Let $\mathbf{x}$ be the p dimensional predictors, and $y \in \{-1, 1\}$ be the corresponding classification label. The classical SVM constructs a hyperplane H_0 to separate the two classes by maximizing the margin, which can be represented as:

$$\boldsymbol{\beta}^T\mathbf{x} + b = 0,$$

such that:

$$\begin{aligned} \boldsymbol{\beta}^T\mathbf{x} + b \geq 1 \quad &\text{for} \quad y = +1, \\ \boldsymbol{\beta}^T\mathbf{x} + b \leq -1 \quad &\text{for} \quad y = -1. \end{aligned}$$

Let H_1 and H_2 be the hyperplanes (Fig. 1.1) separating the classes such that there is no other data point between them. The goal is to maximize the margin M between the two classes. The objective function to be maximized is:

$$\begin{aligned} &\max_{\boldsymbol{\beta}, b} \ M \\ &\text{s.t.} \ \ y_i(\boldsymbol{\beta}^T\mathbf{x}_i + b) \geq 1, i = 1, ..., n. \end{aligned}$$

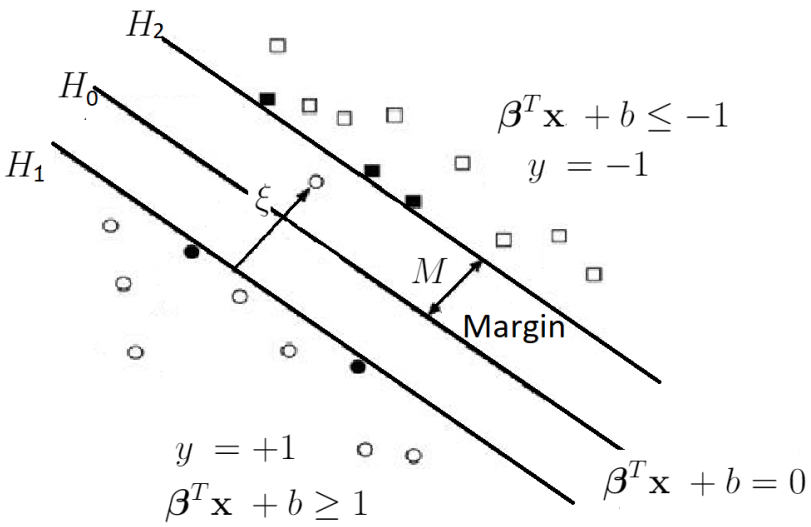

Figure 1.1: The figure shows a linear SVM classifier for two linearly separable classes (square and circles). The solid square and circles represent the support vectors

The margin M is equal to $\frac{2}{||\boldsymbol{\beta}||}$. The objective function can re-written as:

$$\begin{aligned} &\min_{\boldsymbol{\beta},b} \ \frac{1}{2}||\boldsymbol{\beta}||^2 \\ &\text{s.t.} \ \ y_i(\boldsymbol{\beta}^T\mathbf{x}_i + b) \geq 1, i = 1, ..., n. \end{aligned}$$

If the two classes are not linearly separable, a slack variable $\boldsymbol{\xi} = (\xi_1, ..., \xi_n)$ (Figure 1.1) can be introduced to allow some points to be on the wrong side of the hyperplane. Then the modified objective function is:

$$\begin{aligned} &\min_{\boldsymbol{\beta},b} \ \frac{1}{2}||\boldsymbol{\beta}||^2 + C\sum_{i=1}^{n}\xi_i \\ &\text{s.t.} \ \ y_i(\boldsymbol{\beta}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \\ &\qquad \xi_i \geq 0, \ \ i = 1, ..., n. \end{aligned}$$

The parameter C can be tuned using the validation set. If ξ_i is defined as the hinge loss function, then this optimization problem can be re-expressed as:

$$\min_{\boldsymbol{\beta},b} \ \frac{1}{2}||\boldsymbol{\beta}||^2 + C\sum_{i=1}^{n}\text{max}(0, 1 - y_i(\boldsymbol{\beta}^T\mathbf{x}_i + b)) \tag{1.1}$$

Directly solving this problem is difficult because the constraints are quite complex. The mathematical tool of choice for simplifying this problem is the Lagrangian dual formulation (Bertsekas, 1999):

$$L(\alpha) = \sum_{i=1}^{n} \alpha_i - \sum_{i,j=1}^{n} y_i \alpha_i y_j \alpha_j K(x_i, x_j) \tag{1.2}$$

$$\text{s.t.} \quad \sum_i y_i \alpha_i = 0, 0 \leq \alpha_i \leq C, \quad i = 1, ..., n.$$

where α_i are Lagrange multipliers and $K(x_i, x_j) = x_i x_j$ here represent the case of linear classification.

The solution $\alpha_1, ..., \alpha_n$ is computationally easier. $\boldsymbol{\beta}$ and the linear discriminant boundary for the optimal hyperplane is then recovered from $\hat{\boldsymbol{\alpha}}$:

$$\hat{\boldsymbol{\beta}} = \sum_{i}^{n} \hat{\boldsymbol{\alpha}}_i y_j x_i$$

$$\hat{y(x)} = (\hat{\boldsymbol{\beta}}^T \mathbf{x} + \hat{b}) = \sum_{i}^{n} \alpha_i y_j x_i \mathbf{x} + \hat{b}$$

1.1.2. Non-linear Classification

Non-linear separation can be achieved by mapping the original feature space to some higher-dimensional feature space where the training set is separable. This is known as the "kernel trick" (Cristianini, 2001; Hofmann, Schölkopf, and Smola, 2008), which solves the computational problem of dealing with many dimensions, even infinite-dimensional spaces. The learning process using nonlinear SVM consists of two steps: (a) initially, the input vectors are transformed into high-dimensional feature vectors to be linearly separated; (b) secondly, the SVM learning algorithm is applied to find the optimum margin hyperplane in the new feature space. This separating hyperplane is a linear function in the transformed feature space, but its inverse mapping is a nonlinear structure in the original input space.

Let $\Phi : x \to \phi(x)$ denote a nonlinear mapping from the input space to a higher dimensional feature space. The problem formulation corresponds to the equation 1.2, where $K(x_i, x_j)$ is called a kernel function, and $K(x_i, x_j)$ can be defined as $\phi(x_i)^T \phi(x_j)$. Then the hyperplane that corresponds to

the decision boundary in the feature space is defined as $\hat{\boldsymbol{\beta}}^T \phi(x) + b = 0$ and $\hat{\boldsymbol{\beta}} = \sum_i^n \hat{\boldsymbol{\alpha}}_i y_j \phi(x_i)$.

The kernel function K is sometimes more precisely referred to as Mercer kernels, because they must satisfy Mercer's condition (Cristianini, 2001). For any function f with finite norm $\int g(x)^2 dx < \infty$, K must satisfy:

$$\int K(u,v)g(u)g(v)dudv \geq 0$$

The kernel function K must be continuous, symmetric, and have a positive definite matrix. Such a K means that there exists a mapping to a reproducing kernel Hilbert space (a Hilbert space is a vector space closed under inner products) such that the inner product there gives the same value as the function K. If a kernel does not satisfy Mercer's condition, then the corresponding quadratic problem may have no solution. The two commonly used families of kernels are polynomial kernels and radial basis functions. Polynomial kernels are of the form $K(u,v) = (1 + u^T v)^d$, for any positive integer d. The case of $d = 1$ is a linear kernel and the case of $d = 2$ gives a quadratic kernel. The most common form of radial basis function is a Gaussian distribution, calculated as:

$$K(u,v) = e^{-(u-v))^2/2\sigma^2}$$

where σ is the length scale parameter, and can be chosen via cross-validation.

1.2. Incorporation of Prior Knowledge in SVM

In real-world applications, some prior knowledge is usually known and should be integrated into the classification model to improve power in detection. Recently, Lauer and Bloch (2008) provided a comprehensive review on methods for incorporating prior knowledge in SVM for classification . In their paper, the prior knowledge is classified into two categories: class-invariance and knowledge on the data. The class-invariance is the invariance of the class to a transformation of the input pattern. For instance, if an image is slightly rotated or translated it will represent the same information. Knowledge on the data refers to unlabeled samples, imbalance of the training set and quality of the data. Sometimes, poor quality or unbalanced data may mislead the decision of a classifier. To incorporate prior information, there are three main types of methods: sample methods (Schölkopf, Burges, and Vapnik, 1996; Wu and Srihari, 2004), kernel methods (Decoste and Schölkopf, 2002; Wang et al., 2005) and optimization methods (Chapelle and Schölkopf, 2002; Fung, Mangasarian,

and Shavlik, 2003; Graepel and Herbrich, 2004). Sample methods refer to incorporating the prior knowledge either by generating new data or by modifying the way they are taken into account. Kernel methods refer to incorporating the prior knowledge in the kernel function or creating a new kernel. Optimization methods refer to incorporating the prior knowledge in the problem formulation either by adding constraints or by defining a new formulation which includes the prior knowledge.

However, the prior knowledge defined aforementioned is not applicable to some fields. For example, in genomic studies, genes tend to act in groups through pathways, while a single gene may not have a strong impact. So accounting for the relationship between genes has the potential to improve the power in detection of key molecular features and yield biologically meaningful results. Recent work (Chang, Kundu, and Long, 2018; Pan, Xie, and Shen, 2010; Zhao et al., 2016) demonstrated that integrating biological knowledge such as gene or metabolic pathways in predictive modeling offers great promise of improved predictive accuracy. Therefore, a new category of prior knowledge for SVM is investigated in my dissertation project: biological knowledge represented by graphs. Prior biological knowledge usually refers to the structural information among predictors which can be extracted from existing databases (Ashburner et al., 2000; Nishimura, 2001; Ogata et al., 1999). Alternatively, the Gaussian graphical model can be adopted to estimate the graph structure and provide a sparse and interpretable representation of the conditional dependencies found in the data.

1.3. Gaussian Graphical Model

Suppose a graph $\mathcal{G} = \langle V, E \rangle$ is given where $V = \{1, \ldots, p\}$ represents the set of predictors and the edge set $E \subset \{(j,k) : j,k \in V, j \neq k\}$ represents associations between the predictors. Let G be the adjacency matrix of $\mathcal{G}$, the predictors X is assumed to follow a Gaussian graphical model (GMM) with respect to the graph $\mathcal{G}$ (Dempster, 1972). In other words, we have $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \Omega^{-1})$, where the precision matrix $\Omega = (\omega_{jk})$ is such that $\omega_{jk} = 0$ if and only if $g_{jk} = 0$ in G. In the Gaussian graphical model, the graph structure represents conditional dependencies among predictors. The edge between j and k is present if (and only if) the corresponding two predictors are conditionally correlated (dependent). In other words, $g_{jk} = 0$ implies that the predictors j and k are conditionally independent given all other predictors. Because the graphical model estimation corresponds to estimation of a sparse version of Ω, regularization methods are a natural approach. Fig. 1.2

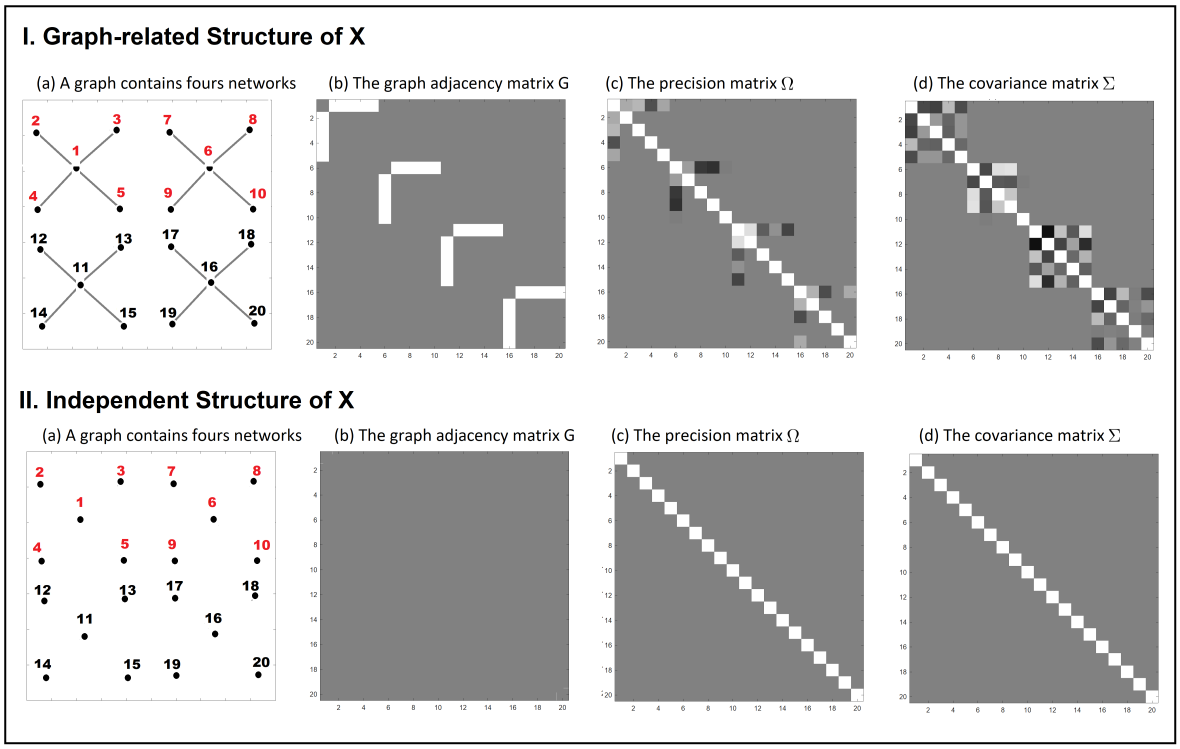

Figure 1.2: The simulated graphs and their corresponding precision matrix Ω and covariance matrix Σ for different structures. The number highlighted in red represents the important features which are relevant with the outcomes.

shows two different structures of X with their corresponding adjacency matrix, precision matrix and covariance matrix.

1.4. Variable Selection in SVM

Variable selection has been widely investigated in model prediction, and it refers to selecting the best subset of predictors among a large set of variables, to provide good predictions and interpretations. As shown in equation 1.1, SVM is equivalent to a regularization framework of loss $+$ penalty, thus, variable selection in SVM, utilizing the prior knowledge of sparsity, can be achieved by imposing appropriate sparsity-inducing penalties. Bradley and Mangasarian (1998), Song et al. (2002), and Zhu et al. (2004) adapted the LASSO techinique (Tibshirani, 1996) into SVM and studied the properties of the L_1 penalized SVM (L_1SVM); however, these L_1SVM variable selection methods do not take advantage of prior knowledge such as structural information among features. Wang, Zhu, and Zou (2006) proposed a double regularization SVM (DrSVM), which combines the L_1 and

L_2 norm to encourage the selection of correlated features. Zou and Yuan (2008) suggested a $L\infty$ penalized SVM when there is prior knowledge about the grouping information of features. Becker et al. (2011) and Zhang et al. (2005) considered the SVM with a non-convex penalty in the application of gene selection (SCADSVM). Despite their success, the SVM variable selection methods suffer from two drawbacks: 1. The flexibility for incorporation of the prior knowledge into SVM is limited. For instance, if the network information of the genes is given, it might be difficult to design the suitable regularization forms for matching the prior information of the correlation between genes; 2. Most of the existing SVM models are focused on point estimation and they do not allow for uncertainty in variable selection and prediction.

1.5. Bayesian Variable Selection in SVM

As seen in the previous section, the regularization approaches based on the frequentist framework have some shortcomings, which naturally lead researchers to explore the Bayesian approach. In the Bayesian framework, the penalty could be replaced by specifying a prior distribution on the parameter. Fortunately, it has been shown that SVM can be reformulated into a MAP (Maximum a Posteriori) estimation in a probabilistic generative model by the technique of data augmentation. Polson and Scott (2011) re-expressed the original SVM by an exponential transformation and derived the pseudo-likelihood as a location-scale mixture of normals, and then introduced auxiliary variables to the pseudo-likelihood to allow drawing samples from the augmented posterior. This work enables Bayesian SVM to provide geometric interpretation, flexible feature modeling, and predictive uncertainty quantification.

In the Bayesian framework, the spike-and-slab prior for variable selection has been widely used. Mitchell and Beauchamp (1988) proposed a spike and slab prior model used for the predictors. The spike component represents the unimportant predictors by placing probability mass at zero, while the slab component represents important predictors by assuming uniform distributions in a wide range. Similarly, George and McCulloch (1993) proposed a stochastic variable selection algorithm, which assumed the predictors to be a mixture of a low and high variance normal prior centered at zero, with the low variance corresponding to the slab and high variance corresponding to the spike. The general idea for the spike-and-slab prior is to introduce indicator variables to determine whether the corresponding predictors will be included in the model. Traditionally, the independent

and identically distributed (iid) Bernoulli priors are assigned to the indicators, while the iid Bernoulli prior may not be able to utilize and incorporate the prior structure information among predictors. In my first project, a Ising prior is proposed, to account for the the pairwise interactions between predictors.

In summary, the goal of my dissertation research is to develop highly accurate, biologically meaningful prediction Bayesian SVM methods to tackle high dimensional data such as genomics data with tens of thousands of variables. These Bayesian SVM methods provide a probabilistic interpretation for SVM and allow direct quantification of the uncertainty of prediction and estimation. In addition, the structural information among the predictors represented by graphs can be easily incorporated in these models to help understand the underlying biological mechanism and improve predictive accuracy. A diagram of our model structure is illustrated in Figure 1.3.

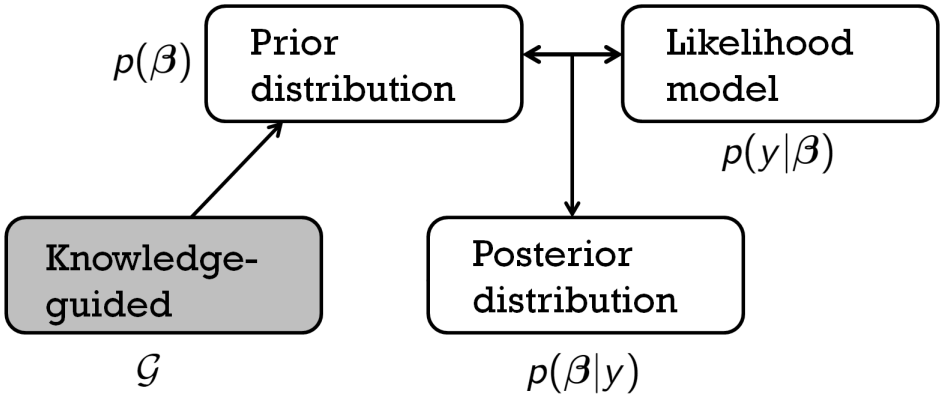

Figure 1.3: The main components our model.

CHAPTER 2

KNOWLEDGE-GUIDED BAYESIAN VARIABLE SELECTION IN SUPPORT VECTOR MACHINE FOR STRUCTURED HIGH-DIMENSIONAL DATA (KBSVM)

2.1. Introduction

The support vector machine (SVM) (Vapnik and Vapnik, 1998) is a popular classification method in data mining and machine learning. It has achieved great successes in various data mining tasks such as image classification, pattern recognition and forecasting (Nayak, Naik, and Behera, 2015; Salcedo-Sanz et al., 2014). Many SVM approaches with feature selection have been introduced in the literature, among which the ones that use a specific penalty on the coefficients (normal vector) are popular. The L_1 norm penalized SVM (L_1SVM) (Bradley and Mangasarian, 1998; Song et al., 2002; Zhu et al., 2004) applies the LASSO penalty (Tibshirani, 1996) into SVM. The SVM with a non-convex penalty (Becker et al., 2011; Zhang et al., 2005) (SCADSVM) adopts the smoothly clipped absolute deviation penalty (Fan, 2001) to alleviate the bias in estimating nonzero coefficients. Double regularization SVM (DrSVM) (Wang, Zhu, and Zou, 2006) combines the L_1 and L_2 norm to encourage the selection of correlated features. L_∞ penalized SVM (Zou and Yuan, 2008) encourages all the features in the same group to be selected simultaneously. These approaches and their variants have proven their superiority during the past two decades. In this era of big data, however, where the multi-omics data need to be analyzed beyond the GWAS or genomic studies, it is imperative that new innovation is required.

In some real world applications, some prior knowledge on data may be available, which can be integrated into the analysis and improve the power of detecting important signals. For example, a comprehensive review (Lauer and Bloch, 2008) summarizes the methods that incorporate such prior knowledge into SVM, while classifying the prior knowledge into two categories: class-invariance and knowledge on the data. The class-invariance stands for the invariance of the class to a transformation of the input pattern, and the knowledge on the data refers to such knowledge as the information in unlabeled samples, the imbalance of the training set, and the quality of the data. This article aims to consider the prior biological knowledge that is represented by the pathway graph information. Enormous genomic studies have revealed that the genes influence phenotypes

through a complex regulatory network represented by a directed acyclic graph, where each gene is expressed by a node and the promotion/inhibition relationships between the genes are indicated by the edges. The network is composed of multiple gene pathways and the knowledge on the pathway graphs is publicly available ("Pathway Databases") and still growing. Recent works (Chang, Kundu, and Long, 2018; Pan, Xie, and Shen, 2010; Stingo et al., 2011; Zhao et al., 2016) have attempted to incorporate the pathway graph information, motivated from its biological interpretation, by encouraging group-wise selection of adjacent predictors. They demonstrate that the incorporation of such prior knowledge offers a great promise toward the improved predictive accuracy and the increased power of detecting key molecular signatures and acting pathways. In addition, the resulting prediction models become more interpretable as they help select key biological pathways and likely lead to idenfication of potential molecular targets for treatments Chuang et al., 2007.

However, only very few works (Zhu, Shen, and Pan, 2009) in the SVM framework can incorporate the prior knowledge on the correlation structure among features. At the same time, most penalization based SVM methods (Becker et al., 2011; Bradley and Mangasarian, 1998; Fan, 2001; Song et al., 2002; Wang, Zhu, and Zou, 2006; Zhang et al., 2005; Zhu et al., 2004; Zhu, Shen, and Pan, 2009; Zou and Yuan, 2008) provide point estimates, failing to systematically quantify the uncertainty of the estimates. Therefore, we propose a knowledge-guided Bayesian SVM (KBSVM), which is a Bayesian approach capable of incorporating the graphical structure of features. As a Bayesian method, our approach can provide not only the uncertainty information but also the ensemble inference, which leads to more accurate and reliable performance in both classification and feature selection. Some Bayesian approaches (Bhosale and Ade, 2014; Luts and Ormerod, 2014; Yang, Pan, and Guo, 2017) have been proposed to perform feature selection by introducing shrinkage priors on the normal vector, but to the best of our knowledge, none of them utilizes the graph structure among the features. Also, note that the exising frequentist approaches (Wang, Zhu, and Zou, 2006; Zhu, Shen, and Pan, 2009; Zou and Yuan, 2008) either force the coefficients to have similar values or apply smoothing between all the member coefficients in a pathway group, which may cause bias. Unlike those works, our approach uses the pathway graph information, which is more refined than the pathway membership information, and encourages only the joint selection among the adjacent features rather than smooths their coefficient estimates. This helps achieve enhanced performance without the expense of bias.

In the proposed model, we employ the spike-and-slab prior (George and McCulloch, 1993) for feature selection. The selection status of each feature is represented by a latent binary variable. The gaussian prior with small variance (spike) is assigned for the inactive coefficient, and the gaussian prior with large variance (slab) is assigned for the active coefficient. This prior shrinks the inactive coefficients toward zero and reduces the bias for the active coefficients. In addition to the spike-and-slab prior, we assign the Ising prior (Ising, 1925) to the latent indicator variables to reflect the graphical structure of the predictors. This prior encourages any pair of predictors which are adjacent on the graph to have the same selection status. Note that (Stingo et al., 2011) uses the Markov random field (MRF) prior for the latent indicator variables, which is similar to the Ising prior. The difference is that, while the MRF prior only has the selected features encourage the selection of the adjacent features, the Ising prior also has the unselected features encourage the deselection of the neignboring features. Therefore, our model prefers both group-wise inclusion and exclusion of adjacent features, which further improves the prediction performance.

We present the Gibbs sampling algorithm (Gilks, Richardson, and Spiegelhalter, 1996) that performs the Bayesian prediction and feature selection. We employ the the state-of-the-art data augmentation techniques (Polson and Scott, 2011) to make our algorithm efficient and easy to implement. Another contribution to the Bayesian SVM literature is that we propose the corrected pseudo-likelihood. Having the proper form of likelihood allows other model parameter to have a better interpretation, which will be elaborated in Section 2.2.1. The performance of the proposed method is evaluated in comparison with other existing SVM methods in terms of prediction and feature selection under extensive simulation scenarios. In addition, we illustrate an application of our method to the analysis of genomic data from a cancer study, further demonstrating its advantage in identifying important features and yielding biologically meaningful results.

The rest of the article is organized as follows. In Sections 2.2, we describe the proposed models and the computing algorithms. In Section 2.3, we conduct simulation to evaluate our approach in comparison with several existing approaches. In Section 2.4, we apply our approach to a TCGA glioblastoma dataset. We conclude with a brief discussion in Section 2.5.

2.2. Methods

2.2.1. Likelihood

Suppose there are n samples in the training set of data where $y_i \in \{-1, 1\}$ are the binary outcome variables and $\mathbf{x}_i$ are the $(p+1)$ dimensional feature vector including the intercept. The classical SVM seeks to find a classification function f to separate the two classes by minimizing

$$\Theta(\beta) = \kappa \sum_{i=1}^{N} \max(1 - y_i f(\mathbf{x}_i), 0) + R(f), \tag{2.1}$$

where $\sum_{i=1}^{N} \max(1 - y_i f(\mathbf{x}_i), 0)$ is the hinge loss function and R is a regularization function controlling the complexity of f. The tuning parameter κ can be seen as part of the regularization parameters. For the linear classifier $f = \mathbf{x}_i'\boldsymbol{\beta}$, minimizing the objective function (2.1) is equivalent to find the mode of the following pseudo-posterior density (Henao, Yuan, and Carin, 2014).

$$\begin{aligned} p(\boldsymbol{\beta}|X, \mathbf{y}, \kappa) &\propto p(\boldsymbol{\beta}) L(\mathbf{y}|X, \boldsymbol{\beta}, \kappa) \\ &\propto p(\boldsymbol{\beta}) \prod_{i=1}^{n} \kappa e^{-2\kappa \max(1 - y_i \mathbf{x}_i'\boldsymbol{\beta}, 0)}. \end{aligned}$$

Note that $\kappa e^{-2\kappa \max(1 - y_i \mathbf{x}_i'\boldsymbol{\beta}, 0)}$ is the pseudo-likelihood contribution from the i-th observation (as it does not sum to a constant) and obviously prefers the coefficients that reduces the hinge loss. Note that this pseudo-likelihood is not exactly same as the one that has been widely used in the Bayesian SVM literature. We correct the one used in Henao, Yuan, and Carin (2014) and Polson and Scott (2011) by multiplying it by κ. This newly proposed pseudo-likelihood gives a plausible interpretation for the parameter κ; the parameter κ learns the overall (average) scale of the errors. In fact, the posterior distribution of κ converges to a degenerate distribution concentrated at 0 under the previous pseudo-likelihood, as the sample size increases. Note also that another important role of the parameter κ is to allow the normal vector $\boldsymbol{\beta}$ to explore its parameter space more freely in MCMC.

We use the Gamma prior for $\kappa \sim \mathcal{G}(a_\kappa, b_\kappa)$, where a_κ and b_κ are hyperparameters representing the shape and the rate parameters of the Gamma distribution, the values of which can be chosen in an uninformative or data-driven manner.

2.2.2. Spike-and-Slab and Ising Prior

As aforementioned, we use the spike-and slab prior for $\boldsymbol{\beta}$ to perform the feature selection. We introduce the latent binary variables γ_j indicating the inclusion of the j-th feature into the model, and assume $\beta_j|\gamma_j \propto N(0, v_j^2)$

$$p(\boldsymbol{\beta}|\boldsymbol{\gamma}) = C \prod_{j=1}^{p+1} v_j^{-\frac{1}{2}} e^{-\frac{\beta_j^2}{2v_j}},$$

where $v_j = \gamma_j \sigma_1^2 + (1-\gamma_j)\sigma_0^2$ with $\sigma_0^2 < \sigma_1^2$ and C is the normalizing constant. If $\gamma_j = 0$, then the prior of β_j has the spike variance $v_j = \sigma_0^2$ and β_j is shrunk toward 0. If $\gamma_j = 1$, then the prior of β_j has the slab variance $v_j = \sigma_1^2$ and β_j is less biased.

Let $\mathcal{G} = \langle V, E \rangle$ be a pathway graph where $V = \{1, \ldots, p+1\}$ is the set of genes and $E \subset \{(j,k) : j,k \in V, j \neq k\}$ be the set of edges representing (partial) correlations among the genes. Let G be the adjacency matrix of $\mathcal{G}$. To incorporate the graph structure between predictors, we use the Ising prior for $\boldsymbol{\gamma}$ given as follows.

$$p(\boldsymbol{\gamma}) = C_{\mu,\eta} e^{-\mu \sum_j \gamma_j + \eta \sum_{j \neq k} G_{jk} \mathbb{I}(\gamma_j = \gamma_k)}, \tag{2.2}$$

where $C_{\mu,\eta}$ is the normalizing constant and $\mathbb{I}(\cdot)$ is the indicator function. The tuning parameters μ controls the sparsity of $\boldsymbol{\gamma}$ and η controls the smoothness of $\boldsymbol{\gamma}$ over E. Note that (2.2) encourages $\gamma_k = 1$ if $\gamma_j = 1$ and $G_{jk} = 1$ and promotes $\gamma_k = 0$ if $\gamma_j = 0$ and $G_{jk} = 1$. Therefore, the group-wise selection of the j-th and the k-th genes are encouraged if there is an edge between them.

The Ising prior is slightly different from the Markov random field prior proposed in the literature earlier Li and Zhang, 2010; Stingo et al., 2011

$$p(\boldsymbol{\gamma}) = C_{\mu,\eta} e^{-\mu \sum_j \gamma_j + \eta \sum_{j \neq k} G_{jk} \gamma_j \gamma_k}, \tag{2.3}$$

Note that (2.3) only encourages $\gamma_k = 1$ if $\gamma_j = 1$ and $G_{jk} = 1$. However, there is little difference from the computational point of view because $\mathbb{I}(\gamma_j = \gamma_k) = 2\gamma_j\gamma_k - \gamma_j - \gamma_k + 1$.

2.2.3. Posterior Inference

Let $\mathbf{z}_i = y_i \mathbf{x}_i$ and $Z = [\mathbf{z}_1, \ldots, \mathbf{z}_n]'$. To facilitate the Bayesian compututation, we use the variable augmentation technique; see, for example, Polson and Scott, 2011.

$$e^{-2\kappa \max(1-\mathbf{z}_i'\boldsymbol{\beta}, 0)} = \int_0^\infty \frac{\sqrt{\kappa}}{\sqrt{2\pi\rho_i}} e^{-\frac{\kappa(\rho_i + 1 - \mathbf{z}_i'\boldsymbol{\beta})^2}{2\rho_i}} d\rho_i. \tag{2.4}$$

Note that (2.4) makes the conditional distribution of $\boldsymbol{\beta}$ become the multivariate Gaussian distribution, which leads to a straightforward Gibbs sampler.

2.2.4. Gibbs Sampling Algorithm

We sample $(\kappa, \boldsymbol{\rho})$ jointly, by first sampling κ with $\boldsymbol{\rho}$ marginalized out and then sampling $\boldsymbol{\rho}$ conditioning on κ (and other parameters). The conditional distribution of κ is given by

$$\kappa | \boldsymbol{\beta}, Z \sim \mathcal{G}\left(a_\kappa + \frac{3n}{2}, b_\kappa + \sum_{i=1}^n \frac{(\rho_i + 1 - \mathbf{z}_i'\boldsymbol{\beta})^2}{2\rho_i} \right).$$

The conditional distribution of ρ_i is given by

$$\rho_i | \boldsymbol{\beta}, \mathbf{z}_i, \kappa \sim \mathcal{GIN}(1/2, \kappa, \kappa(1 - \mathbf{z}_i'\boldsymbol{\beta})^2),$$

where $\mathcal{GIN}(p, a, b)$ stands for the generalized Gaussian distribution. Alternatively, the conditional distribution of ρ_i^{-1} given $(\boldsymbol{\beta}, \mathbf{z}_i, \kappa)$ is an inverse Gaussian distribution, denoted by $\mathcal{IN}$.

$$\rho_i^{-1} | \boldsymbol{\beta}, \mathbf{z}_i, \kappa \sim \mathcal{IN}(|1 - \mathbf{z}_i'\boldsymbol{\beta}|^{-1}, \kappa),$$

where the density function of $\mathcal{IN}(\mu, \lambda)$ is given by

$$f(x; \mu, \lambda) = \frac{\sqrt{\lambda}}{\sqrt{2\pi x^3}} e^{\frac{-\lambda(x-\mu)^2}{2\mu^2 x}}.$$

The conditional distribution of γ_j is given by

$$p(\gamma_j | \beta_j, \boldsymbol{\gamma}_{-j}) \propto v_j^{-1/2} e^{-\frac{\beta_j^2}{2v_j} - \mu\gamma_j + \eta \sum_k G_{jk} I(\gamma_j = \gamma_k)},$$

where $\boldsymbol{\gamma}_{-j} = (\gamma_1, \ldots, \gamma_{j-1}, \gamma_{j+1}, \ldots, \gamma_{p+1})$.

Finally, let $\mathbf{1}$ be a vector of 1's, $D_\rho = \text{diag}(\rho_1, \ldots, \rho_n)$, and $D_v = \text{diag}(v_1, \ldots, v_{p+1})$. The conditional distribution of $\boldsymbol{\beta}$ follows a multivariate Gaussian:

$$\boldsymbol{\beta}|Z, \kappa, \boldsymbol{\rho} \sim \mathcal{N}(\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta),$$

where $\boldsymbol{\mu}_\beta = \kappa(D_v^{-1} + Z'D_\rho^{-1}Z)^{-1}Z'D_\rho^{-1}(\mathbf{1} + \boldsymbol{\rho})$ and $\boldsymbol{\Sigma}_\beta = (D_v^{-1} + \kappa Z'D_\rho^{-1}Z)^{-1}$.

2.2.5. Markov chain Monte Carlo Sampling Algorithm for KBSVM

The latent variable representation form and the full conditional distributions lead to a computationally efficient Gibbs sampler. In the Gibbs sampling scheme, several steps are included to update the variable selection indicators γ conditional on the current $\boldsymbol{\beta}$ and the graph G, to update $\boldsymbol{\beta}$ and covariance matrix, and to sample the latent variables κ and ρ_i. A brief outline of the sampling scheme is given in the succeeding algorithms.

for $t = 1$ **to** T **do**
  Sample $\kappa \propto \mathcal{G}(a_\kappa, b_\kappa + 2\sum_i \max(1 - Z_i\boldsymbol{\beta}, 0))$.
  for $i = 1$ **to** n **do**
    Sample $\rho_i^{-1} \propto \mathcal{IN}(\kappa|1 - Z_i\boldsymbol{\beta}|^{-1}, \kappa^2)$
  end
  for $j = 1$ **to** $p + 1$ **do**
    Sample γ_j from $\pi(\gamma_j|\beta_j, \boldsymbol{\gamma}_{-j}) \propto v_j^{-1/2} \exp\left(-\frac{\beta_j^2}{2v_j} - \mu\gamma_j + \eta\sum_k G_{jk}I(\gamma_j = \gamma_k)\right)$
  end
  Sample $\boldsymbol{\beta} \propto \mathcal{N}((D_v^{-1} + Z'D_\rho^{-1}Z)^{-1}Z'D_\rho^{-1}(J + \kappa\rho), (D_v^{-1} + Z'D_\rho^{-1}Z)^{-1})$
end

Algorithm 1: Full Gibbs sampling algorithm for KBSVM

Beginning from an arbitrary set of initial values, the algorithm iterates until representative samples are obtained from the posterior distribution. Samples from the burn-in period, which are affected by the initial conditions, are discarded, and the remaining samples are used as the basis for inference.

2.3. Simulation Studies

2.3.1. Design of Experiment

We use both the linear discrimination analysis (LDA) model and the probit model to generate correlated data to evaluate the performance of our KBSVM method and make comparisons with other existing methods such as the standard SVM (L_2SVM), L_1SVM, DrSVM and SCADSVM. We generate $m = 100$ datasets, each with a training sample of size $n = 200$, a validation sample of size $n = 200$ and an independent test sample of size $n = 10000$. We specify different combinations of the feature dimension p and the nonzero feature dimension q for different models. To assess the performance of the predictive model, we compute the prediction error (PE), prediction sensitivity (PSEN), prediction specificity (PSPEC), Matthews Correlation Coefficients (MCC), feature selection true positive (FSTP) and feature selection false positive (FSFP) averaged across the $m = 100$ datasets. The approach for obtaining PE is described in the following section. PSEN is calculated as the proportion of positives $(y_i = 1)$ that are correctly identified and PSPEC is calculated as the proportion of negatives $(y_i = -1)$ that are correctly identified. MCC is defined as $\frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$, where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives and FN is the number of false negatives. FSTP and FSFP are the average number of selected relevant and irrelevant features in the training samples.

2.3.2. Parameter Tuning

For each of the existing methods, we use the penalizedSVM R-package Becker et al., 2009 to fit the model on the training datasets, tune the parameters in the validation datasets and obtain the results from the testing datasets. σ_1^2 is set to 100 to account for large variances for the slab. η is set to 1 or 0, to account for the prior knowledge used or not. σ_0^2, μ need to be tuned to achieve the best performance. To tune the parameters σ_0^2 and μ, we apply our algorithm on each training data and draw 1000 samples from the joint posterior distribution of $\boldsymbol{\beta}$ and γ. Each sample of $\boldsymbol{\beta}$ and the corresponding γ values are plugged into the model to make predictions on the validation sample. If $\gamma_j = 1$, the corresponding $\boldsymbol{\beta}_j$ is selected. If $\gamma_j = 0$, the corresponding $\boldsymbol{\beta}_j$ is set to zero. Then the prediction can be obtained by $\hat{\mathbf{y}} = \text{sign}(X\boldsymbol{\beta})$, where X is the observation matrix of the validation sample. PE can be calculated as the number of non-zero elements of $(\mathbf{y} - \hat{\mathbf{y}})$ divided by the number of observations of the validation sample $(n = 200)$. Then the averaged PE

across the 1000 posterior samples will be acquired and used for choosing the optimal parameters, and the corresponding 1000 samples are plugged into the model again to make predictions on the independent test sample. We repeat this procedure on the $m = 100$ datasets to obtain the average PE and the corresponding standard errors.

2.3.3. Simulation I: LDA model in the absence of the graph

The LDA model is used to evaluate the prediction and variable selection performance of our KBSVM method without incorporating the prior graph information. Let $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_p)$, and the same setting of $(\rho = -0.2, p = 400, q = 5)$ is adapted as in Xiang et al. ("Variable selection for support vector machines in moderately high dimensions"). The similar results for the existing methods such as L1SVM, L2SVM and SCADSVM are obtained. Moreover, the cases for $\rho = 0$ and 0.2 is also included to investigate different correlation structure of $\mathbf{X}$ impact on the performance of our method and other methods.

Model: $P(\mathbf{y} = \pm 1) = 0.5$, $X|\mathbf{y} \sim \mathcal{N}(\text{sign}(\mathbf{y})\boldsymbol{\mu}, \Sigma)$, $\boldsymbol{\mu} = (0.1, 0.2, 0.3, 0.4, 0.5, 0, ..., 0)$ and

$$\Sigma = \begin{pmatrix} \begin{pmatrix} 1 & & \rho \\ & \ddots & \\ \rho & & 1 \end{pmatrix}_{q\times q} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}_{p\times p},$$

where $\rho = \pm 0.2$ or 0, $q = 5$ and $p = 400$.

Table 2.1 compares different methods for the LDA model with the negative correlation, independent or positive correlation between genes. The numbers in the parentheses are the corresponding standard errors over the 50 datasets. It is not surprising to see that the performance deteriorates when ρ increases from -0.2 to 0.2 for all the methods, because in general, the variance of $\boldsymbol{\beta}$ is

proportional to the inverse of the covariance matrix Σ. When $\rho = -0.2$,

$$\boldsymbol{\Sigma}^{-1} = \begin{pmatrix} \begin{pmatrix} 1.67 & & 0.83 \\ & \ddots & \\ 0.83 & & 1.67 \end{pmatrix}_{5\times 5} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}_{400\times 400}$$

and when $\rho = 0.2$,

$$\Sigma^{-1} = \begin{pmatrix} \begin{pmatrix} 1.11 & & -0.14 \\ & \ddots & \\ -0.14 & & 1.11 \end{pmatrix}_{5\times 5} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}_{400\times 400}.$$

Therefore, $\boldsymbol{\beta}$ learned from the training set with positive correlation will have smaller variance and may not be particularly stable when making predictions for the testing set. When $\rho = -0.2$, DrSVM has similar performance as L_2SVM and also a very high FSTP because it tends to select more variables. SCADSM and KBSVM achieve significantly lower PE and greater MCC, which may be due to the negative correlation structure, while our method KBSVM has the least PE, largest MCC and highest FSTP. When $\rho = 0$, genes in X are independent, DrSVM still has the highest FSTP, as well as the highest FSFP. PE for SCADSVM and KBSVM are close, while KBSVM has significantly lower FSFP than the other methods. When $\rho = 0.2$, PE and MCC for L_1SVM, SCADSVM and KBSVM are similar, while L_1SVM has the highest FSTP, SCADSVM has the highest FSFP and KBSVM has the moderate FSTP and the lowest FSFP. In sum, Our KBSVM method outperforms the presented methods in terms of PE, PSEN, MCC and FSFP. Even without the guidance of prior knowledge, the performance of our method doesn't degrade.

Method	PE (%)	PSen (%)	PSpec(%)	MCC (%)	FSTP (%)	FSFP (%)
$p = 400$, $q = 5$, $\rho = -0.2$						
L_2SVM	41.73 (0.23)	56.71 (2.62)	59.88 (2.53)	17.79 (0.34)	–	–
L_1SVM	15.63 (0.51)	84.00 (0.81)	84.73 (0.72)	68.92 (1.16)	98.78 (0.72)	5.58 (0.38)
DrSVM	39.24 (0.24)	60.08 (1.32)	61.46 (1.33)	21.88 (0.37)	98.00 (0.83)	61.26 (1.15)
SCADSVM	8.63 (0.45)	90.85 (0.74)	91.89 (0.42)	82.88 (0.85)	98.80 (0.94)	0.14 (0.04)
KBSVM	8.25 (0.32)	91.38 (0.64)	92.11 (0.42)	83.60 (0.74)	99.99 (0.56)	0.09 (0.04)
$p = 400$, $q = 5$, $\rho = 0$						
L_2SVM	42.56 (0.31)	61.59 (3.09)	53.26 (3.19)	16.37 (0.46)	–	–
L_1SVM	33.73 (0.58)	68.36 (1.25)	32.94 (1.18)	32.94 (1.18)	79.00 (3.05)	20.00 (0.77)
DrSVM	40.78 (0.23)	59.87 (1.93)	58.52 (1.85)	19.06 (0.44)	93.60 (1.44)	70.88 (2.05)
SCADSVM	30.09 (0.50)	70.73 (1.30)	69.07 (1.28)	40.38 (0.99)	51.60 (2.13)	1.92 (0.33)
KBSVM	29.93 (0.50)	71.49 (0.96)	68.55 (1.02)	40.48 (1.00)	49.94 (1.95)	0.41 (0.11)
$p = 400$, $q = 5$, $\rho = 0.2$						
L_2SVM	44.12 (0.41)	56.64 (4.08)	55.05 (4.09)	13.31 (0.79)	–	–
L_1SVM	36.44 (0.55)	63.25 (1.17)	63.87 (1.29)	27.48 (1.13)	50.21 (2.65)	8.02 (2.53)
DrSVM	42.18 (0.31)	54.49 (2.77)	61.13 (2.49)	16.65 (0.53)	44.80 (2.48)	3.28 (2.30)
SCADSVM	35.58 (0.79)	63.73 (2.01)	65.01 (1.77)	30.15 (1.53)	45.11 (3.50)	10.43 (3.39)
KBSVM	34.98 (0.64)	64.62 (1.15)	65.41 (1.12)	30.27 (1.34)	40.55 (1.67)	1.87 (0.53)

Table 2.1: Simulation results for linear discrimination model for $\rho = -0.2, 0, 0.2$

2.3.4. Simulation II: Probit model in the presence of the graph

This section is to illustrate how to model the prior structure information and how to incorporate it in our method.

a. Graph simulation

Note that the true correlation structure of the genes is unknown in practice. As mentioned, we use the undirected graph G to represent the relationship between genes. In our simulation, we distinguish the underlying true graph G which is used for generating the data, and the working graph $G*$ which is providing the guidance to KBSVM algorithms.

In our simulation examples, the true graph G is pre-defined. Let $X = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_p) \sim \mathcal{N}(0, \Omega^{-1})$, where the precision matrix $\Omega = (\omega_{ij})$ is such that $(i, j) \notin E$ implies $\omega_{ij} = 0$. We then say that X follows a Gaussian graphical model (GMM) with respect to the graph $\mathcal{G}$. In order to convert the graph $\mathcal{G}$ to the precision matrix Ω, the Gaussian graphical model is adopted and several steps are performed. First, a matrix is created by assigning uniformly distributed random numbers over an interval of $[-1, 1]$ to the off diagonal elements corresponding to the edges in the graph $\mathcal{G}$; second, the absolute value of the lowest eigen-value of the resulting matrix in the first step is obtained

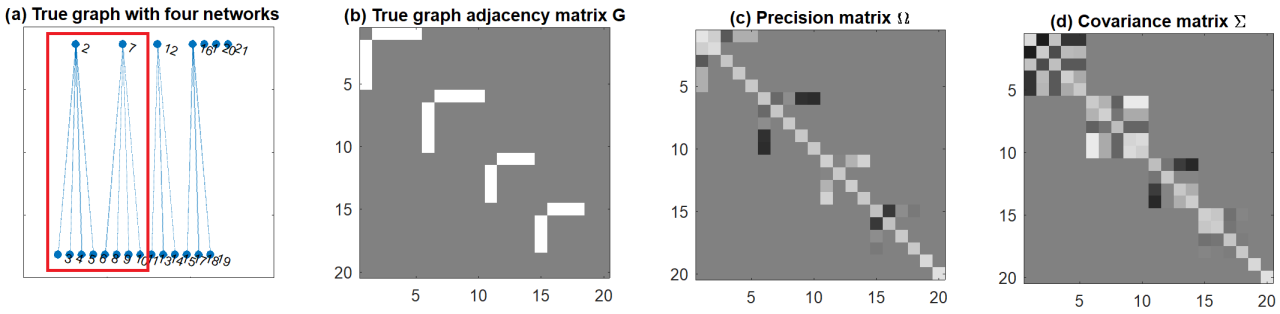

Figure 2.1: The true graph $\mathcal{G}$ and the corresponding adjacency matrix G, precison matrix Ω and covariance matrix Σ

and added to a small positive number, denoted as $|\lambda| + \Delta$; third, the elements on the diagonal of the matrix are reset to $|\lambda| + \Delta$, and therefore, all the eigenvalues of the resulting matrix are positive. Then the precision matrix can be obtained through scaling the resulting matrix by making the diagonal elements equal to 1's. Correspondingly, the covariance matrix Σ can be obtained by normalizing the inverse of the precision matrix. An example of the three matrices are illustrated in Fig. 2.1(b, c, d).

The working graph $G*$ represents the prior knowledge we now have to incorporate into our algorithm, thus it could be the true graph G indicating that the truth is known, a partial graph indicating that the truth is partially known or a noisy graph indicating that the prior knowledge is wrong. To simulation the partial graph, we adopt the Gaussian graphical model and set a threshold value on the precision matrix to remove some weak correlations. We first define a threshold value t, then compare the absolute values of each element of the precision matrix to t: if less than t, the element is set to zero; if equal or greater than t, the element remains the same value. Then the adjacency matrix of the partial graph is acquired by setting all the off-diagonal nonzero values of the resulting matrix to 1's, indicating the connection between nodes, while setting the diagonal elements to 0's. The steps of partial graph generated from the true graph is shown in Fig. 2.2.

To simulate the noisy graph, we can directly work on the lower triangle part of the corresponding adjacency matrix. First, we create a dimension $0_{(p+1)\times(p+1)}$ matrix, define a maximum number of connections n and generate a uniformly distributed random integer k over the interval of $[0, n]$. Second, we count the total number of the elements of the lower triangle part without including the diagonal elements, denoted as m, then generate m standard uniformly distributed random numbers and sort them. Third, the first k elements in the ordered m samples are assigned 1's and the left

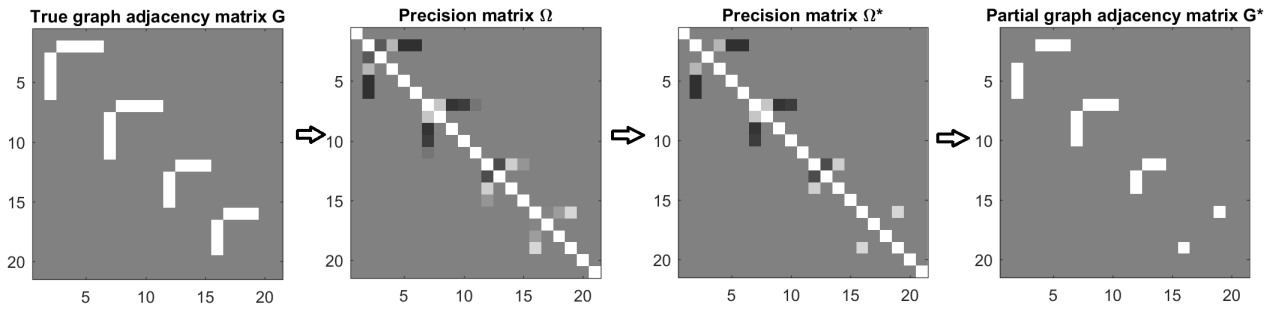

Figure 2.2: The simulation steps of the partial graph $G*$.

elements are assigned 0's. Then we apply some transformations to create a symmetric adjacency matrix from the lower part.

b) Probit model

The probit model is used to demonstrate the benefits of incorporating prior knowledge into our KBSVM method. The model can be written as : $X \sim \mathcal{N}(0, \Sigma)$, $\Sigma = f(G)$, $P(\mathbf{y} = 1|X) = \Phi(X\boldsymbol{\beta}+\beta_0)$. G is the true underlying true structure among predictors. The covariance structure of Σ should have a similar pattern to G, in other words, a function of G. Φ is the CDF of the standard normal distribution. β_0 is the intercept set to 0.5 and $\boldsymbol{\beta} = (0.8, 0.8, .., 0.8, 0.8, 0, , 0)$ is the p-dimension coefficient with the first q non-zero elements.

We specify four settings for our model and compare them to L_2SVM, L_1SVM, DrSVM and SCADSVM. The four settings are: no working graph incorporated ($\eta = 0$), the working graph $G*$ is assigned by a noisy graph (nG), a partial graph (pG) and the true underlying graph (G). Table 2.2 summarizes the simulation results for both $n > p$ and $n < p$ cases. Clearly, for all the cases, when the working graph $G*$ is assigned by the true graph G, our model KBSVM performs the best among the other settings as well as other existing methods. When $p = 20$ and $q = 10$, L_2SVM gives the largest PE and the lowest MCC, the prediction performance for L_1SVM, DrSVM, SCADSVM, KBSVM ($\eta = 0$) and KBSVM ($G* = nG$) are similar, while L_1SVM has a very high FSFP, and tends to select a larger model. When $p = 100$ and $q = 20$, PE for KBSVM($G* = G$) is significantly decreasing comparing to the other settings and other existing methods. When $\eta = 0$, the performance is the worst, among the four settings, but still outperforms L_2SVM, DrSVM and SCADSVM. We also note that L_1SVM still has the highest FSFP, and DrSVM has the second highest FSFP, which case is a little different from the case with $p = 20$. When $p = 500$, the prediction errors of L_2SVM and DrSVM are

similar, L_1SVM and KBSVM ($G* = pG$) are similar, while L_1SVM has the much higher high FSTP and FSFP. SCADSVM and KBSVM ($G* = G$) achieve the best results in terms of PE. In general, our method gives the smallest PE, the greatest MCC, a very low FSFP and BS. Even when $G*$ is assigned by nG, the performance of our method doesn't deteriorate too much.

In addition, we generate a new set of data from the independent correlation structure and thus we only need specify two settings for our model: $\eta = 0$ and $G* = nG$. The results are summarized in Table 2.3. When $p = 20$ and 100, KBSVM($\eta = 0$) outperforms the other methods in terms of PE and MCC. L_1SVM, DrSVM, SCADSVM tend to select more variables with a very high FSFP. Both of two settings for KBSVM give a significantly lower FSFP but keep the relatively high FSTP, showing the consistent ability of feature selection. When $p = 500$, L_1SVM gives the best performance in terms of PE, MCC and FSTP, while our model with $\eta = 0$ achieves satisfactory performance and also agrees with the findings in the LDA model.

In this simulation section, we consider two models under two conditions which are absence of the graph and presence of the graph. We observe that if the graphical network information is associate with the outcome and we utilize the true network information in the model, our KBSVM model outperforms other methods in terms of both prediction and selection accuracy. If the prior graph is not available, the performance doesn't degrade. Such stability is desirable and the results demonstrate encouraging gene selection ability and prediction power for our method.

2.4. Data Analysis

In this section, we apply our methods as well as other existing methods to classify a glioblastoma data set obtained from the Cancer Genome Atlas Network. Glioblastoma is a highly malignant brain tumor, also related to other cancer. This data set includes survival times (Y) and the gene expression levels for $p = 12,999$ genes (X) and 303 glioblastoma patients. For the purpose of classification, we define a new indicator variable Z to account for the one year survival outcome by setting

$$Z = \begin{cases} 1, & Y < 365, \Delta = 0, \\ 0, & Y > 365, \end{cases}$$

Method	PE (%)	PSen (%)	PSpec(%)	MCC (%)	FSTP (%)	FSFP (%)
			$p = 20$, $q = 10$			
L_2SVM	14.10 (0.11)	89.97 (0.31)	80.75 (0.47)	71.52 (0.22)	–	–
L_1SVM	11.82 (0.09)	90.09 (00.22)	85.75 (0.30)	76.06 (0.19)	98.05 (0.51)	52.30 (2.51)
DrSVM	11.92 (0.08)	89.69 (0.23)	86.04 (0.26)	75.88 (0.16)	99.80 (0.14)	18.20 (1.33)
SCADSVM	11.84 (0.11)	89.63 (0.21)	86.30 (0.28)	76.04 (0.23)	98.10 (0.42)	28.60 (3.08)
KBSVM, $\eta = 0$	11.92 (0.11)	89.83 (0.22)	85.86 (0.30)	75.87 (0.22)	96.23 (0.62)	16.44 (2.22)
KBSVM, $G* = nG$	12.02 (0.11)	89.59 (0.24)	85.94 (0.29)	75.69 (0.22)	97.20 (0.45)	25.93 (2.56)
KBSVM, $G* = pG$	11.59 (0.11)	90.00 (0.20)	86.41 (0.30)	76.56 (0.22)	98.36 (0.48)	12.94 (2.06)
KBSVM, $G* = G$	11.55 (0.11)	90.00 (0.22)	86.48 (0.30)	76.64 (0.21)	98.56 (0.41)	11.00 (1.91)
			$p = 100$, $q = 20$			
L_2SVM	20.96(0.11)	83.23 (0.44)	73.87 (0.62)	57.77 (0.27)	–	–
L_1SVM	17.27 (0.19)	84.93 (0.30)	80.03 (0.46)	65.18 (0.39)	90.41 (0.83)	40.57 (1.66)
DrSVM	19.74 (0.15)	83.13 (0.31)	76.73 (0.46)	60.16 (0.30)	84.40 (2.13)	28.43 (2.12)
SCADSVM	18.18 (0.27)	83.89 (0.34)	79.28 (0.51)	63.35 (0.54)	73.65 (1.46)	9.85 (1.63)
KBSVM, $\eta = 0$	17.92 (0.29)	83.67 (0.33)	80.13 (0.42)	63.85 (0.58)	78.71 (1.16)	8.10 (0.56)
KBSVM, $G* = nG$	17.29 (0.25)	84.30 (0.30)	80.76 (0.42)	65.15 (0.50)	79.11 (1.12)	9.63 (0.66)
KBSVM, $G* = pG$	15.76 (0.24)	85.67 (0.27)	82.51 (0.42)	68.25 (0.49)	87.83 (0.91)	7.19 (0.46)
KBSVM, $G* = G$	14.40 (0.11)	87.08 (0.27)	83.79 (0.35)	70.97 (0.41)	96.66 (0.55)	6.69 (0.41)
			$p = 500$, $q = 20$			
L_2SVM	33.61 (0.29)	75.02 (1.92)	56.96 (2.26)	33.44 (0.45)	–	–
L_1SVM	24.34 (0.46)	79.39 (0.97)	71.07 (0.87)	50.87 (0.91)	67.59 (1.47)	9.65 (0.85)
DrSVM	32.26 (0.24)	75.19 (1.28)	58.57 (1.26)	34.75 (0.47)	31.67 (5.06)	1.56 (0.36)
SCADSVM	24.16 (0.56)	77.48 (0.77)	73.83 (1.02)	51.36 (1.12)	48.00 (2.38)	1.38 (0.12)
KBSVM, $\eta = 0$	24.87 (0.54)	76.89 (0.80)	72.97 (1.00)	49.96 (1.08)	45.64 (2.53)	2.06 (0.59)
KBSVM, $G* = nG$	24.67 (0.52)	77.30 (0.66)	72.90 (0.91)	50.28 (1.05)	42.86 (2.04)	1.23 (0.39)
KBSVM, $G* = pG$	24.32 (0.51)	77.72 (0.63)	73.11 (0.81)	50.94 (1.04)	46.60 (2.71)	1.25 (0.22)
KBSVM, $G* = G$	24.11 (0.52)	77.76 (0.53)	73.63 (0.94)	51.56 (1.06)	48.94 (2.50)	1.33 (0.20)

Table 2.2: Comparison of the prediction performance and variable selection when the dimension of predictions p changes from 20 to 500 among different methods. q is the number of relevant variables. $\eta = 0$ represents the working graph $G*$ is not incorporated in our KBSVM model.

Method	PE (%)	PSen (%)	PSpec(%)	MCC (%)	FSTP (%)	FSFP (%)
			$p = 20$, $q = 10$			
L_2SVM	16.74 (0.20)	87.76 (0.55)	77.22 (0.86)	65.90(0.39)	–	–
L_1SVM	14.13 (0.13)	88.40 (0.37)	82.45 (0.45)	71.15 (0.25)	100.00 (0.00)	56.98 (3.12)
DrSVM	14.31 (0.12)	88.20 (0.37)	82.33 (0.48)	70.82 (0.25)	100.00 (0.00)	45.80 (3.47)
SCADSVM	13.89 (1.48)	87.95 (0.37)	83.65 (0.42)	71.71 (0.29)	100.00 (0.00)	25.80 (4.51)
KBSVM, $\eta = 0$	13.67 (0.12)	88.03 (0.34)	84.03 (0.39)	72.15 (0.24)	99.95 (0.03)	10.48 (2.27)
KBSVM, $G* = nG$	13.90 (0.14)	87.91 (0.34)	83.68 (0.42)	71.69 (0.29)	99.88 (0.08)	15.84 (2.97)
			$p = 100$, $q = 20$			
L_2SVM	22.93 (0.19)	83.56 (0.68)	69.08 (0.90)	53.74 (0.37)	–	–
L_1SVM	17.68 (0.25)	85.90 (0.51)	77.90 (0.57)	64.29 (0.51)	99.39 (0.23)	39.69 (2.27)
DrSVM	21.14 (0.21)	82.25 (0.46)	74.68 (0.52)	57.26 (0.43)	99.00 (0.35)	43.38 (2.95)
SCADSVM	19.58 (0.58)	82.66 (0.60)	77.65 (0.87)	60.46 (1.17)	89.50 (1.68)	25.75 (5.08)
KBSVM, $\eta = 0$	16.61 (0.39)	85.07 (0.49)	81.32 (0.53)	66.49 (0.78)	94.48 (0.75)	7.93 (1.05)
KBSVM, $G* = nG$	17.11 (0.42)	84.57 (0.53)	80.82 (0.60)	65.50 (0.85)	93.67 (1.18)	9.34 (1.05)
			$p = 500$, $q = 20$			
L_2SVM	36.16 (0.27)	81.90 (1.09)	41.55 (1.82)	26.27 (0.53)	–	–
L_1SVM	26.33 (0.68)	78.36 (0.68)	67.86 (0.72)	46.65 (0.99)	88.75 (1.30)	18.49 (0.27)
DrSVM	35.43 (0.17)	74.23 (0.90)	52.67 (1.21)	27.90 (0.35)	43.70 (4.51)	9.10 (0.94)
SCADSVM	27.07 (0.70)	76.64 (0.67)	68.34 (1.19)	45.22 (1.44)	71.90 (1.81)	14.98 (0.38)
KBSVM, $\eta = 0$	26.90 (0.61)	76.07 (0.63)	69.43 (0.85)	45.68 (1.27)	64.03 (1.77)	13.33 (0.37)
KBSVM, $G* = nG$	27.95 (0.59)	74.80 (0.66)	68.67 (0.81)	43.55 (1.20)	59.11 (2.16)	12.31 (0.45)

Table 2.3: Comparison of the prediction performance and variable selection when the predictors are independent.

where Δ represents censoring. Those subjects with $Y < 365, \Delta = 1$ are removed so the total number of subjects is 286 with $P(Z = 1) = 45\%$, $P(Z = 0) = 55\%$. First, we use the gene-ranking methods to select important genes. For each gene, the p value is acquired from the logistic regression and the top 1000 genes corresponding to the smallest 1000 p values are selected. Second, we obtain the network G for all the $12,999$ genes from the Kyoto Encyclopedia of Genes and Genomes (KEGG) database, use an algorithm to search the connections within the top 1000 genes, and then map them to the working graph $G*$. We specify two settings ($\eta = 0$ and 1) for our model to compare with other methods. The optimal tuning parameters for each methods are chosen by the minimum 20-fold cross-validation error. The average cross-validation error and the number of selected genes are summarized in Table 2.4.

As can be seen, L_1SVM selects most of the 1000 genes and has a similar performance to L_2SVM. DrSVM and SCADSVM give the very close CV errors while DrSVM select fewer number of genes. Our method KBSVM ($\eta = 1$) achieves the lowest CV error and BS and identifies a moderate number of genes. KBSVM($\eta = 0$) imposes more sparsity on the model and select only 69 genes, yet provides the satisfactory cross-validation error. In addition, all the genes selected by KBSVM ($\eta = 0$) are contained in the set of genes selected by KBSVM ($\eta = 1$), which confirms the stability.

We also conduct the pathway enrichment analysis for the selected genes for our method via ToppGene Suite Chen et al., 2009. When $\eta = 0$, our method doesn't encourage the inclusion of the connected genes, therefore, fewer genes and pathways are detected. However, several important genes are still selected, such as PICK1, IL22, BHLHE40 and NTN1, which are the members of the glioma pathways. When $\eta = 1$, the pathways detected by our method are highly enriched, such as protein processing in endoplasmic reticulum (1.16×10^{-6}), asparagine N-linked glycosylation (6.69×10^{-3}), ATF6 (ATF6-alpha) activates chaperone genes (7.86×10^{-3}), and unfolded protein response (1.08×10^{-2}). The numbers in the parentheses are the Bonferroni-adjusted p values. These pathways were found to be linked with the cancer cell proliferation and survival Clarke et al., 2014; Grantham et al., 2017; Hiramatsu, Joseph, and Lin, 2011; Kurtoglu et al., 2007. Moreover, the most highly enriched diseases are glioblastoma, mammary neoplasms and malignant tumor of colon. Therefore, the detected pathways and diseases further confirm our method can offer great promises of improved power in detection of key molecular signatures and provide valuable insights on biological bases of diseases.

Table 2.4: Results of the analysis of TCGA data. $n = 286$, $p = 1000$.

	CV error (%)	# selected genes
L_2SVM	30.45	1000
L_1SVM	29.85	957
DrSVM	27.52	399
SCADSVM	27.31	864
KBSVM, $\eta = 0$	28.92	69
KBSVM, $\eta = 1$	26.49	821

In sum, for our method KBSVM, when the prior network incorporated, the cross-validation error is reduced and the related pathways are significantly enriched, yielding biologically meaningful results. Therefore, we believe that our method KBSVM enjoys the benefits of incorporating prior knowledge to improve predictive performance.

2.5. Discussions

In this project, we have developed a Knowledge-guided Baysian SVM approach, which allows performing the variable selection and incorporating the prior structural information simultaneously. This method relies on specifying the structural network in the Ising priors combined with the spike-and-slab priors. The numerical results confirm the performance of our method in terms of the improved prediction and variable selection accuracy. However, we expect that the performance will be influenced by the level of agreement between the prior structural information and actual underlying predictive structure. There will be significant gains when the working graph is correctly specified, and a robust performance when the working graph is not incorporated or miss-specified. One of the limitations of our model is that we use the data augmentation technique and introduce more hyper-parameters than other methods. In order to achieve better performance, the hyper-parameter tuning procedure may be computationally expensive, especially in high-dimensional settings.

CHAPTER 3

GRAPH-GUIDED BAYESIAN SVM WITH ADAPTIVE STRUCTURED SHRINKAGE PRIOR FOR HIGH-DIMENSIONAL DATA (ASBSVM)

3.1. Introduction

Recently, support vector machines (SVMs) have been widely used in biomedical studies for building classification models for disease risk, uncovering molecular signatures associated with a disease and identifying potential therapeutic targets (Guyon et al., 2002; Mukherjee et al., 1999). When the sample size is large enough compared to the number of features, the classical SVM has demonstrated its success in serving as a classification tool. The remarkable success of SVM is mainly due to its excellent adaptability to different data sets with the help of highly plausible geometric interpretation, and the quadratic programming formulation which can be implemented efficiently. However, one significant limitation of the standard SVM is that its performance deteriorates when the sample size is small compared to the number of features. In recent genomics studies, for example, gene expression data often involve tens of thousands of genes and a large portion of data are redundant and noisy. This poses a great challenge in detecting the important signals which can be associated with the phenotype.

Recently, several SVM methods have been developed by replacing the penalty functions of the standard L_2SVM to accommodate different purposes. The L_1 norm penalized SVM (L_1SVM) (Bradley and Mangasarian, 1998; Song et al., 2002; Zhu et al., 2004), produces sparse models by adopting the LASSO technique (Tibshirani, 1996) into SVM. However, the L_1SVM does not take correlations among predictors into account. In contrast, double regularization SVM (DrSVM) (Wang, Zhu, and Zou, 2006), which applies the elastic-net penalty to encourage the selection of grouped features; the $L\infty$ penalized SVM (Zou and Yuan, 2008) uses a grouped variable selection scheme such that all features derived from the same factor are include or excluded simultaneously; the smoothly clipped absolute deviation SVM (SCADSVM) uses a non-convex continuous penalty to select correlated features and eliminate biases in estimating nonzero coefficients. Despite their successes, these methods rely solely on the sparse estimation of coefficients and as a result they are still prone to fail to detect the important but weak features.

It is a well known fact that genes lie on a graphical structure and interact with connected genes in biological processes, and the neighboring genes tend to work jointly to influence biological procedures. For example, there are certain pathways associated with cancer risk and the expression levels of the genes in the pathway can be positively/negatively correlated. Most of the individual genes in the pathway often have weak influence, but their aggregated signal can be stronger and hence easier to be detected. Such pathway and graphical knowledge on genes or other entities have been structured and stored in various databases, such as Kyoto Encyclopedia of Genes and Genomes (KEGG) (Ogata et al., 1999), Gene Ontology (GO)(Ashburner et al., 2000) and BioCarta (Nishimura, 2001). It has been shown to be highly valuable to incorporate such graphical knowledge into analysis of gene expression data in relation to disease risk. For example, a network-constrained penalty was used to encourage smoothness of the connected features with respect to a graph (Li and Li, 2008); a group penalty was developed the weighted L_γ norm to realize "grouped" feature selection (Pan, Xie, and Shen, 2010); a nonconvex penalty proposed was then proposed without assuming the coefficients for the connected features being similar (Kim, Pan, and Shen, 2013). In the Bayesian framework, the spike and slab priors combined with the Markov Random Field (MRF) prior were proposed to encourage the joint selection of features (Li and Zhang, 2010; Stingo and Vannucci, 2010). Zhou and Zheng (2013) developed a Bayesian random graph-constrained model to allow uncertainty over the graph.

On the other hand, several Bayesian SVM approaches have been proposed for classification and feature selection. A comprehensive formulation of SVM in the Bayesian setting is given by Mallick, Ghosh, and Ghosh (2005). However, they do not make any attempt at variable selection along with class prediction. Simultaneous gene selection and class prediction in the Bayesian SVM set-up has been discussed in multi-class cases (Chakraborty, 2009). A Bayesian elastic-net model (Li and Zhang, 2010) has been formulated as a prior structure similar to the elastic-net for linear regression problems. A variational inference approach has been proposed in Luts and Ormerod (2014) to provide faster computation. More recently, a knowledge-guided Bayesian linear SVM which enables incorporation of the prior network information among predictors has been proposed, known as KBSVM (Sun et al., 2018), and shown that it outperforms existing SVM methods. However, in high-dimensional setting, the Markov random field prior or the Ising prior used in KBSVM suffers the phase transition problem, where adjacent indicator variables are either extremely correlated or almost uncorrelated, resulting in a very different tuning of the sparsity parameter.

In this paper, we propose a Bayesian shrinkage approach analogous to the work of Chang, Kundu, and Long (2018) in the linear regression framework. The proposed approach assigns Laplace priors to the regression coefficients and incorporates the underlying graph information via a hyper-prior for the shrinkage parameters in the Laplace priors. Specifically, the shrinkage parameters are assigned a log-normal prior specifying the inverse covariance matrix as a graph Laplacian (Chung and Graham, 1997), which has a zero or positive partial correlation depending on whether the corresponding edge is absent or present. This enables smoothing of shrinkage parameters for connected variables in the graph and conditional independence between shrinkage parameters for disconnected variables. Thus, the resulting approach encourages connected variables to have a similar degree of shrinkage in the model without forcing their regression coefficients to be similar in magnitude.

The rest of this article is organized as follows. In Section 2, we describe our model and the MCMC algorithm for posterior inference and prediction. In Section 3, we evaluate the performance of our model in comparison with other existing methods in simulations. In Section 4, we apply our method to a cancer genomics study. In section 5, we summarize our findings and discuss possible future extensions. All derivations and proofs are provided in the Appendix A.

3.2. Methods

3.2.1. Likelihood

Suppose there are n samples in the training set of data where $y_i \in \{-1, 1\}$ are the binary outcome variables and $\mathbf{x}_i$ are the $(p+1)$ dimensional feature vector including the intercept. The classical SVM seeks to find a classification function f to separate the two classes by minimizing

$$\Theta(\beta) = \sum_{i=1}^{N} \max(1 - y_i f(\mathbf{x}_i), 0) + R(f), \tag{3.1}$$

where $\sum_{i=1}^{N} \max(1 - y_i f(\mathbf{x}_i), 0)$ is the hinge loss function and R is a regularization function controlling the complexity of f. For the linear classifier $f = \mathbf{x}_i'\boldsymbol{\beta}$, minimizing the objective function (3.1) is equivalent to find the mode of the following pseudo-posterior density (Henao, Yuan, and Carin,

2014).

$$p(\boldsymbol{\beta}|X,\mathbf{y},\kappa) \propto p(\boldsymbol{\beta})L(\mathbf{y}|X,\boldsymbol{\beta},\kappa) \tag{3.2}$$

$$\propto p(\boldsymbol{\beta})\prod_{i=1}^{n}\kappa e^{-2\kappa\max(1-y_i\mathbf{x}_i'\boldsymbol{\beta},0)}, \tag{3.3}$$

$$\propto p(\boldsymbol{\beta})\prod_{i=1}^{n}\int_0^\infty \frac{\sqrt{\kappa}}{\sqrt{2\pi\rho_i}}e^{-\frac{\kappa(\rho_i+1-y_i\mathbf{x}_i'\boldsymbol{\beta})^2}{2\rho_i}}d\rho_i, \tag{3.4}$$

Note that (3.3) obviously prefers the coefficients that reduces the hinge loss, and is called the pseudo-likelihood as it does not sum to a constant. (3.4) rewrites the likelihood as a location-scale mixture of normals by introducing a latent variable ρ_i to facilitate Gibbs sampling.

3.2.2. Priors for the parameters

We assign the following priors for $\boldsymbol{\beta}$ and the form is taken as

$$p(\boldsymbol{\beta}|\boldsymbol{\lambda}) = \frac{1}{2^p}\prod_{j=1}^{p}\lambda_j e^{-\lambda_j|\beta_j|}. \tag{3.5}$$

If the shrinkage parameters λ_j are homogeneous ($\lambda_j \equiv \lambda$) and fixed, (3.5) boils down to the Bayesian lasso prior. In our model, λ_j are heterogeneous and random, so that they are able to learn the shrinkage level adaptive to the coefficient β_j and the graphical structure incorporated.

We use the lognormal prior for the shrinkage parameters λ_j. That is, we have

$$\log\pi(\boldsymbol{\alpha}|\boldsymbol{\mu},\Omega) = C_\nu + \frac{1}{2}\log|\Omega| - \frac{1}{2\nu}(\boldsymbol{\alpha}-\boldsymbol{\mu})'\Omega(\boldsymbol{\alpha}-\boldsymbol{\mu}), \tag{3.6}$$

where $\boldsymbol{\alpha} = (\log\lambda_1,\ldots,\log\lambda_p)^T$. Here, $\boldsymbol{\mu} = \mu\mathbf{1}$ is the sparsity parameter and ν is the coefficient-adaptivity parameter. Obviously, the larger μ is, the larger λ_j tend to be. Assume for now that $\Omega = I$. We can also see that, the larger ν is, the more volatile λ_j is, which leads to greater sensitivity to the coefficient β_j—hence the name of ν.

The network information is conveyed through Ω, which takes the following form

$$\Omega = \begin{bmatrix} 1+\sum_{j\neq 1}\omega_{1j} & -\omega_{12} & \cdots & -\omega_{1p} \\ -\omega_{21} & 1+\sum_{j\neq 2}\omega_{2j} & \ddots & -\omega_{2p} \\ \vdots & \ddots & \ddots & \vdots \\ -\omega_{p1} & -\omega_{p2} & \cdots & 1+\sum_{j\neq p}\omega_{pj} \end{bmatrix},$$

and we assign the following prior to $\boldsymbol{\omega} = \{\omega_{jk} : j \neq k\}$

$$\pi(\boldsymbol{\omega}) \propto |\Omega|^{-1/2} \prod_{G_{jk}=1} \omega_{jk}^{a_\omega - 1} \exp(-b_\omega \omega_{jk}) 1(\omega_{jk} > 0) \prod_{G_{jk}=0} \delta_0(\omega_{jk}), \tag{3.7}$$

where δ_0 is the Dirac delta function concentrated at 0 and $1(\cdot)$ is the indicator function. Since Ω is symmetric and diagonally dominant, it is guaranteed to be positive definite. According to (**??**), we have $\omega_{jk} = 0$ if $G_{jk} = 0$ and $\omega_{jk} > 0$ if $G_{jk} = 1$. In other words, the shrinkage parameters λ_j and λ_k have a positive partial correlation if predictors j and k are connected and have a zero partial correlation otherwise. The magnitudes of the positive partial correlations are automatically learned from the data through the normal vector coefficients, with a higher partial correlation leading to the smoothing of corresponding shrinkage parameters.

Our framework has several appealing features. First, a higher positive partial correlation between two connected predictors results in an increased probability of having both predictors included or excluded. This is more appealing when both variables are important or unimportant. Second, in the case where one of the connected predictors is important and the other is not, the method can learn from the data and impose a weak partial correlation, thereby enabling the corresponding shrinkage parameters to act in a largely uncorrelated manner. Finally, the selection of unconnected variables is guided by shrinkage parameters which are partially uncorrelated.

The prior in (3.7) involves a shape parameter a_ω and the rate parameter b_ω, which serve the similar roles as those of the gamma distribution. Note that they directly regulate the correlations ω_{jk} between the elements of $\boldsymbol{\alpha}$. In order for the aforementioned features to work as expected, two conditions must be met. First, the mean of ω_{jk} must be large enough to encourage strong correlation between shrinkage parameters for connected variables. At the same time, the variance of ω_{jk} must be large enough so that ω_{jk} can take a small value in case only one of j and k is an informative

predictor while the other is uninformative. Chang et al. (Chang, Kundu, and Long, 2018) suggests that $2 \leq a_\omega \leq 4$ and $b_\omega = 1$ should work for a broad range of scenarios, although more general choices are also possible.

3.2.3. Posterior Inference

Note that it is helpful to express the Laplace prior as a location-scale mixture of normals.

$$e^{-\lambda_j|\beta_j|} = \int_0^\infty \frac{\lambda_j}{\sqrt{2\pi\tau_j}} \exp\left(-\frac{\lambda_j^2\tau_j^2 + \beta_j^2}{2\tau_j}\right) d\tau_j. \tag{3.8}$$

This faciliate the sampling of β_j. The full pesudo-posterior density is given by

$$\begin{aligned} p(\kappa, \boldsymbol{\rho}, \boldsymbol{\tau}, \boldsymbol{\alpha}, \boldsymbol{\omega}, \boldsymbol{\beta}|\mathbf{y}, K) \propto & \prod_{i=1}^n \frac{\sqrt{\kappa}}{\sqrt{2\pi\rho_i}} e^{-\frac{\kappa(\rho_i+1-y_i\mathbf{x}_i'\boldsymbol{\beta})^2}{2\rho_i}} \times \kappa^{a_\kappa-1}e^{-\kappa b_\kappa} \\ & \times \prod_{j=1}^{p+1} \frac{e^{2\alpha_j}}{\sqrt{2\pi\tau_j}} \exp\left(-\frac{e^{2\alpha_j}\tau_j^2 + \beta_j^2}{2\tau_j}\right) \\ & \times \exp\left(-\frac{1}{2\nu}(\boldsymbol{\alpha}-\boldsymbol{\mu})'\Omega(\boldsymbol{\alpha}-\boldsymbol{\mu})\right) \\ & \times \omega_{jk}^{a_\omega-1} \exp(-b_\omega\omega_{jk}). \end{aligned}$$

It is not stratightforward to directly sample the model paramters from this complex distribution. Therefore, we use the Markov chain Monte Carlo sampling procedure. In particular, Metropolis-Hastings (MH) sampling (Gelfand and Smith, 1990) within Gibbs sampling algorithms (Metropolis et al., 1953) is used. We list the conditional distributions and illustrate the MH procedures in this section.

The conditional distribution of κ is given by

$$\kappa|\boldsymbol{\beta}, X, \mathbf{y}, \boldsymbol{\rho} \sim \mathcal{G}\left(a_\kappa + \frac{3n}{2}, b_\kappa + \sum_{i=1}^n \frac{(\rho_i + 1 - y_i\mathbf{x}_i'\boldsymbol{\beta})^2}{2\rho_i}\right) \tag{3.9}$$

Note that this sampling step can be replaced by the following, as the augmented variables ρ_j can be marginalized.

$$\kappa|\boldsymbol{\beta}, X, \mathbf{y} \sim \mathcal{G}\left(a_\kappa + n, b_\kappa + 2\sum_{i=1}^n \max(1 - y_i\mathbf{x}_i'\boldsymbol{\beta}, 0)\right). \tag{3.10}$$

The conditional distribution of ρ_i is given by

$$\rho_i|\boldsymbol{\beta}, \mathbf{x}_i, y_i, \kappa \sim \mathcal{GIN}(\frac{1}{2}, \kappa, \kappa(1 - y_i\mathbf{x}_i'\boldsymbol{\beta})^2). \tag{3.11}$$

Here, $\mathcal{GIN}$ stands for the generalized inverse gaussian distribution. Note that it is equivalent to sample ρ_i^{-1} from the inverse Gaussian distribution, denoted by $\mathcal{IN}$, as follows.

$$\rho_i^{-1}|\boldsymbol{\beta}, \mathbf{x}_i, y_i, \kappa \sim \mathcal{IN}(|1 - y_i\mathbf{x}_i'\boldsymbol{\beta}|^{-1}, \kappa). \tag{3.12}$$

Note that density function of $\mathcal{IN}(\mu, \lambda)$ is given by

$$f(x; \mu, \lambda) = \frac{\sqrt{\lambda}}{\sqrt{2\pi x^3}} e^{\frac{-\lambda(x-\mu)^2}{2\mu^2 x}}.$$

Similarly, the conditional distribution of τ_j^{-1} is given by

$$\tau_j^{-1}|\beta_j, \lambda_j \sim \mathcal{IN}(\lambda_j/|\beta_j|, \lambda_j^2). \tag{3.13}$$

The conditional distribution of $\boldsymbol{\beta}$ follows the multivariate Gaussian distribution.

$$\boldsymbol{\beta}|X, \mathbf{y}, \kappa, \boldsymbol{\rho}, \boldsymbol{\tau} \sim \mathcal{N}((D_\tau^{-1} + \kappa X' D_\rho^{-1} X)^{-1} \kappa Z' \mathbf{1}, (D_\tau^{-1} + \kappa X' D_\rho^{-1} X)^{-1}), \tag{3.14}$$

where $D_\rho = \mathrm{diag}(\rho_1, \ldots, \rho_n)$, $D_\tau = \mathrm{diag}(\tau_1, \ldots, \tau_{p+1})$, $\mathbf{1}$ is a vector of 1's, and Z is an $n \times (p+1)$ matrix with the ith row $\mathbf{z}_i = (1 + \rho_i^{-1}) y_i \mathbf{x}_i$.

The conditional distribution of ω_{jk} follows the Gamma distribution. If $(j, k) \in E$ with $j < k$, we have

$$\omega_{jk}|\boldsymbol{\alpha} \sim \mathcal{G}(a_\omega, b_\omega + \frac{1}{2\nu}(\alpha_j - \alpha_k)^2). \tag{3.15}$$

If $(j, k) \notin E$, we have $\omega_{jk} = 0$. For $j > k$, we have $\omega_{jk} = \omega_{kj}$.

Finally, the conditional distribution $\boldsymbol{\alpha}$ is given by

$$\pi(\boldsymbol{\alpha}|\boldsymbol{\tau}, \boldsymbol{\omega}) \propto [\prod_{j=1}^{p+1} \exp\left(2\alpha_j - \frac{e^{2\alpha_j}\tau_j}{2}\right)] \exp\left(-\frac{1}{2\nu}(\boldsymbol{\alpha} - \boldsymbol{\mu})'\Omega(\boldsymbol{\alpha} - \boldsymbol{\mu})\right) \tag{3.16}$$

Since $\pi(\boldsymbol{\alpha}|\boldsymbol{\tau},\boldsymbol{\omega})$ has an unknown density form, we resort to the MH algorithm.

We use the Laplace approximation to find a good proposal distribution for $\boldsymbol{\alpha}$. That is, the proposal density $q(.|\boldsymbol{\alpha},\boldsymbol{\tau},\boldsymbol{\omega})$ is as follows.

$$q(.|\boldsymbol{\alpha},\boldsymbol{\tau},\boldsymbol{\omega}) = \mathcal{N}(\boldsymbol{\alpha} - H^{-1}_{\boldsymbol{\tau},\boldsymbol{\omega}}(\boldsymbol{\alpha})g_{\boldsymbol{\tau},\boldsymbol{\omega}}(\boldsymbol{\alpha})/c, H^{-1}_{\boldsymbol{\tau},\boldsymbol{\omega}}(\boldsymbol{\alpha})/c), \tag{3.17}$$

where $g_{\boldsymbol{\tau},\boldsymbol{\omega}}(\boldsymbol{\alpha})$ and $H_{\boldsymbol{\tau},\boldsymbol{\omega}}(\boldsymbol{\alpha})$ are the gradient vector and the Hessian matrix of the negative conditional log-density with respect to $\boldsymbol{\alpha}$.

$$\begin{aligned} g_{\boldsymbol{\tau},\boldsymbol{\omega}}(\boldsymbol{\alpha}) &= \Omega\boldsymbol{\alpha}/\nu + \boldsymbol{\delta} - (2+\mu/\nu)\mathbf{1}, \\ H_{\boldsymbol{\tau},\boldsymbol{\omega}}(\boldsymbol{\alpha}) &= \Omega/\nu + 2\mathrm{diag}(\boldsymbol{\delta}), \end{aligned}$$

where $\boldsymbol{\delta} = (\tau_1 e^{2\alpha_1}, \ldots, \tau_{p+1}e^{2\alpha_{p+1}})^T$. Here, c controls the acceptrance rate of the MH algorithm. As c increases, the proposal is more concentrated to the current value of $\boldsymbol{\alpha}$.

Let $\boldsymbol{\alpha}^{t-1}$ be the last state of $\boldsymbol{\alpha}$, we draw a sample $\boldsymbol{\alpha}^*$ from the proposal distribution $q(.|\boldsymbol{\alpha}^{t-1},\boldsymbol{\tau}^{t-1},\boldsymbol{\omega}^{t-1})$. The proposal is then accepted with probability

$$\min\left(1, \frac{\pi(\boldsymbol{\alpha}^*|\boldsymbol{\tau}^{t-1},\boldsymbol{\omega}^{t-1})q(\boldsymbol{\alpha}^{t-1}|\boldsymbol{\alpha}^*,\boldsymbol{\tau}^{t-1},\boldsymbol{\omega}^{t-1})}{\pi(\boldsymbol{\alpha}^{t-1}|\boldsymbol{\tau}^{t-1},\boldsymbol{\omega}^{t-1})q(\boldsymbol{\alpha}^*|\boldsymbol{\alpha}^{t-1},\boldsymbol{\tau}^{t-1},\boldsymbol{\omega}^{t-1})}\right). \tag{3.18}$$

Derivations are provided in Appendix. The Markov chain Monte Carlo sampling algorithm is described in Algorithm 2

3.3. Simulation Studies

3.3.1. Design of Experiment

In this section, we study the performance of our ASBSVM methods through the simulated probit model. We simulate the examples for both the graph ($\mathcal{G}$) related covariance structure and independent covariance struture for the input features. In each experimental setting, we generate 100 datasets, each with a training sample for fitting, a validation sample for tuning and an independent test sample for estimating the prediction error (PE), prediction sensitivity (PSEN), prediction specificity (PSPEC), Matthews Correlation Coefficients (MCC), feature selection true positive (FSTP)

for $t = 1$ **to** $T - 1$ **do**
  Sample $\kappa \sim \mathcal{G}\left(a_\kappa + \frac{3n}{2}, b_\kappa + \sum_{i=1}^{n} \frac{(\rho_i + 1 - y_i \mathbf{x}_i' \boldsymbol{\beta})^2}{2\rho_i}\right)$.
  for $i = 1$ **to** n **do**
    Sample $\rho_i^{-1} \sim \mathcal{IN}(|1 - y_i \mathbf{x}_i \boldsymbol{\beta}|^{-1}, \kappa)$
  end
  Sample $\boldsymbol{\beta} \sim \mathcal{N}((D_\tau^{-1} + \kappa Z' D_\rho^{-1} Z)^{-1} \kappa Z' D_\rho^{-1}(J + \boldsymbol{\rho}), (D_\tau^{-1} + \kappa Z' D_\rho^{-1} Z)^{-1})$
  for $j = 1$ **to** p **do**
    Sample $\tau_j^{-1} \sim \mathcal{IN}(e^{\alpha_j}/|\beta_j|, e^{2\alpha_j})$
  end
  for $j = 1$ **to** p **do**
    for $k = j + 1$ **to** p **do**
      $\omega_{jk} \sim G_{jk} \times \mathcal{G}(a_\omega, b_\omega + \frac{1}{2\nu}(\alpha_j - \alpha_k)^2)$,
    end
  end
  Generate a proposal $\boldsymbol{\alpha}^* \sim q(.|\boldsymbol{\alpha}^{t-1})$.
  Generate $u \sim U(0, 1)$.
  if $u < min(1, \frac{\pi(\boldsymbol{\alpha}^*) q(\boldsymbol{\alpha}^{t-1}|\boldsymbol{\alpha}^*)}{\pi(\boldsymbol{\alpha}^{t-1}) q(\boldsymbol{\alpha}^*|\boldsymbol{\alpha}^{t-1})})$ **then**
    $\boldsymbol{\alpha}^t \leftarrow \boldsymbol{\alpha}^*$;
  else
    $\boldsymbol{\alpha}^t \leftarrow \boldsymbol{\alpha}^{t-1}$;
  end
end

Algorithm 2: MH algorithm for ASBSVM

and feature selection false positive (FSFP).

The prediction sensitivity is calculated as the proportion of positives $(y = 1)$ that are correctly identified and the prediction specificity is calculated as the proportion of negatives $(y = -1)$ that are correctly identified. MCC is defined as $\frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$, where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives and FN is the number of false negatives.

The sample size for training, validation and testing data is 200, and the feature dimension p is set at 120 and 480, representing both $n > p$ and $n < p$ cases. We also compare our results with L_1SVM, L_2SVM, DrSVM, SCADSVM and KBSVM with knowledge guided ($G^* = G$, details can be been in Chapter 2). We summarize the average PE, PSEN, PSPEC, MCC, FSTP and FSFP over the 100 datasets in Table 1.

3.3.2. Parameter Tuning

For L_1SVM, L_2SVM, DrSVM and SCADSVM, we use the penalizedSVM R-package (Becker et al., 2009) to tune the parameters in the validation datasets. For KBSVM, three parameters (μ, η and σ_0^2) need to be tuned. For our method ASBSVM, MCMC samples cannot take exact zeroes under a Laplace prior. To perform feature selection, we use two strategies, one is to include all the features (labeled as 'a' in Table 1), the other one is to treat the cut-off values as tuning parameters (labeled as 'b' in Table 1). We set $(a_\omega, b_\omega) = (2, 1)$ for ω_{jk}, which is fairly uninformative. The remaining parameters μ and ν are chosen by validation method.

3.3.3. A simulation dataset generated from the underlying graph

The probit model is used to demonstrate the benefits of incorporating prior network information into our ASBSVM method. The general idea is that the covariance structure of the simulated data has the graph information embedded, which mimics the genetic data with underlying interactions between genes. If we utilize the known graph to guide our algorithm, we should be able to improve the prediction performance and identify the relevant features. As mentioned, we use the graph $\mathcal{G}$ to represent the network among predictors. The model can be written as : $X \sim MN(0, \Omega^{-1})$, $\Omega = f(\mathcal{G})$, $P(Y = 1|X) = \Phi(X\boldsymbol{\beta}+\beta_0)$. $\mathcal{G}$ is the underlying true structure among predictors. Φ is the CDF of the standard normal distribution. β_0 is the intercept set to 0.5 and $\boldsymbol{\beta} = (0.8, 0.8, .., 0.8, 0.8, 0, , 0)$ is the p-dimension coefficient with the first q non-zero elements. The precision matrix $\Omega = (\omega_{ij})$ is the inverse of the covariance matrix.

Here we adopt the Gaussian graphical model and allow the precision matrix to represent the connection strength between predictors. Thus, the precision matrix Ω of X should have a similar pattern to G which is the adjacency matrix of $\mathcal{G}$. Fig. 3.1 shows the procedure of how to generate the covariance matrix from the graph $\mathcal{G}$. First, we pre-define a undirected acyclic graph $\mathcal{G}$, which has $p = 120$ predictors and the first $q = 12$ are the important features, then we generate the corresponding adjacency matrix G which is a symmetric $p \times p$ matrix, with element "1" representing the edge between connected predictors and "0" representing no edges. Note the diagonal elements of G are 0 because each predictor itself is not connected. Second, we generate the same size $p \times p$ matrix with random numbers over an interval of $[-1, 1]$ for each edge. Third, the smallest eigen-value of the resulting matrix is calculated. If it is positive, the precision matrix is obstained

and guaranteed to be positive denite; if negative, some small number is added on the diagonal of the resulting matrix to make it positive denite. The covariance matrix is acquired by normalizing the inverse of the precision matrix.

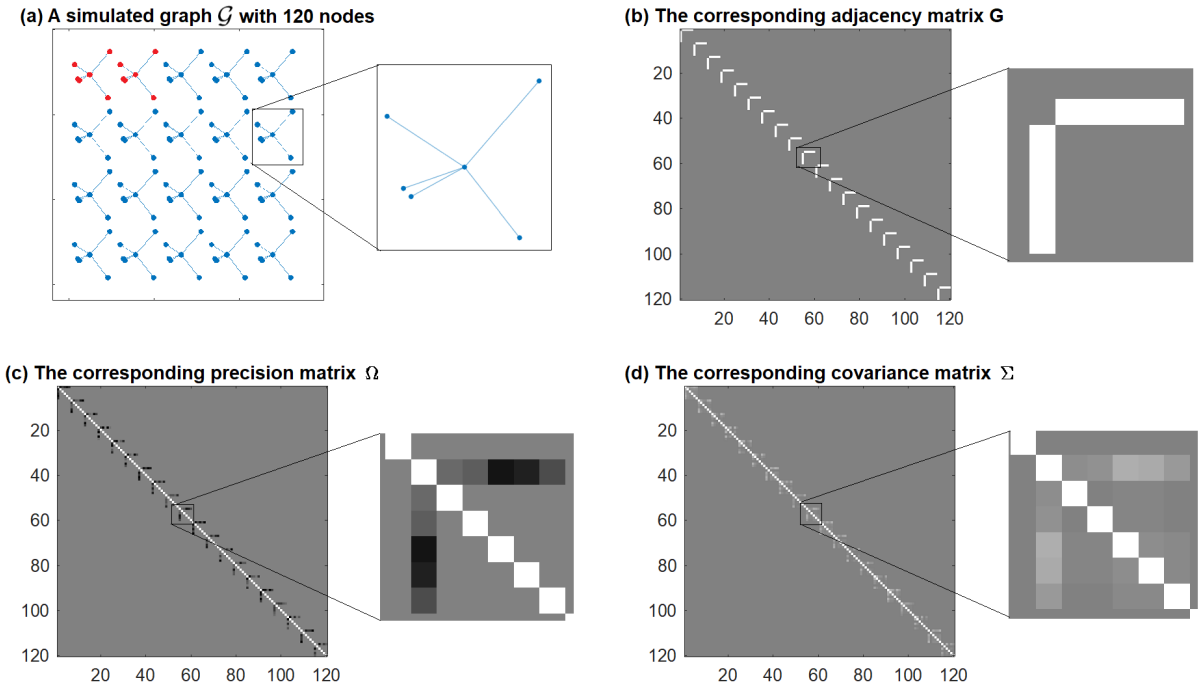

Figure 3.1: The simulated graph $\mathcal{G}$ contains 20 sub-networks, with 6 nodes in each sub-network (a), the corresponding adjacency matrix G (b), precision matrix Ω (c) and covariance matrix Σ (d)

3.3.4. Working graph simulation

Once the simulated data with graph-related covariance structure is acquired, we use the graph to guide our algorithm to make classification and feature selection. However, the graph might not be correct in practice. For example, in genetic study, the pathways in the database might be incomplete or noisy. To mimic these situations, we define the working graph adjacency matrix G^* under three conditions: 1. G indicating that the truth is known, 2. a partial graph (pG) indicating that the truth is partially known or 3. a noisy graph (nG) indicating that the graph is completely random.

The partial graph can be generated by removing some weak signals of the original precision matrix Ω (Fig. 3.2(b)). If the absolute value of the elements of Ω less than a pre-set value, then they are set to zero; The resulting matrix Ω^* is converted to the binary matrix with zeros and ones, and the diagonal elements are set to zero as shown in 3.2(d).

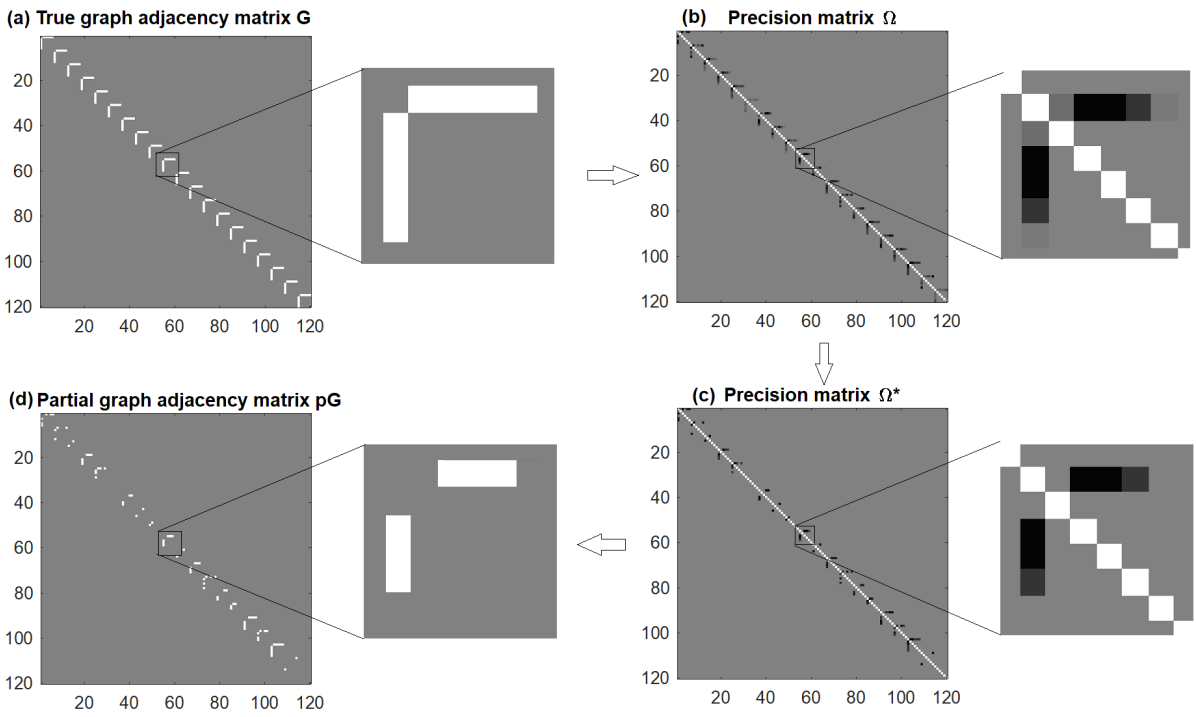

Figure 3.2: The procedure of generating partial graph pG.

The noisy graph can be generated by randomly assigning "1" on the lower triangle part of the corresponding adjacency matrix and making the upper part the same as the lower part.

3.3.5. Simulation results

In Table 3.1, we present the average prediction error of the competing methods. We report our ASBSVM results under three different conditions ($G^* = nG, pG, G$) by two scenarios (a and b). The number in parentheses is the standard error. "–" in the last two column represents no simultaneous feature selection performed. Ideally, we would like to include only the correct features in a model. If a model includes too many features then although it would be possible to capture all the true covariates but too many noise features will reduce the prediction accuracy.

The prediction error as reported in Table 3.1 for both $n > p$ and $n < p$ cases indicate that our ASBSVM model consistently outperforms other existing methods in terms of the lowest average prediction error when the working graph is correctly specified ($G* = G$). Moreover, the performance of our proposed model with the strategy b, which consider the cut off values as tuning parameters is even better than the strategy a. It is particularly clear that the proposed strategy $ASBSVM^b$ of

selecting features and fitting them for predicition is highly reliable. The L_1SVM, SCADSVM and KBSVM which do a simultaneous feature selection and classification are also quite effective in both settings.

When the graph is partially specified or noisy, the performance of our method doesn't degrade too much, and still outperforms L_2SVM and SCADSVM, demonstrating its robustness to mis-specified graph information. The robustness comes from the ability to adaptively learn the correlation between shrinkage parameters. In addition, in the high dimensional setting ($p = 480$), FSTP for the existing methods dramatically decrease, particularly for L_2SVM and DrSVM. While our proposed method drops about only 12%, which demonstrates the stability of our method. One of the reasons is that the proposed method learns small values of the partial correlations between pairs of connected important and unimportant variables resulting in weak smoothing, and imposes stronger partial correlations for other sets of connected variables, which enable accurate variable selection and prediction.

In this simulation section, we consider two strategies under three conditions for both low and high dimensional settings. We observe that if the graphical network information is associate with the outcome and we incorporate the true network information in the model, our ASBSVM model outperforms other methods in terms of both prediction and feature selection accuracy. If the prior graph is mis-specified, the performance doesnt severely deteriorate. Such stability is desirable and the results demonstrate encouraging gene selection ability and prediction power for our method.

3.4. Data Analysis

Glioblastoma is one of the most common and aggressive form of primary brain cancers in human adults, and it is also related to other cancer development. In this section, we apply the proposed methods as well as other existing methods to examine the impact of protein levels on glioblastoma survival. The data set obtained from the Cancer Genome Atlas Network (Verhaak et al., 2010) includes survival times (T) and the gene expression levels of $p = 12,999$ genes for 303 glioblastoma subjects. We are interested in making predition on the one year survival status. The survival label

Method	PE (%)	PSen (%)	PSpec(%)	MCC (%)	FSTP (%)	FSFP (%)
$p = 120$, $q = 12$						
L_2SVM	22.88 (0.16)	80.89 (0.49)	72.84 (0.63)	54.28 (0.32)	–	–
DrSVM	22.60 (0.18)	80.73 (0.46)	73.61 (0.60)	54.82 (0.35)	79.17 (3.28)	17.57 (3.81)
SCADSVM	19.92 (0.29)	81.52 (0.41)	78.41 (0.49)	60.09 (0.58)	76.83 (1.61)	7.20 (1.64)
L_1SVM	18.46 (0.19)	83.36 (0.36)	79.46 (0.41)	63.03 (0.38)	92.42 (1.96)	24.64 (1.38)
KBSVM, $G* = G$	18.41 (0.40)	83.32 (0.44)	80.25 (0.56)	63.71 (0.80)	92.83 (1.59)	28.89 (1.34)
$ASBSVM^a$, $G* = nG$	19.63 (0.24)	79.73 (0.32)	81.02 (0.35)	60.72 (0.49)	–	–
$ASBSVM^b$, $G* = nG$	19.19 (0.24)	80.00 (0.29)	81.69 (0.35)	61.65 (0.48)	90.67 (1.14)	24.28 (2.64)
$ASBSVM^a$, $G* = pG$	19.10 (0.19)	80.18 (0.30)	81.76 (0.31)	61.91 (0.39)	–	–
$ASBSVM^b$, $G* = pG$	18.75 (0.20)	80.42 (0.29)	82.15 (0.37)	62.55 (0.42)	92.25 (0.97)	26.45 (2.92)
$ASBSVM^a$, $G* = G$	18.34 (0.23)	81.27 (0.30)	82.08 (0.38)	63.33 (0.44)	–	–
$ASBSVM^b$, $G* = G$	18.13 (0.20)	81.16 (0.29)	82.65 (0.37)	63.79 (0.41)	92.25 (1.01)	18.07 (2.39)
$p = 480$, $q = 24$						
L_2SVM	30.54 (1.76)	74.61 (0.89)	63.86 (1.10)	39.47 (0.30)	–	–
DrSVM	30.37 (0.13)	73.39 (0.69)	65.57 (0.78)	39.50 (0.27)	38.25 (3.16)	1.95 (0.26)
SCADSVM	25.85 (0.29)	75.36 (0.46)	72.85 (0.50)	48.34 (0.58)	48.42 (1.78)	6.39 (1.78)
L_1SVM	22.95 (0.24)	78.92 (0.50)	75.02 (0.53)	54.22 (0.49)	67.67 (2.34)	8.65 (0.49)
KBSVM, $G* = G$	22.90 (0.21)	78.09 (0.52)	78.93 (0.50)	55.16 (0.42)	63.35 (2.58)	1.57 (0.29)
$ASBSVM^a$, $G* = nG$	24.07 (0.24)	75.57 (0.30)	76.33 (0.32)	51.88 (0.48)	–	–
$ASBSVM^b$, $G* = nG$	23.20 (0.24)	75.93 (0.29)	77.74 (0.32)	53.65 (0.48)	74.79 (1.52)	14.29 (1.80)
$ASBSVM^a$, $G* = pG$	23.50 (0.24)	76.02 (0.29)	77.04 (0.32)	53.03 (0.47)	–	–
$ASBSVM^b$, $G* = pG$	22.99 (0.25)	76.10 (0.27)	78.02 (0.31)	54.08 (0.49)	79.71 (1.43)	19.39 (2.34)
$ASBSVM^a$, $G* = G$	22.84 (0.24)	76.77 (0.29)	77.56 (0.32)	54.31 (0.49)	–	–
$ASBSVM^b$, $G* = G$	21.88 (0.23)	77.37 (0.29)	78.95 (0.26)	56.28 (0.45)	80.33 (1.39)	16.82 (2.03)

Table 3.1: Comparison of the prediction performance for different p and q with graph related covariance structure among X .

y_i for each subject i is defined as

$$y_i = \begin{cases} 1, & T_i < 365, \Delta_i = 0, \\ -1, & T_i > 365, \end{cases}$$

where Δ_i represents censoring for each subject i. Those subjects with $T_i < 365, \Delta_i = 1$ are removed so the total number of subjects is 286 with 45% dead $(y_i = 1)$ and 55% alive $(y_i = -1)$.

To focus our analysis on the 500 genes which have the most impact on the survival status, an univariate logistic regression model is fit for each gene expression level x_j:

$$log \frac{p(y = 1)}{1 - p(y = 1)} = \beta_0 + \beta_1 x_j \tag{3.19}$$

The p value for each gene expression level x_j is acquired and ranked ascendingly, where the top 500 genes are selected. The corresponding expression levels X and survival labels y are used to apply our methods ($ASBSVM^a$ and $ASBSVM^b$) and other methods.

The prior knowledge on the graphical structure between these 500 genes is retrieved from the Kyoto Encyclopeida of Genes and Genomes (KEGG) database (Ogata et al., 1999). The corresponding adjacency matrix G^* is generated and incoporated in our proposed methods.

In Table 3.2, we provide a comparative performance of our model ASBSVM with the other existing methods in terms of the average cross-validation (CV) error and the number of selected genes. The optimal tuning parameters for each methods are chosen by the minimum 5-fold cross-validation error. Note that L2SVM and $ASBSVM^a$ don't perform feature selection, so all the 500 genes are included. L_1SVM produces a slightly better performance than L_2SVM while the cv error of them are both around 30%. The performance of DrSVM and SCADSVM is similar in terms of CV error but DrSVM select fewer number of genes. When incorporating G^* extracted from the database, both our proposed methods ($ASBSVM^b$, $ASBSVM^b$) and KBSVM produce a lower CV error comparing to the other four methods, which suggests that the prior graph improves the prediction accuracy. Particularly, comparing to L_2SVM, L_1SVM and SCADSVM, our model $ASBSVM^b$ produces the smallest prediction error and a relatively sparser model when using the cut-off values as tuning parameters.

To validate the genes selected by our method, a gene list enrichment analysis is conducted via the ToppGene Suite (Chen et al., 2009). A number of enriched pathways are identified such as Protein processing in endoplasmic reticulum (1.98×10^{-7}), extracellular matrix organization (3.45×10^{-5}), and unfolded protein response (5.60×10^{-5}). The numbers in the parentheses is the Bonferrorni-adjusted p value. These pathways were found to be linked with the cancer cell proliferation and survival (Clarke et al., 2014; Grantham et al., 2017; Hiramatsu, Joseph, and Lin, 2011; Kurtoglu et al., 2007). Moveover, the most highly enriched diseases are glioblastoma and acute promyelocytic Leukemia. Therefore, the detected pathways and diseases further confirm our method can offer great promises of improved power in detection of key molecular signatures and provide valuable insights on biological bases of diseases.

3.5. Discussions

In this article we have developed a graph-guided Bayesian SVM approach, which can incorporate the structural information between covariates in high dimensional settings. The approach relies on specifying informative priors on the log-shrinkage parameters of the Laplace priors on the re-

n=286, p=500	CV error (%)	# genes selected
L_2SVM	31.27	500
L_1SVM	29.55	468
DrSVM	27.26	369
SCADSVM	27.94	492
KBSVM	25.87	439
$ASBSVM^a$	25.53	500
$ASBSVM^b$	25.19	460

Table 3.2: Results for glioblastoma data.

gression coefficients, which results in adaptive regularization. The numerical results confirm the performance of our method in terms of the improved prediction and variable selection accuracy. Our method yields significant performance when the working graph is correctly specified, and is fairly robust when the working graph is mis-specified. One limitation of our model is that when the number of features is very large, MCMC samples might be slow to converge,and tuning parameters is also computationally expensive. Instead of drawing samples from the MCMC step, we can combine our model with the EM algorithm to obtain the MAP estimate, which leads to scalability to ultra-high dimensional settings. Another potential avenue is to extend the approach to more general classes of priors on the shrinkage parameters, which will translate to more diverse penalties. We hope to tackle these issues as future research questions of interest.

CHAPTER 4

BAYESIAN NON-LINEAR SUPPORT VECTOR MACHINE FOR HIGH-DIMENSIONAL DATA WITH INCORPORATION OF GRAPH INFORMATION ON FEATURES (BNSVM)

4.1. Introduction

Recently, rapid advances in high-throughput technologies have generated a large amount of omics data such as gene expressions data. As a result, new challenges have emerged related to the analysis and interpretation of such omics data. For instance, in genomics studies, the number of gene expression features is often much larger than the sample size. Because of this high dimensionality, one of the challenges is to avoid over-fitting the data. Another challenge is feature selection, i.e., selection of a subset of informative features, leading to more interpretable results.

The linear support vector machine (SVM) is a popular technique to handle such high dimensionality and has been extended to select informative features by applying penalties on the coefficients such as L_1SVM (Bradley and Mangasarian, 1998; Song et al., 2002; Zhu et al., 2004) implementing the LASSO technique (Tibshirani, 1996) into SVM, DrSVM (Wang, Zhu, and Zou, 2006) combining the L_1 and ridge penalties to encourage the selection of correlated features, and SCADSVM (Becker et al., 2011; Zhang et al., 2005) adoptting smoothly clipped absolute deviation penalty (Fan, 2001) to into SVM. In addition to frequentist approaches, Bayesian SVM with variable selection methods have received much attention recently with many successful applications (Luts and Ormerod, 2014; Marchiori and Sebag, 2005). Bayesian approaches can naturally incorporate the prior knowledge and make posterior inference explaining uncertainty of model parameters. Recently, a knowledge-guided Bayesian linear SVM (Sun et al., 2018), enables incorporation of the prior network information among predictors, known as KBSVM, and shows that it outperforms a number of preexisting linear SVM approaches that do not take advantage of such knowledge.

However, if the data are not linearly separable, the existing linear SVM methods may not be adequate. To address this problem, the non-linear separation for SVM can be realized by mapping the original data into some high dimensional feature space where the data is linearly separable and constructing an optimal hyperplane in this space. This mapping is performed by a kernel function,

which is referred to as ”the kernel trick” (Vapnik and Vapnik, 1998). Although it is more challenging to perform feature selection for non-linear SVM, some techniques based on the frequentist framework have been developed. Weston et al. (2001) reduced the feature dimensions by minimizing bounds on the leave-one-out error via a gradient approach; Zhang (2006) implemented a smoothing spline ANOVA framework to conduct simultaneous classification and feature selection; Mangasarian and Kou (2007) proposed an approach that inserts a diagonal indicator matrix into the non-linear kernel and minimizes the objective function as well as the number of features selected. There is little work on feature selection in the Bayesian non-linear SVM framework, to the best of our knowledge.

In this work, we propose to incorporate the prior graph information such as pathways from functional genomics to further guide feature selection. One of the primary motivations for incorporating such pathways information is that weak signals are often grouped into pathways and accounting for the structure information among them has the potential to increase power of detecting key signatures and yield biologically more meaningful results. Such informative priors for related features often lie on undirected acyclic graphs where nodes represent genes and edges represent functional interactions between genes. Some recent works use the known graph or network information describing the relationships between features to guide feature selection, which leads to improvement in prediction and feature selection, especially for high dimensional data. For example, Li and Li (2008) proposed a network-constrained penalty to encourage smoothness of the connected features with respect to a graph; Pan, Xie, and Shen (2010) developed a group penalty using the weighted L_γ norm to realize ”grouped” feature selection; Kim, Pan, and Shen (2013) proposed a nonconvex penalty without assuming the coefficients for the connected features being similar. In the Bayesian framework, Li and Zhang (2010) and Stingo and Vannucci (2010) proposed spike and slab priors combined with the Markov Random Field (MRF) prior to encourage the joint selection of features. Zhou and Zheng (2013) developed a Bayesian random graph-constrained model to allow uncertainty over the graph. More recently, Chang, Kundu, and Long (2018) developed a Bayesian shrinkage approach by assigning independent Laplace priors on the regression coecients, while incorporating the graph information via the hyperprior imposed on the shrinkage parameters of the Laplace distributions.

However, the above-mentioned approaches for incorporating graph information are only applicable

to linear models and no existing work has been developed for non-linear SVM. To fill this important gap, we propose a Bayesian non-linear SVM feature selection method with incorporation of the graph information (BNSVM). The non-linear classifier in our model is assumed to be drawn from a zero mean Gaussian process (Henao, Yuan, and Carin, 2014) with a special covariance matrix. This covariance matrix is constructed by the usual non-linear kernel function embedded with latent binary variables representing the selection status of features. Furthermore, Ising priors are assigned to the latent binary variables to incorporate the graphical structure of the features. This Ising prior allows our model to encourage both group-wise inclusion and exclusion of neighboring features, and therefore, further improve the prediction performance.

By using the data augmentation techniques developed by Polson and Scott (2011), we re-express the likelihood, incorporate the graph-guided priors, and employ the Metropolis Hastings (MH) sampling within Gibbs sampling algorithm to perform Bayesian inference and prediction. The performance of our method is investigated by extensive simulation studies in comparison with L_1SVM, standard linear SVM (L_2SVM) and the knowledge-guided Bayesian linear SVM (KBSVM) method described in Chapter 2 as well as the non-linear SVM (Kernel-SVM) methods in terms of prediction and feature selection. Of note, among these existing methods, only L_1SVM and KBSVM can perform feature selection. We also apply our methods to a glioblastoma cancer study with a large number of genes, and construct a classification model to predict the survival status and identify the subset of genes that are predictive of patient survival.

The remainder of the paper is organzed as follows. Section 4.2 describes the Bayesian model and prediction. Section 4.3 and Section 4.4 present the simulation study and the real data application, respectively. Section 4.5 concludes the paper with brief discussion remarks.

4.2. Methods

4.2.1. Likelihood

Let $D = \{\mathbf{x}_i, y_i\}_{i=1}^n$ be n samples in the training set, where $\mathbf{x}_i \in \mathbb{R}^p$ are the feature inputs and $y_i \in \{-1, 1\}$ are the corresponding binary labels. Our task is to learn a classification rule from the training set so that we can assign a class label to any new subject observed in the future. The linear SVM is a large margin classifier which seperates two classes by maximizing the margin between

them. The classical SVM seeks to find the optimal function f by solving the following regularized hinge loss objective:

$$\widehat{f} = \arg\min_{f} \left(\sum_{i=1}^{N} (1 - y_i f(\mathbf{x}_i))_+ + \lambda R(f) \right), \tag{4.1}$$

where $(x)_+ \equiv \max(x, 0)$ and $R(f)$ is a regularization term reflecting the complexity of f. λ is a tuning parameter controlling the tradeoff between error minimization and the complexity of f. The classification function is then given by sign$(\widehat{f}(\mathbf{x}))$. Note that the optimization problem (4.1) is equivalent to finding the mode of the following pseudo-posterior:

$$\pi(f|X, \mathbf{y}, \nu) \propto \pi(f|X) L(\mathbf{y}|X, f, \nu) \tag{4.2}$$

$$\propto \pi(f|X) \prod_{i=1}^{n} \nu e^{-2\nu \max(1 - y_i f(\mathbf{x}_i), 0)} \tag{4.3}$$

$$\propto \pi(f|X) \prod_{i=1}^{n} \int_0^\infty \frac{\sqrt{\nu}}{\sqrt{2\pi\rho_i}} e^{-\frac{\nu(\rho_i + 1 - y_i f(\mathbf{x}_i))^2}{2\rho_i}} d\rho_i \tag{4.4}$$

The proposed pseudo-likelihood enables ν in (3) to learn the overall scale of the errors and a Gamma prior $\mathcal{G}(a_\nu, b_\nu)$ is assigned for ν. a_ν represents the shape parameter, b_ν represents the rate parameters of the Gamma distribution, and the values can be tuned in an uninformative or data-driven manner. (4) re-expresses (3) as a location-scale mixture of Gaussians by introducing a latent variable ρ_i (Polson et al. Polson and Scott, 2011) to facilitates Gibbs sampling. Following Henao, Yuan, and Carin (2014), we assume the non-linear classifiers $f(\mathbf{x}_i)$ to be drawn from a zero-mean Gaussian process $\mathcal{GP}(\mathbf{0}, K)$. The details are given in the following session.

4.2.2. Prior for the non-linear classifier $f(\mathbf{x})$

A Gaussian process is a collection of random variables where the joint distribution of any combination of the variables is Gaussian. Such a process $f(\mathbf{x})$ is completely determined by its mean function $\mu(\mathbf{x}) = E(f(\mathbf{x}))$ and the covariance kernel function $k(\mathbf{x}, \mathbf{x}') = E[(f(\mathbf{x}) - \mu(\mathbf{x}))(f(\mathbf{x}') - \mu(\mathbf{x}'))]$. We assume the prior mean functions $\mu(\mathbf{x})$ to be zero, which implies that there is no preference of positive or negative values for the mean given no data.

In our case, the random variables $f_1, \ldots, f_n$ replace $f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)$ and their prior covariance matrix $K \equiv K(\boldsymbol{\gamma}) \equiv K(\boldsymbol{\gamma}, X)$ is given by an $n \times n$ matrix with non-linear kernels $k(\mathbf{x}_i, \mathbf{x}_j) = \phi^{\sum_{l=1}^{p} \gamma_l (x_{il} - x_{jl})^2}$, where $\phi \in (0, 1)$ and γ_l is the latent binary variable indicating the inclusion or

exclusion of the lth feature of the model. This covariance structure encourages f_i and f_j to be highly correlated when $\mathbf{x}_i$ and $\mathbf{x}_j$ are close, or uncorrelated if $\mathbf{x}_i$ and $\mathbf{x}_j$ are far enough. The parameter ϕ determines the sensitivity of correlation to the distance. We assign the uniform distribution as the prior for ϕ.

The contribution of our work is to insert a binary variable γ_l for each feature l into the kernel function controlling the number of features used in the model.

4.2.3. Ising prior for $\boldsymbol{\gamma}$

γ_l plays an important role of performing selection of each feature l. Usually, the iid Bernoulli prior is assigned for $\boldsymbol{\gamma}$, allowing the predictors to be independently selected, while ignoring the underlying structure information among predictors. The prior structrual information of predictors is represented by a graph $\mathcal{G} = \langle V, E \rangle$, where $V = \{1, \ldots, p\}$ represents the set of predictors and the edge set $E \subset \{(j,k) : j,k \in V, j \neq k\}$ represents associations between the predictors. To take into account the fact that adjacent features are likely to influene the response jointly, we take the Ising prior for $\boldsymbol{\gamma}$ given as follows.

$$\pi(\boldsymbol{\gamma}) = C_{\mu,\eta} e^{-\mu \sum_j \gamma_j + \eta \sum_{j \neq k} G_{jk} \mathbb{I}(\gamma_j = \gamma_k)}, \tag{4.5}$$

where $C_{\mu,\eta}$ is the normalizing constant and $\mathbb{I}(\cdot)$ is the indicator function. Here, G is the adjacency matrix of $\mathcal{G}$; G_{jk} indicates the presence of an edge between the predictors j and k. The tuning parameters μ controls the sparsity of $\boldsymbol{\gamma}$ and η controls the smoothness of $\boldsymbol{\gamma}$ over E. When $\gamma_j = 1$, its neighbors are more likely to stay at "1". Similarly, when $\gamma_j = 0$, its neighbors are likely to stay at "0". Thus, the Ising prior will encourage the group-wise feature selection of the j-th and the k-th features if there is an edge between them.

4.2.4. Posterior Inference and Computation

The full data pseudo-posterior distribution is given by

$$\begin{aligned}\pi(\nu,\boldsymbol{\rho},\boldsymbol{\gamma},\phi,\mathbf{f}|\mathbf{y},X) \propto & \prod_{i=1}^{n}\frac{\sqrt{\nu}}{\sqrt{2\pi\rho_i}}e^{-\frac{\nu(\rho_i+1-y_if_i)^2}{2\rho_i}} \\ & \times e^{-\mu\sum_j\gamma_j+\eta\sum_{j\neq k}G_{jk}\mathbb{I}(\gamma_j=\gamma_k)} \\ & \times |K|^{-\frac{1}{2}}e^{-\frac{1}{2}\mathbf{f}^TK^{-1}\mathbf{f}}\times\nu^{a_\nu-1}e^{-b_\nu\nu}.\end{aligned}$$

MCMC is implemented by Metropolis Hastings within Gibbs algorithm. Most of the conditional distributions can be easily sampled. We have

$$\nu|\mathbf{f},\mathbf{y},\boldsymbol{\rho}\sim\mathcal{G}\left(a_\nu+\frac{3n}{2},b_\nu+\sum_{i=1}^{n}\frac{(\rho_i+1-y_if_i)^2}{2\rho_i}\right), \tag{4.6}$$

and we have

$$\rho_i|\mathbf{f},\mathbf{y},\nu\sim\mathcal{GIN}(1/2,\nu,\nu(1-y_if_i)^2),$$

where $\mathcal{GIN}(p,a,b)$ stands for the Generalized Inverse Gaussian distribution. Alternatively, the conditional distribution of ρ_i^{-1} is an inverse Gaussian distribution, denoted by $\mathcal{IN}$.

$$\rho_i^{-1}|\mathbf{f},\mathbf{y},\nu\sim\mathcal{IN}(|1-y_if_i|^{-1},\nu), \tag{4.7}$$

where the density function of $\mathcal{IN}(\mu,\lambda)$ is defined as below:

$$f(x;\mu,\lambda)=\frac{\sqrt{\lambda}}{\sqrt{2\pi x^3}}e^{\frac{-\lambda(x-\mu)^2}{2\mu^2x}}.$$

The conditional distribution of γ_j follows the Bernoulli distribution which is given by

$$\gamma_j|\boldsymbol{\gamma}_{-j},\mathbf{f},X\sim Ber\left(\frac{\Pi(\gamma_j=1,\boldsymbol{\gamma}_{-j})}{\Pi(\gamma_j=1,\boldsymbol{\gamma}_{-j})+\Pi(\gamma_j=0,\boldsymbol{\gamma}_{-j})}\right), \tag{4.8}$$

where $\boldsymbol{\gamma}_{-j}=(\gamma_1,\ldots,\gamma_{j-1},\gamma_{j+1},\ldots,\gamma_p)$ and $\Pi(\boldsymbol{\gamma})=|K(\boldsymbol{\gamma})|^{-\frac{1}{2}}e^{-\frac{1}{2}\mathbf{f}^TK(\boldsymbol{\gamma})^{-1}\mathbf{f}}e^{-\mu\gamma_j+\eta\sum_{k\neq j}G_{jk}I(\gamma_j=\gamma_k)}$.

The conditional distribution of ϕ is given by

$$\pi(\phi|\mathbf{f},\boldsymbol{\gamma},X) \propto |K|^{-\frac{1}{2}} e^{-\frac{1}{2}\mathbf{f}^T K^{-1}\mathbf{f}} \tag{4.9}$$

We use the Metropolis-Hasting algorithm to draw samples from $\pi(\phi|\mathbf{f},\boldsymbol{\gamma},X)$. Since $\phi \in (0,1)$, the logit normal distribution $q(x;\mu,\tau^2) = \frac{1}{\tau\sqrt{2\pi}}\frac{1}{x(1-x)}e^{-\frac{(\text{logit}(x)-\mu)^2}{2\tau^2}}$ is chosen as the proposed distribution. Assume the last state is ϕ^{t-1}, we draw a sample ϕ^* from $q(x;\phi^{t-1},\tau^2)$, where τ^2 is set to a value that keeps the acceptance rate around 40%. We accept or reject the current proposed ϕ^* with probability α_t.

$$\alpha_t = \min\left(1, \frac{\pi(\phi^*|\mathbf{f},\boldsymbol{\gamma},X)\phi^*(1-\phi^*)}{\pi(\phi^{t-1}|\mathbf{f},\boldsymbol{\gamma},X)\phi^{t-1}(1-\phi^{t-1})}\right)$$

Let $D_\rho = \text{diag}(\rho_1,\ldots,\rho_n)$ and $\mathbf{z}$ be a vector with entries $z_i = (1+\rho_i^{-1})y_i$. The conditional distribution for $\mathbf{f}$ is a multivariate Gaussian:

$$\mathbf{f}|\mathbf{y},\nu,\boldsymbol{\rho},K \sim \mathcal{N}(\nu(\nu D_\rho^{-1}+K^{-1})^{-1}\mathbf{z},(\nu D_\rho^{-1}+K^{-1})^{-1}) \tag{4.10}$$

for $t = 1$ **to** T **do**
  Sample $\nu \sim \mathcal{G}\left(a_\nu + \frac{3n}{2}, b_\nu + \sum_{i=1}^n \frac{(\rho_i+1-y_i f_i)^2}{2\rho_i}\right)$;
  for $i = 1$ **to** n **do**
    Sample $\rho_i^{-1} \sim \mathcal{IN}(|1-y_i f(x_i)|^{-1},\nu)$;
  end
  for $i = 1$ **to** p **do**
    Sample $\gamma_j \sim Ber\left(\frac{\Pi(\gamma_j=1,\boldsymbol{\gamma}_{-j})}{\Pi(\gamma_j=1,\boldsymbol{\gamma}_{-j})+\Pi(\gamma_j=0,\boldsymbol{\gamma}_{-j})}\right)$;
  end
  Generate a proposal $\phi^* \sim q(\phi;\phi^{t-1},\tau^2)$;
  Generate $u \sim U(0,1)$;
  if $u < \min\left(1, \frac{\pi(\phi^*|\mathbf{f},\boldsymbol{\gamma},X)\phi^*(1-\phi^*)}{\pi(\phi^{t-1}|\mathbf{f},\boldsymbol{\gamma},X)\phi^{t-1}(1-\phi^{t-1})}\right)$ **then**
    $\phi^t \leftarrow \phi^*$;
  else
    $\phi^t \leftarrow \phi^{t-1}$;
  end
  Sample $\mathbf{f} \sim \mathcal{N}(\nu(\nu D_\rho^{-1}+K^{-1})^{-1}\mathbf{z},(\nu D_\rho^{-1}+K^{-1})^{-1})$;
end

Algorithm 3: MCMC algorithm for BNSVM.

4.2.5. Prediction

To predict the label for a new testing case, the inference is divided into two steps: first computing the distribution of the latent non-linear function f^* corresponding to the test case; second cpmputing the predictive label probabilities of the test case.

Let $\boldsymbol{\theta} = (\gamma, \nu, \boldsymbol{\rho}, \phi)^T$ denote the vector of model parameters. The predictive distribution of f^* for a new testing data vector $\mathbf{x}^*_{p\times 1}$ given the training dataset X, $\mathbf{y}$ and $\boldsymbol{\theta}$ can be written as

$$f^*|\mathbf{x}^*, X, \mathbf{y}, \boldsymbol{\theta} \sim \mathcal{N}(\mu, \sigma^2), \tag{4.11}$$

where

$$\begin{aligned}
\mathbf{k}^* &= (k(\mathbf{x}^*, \mathbf{x}_1), \ldots, k(\mathbf{x}^*, \mathbf{x}_n))^T, \\
k^* &= k(\mathbf{x}^*, \mathbf{x}^*), \\
\Sigma &= (K + \nu^{-1} D_\rho)^{-1}, \\
\mu &= \mathbf{k}^{*T} \Sigma D_\rho \mathbf{z}, \\
\sigma^2 &= k^* - \mathbf{k}^{*T} \Sigma \mathbf{k}^*.
\end{aligned}$$

Then, we can use the probit link to compute the conditional predictive class probabilities.

$$\begin{aligned}
\pi(y^* = 1|\mathbf{x}^*, X, \mathbf{y}, \boldsymbol{\theta}) &= \int \Phi(f^*) \pi(f^*|\mathbf{x}^*, X, \mathbf{y}, \boldsymbol{\theta}) df^* \\
&= \Phi\left(\frac{\mathbf{k}^{*T} \Sigma D_\rho \mathbf{z}}{\sqrt{1 + k^* - \mathbf{k}^{*T} \Sigma \mathbf{k}^*}} \right).
\end{aligned} \tag{4.12}$$

The derivations of (4.11) and (4.12) can be found in Appendix A. To estimate the marginal predictive probability $\pi(y^* = 1|\mathbf{x}^*, X, \mathbf{y})$, the MCMC samples of $\boldsymbol{\theta}$ are used.

$$\widehat{p} = \frac{1}{M} \sum_{m=1}^{M} p(y^* = 1|\mathbf{x}^*, X, \mathbf{y}, \boldsymbol{\theta}_m), \tag{4.13}$$

where $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M$ are the MCMC samples of $\boldsymbol{\theta}$.

The prediction error can be measured by the cross-entropy between the predictive probabilities and

the actual class.

$$\mathrm{CE} = -\sum_{i=1}^{N} \mathbb{I}(y_i^* = 1) \log \widehat{p}_i - \sum_{i=1}^{N} \mathbb{I}(y_i^* = -1) \log(1 - \widehat{p}_i),$$

where N is the sample size of testing data. The decision can be made by $\widehat{y}_i = \mathrm{sign}(\widehat{p}_i - 0.5)$ and the associated prediction error can be reported as follows.

$$\mathrm{PE} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(\widehat{y}_i \neq y_i^*). \tag{4.14}$$

4.3. Simulation studies

4.3.1. Design of Experiment

In this section, we study the performance of our BNSVM methods through the simulated additive models. We simulate the examples for both the graph-related covariance structure and independent covariance structure for the input features. In each experimental setting, we generate 100 datasets, each with a training sample for fitting, a validation sample for tuning and an independent test sample for computing the following performance metrics: the prediction error (PE), prediction sensitivity (PSEN), prediction specificity (PSPEC), Matthews Correlation Coefficients (MCC), feature selection true positive (FSTP) and feature selection false positive (FSFP).

For comparision, prediction errors are reported by (4.14) as not all methods provide probabilistic prediciton. The prediction sensitivity is calculated as the proportion of positives $(y = 1)$ that are correctly identified and the prediction specificity is calculated as the proportion of negatives $(y = -1)$ that are correctly identified. MCC is defined as

$$\frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{4.15}$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives and FN is the number of false negatives. FSTP is the percentage of important features selected by the model among the total important features and FSFP is the percentage of unimportant features selected by the model among the total unimportant features.

The sample size for training, validation and testing data is 200, and the feature dimension p is set at 20, 100 and 500, representing both $n > p$ and $n < p$ cases. In addition to L_1-SVM, L_2-SVM (the linear SVM with L_2 penalty) and Kernel-SVM (the standard non-linear SVM), our method is compared with KBSVM with the correctly specified graph (KBSVM, $G^* = G$) and a simplified version of KBSVM without using graph information (KBSVM, $\eta = 0$). Of note, Sun et al. Sun et al., 2018 showed that KBSVM outperforms several penalized linear SVM methods such as Dr-SVM and SCAD-SVM, and provided additional details of KBSVM. Tables 1 and 2 present the average performance metrics over the 100 simulated datasets for each simulation setting.

4.3.2. Parameter Tuning

For L_1SVM, L_2SVM and Kernel-SVM, we use the penalizedSVM R-package Becker et al., 2009 to tune the parameters in the validation datasets. For KBSVM, three parameters (μ, η and σ_0^2) need to be tuned. For our method BNSVM, when $\eta = 0$, representing no graph incorporated, only μ needs to be tuned; when $\eta \neq 0$, representing the graph information is incorporated, two parameters (μ, η) need to be tuned to achieve the best performance in terms of PE.

4.3.3. Generating data from an underlying true graph

The additive model is used to demonstrate the benefits of incorporating prior network information into our BNSVM method. The general idea is that the covariance structure of the simulated data has the graph information embedded, which mimics the genetic data with underlying interactions between genes. If we utilize the known graph to guide our algorithm, we should be able to improve the prediction performance and identify the relevant features. As mentioned, we use the undirected graph $\mathcal{G}$ to represent the network among predictors. The model can be written as : $\mathbf{x} \sim \mathcal{N}(0, \Omega_{p\times p}^{-1})$, $f(\mathbf{x}) = x_1^2 + x_2^2 + .. + x_q^2$, where $x_1, ..., x_q$ are the first q dimensions of $\mathbf{x}$ and f is only relevant with the first q features. The binary response y is determined by a cut-off value of $f(\mathbf{x})$, which divides the two classes almost equally. The precision matrix $\Omega = (\omega_{ij})$ is the inverse of the covariance matrix with each entry

$$\omega_{ij} = \begin{cases} 0, & (i,j) \notin E \\ \rho_{ij} \in [-1,1], & \text{otherwise} \end{cases}$$

Here we adopt the Gaussian graphical model and allow the precision matrix to represent the connection strength between predictors. Thus, the precision matrix Ω of $\mathbf{x}$ should have a similar pattern

to G which is the adjacency matrix G.

Fig. 4.1 shows the procedure of how to generate the covariance matrix from the graph. First, we pre-define a undirected acyclic graph $\mathcal{G}$, which has $p = 20$ predictors and the first $q = 10$ are the important features, then we generate the corresponding adjacency matrix G which is a symmetric $p \times p$ matrix, with element "1" representing the edge between connected predictors and "0" representing no edges. Note the diagonal elements of G are 0 because each predictor itself is not connected. Second, we generate the same size $p \times p$ matrix with random numbers over an interval of $[-1, 1]$ for each edge. Third, the smallest eigenvalue of the resulting matrix is calculated, if it is positive, the precision matrix can be acquired by resaling the diagonal elements to be 1; if negative, some small number is added on the diagonal of the resulting matrix to make it positive definite, then rescale and obtain the precision matrix. The covariance matrix is acquired by inverting the precision matrix.

4.3.4. Generating working graph

Once the simulated data with graph-related covariance structure is acquired, we use the graph to guide our algorithm to make classification and feature selection. However, the graph might not be correct in practice. For example, in genetic study, the pathways in the database might be incomplete or noisy. To mimic these situations, we define the working graph adjacency matrix G^* under three conditions: 1. G indicating that the truth is known, 2. a partial graph (pG) indicating that the truth is partially known or 3. a noisy graph (nG) indicating that the graph is completely random. Fig. 4.2 shows the three conditions.

The partial graph can be generated by removing some weak signals of the original precision matrix Ω (Fig. 4.2(b)). If the absolute value of the elements of Ω less than a pre-set value, then they are set to zero; The resulting matrix is converted to the binary matrix with zeros and ones, and the diagonal elements are set to zero as shown in Fig. 4.2(d).

The noisy graph can be generated by randomly assigning "1" on the lower triangle part of the corresponding adjacency matrix and making the upper part the same as the lower part. A noisy graph example is shown in Figure 4.2(e).

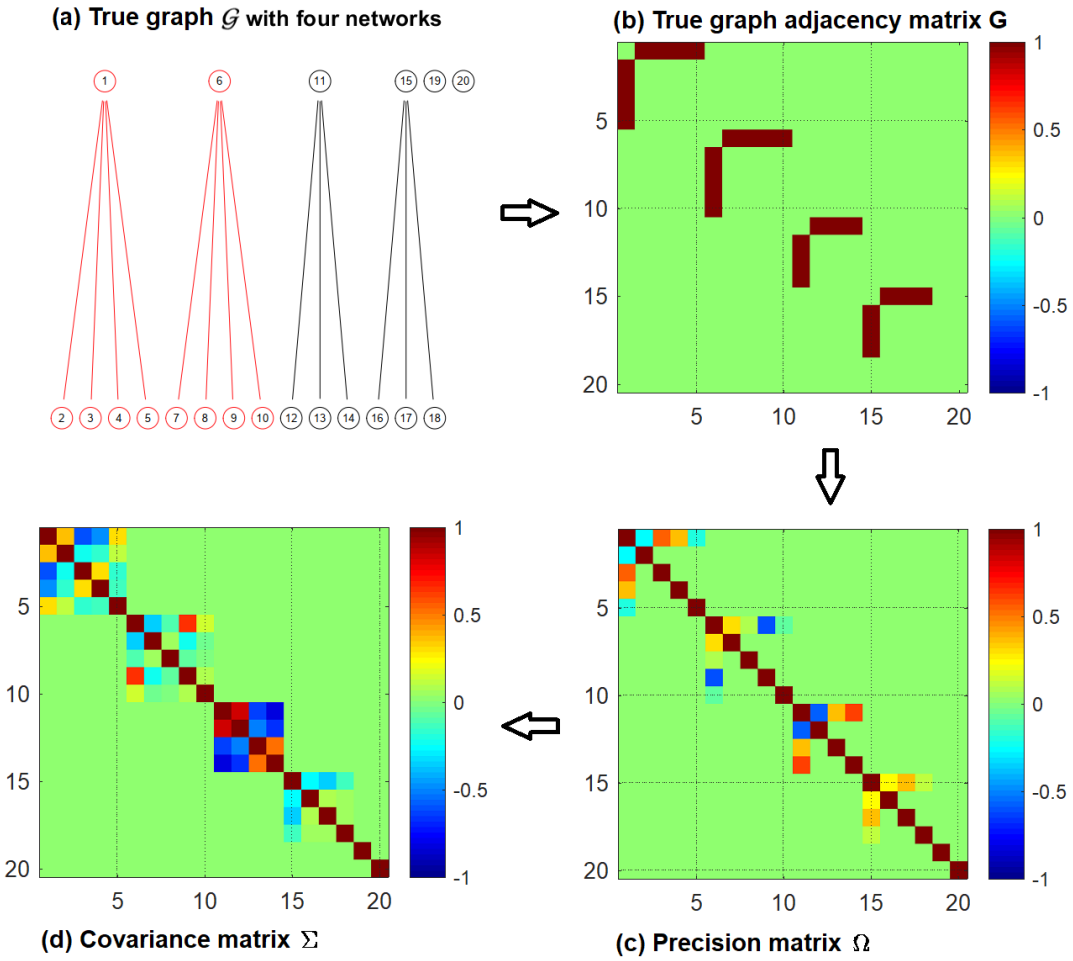

Figure 4.1: The true graph $\mathcal{G}$ (a) with the subsets in red are the relevant q features, the corresponding adjacency matrix G (b), precision matrix Ω (c) and covariance matrix Σ (d)

4.3.5. Simulation results

Four settings are specified for our method BNSVM: no working graph incorporated ($\eta = 0$), the working graph G^* is assigned by a noisy graph (nG), a partial graph (pG) and the true underlying graph (G). Two settings are specified for KBSVM: no working graph incorporated ($\eta = 0$), the working graph G^* is assigned by the true graph. Table 4.1 summarizes the simulation results for both $n > p$ and $n < p$ cases.

When $p = 20$ and $q = 10$, our method BNSVM($G^* = G$) gives the smallest PE and the highest MCC. The prediction performance for the four linear methods L_1SVM, L_2SVM, KBSVM($\eta = 0$) and KBSVM($G^* = G$) are similar. The non-linear Kernel-SVM gives a relatively low PE; while even

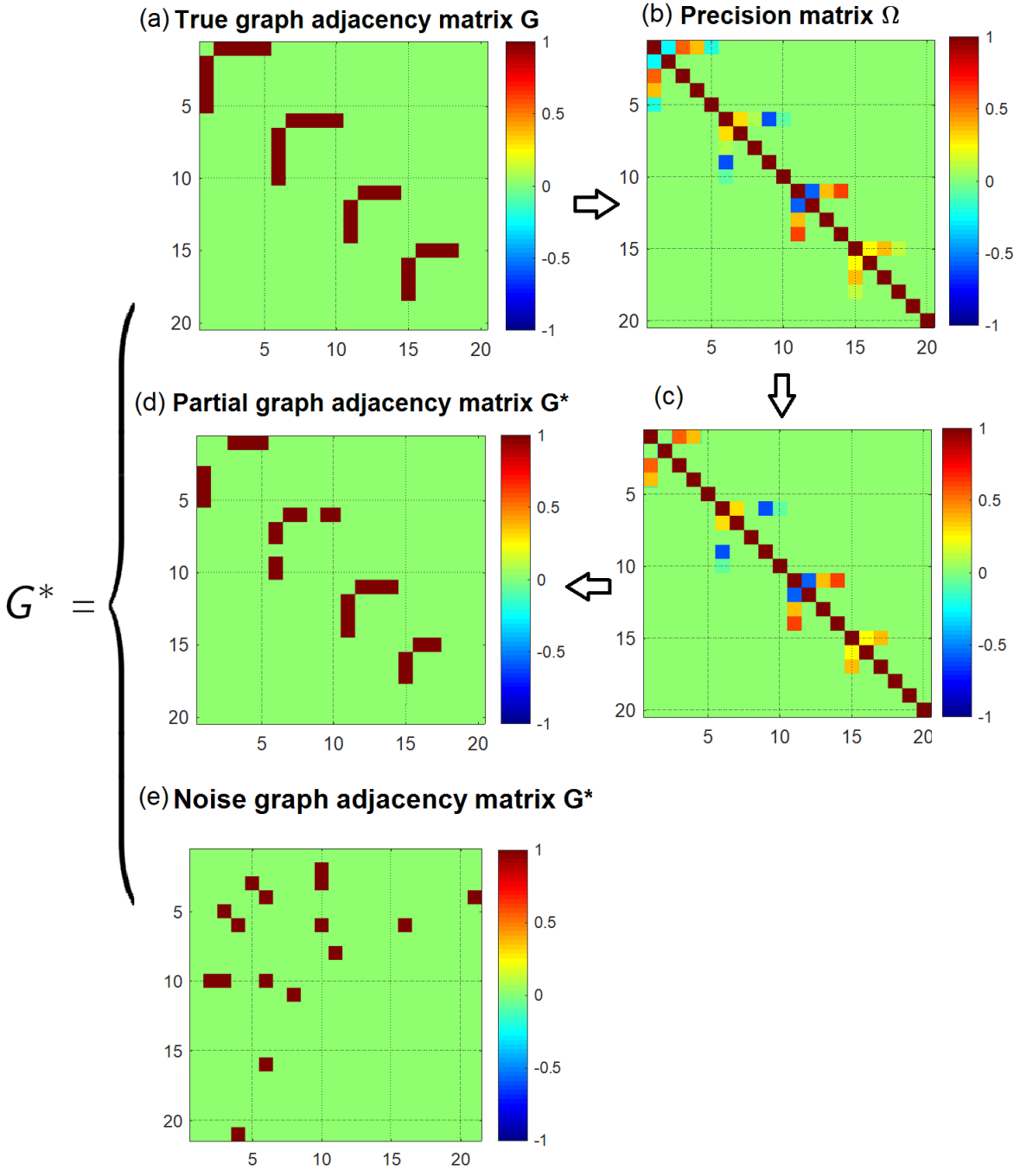

Figure 4.2: Three settings for working graph $G*$.

no graph guided ($\eta = 0$), our method has a lower PE and high MCC, comparing to the settings of $G^* = nG$ and $G^* = pG$. When the working graph is assigned by nG, PE increase about 15% comparing to the setting of BNSVM($G^* = G$), while still gives satisfactory MCC, FSTP and FSFP.

When the working graph is assigned by the true graph G, the lowest PE is achieved, moreover, it discovers all the relevant features while with the lowest FSFP at 7%. Note that when the graph is incorporated, the PE of the KBSVM drops 2% and the FSTP increases 9%, which confirms the advantage of incorporating prior knowledge even the wrong model is applied.

When $p = 100$ and $q = 20$, PE for BNSVM($G^* = G$) still gives the smallest PE comparing to the other settings and other existing methods. When the working graph is assigned by nG, the performance is very close to Kernel-SVM. If no working graph is incorporated, PE doesn't inclease a lot comparing to the case when true graph is assigned. This observation may indicate that even without the graph guidance, our algorithm still work well. If the prior knowledge is not certain, we prefer not use it. We also note that when the dimension p increases, both FSTP and FSFP decrease because the ratio of the relevant features becomes small comparing to the total number of dimensions.

When $p = 500$, since our training sample size is $n = 200$, this is the $n < p$ case. We see that all the methods generate a relatively high PE and low MCC. The performance of L_1SVM, L_2SVM, KBSVM and Kernel-SVM are similar, give the PE around 45%, while PE of our methods ($\eta = 0$, $G^* = pG$ and $G^* = G$) is below 40%. In general, our method gives the smallest PE, the greatest MCC, and high FSTP. Even when G^* is assigned by nG, the performance of our method doesn't deteriorate too much.

In addition, we simulate a new dataset which has the independent covariance structure ($\mathbf{x} \sim \mathcal{N}(0, \boldsymbol{I}_{p\times p})$). In the graphical view, all the predictors are isolated without connections. Therefore, we only need specify two settings to test for our model: $\eta = 0$ and $G^* = nG$ and one setting $\eta = 0$ for KBSVM. The results are summarized in Table 4.2. When $p = 20$, Kernel-SVM performs the best in terms of PE, our method ($\eta = 0$) achieves a similar PE but has the highest MCC. When $p = 100$ and $p = 500$, we observe that our method ($\eta = 0$) performs the best among the other settings as well as other existing methods.

In the simulation section, we consider two datasets with different covariance structures (i.e correlated or independent). We have found that the existing linear SVM methods (i.e. L_1SVM, L_2SVM, KBSVM) don't perform well for complex data, as well as the non-linear kernel-SVM in high dimension settings. If the working graph is partially or fully correctly specified, our method outperforms

Method	PE (%)	PSen (%)	PSpec(%)	MCC (%)	FSTP (%)	FSFP (%)
			$p = 20$, $q = 10$			
L_1SVM	48.30 (1.15)	43.41 (3.61)	59.80 (4.67)	4.15 (2.80)	53.00 (11.35)	61.00 (11.81)
L_2SVM	49.20 (0.86)	46.55 (2.30)	54.93 (2.75)	1.56 (1.72)	–	–
KBSVM, $\eta = 0$	46.10 (0.60)	39.87 (3.21)	62.61 (2.74)	9.38 (0.80)	22.00 (8.24)	14.00 (9.70)
KBSVM, $G* = G$	45.15 (0.58)	40.09 (4.92)	62.83 (5.68)	10.82 (1.38)	24.00 (8.41)	12.00 (9.23)
Kernel-SVM	17.35 (0.62)	79.90 (0.98)	85.31 (1.41)	65.49 (1.27)	–	–
BNSVM, $\eta = 0$	16.00 (0.73)	67.04 (2.49)	91.73 (1.17)	60.85 (2.92)	95.00 (2.52)	11.00 (1.68)
BNSVM, $G* = nG$	17.80 (0.83)	70.15 (1.10)	93.11 (1.33)	66.01 (1.74)	95.00 (3.20)	10.00 (1.98)
BNSVM, $G* = pG$	16.35 (0.89)	72.73 (1.43)	94.29 (0.83)	68.83 (1.70)	98.00 (1.25)	11.00 (2.19)
BNSVM, $G* = G$	15.50 (0.52)	73.54 (1.06)	94.18 (1.01)	69.42 (1.50)	100.00 (0.00)	7.00 (2.00)
			$p = 100$, $q = 20$			
L_1SVM	46.60 (0.81)	53.75 (1.60)	53.01 (2.14)	6.79 (1.61)	59.50 (9.01)	62.00 (7.98)
L_2SVM	48.75 (0.83)	46.43 (0.94)	51.00 (1.54)	1.15 (1.82)	–	–
KBSVM, $\eta = 0$	46.50 (0.68)	40.63 (6.62)	56.77 (1.82)	7.57 (1.31)	42.00 (8.30)	36.13 (9.85)
KBSVM, $G* = G$	45.50 (0.61)	46.77 (5.01)	61.64 (4.16)	9.08 (1.19)	25.00 (6.70)	21.12 (7.11)
Kernel-SVM	34.75 (1.01)	63.42 (2.62)	66.81 (1.99)	30.55 (2.07)	–	–
BNSVM, $\eta = 0$	30.95 (1.36)	55.31 (3.47)	78.67 (6.96)	42.13 (3.16)	62.00 (11.84)	8.25 (6.96)
BNSVM, $G* = nG$	35.10 (1.90)	43.00 (4.37)	75.16 (2.91)	34.94 (3.75)	56.00 (8.43)	12.25 (3.86)
BNSVM, $G* = pG$	32.50 (1.12)	52.83 (2.30)	85.15 (3.40)	37.19 (3.09)	64.50 (8.99)	5.75 (1.33)
BNSVM, $G* = G$	27.50 (0.94)	74.05 (3.78)	71.01 (3.79)	46.32 (1.66)	66.50 (6.78)	5.13 (1.32)
			$p = 500$, $q = 20$			
L_1SVM	46.00 (1.19)	52.21 (1.77)	55.86 (1.31)	8.03 (2.39)	20.50 (4.33)	21.34 (3.31)
L_2SVM	48.30 (1.13)	47.56 (2.04)	49.25 (1.49)	3.44 (2.27)	–	–
KBSVM, $\eta = 0$	45.90 (1.01)	42.47 (7.16)	51.75 (2.10)	7.70 (2.04)	44.00 (12.20)	37.10 (13.07)
KBSVM, $G* = G$	44.90 (0.71)	44.68 (6.00)	55.92 (1.37)	12.31 (0.83)	48.00 (13.33)	46.73 (13.69)
Kernel-SVM	45.05(0.76)	58.02 (4.47)	51.78 (5.33)	10.20 (1.52)	–	–
BNSVM, $\eta = 0$	36.40 (0.82)	55.13 (1.98)	74.00 (2.87)	34.19 (2.64)	62.00 (8.67)	1.27 (0.31)
BNSVM, $G* = nG$	42.50 (0.38)	48.80 (2.45)	69.07 (1.69)	31.82 (2.70)	53.00 (7.69)	3.33 (0.41)
BNSVM, $G* = pG$	37.56 (1.32)	52.53 (3.11)	74.95 (2.53)	35.47 (2.38)	63.00 (8.62)	1.56 (0.31)
BNSVM, $G* = G$	33.35 (1.01)	57.48 (2.50)	78.09 (2.50)	36.73 (2.84)	65.50 (8.85)	1.40 (0.29)

Table 4.1: Comparison of the prediction performance and variable selection when the dimension of predictions p changes from 20 to 500 among different methods. q is the number of relevant variables. $\eta = 0$ represents the working graph $G*$ is not incorporated in KBSVM and BNSVM methods.

all the other methods in terms of both prediction and selection accuracy. If the working graph is not available or noisy, the performance is comparable to the kernel-SVM in low dimension settings (i.e $p = 20, 100$), while perform better in the high dimension setting ($p = 500$).

4.4. Data Analysis

Glioblastoma is known as one of the most aggressive brain cancers, only 12% of the samples were censored and it is also related to other cancer development. We obtained a glioblastoma data set from the Cancer Genome Atlas Network Verhaak et al., 2010. This data set includes survival times (T) and the gene expression levels of $p = 12,999$ genes (X) for 303 glioblastoma patients. To perform the classification, a new indicator variable Z is defined to denote the one year survival

Method	PE (%)	PSen (%)	PSpec(%)	MCC (%)	FSTP (%)	FSFP (%)
			$p = 20$, $q = 10$			
L_1SVM	47.89 (0.71)	41.12 (3.24)	63.98 (3.64)	4.22 (1.00)	42.22 (9.97)	41.11 (8.65)
L_2SVM	48.55 (1.04)	41.06 (1.94)	56.21 (2.87)	3.12 (2.12)	–	–
KBSVM, $\eta = 0$	45.55 (0.73)	33.79 (4.20)	74.83 (3.43)	9.58 (1.35)	28.00 (7.23)	19.00 (9.63)
Kernel-SVM	20.20 (0.68)	79.30 (1.02)	80.29 (1.48)	50.74 (1.35)	–	–
BNSVM, $\eta = 0$	20.95 (1.06)	86.00 (2.41)	72.11 (3.00)	59.43 (2.02)	97.00 (2.00)	20 (5.76)
BNSVM, $G* = nG$	21.60 (2.49)	84.30 (1.32)	72.58 (5.05)	57.50 (4.71)	90.00 (8.39)	25.00 (6.13)
			$p = 100$, $q = 20$			
L_1SVM	47.70 (0.92)	51.38 (1.70)	54.96 (1.06)	4.55 (1.84)	69.00 (7.38)	70.62 (6.56)
L_2SVM	50.00 (1.22)	47.18 (1.50)	52.39 (1.61)	3.18 (2.22)	–	–
KBSVM, $\eta = 0$	46.20 (0.74)	48.42 (4.19)	57.02 (3.63)	5.47 (2.25)	43.00 (11.57)	44.12 (11.67)
Kernel-SVM	38.95 (1.33)	65.91 (4.68)	55.77 (3.01)	22.86 (3.14)	–	–
BNSVM, $\eta = 0$	32.87 (0.95)	55.53 (4.15)	59.38 (2.07)	34.32 (1.58)	63.50 (10.76)	1.85 (0.34)
BNSVM, $G* = nG$	35.88 (0.88)	43.83 (4.13)	71.99 (4.44)	29.28 (3.79)	54.50 (12.74)	22.63 (12.18)
			$p = 500$, $q = 20$			
L_1SVM	45.50 (0.91)	53.24 (1.02)	54.07 (1.80)	9.04 (1.80)	21.00 (4.00)	20.90 (2.43)
L_2SVM	49.85 (1.13)	48.98 (1.73)	51.49 (1.01)	2.32 (2.16)	–	–
KBSVM, $\eta = 0$	44.85 (0.67)	42.38 (3.87)	56.05 (1.62)	8.83 (1.53)	60.00 (12.60)	55.62 (13.98)
Kernel-SVM	45.35 (0.88)	47.66 (4.70)	62.54 (4.12)	10.69 (1.51)	–	–
BNSVM, $\eta = 0$	36.97 (0.76)	49.11 (1.52)	66.06 (2.58)	30.20 (1.22)	59.50 (9.41)	1.85 (0.34)
BNSVM, $G* = nG$	41.67 (0.89)	44.80 (1.93)	65.49 (3.08)	27.98 (1.04)	56.00 (9.68)	2.04 (0.31)

Table 4.2: Comparison of the prediction performance and variable selection when the predictors are independent.

outcome:

$$Y = \begin{cases} 1, & T < 365, \Delta = 0, \\ 0, & T > 365, \end{cases}$$

where Δ represents censoring. Those subjects with $T < 365, \Delta = 1$ are removed so the total number of subjects is 286 with $P(Y = 1) = 45\%$, $P(Y = 0) = 55\%$. In this section, we apply our methods and other existing methods to classify the survival status of the glioblastoma patients.

First, we fit an univariate logistic regression model for each gene x:

$$log \frac{p(x)}{1 - p(x)} = \beta_0 + \beta_1 x \tag{4.16}$$

We calculate the p value from the logistic regression and create a list of ascend ordered p-value as well as the corresponding genes. Second, we use the gene-ranking methods to select important genes, for example, we select the top 500 genes in the list and generate the new feature inputs from the gene expression levels.

Second, we create a graph G for all the $12,999$ genes from the Kyoto Encyclopedia of Genes and Genomes (KEGG) database, and we use an algorithm to retrieve the connections within the top

Table 4.3: Results of the analysis of TCGA data. $n = 286$, $p = 500$.

	CV error (%)	# selected genes
L_1SVM	29.55	468
L_2SVM	31.27	500
KBSVM, $\eta = 0$	26.94	233
KBSVM, $G^* = G$	25.87	439
Kernel-SVM	25.86	500
BNSVM, $\eta = 0$	25.00	67
BNSVM, $G^* = G$	21.43	232

500 genes, and then map them to the working graph G^*.

We specify two settings ($\eta = 0$ and $G^* = G$) for both our model and KBSVM (Sun et al. Sun et al., 2018) to compare with other methods. The optimal tuning parameters for each methods are chosen by the minimum 5-fold cross-validation error. The average cross-validation error and the number of selected genes are summarized in Table 4.3. Note that L_2SVM doesn't perform feature selection, so all the 500 genes are selected. L_1SVM selects most of the 500 genes and has a similar performance to L_2SVM. Kernel-SVM achieves a lower cross-validation error, which may suggest that the non-linear classifier is more plausible. KBSVM with graph incorporated ($G^* = G$) produces a lower cross-validation error comparing to the no graph guided one, which may indicate that the prior graph improves the prediction accuracy. For our method BNSVM, if the graph is not incorporated ($\eta = 0$), fewer genes are detected, but still provides the satisfactory cross-validation error; if the graph is incorporated ($G^* = G$), more genes are detected and achieves the smallest cross-validation error.

To validate the selected genes by our method, a pathway enrichment analysis is conducted via ToppGene Suite (Chen et al. Chen et al., 2009). If no graph information is provided ($\eta = 0$), only 67 genes including several key genes such as PICK1 and IL22 that are members of glioma-related pathways are selected and no significant pathways are identified. If the graph information is provided ($G^* = G$), our BNSVM method encourages the selection of the connected genes. As a result, a number of pathways are enriched in the set of selected genes, such as extracellular matrix organization (4.18×10^{-2}), protein processing in endoplasmic reticulum (3.29×10^{-4}), and unfolded protein response (9.34×10^{-2}), where the numbers in the parentheses are the Bonferroni-adjusted p values. The pathways detected have been found to be related to cancer cell development and survival (Clarke et al., 2014; Hiramatsu, Joseph, and Lin, 2011; Koh et al., 2018; Pointer et al.,

2016). Moreover, the most highly enriched diseases are malignant neoplasm of ovary, glioblastoma, and malignant tumor of colon. These results further confirm that the integration of existing biological knowledge yields biologically more meaningful results.

4.5. Discussions

This chapter introduces a knowledge-guided Bayesian non-linear SVM approach that uses the structural information between features to guide feature selection and is more robust than the existing linear SVM methods. Our numerical studies demonstrate that the proposed method outperforms several existing methods including the knowledge-guided Bayesian linear SVM and the standard nonlinear SVM in terms of prediction and feature selection. In the analysis of the real gene expression data, our results suggest that the integration of prior biological knowledge into our model leads to an increased ability to identify important genes, and yields biologically meaningful results and improved prediction accuracy.

CHAPTER 5

JOINT BAYESIAN VARIABLE SELECTION AND GRAPH ESTIMATION FOR NON-LINEAR SUPPORT VECTOR MACHINE (JBNSVM)

5.1. Introduction

With the advances made in the last few years in microarray technology, we are able to monitor expression measurements for tens of thousands of genes simultaneously. Several studies using microarrays to profile colon, breast and other tumors have demonstrated the potential power of expression profiling for classification (Alon et al., 1999; Hedenfalk et al., 2001). Due to the high cost, we can afford a very small number of samples, mostly less than hundred, and a key goal is to perform tumor classification in such a large p (genes), small n (samples) data pattern. In addition, there is increasing evidence from genomics studies that genes affect phenotypes through complex molecular networks or pathways, while the expression levels of individual genes in the pathways have relatively weak signals. However, the grouped signals in the pathways can be considerably stronger. So accounting for the relationships among the genes has the potential to increase power to detect true associations and yield biologically more meaningful results. Along with the tumor classification, an important task is to identify the genes that lie in the networks associated with tumor progression.

Support vector machine (SVM) (Vapnik and Vapnik, 1998) has been widely used to handle such large gene expression data sets for accurate classification. Moreover, the penalty based SVM shrinkage methods can deal with variable selection and classification simultaneously. Bradley and Mangasarian (1998), Song et al. (2002) and Zhu et al. (2004) applied the LASSO technique (Tibshirani, 1996) into SVM (L_1SVM). However, the L_1SVM does not take correlations among predictors into account. In contrast, Wang, Zhu, and Zou (2006) proposed a double regularization SVM, which combines the L_1 and ridge penalties to encourage the selection of correlated features; Zou and Yuan (2008) suggested the L_∞ penalized SVM to encourage all the features in the same group to be selected simultaneously; Zhang et al. (2005) and Becker et al. (2011) adopted the smoothly clipped absolute deviation penalty (Fan, 2001) to alleviate biases in estimating nonzero coefficients. Extending the idea of grouping to gene networks, Zhu, Shen, and Pan (2009) proposed a network-

based SVM, which considers any two neighboring genes in a network as one group, and integrates the network information to build classifiers. In addition to frequentist approaches, Bayesian SVM with variable selection methods have received much attention recently with many successful applications. The Bayesian methods are natural to incorporate the prior knowledge and make posterior inference on uncertainty of variable selection. Most recently, Sun et al. (2018) proposed a knowledge-guided Bayesian linear SVM which combines the spike-and-slab with Ising priors to enable incorporation of the prior network information among predictors; and Chang, Kundu, and Long (2018) developed a Bayesian shrinkage prior which smoothed shrinkage parameters of connected nodes to a similar degree for structural variable selection in the linear regression setting.

Apart from the SVM methods, there are a few penalty based linear formulations using the known graph or network information describing the relationships between features to guide feature selection, which leads to improvement in prediction and feature selection. For example, Li and Li (2008) proposed a network-constrained penalty to encourage smoothness of the connected features with respect to a graph; Pan, Xie, and Shen (2010) developed a group penalty using the weighted L_γ norm to realize "grouped" feature selection; Kim, Pan, and Shen (2013) proposed a nonconvex penalty without assuming the coefficients for the connected features being similar. In the Bayesian framework, Li and Zhang (2010) and Stingo and Vannucci (2010) proposed spike and slab priors combined with the Markov Random Field (MRF) prior to encourage the joint selection of features. Zhou and Zheng (2013) developed a Bayesian random graph-constrained model to allow uncertainty over the graph.

These network-based approaches have shown that incorporating network or graph information not only improves predictive performance and reproducibility, but also sheds biological insights into molecular mechanisms underlying the clinical outcome. However, they have used a linear model to establish the relationship between the genes and the cancer types, while how the genes finally explain the tumor behavior often cannot be tracked down by a simple linear structure. To address this problem, a more general form, the non-linear relationship between genes and tumor progression, can be formulated in SVM by mapping the original data into some high dimensional feature space where the data is linearly separable and constructing an optimal hyperplane in this space. This mapping is performed by a kernel function, which is referred to as "the kernel trick" (Vapnik and Vapnik, 1998). Most recently, a graph-guided Non-linear Bayesian SVM which enables incorpora-

tion of the prior network information among predictors, known as BNSVM, and showed that BNSVM outperforms a number of penalized linear and non-linear kernel SVM methods in numerical studies (details can been in Chapter 4).

However, the aforementioned methods assume the prior network is known or given as a *priori*. Although there are several available databases storing biological knowledge on pathways/networks (Ashburner et al., 2000; Nishimura, 2001; Ogata et al., 1999), the available reference networks may be incomplete or inappropriate for the experimental condition or set of subjects under study. Unlike the aforementioned approaches incorporating prior knowledge to perform feature selection and prediction, we are interested in estimating the graph from data, and incorporating the uncertainty of the graph estimation into the model to improve prediction and features selection. Importantly, our proposed method can estimate the graph and perform feature selection simultaneously, which is different from the work using two stage procedure, first estimating the graph and then using the graph to select relevant features (Kundu et al., 2018).

To estimate the graph from the data at hand, several models have been developed. For example, Dobra (2009) proposed estimating a network among relevant predictors by first performing a stochastic search in the regression setting to identify sets of predictors with high posterior probability, then applying a Bayesian model averaging approach to estimate a dependency network given these results. Liu et al. (2014) propose a Bayesian regularization graph Laplacian approach which uses the graph Laplacian matrix to specify a prior precision matrix of regression coefficients. Different from the aforementioned approaches, we extend the model BNSVM (details can been in Chapter 4) and propose a new joint Bayesian non-linear SVM model (JBNSVM) to infer a sparse network among the predictors and perform variable selection by incorporating the estimated network simultaneously. The predictors are assigned by Gaussian priors through the Gaussian graphical model, in which the precision matrix is related to the graphical structure. Similar to BNSVM, the non-linear classifier is assumed to be drawn from a zero mean Gaussian process (Henao, Yuan, and Carin, 2014) with a special covariance matrix. This covariance matrix is constructed by the usual non-linear kernel function embedded with latent binary variables representing the selection status of features. Furthermore, Ising priors are assigned to the latent binary variables to incorporate the graphical structure of the features. The performance of our method is investigated by extensive simulation studies in comparison with L_1SVM, standard linear SVM (L_2SVM), the non-linear SVM

(Kernel-SVM) and BNSVM methods in terms of prediction and feature selection. The proposed approach not only offers good performance in terms of selection and prediction but also provides insight into the relationships among important variables and allows the identification of related predictors that jointly impact the response. In addition, because we take a Bayesian approach to the problem of joint variable and graphical model selection, we are able to fully account for uncertainty over both the selection of variables and of the graph.

The remainder of the paper is organized as follows. In Sections 5.2, we describe the proposed joint Bayesian model and the MCMC algorithm for posterior inference and prediction. In Section 5.3, we conduct simulation studies to evaluate our approach in comparison with several existing approaches. In Section 5.4, we apply our approach to a TCGA glioblastoma dataset. We conclude with a brief discussion and future works in Section 5.5.

5.2. Methods

5.2.1. Proposed joint model

Suppose there are n samples in the training set of data where $y_i \in \{-1, 1\}$ are the binary outcome variables and $\mathbf{x}_i$ are the p dimensional feature vector. In our modeling approach, we consider both the response $\mathbf{y}_{n\times 1}$ and the predictors $X_{n\times p}$ to be random variables, so our likelihood is the joint distribution $\pi(\mathbf{y}, X)$, which can be factored into the conditional distribution of $\mathbf{y}$ given X and the marginal distribution of X as below:

$$\pi(\mathbf{y}, X) = \pi(\mathbf{y}|X) \times \pi(X) \tag{5.1}$$

We re-express $\pi(\mathbf{y}|X)$ as a location-scale mixture of normals Henao, Yuan, and Carin, 2014; Polson and Scott, 2011 by introducing a latent variable ρ_i, then

$$\pi(\mathbf{y}|X) = \prod_{i=1}^{n} \int_0^\infty \frac{\sqrt{\kappa}}{\sqrt{2\pi\rho_i}} \exp\left(-\frac{\kappa(\rho_i + 1 - y_i f(\mathbf{x}_i))^2}{2\rho_i}\right) d\rho_i. \tag{5.2}$$

And $\pi(X)$ is assumed to be a centered multivariate normal distribution as

$$\mathbf{x}_i \sim \mathcal{N}(0, \Omega^{-1}) \tag{5.3}$$

where $\Omega = \Sigma^{-1}$ is the precision matrix.

5.2.2. Prior for the non-linear classifier $f(\mathbf{x})$ in Equation 5.2

A Gaussian process is a collection of random variables, and the joint distribution of any of these variables is Gaussian. Such a process $f(\mathbf{x})$ is completely determined by its mean function $\mu(\mathbf{x}) = E(f(\mathbf{x}))$ and the covariance kernel function $k(\mathbf{x}, \mathbf{x}') = E[(f(\mathbf{x}) - \mu(\mathbf{x}))(f(\mathbf{x}') - \mu(\mathbf{x}'))]$. We can assume the mean functions $\mu(\mathbf{x})$ to be zero, because it is easy to subtract this off if we know a priori any deviation from zero. The zero mean assumption implies that there is no preference of positive or negative values for the mean given no data.

In our case, the random variables are $f_1, ..., f_i, ..., f_n$ corresponds to $f(\mathbf{x}_1), ...,$ $f(\mathbf{x}_i), ..., f(\mathbf{x}_n)$, which are evaluated at the n data points. The covariance matrix K between $f_1, ..., f_n$ is defined as

$$\begin{pmatrix} 1 & \phi^{\sum_{l=1}^{p} \gamma_l (x_{1l} - x_{2l})^2} & \dots & \phi^{\sum_{l=1}^{p} \gamma_l (x_{1l} - x_{nl})^2} \\ \phi^{\sum_{l=1}^{p} \gamma_l (x_{2l} - x_{1l})^2} & 1 & & \vdots \\ \vdots & & \ddots & \\ \phi^{\sum_{l=1}^{p} \gamma_l (x_{nl} - x_{1l})^2} & \dots & & 1 \end{pmatrix},$$

where $\phi \in (0, 1)$ and γ_l is the latent binary variable indicating the inclusion or exclusion of the lth feature of the model. This covariance structure encourages f_i and f_j to be highly correlated when $\mathbf{x}_i$ and $\mathbf{x}_j$ are close, or uncorrelated if $\mathbf{x}_i$ and $\mathbf{x}_j$ are far enough. The parameter ϕ determines the sensitivity of correlation to the distance. We assign the uniform distribution as the prior for ϕ.

5.2.3. Markov Random Field (MRF) priors for $\boldsymbol{\gamma}$

In the covariance matrix, γ_l plays an important role to perform selection of each feature l. Usually, the iid Bernoulli prior is assigned for $\boldsymbol{\gamma}$, which is equivalent to assuming the predictors are independently chosen, but ignoring the underlying structure information among predictors. Instead of an independent prior, we propose a prior to tie the selection of predictors to the presence of edges

relating them in the graph G. To accomplish this, we rely on an MRF prior favoring the inclusion of variables that are linked to other predictors in the network. MRF priors have been utilized in the variable selection context (Li and Zhang, 2010; Stingo and Vannucci, 2010). However, unlike these authors, who assume that the structure of the network among predictors is known, we incorporate inference of the network structure from X. Recalling that $G_{jk} \in \{0, 1\}$ indicates the presence of edge (j, k) in the graph G, we have the MRF prior distribution over $\boldsymbol{\gamma}$ taken as

$$p(\boldsymbol{\gamma}|G, \mu, \eta) = C_1(G, \mu, \eta)^{-1} e^{-\mu \sum_j \gamma_j + \eta \sum_{j \neq k} G_{jk} \gamma_j \gamma_k}, \tag{5.4}$$

where the tuning parameters μ controls the sparsity of $\boldsymbol{\gamma}$ and η controls the smoothness of $\boldsymbol{\gamma}$ over E. $C_1(G, \mu, \eta)$ is the constant that normalizes the prior density of $\boldsymbol{\gamma}$ given G. The prior linking variable and edge selection reflects a preference for the inclusion of connected predictors in the model by incorporating an MRF on the variable selection indicators that utilize the estimated network among predictors. The proposed model is therefore appropriate for datasets where the predictors that affect the outcome of interest are in fact connected through a network.

5.2.4. Graph selection prior on Ω and G

The goal of the graph selection is to allow inference on the network G among predictors X. The prior distribution on the precision matrix Ω discussed in (5.3) combines an exponential prior on the diagonal entries with a mixture of normals on the off-diagonal entries of to allow the entries for selected edges to have a larger variance than that of non-selected edges:

$$\pi(\Omega|G, v_0, v_1, \lambda) = C_2(G, v_0, v_1, \lambda)^{-1} \prod_{j<k} \mathcal{N}(\omega_{jk}|0, v^2_{G_{jk}}) \prod_j Exp(\omega_{jj}|\frac{\lambda}{2}) I_{\{\Omega \in M^+\}} \tag{5.5}$$

where $v_0 > 0$ is small, $v_1 > 0$ is large, $\lambda > 0$ and $I_{\{\Omega \in M^+\}}$ is an indicator function that restricts the prior to the space of symmetric-positive definite matrices. Note that $C_2(G, v_0, v_1, \lambda)$ is the normalizing constant. By choosing v_0 to be small, we ensure that ω_{jk} will be close to 0 for non-selected edges. For selected edges, a large value of v_1 allows ω_{jk} to have more substantial magnitude. In the second level of the hierarchy, we place a prior on the edge inclusion indicators G_{jk}:

$$\pi(G|p_0, v_0, v_1, \lambda, \mu, \eta) \propto C_2(G, v_0, v_1, \lambda) C_1(G, \mu, \eta) \prod_{j<k} \{p_0^{g_{jk}} (1 - p_0)^{1 - G_{jk}}\} \tag{5.6}$$

where p_0 reflects the prior probability of edge inclusion and can be fixed.

5.2.5. Posterior Inference

Together with the observed data, prior distributions are converted to posterior distributions through the use of Bayes theorem. The joint posterior distribution for the set of all parameters $\boldsymbol{\theta} = (\nu, \boldsymbol{\rho}, \boldsymbol{\gamma}, \phi, \mathbf{f}, \Omega, G)$ is written as

$$\begin{aligned}
\pi(\boldsymbol{\theta}|\mathbf{y}, X) &\propto \pi(\mathbf{y}|\mathbf{f}, \nu, \boldsymbol{\rho})\pi(\boldsymbol{\rho})\pi(\mathbf{f}|X, \boldsymbol{\gamma}, \phi)\pi(X|\Omega)\pi(\boldsymbol{\gamma}|G)p(\Omega|G)\pi(G)\pi(\nu) \\
&\propto \prod_{i=1}^{n} \frac{\sqrt{\nu}}{\sqrt{2\pi\rho_i}} e^{-\frac{\nu(\rho_i+1-y_i f(\mathbf{x}_i))^2}{2\rho_i}} \times |K|^{-\frac{1}{2}} e^{-\frac{1}{2}\mathbf{f}^T K^{-1}\mathbf{f}} \\
&\times |\Omega|^{\frac{n}{2}} e^{-\frac{1}{2}X'\Omega X} \times e^{-\mu\sum_j \gamma_j + \eta\sum_{j\neq k} G_{jk}\gamma_j\gamma_k} \\
&\times \prod_{j<k} \mathcal{N}(\omega_{jk}|0, v^2_{G_{jk}}) \prod_j Exp(\omega_{jj}|\frac{\lambda}{2}) I_{\{\Omega\in M^+\}} \\
&\times \prod_{j<k} p_0^{G_{jk}}(1-p_0)^{1-G_{jk}} \times \nu^{a_\nu - 1} e^{-b_\nu \nu}.
\end{aligned}$$

MCMC is implemented by Metropolis Hastings within Gibbs algorithm. Most of the conditional distributions can be easily sampled. We have

$$\nu|\mathbf{f}, X, \mathbf{y}, \boldsymbol{\rho} \sim \mathcal{G}\left(a_\nu + \frac{3n}{2}, b_\nu + \sum_{i=1}^{n} \frac{(\rho_i + 1 - y_i f(\mathbf{x}_i))^2}{2\rho_i}\right), \tag{5.7}$$

and we have

$$\rho_i|\mathbf{f}, \mathbf{y}, \nu \sim \mathcal{GIN}(1/2, \nu, \nu(1-y_i f_i)^2),$$

where $\mathcal{GIN}(p, a, b)$ stands for the Generalized Inverse Gaussian distribution. Alternatively, the conditional distribution of ρ_i^{-1} is an inverse Gaussian distribution, denoted by $\mathcal{IN}$.

$$\rho_i^{-1}|\mathbf{f}, \mathbf{y}, \nu \sim \mathcal{IN}(|1-y_i f_i|^{-1}, \nu), \tag{5.8}$$

where the density function of $\mathcal{IN}(\mu, \lambda)$ is defined as below:

$$f(x; \mu, \lambda) = \frac{\sqrt{\lambda}}{\sqrt{2\pi x^3}} e^{\frac{-\lambda(x-\mu)^2}{2\mu^2 x}}.$$

The conditional distribution of γ_j follows the Bernoulli distribution which is given by

$$\gamma_j | \boldsymbol{\gamma}_{-j}, \phi, G, \mathbf{f}, X \sim Ber\left(\frac{\Pi_j(\gamma_j = 1, \boldsymbol{\gamma}_{-j})}{\Pi_j(\gamma_j = 1, \boldsymbol{\gamma}_{-j}) + \Pi_j(\gamma_j = 0, \boldsymbol{\gamma}_{-j})}\right), \tag{5.9}$$

where $\boldsymbol{\gamma}_{-j} = (\gamma_1, \ldots, \gamma_{j-1}, \gamma_{j+1}, \ldots, \gamma_p)$ and $\Pi_j(\boldsymbol{\gamma}) = |K(\boldsymbol{\gamma})|^{-\frac{1}{2}}$
$e^{-\frac{1}{2}\mathbf{f}^T K(\boldsymbol{\gamma})^{-1}\mathbf{f}} e^{-\mu\gamma_j + \eta\gamma_j \sum_k g_{jk}\gamma_k}$.

The conditional distribution of ϕ is given by

$$\pi(\phi | \mathbf{f}, \boldsymbol{\gamma}, X) \propto |K|^{-\frac{1}{2}} e^{-\frac{1}{2}\mathbf{f}^T K^{-1}\mathbf{f}} \tag{5.10}$$

We use the Metropolis-Hasting algorithm to draw samples from $\pi(\phi|\mathbf{f}, \boldsymbol{\gamma}, X)$. Since $\phi \in (0, 1)$, the logit normal distribution $q(x; \mu, \tau^2) = \frac{1}{\tau\sqrt{2\pi}} \frac{1}{x(1-x)} e^{-\frac{(\text{logit}(x)-\mu)^2}{2\tau^2}}$ is chosen as the proposed distribution. Assume the last state is ϕ^{t-1}, we draw a sample ϕ^* from $q(x; \phi^{t-1}, \tau^2)$, where τ^2 is set to a value that keeps the acceptance rate around 40%. We accept or reject the current proposed ϕ^* with probability α_t.

$$\alpha_t = \min\left(1, \frac{\pi(\phi^*|\mathbf{f}, \boldsymbol{\gamma}, X)\phi^*(1-\phi^*)}{\pi(\phi^{t-1}|\mathbf{f}, \boldsymbol{\gamma}, X)\phi^{t-1}(1-\phi^{t-1})}\right)$$

Let $D_\rho = \text{diag}(\rho_1, \ldots, \rho_n)$ and $\mathbf{z}$ be a vector with entries $z_i = (1+\rho_i^{-1})y_i$. The conditional distribution for $\mathbf{f}$ is a multivariate Gaussian:

$$\mathbf{f}|\mathbf{y}, \nu, \boldsymbol{\rho}, K \sim \mathcal{N}(\nu(\nu D_\rho^{-1} + K^{-1})^{-1}\mathbf{z}, (\nu D_\rho^{-1} + K^{-1})^{-1}). \tag{5.11}$$

The conditional distribution of G is given by

$$\pi(G|\boldsymbol{\gamma}, v_0, v_1, \lambda, p_0) \propto \prod_{j<k} \mathcal{N}(\omega_{jk}|0, v_{G_{jk}}^2) p_0^{G_{jk}} (1-p_0)^{1-G_{jk}} e^{\eta G_{jk}\gamma_j\gamma_k}, \tag{5.12}$$

and each entry G_{jk} of G follows the Bernoulli distribution with

$$\pi(G_{jk}=1|\boldsymbol{\gamma},v_0,v_1,\lambda,p_0)=\frac{\mathcal{N}(\omega_{jk}|0,v_1^2)p_0e^{\eta\gamma_j\gamma_k}}{\mathcal{N}(\omega_{jk}|0,v_1^2)p_0e^{\eta\gamma_j\gamma_k}+\mathcal{N}(\omega_{jk}|0,v_0^2)(1-p_0)}. \tag{5.13}$$

The full conditionals for Ω is given as

$$p(\Omega|G,v_0,v_1,\lambda,\mathbf{y},X)\propto|\Omega|^{\frac{n}{2}}\exp\{-\text{tr}(\frac{1}{2}X'X\Omega)\}\prod_{j<k}\mathcal{N}(\omega_{jk}|0,v^2_{G_{jk}})\prod_j Exp(\omega_{jj}|\frac{\lambda}{2})I_{\{\Omega\in M^+\}}. \tag{5.14}$$

We can use the block Gibbs sampler (Wang, 2015) to sample Ω. The details are provided in Appendix. The Metropolis-Hastings within Gibbs sampling approach is provided in Algorithm 4.

for $t=1$ **to** T **do**
  Sample $\nu\sim\mathcal{G}\left(a_\nu+\frac{3n}{2},b_\nu+\sum_{i=1}^n\frac{(\rho_i+1-y_if_i)^2}{2\rho_i}\right)$;
  for $i=1$ **to** n **do**
    Sample $\rho_i^{-1}\sim\mathcal{IN}(|1-y_if(x_i)|^{-1},\nu)$;
  end
  for $i=1$ **to** p **do**
    Sample $\gamma_j\sim Ber\left(\frac{\Pi_j(\gamma_j=1,\boldsymbol{\gamma}_{-j})}{\Pi_j(\gamma_j=1,\boldsymbol{\gamma}_{-j})+\Pi_j(\gamma_j=0,\boldsymbol{\gamma}_{-j})}\right)$;
  end
  for $i=1$ **to** p **do**
    Sample $g_{jk}\sim Ber\left(\frac{\mathcal{N}(\omega_{jk}|0,v_1^2)p_0e^{\eta\gamma_j\gamma_k}}{\mathcal{N}(\omega_{jk}|0,v_1^2)p_0e^{\eta\gamma_j\gamma_k}+\mathcal{N}(\omega_{jk}|0,v_0^2)(1-p_0)}\right)$;
    Sample $(\boldsymbol{\omega}_{-j,j},\omega_{jj})\sim\mathcal{N}(-C_j\mathbf{s}_{-j,j},C_j)\mathcal{G}(\frac{n}{2}+1,\frac{s_{jj}+\lambda}{2})$;
  end
  Generate a proposal $\phi^*\sim q(\phi;\phi^{t-1},\tau^2)$;
  Generate $u\sim U(0,1)$;
  if $u<\min\left(1,\frac{\pi(\phi^*|\mathbf{f},\boldsymbol{\gamma},X)\phi^*(1-\phi^*)}{\pi(\phi^{t-1}|\mathbf{f},\boldsymbol{\gamma},X)\phi^{t-1}(1-\phi^{t-1})}\right)$ **then**
    $\phi^t\leftarrow\phi^*$;
  else
    $\phi^t\leftarrow\phi^{t-1}$;
  end
  Sample $\mathbf{f}\sim\mathcal{N}(\nu(\nu D_\rho^{-1}+K^{-1})^{-1}\mathbf{z},(\nu D_\rho^{-1}+K^{-1})^{-1})$;
end

Algorithm 4: MCMC algorithm for JBNSVM.

5.2.6. Prediction

To predict the label for a new testing case, the inference is divided into two steps: first computing the distribution of the latent non-linear function f^* corresponding to the test case; second computing the predictive label probabilities of the test case.

Let $\boldsymbol{\theta} = (\boldsymbol{\gamma}, \nu, \boldsymbol{\rho}, \phi)^T$ denote the vector of model parameters. The predictive distribution of f^* for a new testing data vector $\mathbf{x}^*_{p\times 1}$ given the training dataset X, $\mathbf{y}$ and $\boldsymbol{\theta}$ can be written as

$$f^*|\mathbf{x}^*, X, \mathbf{y}, \boldsymbol{\theta} \sim \mathcal{N}(\mu, \sigma^2), \tag{5.15}$$

where

$$\begin{aligned}
\mathbf{k}^* &= (k(\mathbf{x}^*, \mathbf{x}_1), \ldots, k(\mathbf{x}^*, \mathbf{x}_n))^T, \\
k^* &= k(\mathbf{x}^*, \mathbf{x}^*), \\
\Sigma &= (K + \nu^{-1} D_\rho)^{-1}, \\
\mu &= \mathbf{k}^{*T} \Sigma D_\rho \mathbf{z}, \\
\sigma^2 &= k^* - \mathbf{k}^{*T} \Sigma \mathbf{k}^*.
\end{aligned}$$

Then, we can use the probit link to compute the conditional predictive class probabilities.

$$\begin{aligned}
\pi(y^* = 1|\mathbf{x}^*, X, \mathbf{y}, \boldsymbol{\theta}) &= \int \Phi(f^*) \pi(f^*|\mathbf{x}^*, X, \mathbf{y}, \boldsymbol{\theta}) df^* \\
&= \Phi\left(\frac{\mathbf{k}^{*T} \Sigma D_\rho \mathbf{z}}{\sqrt{1 + k^* - \mathbf{k}^{*T} \Sigma \mathbf{k}^*}} \right).
\end{aligned} \tag{5.16}$$

To estimate the marginal predictive probability $\pi(y^* = 1|\mathbf{x}^*, X, \mathbf{y})$, the MCMC samples of $\boldsymbol{\theta}$ are used.

$$\widehat{p} = \frac{1}{M} \sum_{m=1}^{M} p(y^* = 1|\mathbf{x}^*, X, \mathbf{y}, \boldsymbol{\theta}_m), \tag{5.17}$$

where $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M$ are the MCMC samples of $\boldsymbol{\theta}$.

The prediction error can be measured by the cross-entropy between the predictive probabilities and the actual class.

$$\mathrm{CE} = -\sum_{i=1}^{N} \mathbb{I}(y_i^* = 1) \log \widehat{p}_i - \sum_{i=1}^{N} \mathbb{I}(y_i^* = -1) \log(1 - \widehat{p}_i),$$

where N is the sample size of testing data. The decision can be made by $\widehat{y}_i = \mathrm{sign}(\widehat{p}_i - 0.5)$ and

the associated prediction error can be reported as follows.

$$\mathrm{PE} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(\widehat{y}_i \neq y_i^*). \tag{5.18}$$

5.3. Simulation studies

5.3.1. Design of Experiment

In this section, we study the performance of our JBNSVM method through the simulated additive models and compare with the following existing methods: L_1SVM, L_2SVM, Kernel-SVM and BNSVM. Of note, Chapter 4 showed that BNSVM outperforms several penalized linear SVM methods such as L_1SVM and L_2SVM, and non-linear kernel SVM. The reason we compare the proposed JBNSVM with BNSVM is that when the prior knowledge is partially correct or incorrectly specified, the prediction accuracy for BNSVM may be deteriorated while JBNSVM doesn't subject to the prior knowledge and should produce a stable performance.

Following the settings in Chapter 4, we simulate the additive model: $\mathbf{x} \sim \mathcal{N}(0, \Omega_{p\times p}^{-1})$, $f(\mathbf{x}) = x_1^2 + x_2^2 + .. + x_q^2$, where $x_1, ..., x_q$ are the first q dimensions of $\mathbf{x}$ and f is only relevant with the first q features. The binary response y is determined by a cut-off value of $f(\mathbf{x})$, which divides the two classes almost equally. We simulate the examples for both the graph-related covariance structure and independent covariance structure for the input features. The precision matrix Ω and covariance matrix Σ are generated from a simulated graph G.

In each experimental setting, we generate 100 datasets, each with a training sample for fitting, a validation sample for tuning and an independent test sample. The sample size for training, validation and testing data is 200, and the feature dimension p is set at 20, 100. The performance metrics: the prediction error (PE), prediction sensitivity (PSEN), prediction specificity (PSPEC), Matthews Correlation Coefficients (MCC), feature selection true positive (FSTP) and feature selection false positive (FSFP). For comparision, prediction errors are reported by (5.18) as not all methods provide probabilistic prediction. The prediction sensitivity is calculated as the proportion of positives $(y = 1)$ that are correctly identified and the prediction specificity is calculated as the proportion of

negatives $(y = -1)$ that are correctly identified. MCC is defined as

$$\frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{5.19}$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives and FN is the number of false negatives. FSTP is the percentage of important features selected by the model among the total important features and FSFP is the percentage of unimportant features selected by the model among the total unimportant features.

5.3.2. Parameter Tuning

For L_1SVM, L_2SVM and Kernel-SVM, we use the penalizedSVM R-package (Becker et al., 2009) to tune the parameters in the validation datasets. For BNSVM, there are four settings: BNSVM($\eta = 0$), representing no graph incorporated, only μ needs to be tuned. When $\eta \neq 0$, three settings includes BNSVM($G^* = nG$), representing the noisy graph guided model; BNSVM($G^* = pG$), representing a partial graph guided model; BNSVM($G^* = G$), representing a true graph guided model. Since there working graphs are incorporated into the model, two parameters (μ, η) need to be tuned to achieve the best performance in terms of PE.

5.3.3. Simulation Results

Table 5.1 and Table 5.2 summarize the results for two linear SVM (L_1SVM and L_2SVM) and three non-linear SVM (Kernel-SVM, BNSVM and JBNSVM) under different data covariance structure. We report the mean and standard error over 100 datasets for each metric we choose to compare in the result table. Obviously, all the non-linear methods work much better than the linear methods due to the simulated non-linear data structure. In most settings, the BNSVM approach with incorporating the true graph G has the best performance regarding to PE, MCC and FSFP. However, if the working graph is not correctly specified, our proposed method JBNSVM outperforms other methods such as BNSVM($\eta = 0$), BNSVM($G^* = nG$) and BNSVM($G^* = pG$). These facts show that in most conditions, our prior knowledge may be not fully completed or certain, instead of incorporating such uncertain knowledge, we should infer the network structure from data and incorporate the inference to guide feature selection and prediction. In addition, the proposed model JBNSVM is able to estimate the structure among between predictors with high accuracy which provides insight

into the relationships among important predictors. The true graphs and the estimated graphs for $p = 20, 100$ are illustrated in Figure 5.1.

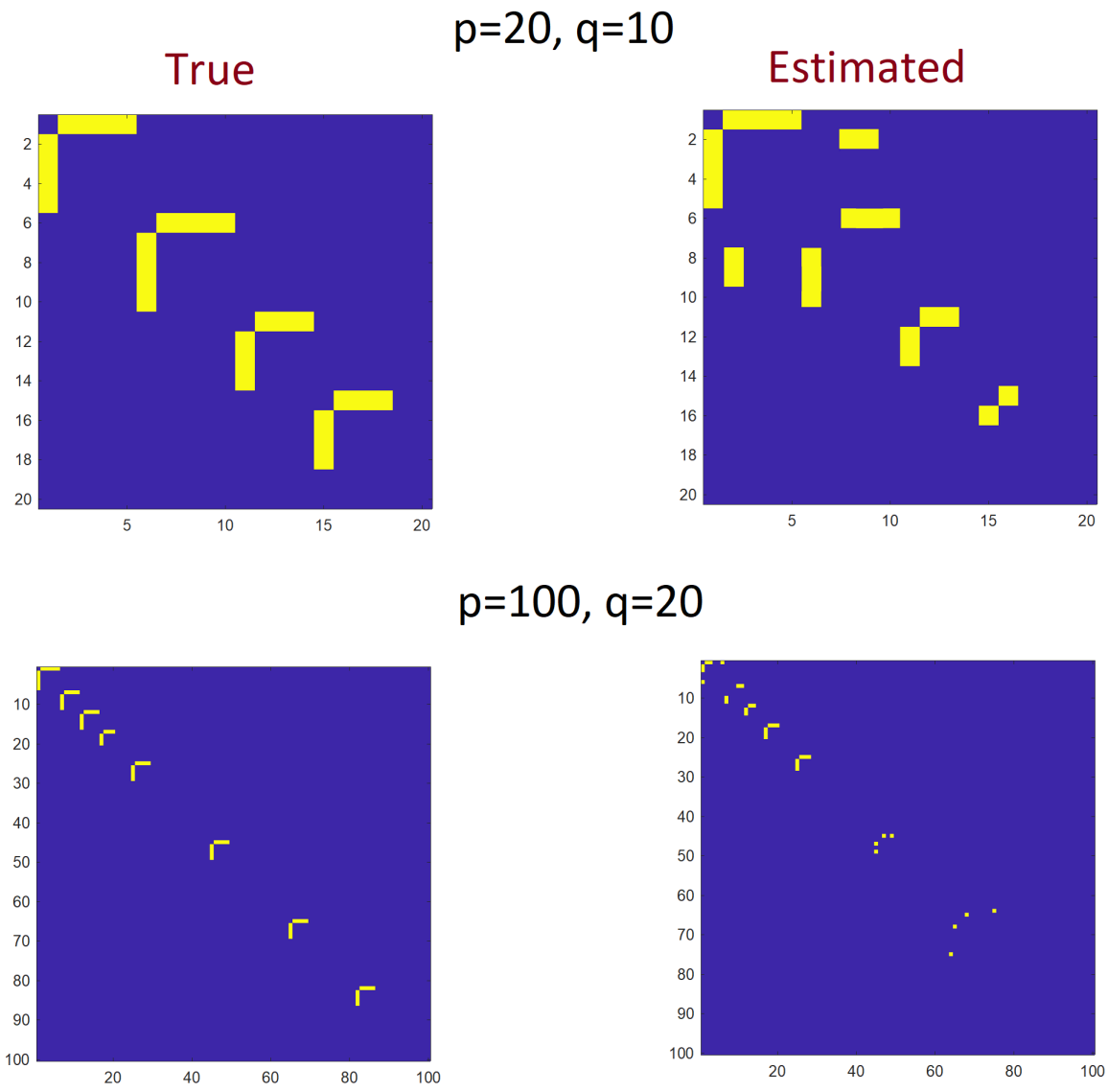

Figure 5.1: The true graphs and estimated graphs.

5.4. Data Analysis

In the real data application, we use the TCGA glioblastoma cancer gene expression dataset with 286 subjects and 12,999 genes in the network. The response variable we consider here is the one year survival status. In this section, we apply our methods and other existing methods to classify

Table 5.1: Simulation results for correlated structure among features. – indicates no feature selection for the corresponding method.

Method	PE (%)	PSen (%)	PSpec(%)	MCC (%)	FSTP (%)	FSFP (%)
			$p = 20$, $q = 10$			
L_1SVM	48.30 (1.15)	43.41 (3.61)	59.80 (4.67)	4.15 (2.80)	53.00 (11.35)	61.00 (11.81)
L_2SVM	49.20 (0.86)	46.55 (2.30)	54.93 (2.75)	1.56 (1.72)	–	–
Kernel-SVM	17.35 (0.62)	79.90 (0.98)	85.31 (1.41)	65.49 (1.27)	–	–
BNSVM($\eta = 0$)	16.00 (0.73)	67.04 (2.49)	91.73 (1.17)	60.85 (2.92)	95.00 (2.52)	11.00 (1.68)
BNSVM($G^* = nG$)	17.80 (0.83)	70.15 (1.10)	93.11 (1.33)	66.01 (1.74)	95.00 (3.20)	10.00 (1.98)
BNSVM($G^* = pG$)	16.35 (0.89)	72.73 (1.43)	94.29 (0.83)	68.83 (1.70)	98.00 (1.25)	11.00 (2.19)
BNSVM($G^* = G$)	15.50 (0.52)	73.54 (1.06)	94.18 (1.01)	69.42 (1.50)	100.00 (0.00)	7.00 (2.00)
JBNSVM	15.85 (0.60)	76.12 (1.13)	91.98 (1.07)	69.19 (1.21)	99.00 (0.94)	25.00 (3.45)
			$p = 100$, $q = 20$			
L_1-SVM	46.60 (0.81)	53.75 (1.60)	53.01 (2.14)	6.79 (1.61)	59.50 (9.01)	62.00 (7.98)
L_2-SVM	48.75 (0.83)	46.43 (0.94)	51.00 (1.54)	1.15 (1.82)	–	–
Kernel-SVM	34.75 (1.01)	63.42 (2.62)	66.81 (1.99)	30.55 (2.07)	–	–
BNSVM($\eta = 0$)	30.95 (1.36)	55.31 (3.47)	78.67 (6.96)	42.13 (3.16)	62.00 (11.84)	8.25 (6.96)
BNSVM($G^* = nG$)	35.10 (1.90)	43.00 (4.37)	75.16 (2.91)	34.94 (3.75)	56.00 (8.43)	12.25 (3.86)
BNSVM($G^* = pG$)	32.50 (1.12)	52.83 (2.30)	85.15 (3.40)	37.19 (3.09)	64.50 (8.99)	5.75 (1.33)
BNSVM($G^* = G$)	27.50 (0.94)	74.05 (3.78)	71.01 (3.79)	46.32 (1.66)	66.50 (6.78)	5.13 (1.32)
JBNSVM	27.60 (1.54)	70.78 (1.67)	73.96 (1.50)	44.77 (3.09)	54.50 (4.10)	2.88 (0.58)

Table 5.2: Simulation results for independent structure among features. – indicates no feature selection for the corresponding method.

Method	PE (%)	PSen (%)	PSpec(%)	MCC (%)	FSTP (%)	FSFP (%)
			$p = 20$, $q = 10$			
L_1-SVM	47.89 (0.71)	41.12 (3.24)	63.98 (3.64)	4.22 (1.00)	42.22 (9.97)	41.11 (8.65)
L_2-SVM	48.55 (1.04)	41.06 (1.94)	56.21 (2.87)	3.12 (2.12)	–	–
KBSVM	45.55 (0.73)	33.79 (4.20)	74.83 (3.43)	9.58 (1.35)	28.00 (7.23)	19.00 (9.63)
Kernel-SVM	20.20 (0.68)	79.30 (1.02)	80.29 (1.48)	50.74 (1.35)	–	–
BNSVM, $G^* = nG$	21.60 (2.49)	84.30 (1.32)	72.58 (5.05)	57.50 (4.71)	90.00 (8.39)	25.00 (6.13)
BNSVM, $G = I$	20.95 (1.06)	86.00 (2.41)	72.11 (3.00)	59.43 (2.02)	97.00 (2.00)	20 (5.76)
JBNSVM	20.00 (0.52)	78.49 (1.43)	81.51 (1.29)	60.21 (1.02)	99.00 (0.94)	49.00 (10.12)
			$p = 100$, $q = 20$			
L_1SVM	47.70 (0.92)	51.38 (1.70)	54.96 (1.06)	4.55 (1.84)	69.00 (7.38)	70.62 (6.56)
L_2SVM	50.00 (1.22)	47.18 (1.50)	52.39 (1.61)	3.18 (2.22)	–	–
KBSVM,	46.20 (0.74)	48.42 (4.19)	57.02 (3.63)	5.47 (2.25)	43.00 (11.57)	44.12 (11.67)
Kernel-SVM	38.95 (1.33)	65.91 (4.68)	55.77 (3.01)	22.86 (3.14)	–	–
BNSVM, $G^* = nG$	35.88 (0.88)	43.83 (4.13)	71.99 (4.44)	29.28 (3.79)	54.50 (12.74)	22.63 (12.18)
BNSVM, $G = I$	31.38 (0.74)	61.91 (3.06)	60.58 (2.21)	35.15 (1.41)	68.90 (10.01)	11.11 (4.23)
JBNSVM	31.65 (1.22)	60.34 (1.11)	66.37 (1.89)	36.79 (2.41)	60.50 (3.83)	8.25 (1.99)

the survival status of the glioblastoma patients. Because our proposed model JBNSVM can only perform on small number of genes so we screen the top important genes and the steps are provides as follows: First, we fit an univariate logistic regression model for each gene x:

$$log\frac{p(x)}{1-p(x)} = \beta_0 + \beta_1 x \tag{5.20}$$

We calculate the p value from the logistic regression and create a list of ascend ordered p-value as well as the corresponding genes. Second, we use the gene-ranking methods to select important genes, for example, we select the top 100 genes in the list and generate the new feature inputs from

Table 5.3: Results of the analysis of TCGA data. $n = 286$, $p = 100$.

	PE (%)	PSen (%)	PSpec (%)	MCC (%)
L_1SVM	33.33	47.50	80.36	29.63
L_2SVM	38.69	39.57	67.90	20.43
Kernel-SVM	29.17	55.00	82.14	38.84
BNSVM($\eta = 0$)	32.29	57.50	75.00	32.92
BNSVM(G^*)	32.29	55.00	76.79	32.56
JBNSVM	28.13	62.50	78.57	41.60

the gene expression levels.

Second, we create a graph G for all the $12,999$ genes from the Kyoto Encyclopedia of Genes and Genomes (KEGG) database, and we use an algorithm to retrieve the connections within the top 100 genes, and then map them to the working graph G^*. Of note, the working G^* is very sparse with only two edges detected. It is almost equivalent to no graph guided when we apply the BNSVM method. This is also the main reason for us to develop JBNSVM to apply the conditions when no prior knowledge is available.

Third, we split the total samples of $n = 286$ into training and testing datasets. The sample size for training is 190. During the training procedure, we choose five pairs of (v_0, v_1) as the tuning parameters resulting in different levels of sparsity of graph. The optimal tuning parameters for each methods are chosen by the minimum 5-fold cross-validation error.

The testing error are summarized in Table 5.3. L_1SVM and L_2SVM have the larger prediction error, which may suggest that the linear separation is not suitable for the real gene expression data. BNSVM(G^*) has very similar performance to and BNSVM($\eta = 0$) due to the sparse graph extracted from KEGG with only two connections. Kernel-SVM achieves a lower cross-validation error, which may suggest that the non-linear classifier is more plausible. JBNSVM gives the best performance with the smallest PE and the highest MCC.

5.5. Discussions

In this work, we have developed a novel-modeling strategy to simultaneously select graph-related features and learn the structure among them. Our approach is fully Bayesian and therefore allow us to account for uncertainty over both feature and graph selections. Through simulations, we have demonstrated that this approach can achieve improved selection and prediction accuracy over

competing existing methods. We have illustrated this method with an application to the glioblastoma survival studies. We have found our method to provide satisfactory results in settings with $p < n$ cases. As more computationally efficient approaches for Bayesian estimation of Gaussian graphical models are developed, these can easily be merged into our framework, enabling the analysis of a much larger number of predictors.

CHAPTER 6

SUMMARY AND FUTURE WORK

6.1. Summary

In this dissertation, I have developed novel Bayesian SVM methods that enable simultaneous parameter estimation and feature selection guided by the graphical structure among predictors. In the first study, the proposed method uses the spike-and-slab prior for feature selection, combined with the Ising prior that encourages group-wise selection of the predictors adjacent to each other on the known graph. In the second study, the proposed method assigns Laplace priors to the regression coefficients and incorporates the underlying graph information via a hyper-prior for the shrinkage parameters in the Laplace priors. This enables smoothing of shrinkage parameters for connected variables in the graph and conditional independence between shrinkage parameters for disconnected variables. In the third study, we extend the linear SVM to the non-linear SVM by inserting a special covariance matrix, which is constructed by a non-linear kernel function embedded with latent binary variables representing the selection status of features. The graphical structure among features is incorporated using again the Ising prior. Unlike the aforementioned studies that assume the prior graph information is fully known, the fourth study develops a new joint Bayesian non-linear SVM model to infer a sparse graph among the predictors and perform variable selection by incorporating the estimated graph simultaneously. This joint model is useful when the available reference graphs are inaccurate or inappropriate for the experimental condition, which is often the case in practice. The performance of all the proposed methods is evaluated in comparison with existing SVM methods in terms of prediction and feature selection in extensive simulations. These methods are also illustrated in analysis of genomic data from cancer studies, demonstrating their advantage in generating biologically meaningful results and identifying potentially important features.

6.2. Future work

In addition to the aforementioned work that we have done, future work may include extending the current approach to more general types of outcomes such as categorical or continuous variables although the complexity of the optimization problem may increase.

APPENDIX

NOTATION

A.1. Taylor's expansion in Chapter 3 MH algorithm

MH algorithm uses as proposal function a multivariate Gaussian fitted locally to the distribution being sampled. This Gaussian fit is based on the following Taylors series expansion to approximate the log density of $\boldsymbol{\alpha}$

$$log\pi(\boldsymbol{\alpha}) \propto f(\boldsymbol{\alpha}) = \sum_{i}^{p+1} (\alpha_j - \frac{\tau_j}{2} e^{2\alpha_j}) - \frac{1}{2\nu}(\boldsymbol{\alpha} - \boldsymbol{\mu})' \Omega (\boldsymbol{\alpha} - \boldsymbol{\mu}) \tag{A.1}$$

$$\approx f(\boldsymbol{\alpha}_0) + g'(\boldsymbol{\alpha}_0)(\boldsymbol{\alpha} - \boldsymbol{\alpha}_0) + \frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}_0)' c H(\boldsymbol{\alpha}_0)(\boldsymbol{\alpha} - \boldsymbol{\alpha}_0)$$

where g and H stand for the gradient vector and Hessian matrix for f, respectively. The constant c is controlling the step of the hessian matrix, when c is 1, it is the ordinary Taylor's expansion. If we assume that f represents the logarithm of a concave probability distribution function (PDF), then the above approximation is equivalent to fitting the PDF (which we call F) with a multivariate Gaussian:

$$F(\boldsymbol{\alpha}) = \frac{1}{(2\pi)^{k/2} |\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\theta})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\alpha} - \boldsymbol{\theta})} \tag{A.2}$$

From comparing equations A.1 and A.2 it is obvious that the precision matrix is the same as the negative Hessian: $\boldsymbol{\Sigma}^{-1} = -cH(\boldsymbol{\alpha}_0)$. To find the mean of the fitted Gaussian, we observe that Gaussian mean maximizes the PDF (and its log). Therefore, finding the mean is equivalent to maximizing equation A.2. After some calculus, we arrive at:

$$\boldsymbol{\theta} = \boldsymbol{\alpha}_0 - H^{-1}(\boldsymbol{\alpha}_0) g(\boldsymbol{\alpha}_0)/c \tag{A.3}$$

Let J be the vector of 1 and $T = \begin{pmatrix} \tau_1 e^{2\alpha_1} & & \\ & \ddots & \\ & & \tau_{p+1} e^{2\alpha_{p+1}} \end{pmatrix}$, then

$$g(\boldsymbol{\alpha}_0) = (I - T_0)J - \frac{\Omega}{\nu}(\boldsymbol{\alpha}_0 - \mu), H(\boldsymbol{\alpha}_0) = -2T_0 - \frac{\Omega}{\nu}$$

where g and H stand for the gradient vector and Hessian matrix for f, respectively. T_0 represents the T matrix evaluated at α_0. If we assume that f is the logarithm of $\pi(\boldsymbol{\alpha})$, then $\pi(\boldsymbol{\alpha})$ is equivalent to a multivariate Gaussian

$$\begin{aligned} \pi(\boldsymbol{\alpha}) &\propto exp\{f(\alpha_0) + ((I - T_0)J - \frac{\Omega}{\nu}(\alpha_0 - \mu))(\alpha - \alpha_0) - \frac{1}{2}(\alpha - \alpha_0)'(2cT_0 + \frac{c\Omega}{\nu})(\alpha - \alpha_0)\} \\ &\propto exp\{-\alpha'(cT_0 + \frac{c\Omega}{2\nu})\alpha + \alpha_0'(2cT_0 + \frac{(c-1)\Omega}{\nu})\alpha + (J' - J'T_0 + J'\frac{\mu}{\nu})\alpha)\} \end{aligned}$$

A.2. The prediction formula for new testing point $\mathbf{x}^*$ in (4.11)

Consider a new testing data vector $\mathbf{x}^*$, the corresponding f^* is the value of non-linear function $f(x)$ evaluated at $\mathbf{x}^*$. The joint joint prior distribution of the training outputs $\mathbf{f}$, and the test outputs f^* according to the prior is

$$\begin{bmatrix} \mathbf{f} \\ f^* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K & \mathbf{k}^* \\ \mathbf{k}^{*T} & k^* \end{bmatrix}\right) \tag{A.4}$$

The the conditional distribution of f^* given $\mathbf{f}$ follows

$$f^*|\mathbf{x}^*, X, \boldsymbol{\theta}, \mathbf{f} \sim \mathcal{N}(\mathbf{k}^{*T}K^{-1}\mathbf{f}, k^* - \mathbf{k}^{*T}K^{-1}\mathbf{k}^*). \tag{A.5}$$

Since the conditional distribution for $\mathbf{f}$ is given by (4.10), we have

$$f^*|\mathbf{x}^*, X, \mathbf{y}, \boldsymbol{\theta} \sim \mathcal{N}(\mathbf{k}^{*T}\Sigma D_\rho \mathbf{z}, k^* - \mathbf{k}^{*T}\Sigma\mathbf{k}^*). \tag{A.6}$$

Hence, (4.11) holds.

Note that the first equality in (4.12) implies that $\pi(y^* = 1|\mathbf{x}^*, X, \mathbf{y}, \boldsymbol{\theta}) = P(u < f^*)$ where $u \sim$

$\mathcal{N}(0,1)$, f^* follows , and u and f^* are independent. Since

$$f^* - u \sim \mathcal{N}(\mathbf{k}^{*T}\Sigma D_\rho \mathbf{z}, 1 + k^* - \mathbf{k}^{*T}\Sigma \mathbf{k}^*),$$

the second equality in (4.12) follows.

A.3. Fast block Gibbs sampler for Ω in Chapter 5 (5.14)

We define $V = (v_{z_{ij}}^2)$ as a $p \times p$ symmetric matrix with zeros in the diagonal entries and $(v_{z_{ij}}^2)_{i<j}$ in the upper diagonal entries. $S = X'X$. Partition Ω, S and V as follows:

$$\Omega = \begin{pmatrix} \Omega_{11} & \boldsymbol{\omega}_{12} \\ \boldsymbol{\omega}_{12}' & \omega_{22} \end{pmatrix}, \quad S = \begin{pmatrix} S_{11} & \mathbf{s}_{12} \\ \mathbf{s}_{12}' & s_{22} \end{pmatrix}, \quad V = \begin{pmatrix} V_{11} & \mathbf{v}_{12} \\ \mathbf{v}_{12}' & v_{22} \end{pmatrix} \tag{A.7}$$

where $(\boldsymbol{\omega}_{12}, \omega_{22})$, $(\mathbf{s}_{12}, s_{22})$, $(\mathbf{v}_{12}, v_{22})$ are the last column of Ω, S and V respectively. The conditional distribution of $(\boldsymbol{\omega}_{12}, \omega_{22})$ in Ω is

$$p(\boldsymbol{\omega}_{12}, \omega_{22} | \Omega_{11}, X, G) \propto (\omega_{22} - \boldsymbol{\omega}_{12}'\Omega_{11}^{-1}\omega_{12})^{\frac{n}{2}} \exp[-\frac{1}{2}\{\boldsymbol{\omega}_{12}' D^{-1} \boldsymbol{\omega}_{12} + 2\mathbf{s}_{12}'\boldsymbol{\omega}_{12} + (s_{22}+\lambda)\omega_{22}\}]$$

where $D = \text{diag}(\mathbf{v}_{12})$. Consider a change of variables $(\boldsymbol{\omega}_{12}, \omega_{22}) \to (\boldsymbol{\mu} = \boldsymbol{\omega}_{12}, v = \omega_{22} - \boldsymbol{\omega}_{12}'\Omega_{11}^{-1}\omega_{12})$, whose Jacobian is a constant not involving $(\boldsymbol{\mu}, v)$. So

$$p(\boldsymbol{\mu}, v | \Omega_{11}, X, G) \propto v^{\frac{n}{2}} \exp(-\frac{s_{22}+\lambda}{2} v) \exp(-\frac{1}{2}[\boldsymbol{\mu}'\{D^{-1} + (s_{22}+\lambda)\Omega_{11}^{-1}\}\boldsymbol{\mu} + 2\mathbf{s}_{12}'\boldsymbol{\mu}]) \tag{A.8}$$

This implies that:

$$(\boldsymbol{\mu}, v) | (\Omega_{11}, X, G) \propto \mathcal{N}(-C\mathbf{s}_{12}, C)\mathcal{G}(\frac{n}{2}+1, \frac{s_{22}+\lambda}{2}) \tag{A.9}$$

where $C = \{(s_{22}+\lambda)\Omega_{11}^{-1} + D^{-1}\}^{-1}$. Using this method, we can permute any column to attain the full conditional used to generate $\Omega | X, G$.

BIBLIOGRAPHY

Alon, U, Barkai, N, Notterman, DA, Gish, K, Ybarra, S, Mack, D, and Levine, AJ (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences* 96.12, 6745–6750.

Ashburner, M, Ball, CA, Blake, JA, Botstein, D, Butler, H, Cherry, JM, Davis, AP, Dolinski, K, Dwight, SS, Eppig, JT, et al. (2000). Gene ontology: tool for the unification of biology. *Nature genetics* 25.1, 25.

Becker, N, Werft, W, Toedt, G, Lichter, P, and Benner, A (2009). penalizedSVM: a R-package for feature selection SVM classification. *Bioinformatics* 25.13, 1711–1712.

Becker, N, Toedt, G, Lichter, P, and Benner, A (2011). Elastic SCAD as a novel penalization method for SVM classification tasks in high-dimensional data. *BMC bioinformatics* 12.1, 138.

Bertsekas, DP (1999). *Nonlinear programming*. Athena scientific Belmont.

Bhosale, D and Ade, R (2014). Feature Selection based Classification using Naive Bayes, J48 and Support Vector Machine. *International Journal of Computer Applications (0975- 8887) Volume* 99.

Bradley, PS and Mangasarian, OL (1998). "Feature selection via concave minimization and support vector machines." In: *ICML*. Vol. 98, 82–90.

Chakraborty, S (2009). Bayesian binary kernel probit model for microarray based cancer classification and gene selection. *Computational Statistics & Data Analysis* 53.12, 4198–4209.

Chang, C, Kundu, S, and Long, Q (2018). Scalable Bayesian variable selection for structured high-dimensional data. *Biometrics, in press.*

Chapelle, O and Schölkopf, B (2002). "Incorporating invariances in non-linear support vector machines". In: *Advances in neural information processing systems*, 609–616.

Chen, J, Bardes, EE, Aronow, BJ, and Jegga, AG (2009). ToppGene Suite for gene list enrichment analysis and candidate gene prioritization. *Nucleic acids research* 37.suppl_2, W305–W311.

Chuang, HY, Lee, E, Liu, YT, Lee, D, and Ideker, T (2007). Network-based classification of breast cancer metastasis. *Molecular systems biology* 3.1, 140.

Chung, FR and Graham, FC (1997). *Spectral graph theory*. 92. American Mathematical Soc.

Clarke, HJ, Chambers, JE, Liniker, E, and Marciniak, SJ (2014). Endoplasmic reticulum stress in malignancy. *Cancer cell* 25.5, 563–573.

Cristianini, N (2001). Support vector and kernel machines. *Tutorial at ICML*.

Decoste, D and Schölkopf, B (2002). Training invariant support vector machines. *Machine learning* 46.1, 161–190.

Dempster, AP (1972). Covariance selection. *Biometrics*, 157–175.

Dobra, A (2009). Variable selection and dependency networks for genomewide data. *Biostatistics* 10.4, 621–639.

F., SC. Pathway Databases. *Annals of the New York Academy of Sciences* 1020.1 (), 77–91.

Fan, J (2001). Runze Li Variable selection via penalized likelihood. *J. Amer. Stat. Assoc*.

Fung, GM, Mangasarian, OL, and Shavlik, JW (2003). Knowledge-based nonlinear kernel classifiers. In: *Learning Theory and Kernel Machines*. Springer, 102–113.

Gelfand, AE and Smith, AF (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association* 85.410, 398–409.

George, EI and McCulloch, RE (1993). Variable selection via Gibbs sampling. *Journal of the American Statistical Association* 88.423, 881–889.

Gilks, WR, Richardson, S, and Spiegelhalter, DJ (1996). Introducing markov chain monte carlo. *Markov chain Monte Carlo in practice* 1, 19.

Graepel, T and Herbrich, R (2004). "Invariant pattern recognition by semi-definite programming machines". In: *Advances in neural information processing systems*, 33–40.

Grantham, NS, Reich, BJ, Borer, ET, and Gross, K (2017). MIMIX: a Bayesian Mixed-Effects Model for Microbiome Data from Designed Experiments. *arXiv preprint arXiv:1703.07747*.

Guyon, I, Weston, J, Barnhill, S, and Vapnik, V (2002). Gene selection for cancer classification using support vector machines. *Machine learning* 46.1-3, 389–422.

Hedenfalk, I, Duggan, D, Chen, Y, Radmacher, M, Bittner, M, Simon, R, Meltzer, P, Gusterson, B, Esteller, M, Raffeld, M, et al. (2001). Gene-expression profiles in hereditary breast cancer. *New England Journal of Medicine* 344.8, 539–548.

Henao, R, Yuan, X, and Carin, L (2014). "Bayesian nonlinear support vector machines and discriminative factor modeling". In: *Advances in Neural Information Processing Systems*, 1754–1762.

Hiramatsu, N, Joseph, VT, and Lin, JH (2011). Monitoring and manipulating mammalian unfolded protein response. In: *Methods in enzymology*. Vol. 491. Elsevier, 183–198.

Hofmann, T, Schölkopf, B, and Smola, AJ (2008). Kernel methods in machine learning. *The annals of statistics*, 1171–1220.

Ising, E (1925). Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik* 31.1, 253–258.

Kim, S, Pan, W, and Shen, X (2013). Network-based penalized regression with application to genomic data. *Biometrics* 69.3, 582–593.

Koh, I, Cha, J, Park, J, Choi, J, Kang, SG, and Kim, P (2018). The mode and dynamics of glioblastoma cell invasion into a decellularized tissue-derived extracellular matrix-based three-dimensional tumor model. *Scientific reports* 8.1, 4608.

Kundu, S, Cheng, Y, Shin, M, Manyam, G, Mallick, BK, and Baladandayuthapani, V (2018). Bayesian variable selection with graphical structure learning: Applications in integrative genomics. *PloS one* 13.7, e0195070.

Kurtoglu, M, Gao, N, Shang, J, Maher, JC, Lehrman, MA, Wangpaichitr, M, Savaraj, N, Lane, AN, and Lampidis, TJ (2007). Under normoxia, 2-deoxy-D-glucose elicits cell death in select tumor types not by inhibition of glycolysis but by interfering with N-linked glycosylation. *Molecular cancer therapeutics* 6.11, 3049–3058.

Lauer, F and Bloch, G (2008). Incorporating prior knowledge in support vector machines for classification: A review. *Neurocomputing* 71.7, 1578–1594.

Li, C and Li, H (2008). Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics* 24.9, 1175–1182.

Li, F and Zhang, NR (2010). Bayesian variable selection in structured high-dimensional covariate spaces with applications in genomics. *Journal of the American statistical association* 105.491, 1202–1214.

Liu, F, Chakraborty, S, Li, F, Liu, Y, Lozano, AC, et al. (2014). Bayesian regularization via graph Laplacian. *Bayesian Analysis* 9.2, 449–474.

Luts, J and Ormerod, JT (2014). Mean field variational Bayesian inference for support vector machine classification. *Computational Statistics & Data Analysis* 73, 163–176.

Mallick, BK, Ghosh, D, and Ghosh, M (2005). Bayesian classification of tumours by using gene expression data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.2, 219–234.

Mangasarian, OL and Kou, G (2007). "Feature selection for nonlinear kernel support vector machines". In: *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*. IEEE, 231–236.

Marchiori, E and Sebag, M (2005). "Bayesian learning with local support vector machines for cancer classification with gene expression data". In: *Workshops on Applications of Evolutionary Computation*. Springer, 74–83.

Metropolis, N, Rosenbluth, AW, Rosenbluth, MN, Teller, AH, and Teller, E (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics* 21.6, 1087–1092.

Mitchell, TJ and Beauchamp, JJ (1988). Bayesian variable selection in linear regression. *Journal of the American Statistical Association* 83.404, 1023–1032.

Mukherjee, S, Tamayo, P, Slonim, D, Verri, A, Golub, T, Mesirov, J, and Poggio, T (1999). *Support vector machine classification of microarray data*. Tech. rep. AI Memo 1677, Massachusetts Institute of Technology.

Nayak, J, Naik, B, and Behera, H (2015). A comprehensive survey on support vector machine in data mining tasks: applications & challenges. *International Journal of Database Theory and Application* 8.1, 169–186.

Nishimura, D (2001). BioCarta. *Biotech Software & Internet Report: The Computer Software Journal for Scient* 2.3, 117–120.

Ogata, H, Goto, S, Sato, K, Fujibuchi, W, Bono, H, and Kanehisa, M (1999). KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic acids research* 27.1, 29–34.

Pan, W, Xie, B, and Shen, X (2010). Incorporating predictor network in penalized regression with application to microarray data. *Biometrics* 66.2, 474–484.

Pointer, KB, Clark, PA, Schroeder, AB, Salamat, MS, Eliceiri, KW, and Kuo, JS (2016). Association of collagen architecture with glioblastoma patient survival. *Journal of neurosurgery* 126.6, 1812–1821.

Polson, NG, Scott, SL, et al. (2011). Data augmentation for support vector machines. *Bayesian Analysis* 6.1, 1–23.

Salcedo-Sanz, S, Rojo-Álvarez, JL, Martínez-Ramón, M, and Camps-Valls, G (2014). Support vector machines in engineering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 4.3, 234–267.

Schölkopf, B, Burges, C, and Vapnik, V (1996). Incorporating invariances in support vector learning machines. *Artificial Neural NetworksICANN 96*, 47–52.

Song, M, Breneman, CM, Bi, J, Sukumar, N, Bennett, KP, Cramer, S, and Tugcu, N (2002). Prediction of protein retention times in anion-exchange chromatography systems using support vector regression. *Journal of chemical information and computer sciences* 42.6, 1347–1357.

Stingo, FC and Vannucci, M (2010). Variable selection for discriminant analysis with Markov random field priors for the analysis of microarray data. *Bioinformatics* 27.4, 495–501.

Stingo, FC, Chen, YA, Tadesse, MG, and Vannucci, M (2011). Incorporating biological information into linear models: A Bayesian approach to the selection of pathways and genes. *The annals of applied statistics* 5.3.

Sun, W, Chang, C, Zhao, Y, and Long, Q (2018). "Knowledge-Guided Bayesian Support Vector Machine for High-Dimensional Data with Application to Analysis of Genomics Data". In: *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 1484–1493.

Tibshirani, R (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267–288.

Vapnik, VN and Vapnik, V (1998). *Statistical learning theory*. Vol. 1. Wiley New York.

Verhaak, RG, Hoadley, KA, Purdom, E, Wang, V, Qi, Y, Wilkerson, MD, Miller, CR, Ding, L, Golub, T, Mesirov, JP, et al. (2010). Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR, and NF1. *Cancer cell* 17.1, 98–110.

Wang, H et al. (2015). Scaling it up: Stochastic search structure learning in graphical models. *Bayesian Analysis* 10.2, 351–377.

Wang, L, Gao, Y, Chan, KL, Xue, P, and Yau, WY (2005). “Retrieval with knowledge-driven kernel design: an approach to improving SVM-based CBIR with relevance feedback”. In: *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*. Vol. 2. IEEE, 1355–1362.

Wang, L, Zhu, J, and Zou, H (2006). The doubly regularized support vector machine. *Statistica Sinica*, 589–615.

Weston, J, Mukherjee, S, Chapelle, O, Pontil, M, Poggio, T, and Vapnik, V (2001). “Feature selection for SVMs”. In: *Advances in neural information processing systems*, 668–674.

Wu, X and Srihari, R (2004). “Incorporating prior knowledge with weighted margin support vector machines”. In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 326–333.

Xiang, Z, Yichao, W, Lan, W, and Runze, L. Variable selection for support vector machines in moderately high dimensions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 78.1 (), 53–76.

Yang, X, Pan, W, and Guo, Y (2017). Sparse Bayesian classification and feature selection for biological expression data with high correlations. *PloS one* 12.12, e0189541.

Zhang, HH (2006). Variable selection for support vector machines via smoothing spline ANOVA. *Statistica Sinica*, 659–674.

Zhang, HH, Ahn, J, Lin, X, and Park, C (2005). Gene selection using support vector machines with non-convex penalty. *bioinformatics* 22.1, 88–95.

Zhao, Y, Chung, M, Johnson, BA, Moreno, CS, and Long, Q (2016). Hierarchical feature selection incorporating known and novel biological information: Identifying genomic features related to prostate cancer recurrence. *Journal of the American Statistical Association* 111.516, 1427–1439.

Zhou, H and Zheng, T (2013). Bayesian hierarchical graph-structured model for pathway analysis using gene expression data. *Statistical applications in genetics and molecular biology* 12.3, 393–412.

Zhu, J, Rosset, S, Tibshirani, R, and Hastie, TJ (2004). “1-norm support vector machines”. In: *Advances in neural information processing systems*, 49–56.

Zhu, Y, Shen, X, and Pan, W (2009). Network-based support vector machine for classification of microarray samples. *BMC bioinformatics* 10.1, S21.

Zou, H and Yuan, M (2008). The F-norm support vector machine. *Statistica Sinica*, 379–398.