Sparse Bayesian inference using reduced-rank regression approaches

by

Dunfu Yang

B.S., Shandong University of Finance and Economics, China, 2015

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Statistics
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2022

Abstract

In multivariate regression analysis, reduced-rank regression (RRR) has received considerable attention as a powerful way of improving estimation and prediction performances. In this dissertation, we aim to address challenges of dimension reduction associated with rank selection and variable selection in RRR. Our proposed methods are developed in a Bayesian framework so that the uncertainties of rank selection and variable selection can be integrated out via marginalization. We pay special attention to high-dimensional problems in which the number of potential predictors is greater than the sample size. We propose new posterior computation schemes to tackle high-dimensional data challenges under the RRR framework. A great merit of our proposed methods is that they are applicable to a variety of statistical models and machine learning methods including generalized linear models and support vector machines. In addition, various posterior sampling strategies are proposed for handling a variety of rank selection and variable selection problems. To investigate the performance of our proposed methods, simulation study and real data analysis are extensively implemented.

This dissertation consists of five chapters. In Chapter 1, we discuss the background and motivation of our study. In Chapter 2, we develop a fully Bayesian approach for high-dimensional RRR problems. In Chapter 3, we propose a multivariate extension of generalized linear models using the sparse RRR idea to handle various data types, including binary, count, and continuous responses. In Chapter 4, we develop a new support vector machine approach for multivariate binary outcomes by incorporating the RRR scheme into the Bayesian support vector machine framework. In Chapter 5, we discuss some remarks and future directions.

Sparse Bayesian inference using reduced-rank regression approaches

by

Dunfu Yang

B.S., Shandong University of Finance and Economics, China, 2015

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Statistics
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2022

Approved by:

Co-Major Professor
Gyuhyeong Goh

Approved by:

Co-Major Professor
Haiyan Wang

Copyright

Abstract

In multivariate regression analysis, reduced-rank regression (RRR) has received considerable attention as a powerful way of improving estimation and prediction performances. In this dissertation, we aim to address challenges of dimension reduction associated with rank selection and variable selection in RRR. Our proposed methods are developed in a Bayesian framework so that the uncertainties of rank selection and variable selection can be integrated out via marginalization. We pay special attention to high-dimensional problems in which the number of potential predictors is greater than the sample size. We propose new posterior computation schemes to tackle high-dimensional data challenges under the RRR framework. A great merit of our proposed methods is that they are applicable to a variety of statistical models and machine learning methods including generalized linear models and support vector machines. In addition, various posterior sampling strategies are proposed for handling a variety of rank selection and variable selection problems. To investigate the performance of our proposed methods, simulation study and real data analysis are extensively implemented.

This dissertation consists of five chapters. In Chapter 1, we discuss the background and motivation of our study. In Chapter 2, we develop a fully Bayesian approach for high-dimensional RRR problems. In Chapter 3, we propose a multivariate extension of generalized linear models using the sparse RRR idea to handle various data types, including binary, count, and continuous responses. In Chapter 4, we develop a new support vector machine approach for multivariate binary outcomes by incorporating the RRR scheme into the Bayesian support vector machine framework. In Chapter 5, we discuss some remarks and future directions.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to express my appreciation to Dr. Gyuhyeong Goh, my co-major advisor, for all this knowledge, guidance and suggestions. He was always available when I had doubts or questions. I learned a lot from him through many helpful discussions. My appreciation also goes to Dr. Haiyan Wang, another of my co-major advisor. She has given me a lot of valuable advises to my projects as well as my writings. I would also like to thank Dr. Jingru Mu and Dr. Ming-shun Chen for serving on my committee, and for their valuable guidance through my study process.

I would like to thank the Department of Statistics for offering me graduate assistantship so that I could come to the states and complete my graduate studies at Kansas State University. I would like to thank everyone in the department for their kindness, thank all the professors for their excellent courses and for their help. Finally, many thanks to my family for their endless love, support, understanding and encouragement. I would never have been able to finish my graduate studies without their support.

Last but not least, this dissertation has been supported by the Arthur D and Lavonia B Dayton Scholarship in 2019, the Lolafaye Coyne Statistics Graduate Scholarship in 2019 and 2020, as well as the Holly and Beth Fryer Scholarship in 2021. I would like to thank the families of Coyne, Fryer and Dayton for providing me scholarships and assistantship. These financial support helped me a lot through my study. Without their support, it would have been harder to complete my research in a timely manner.

Chapter 1

Introduction

In multivariate regression analysis, we often encounter situations where multiple response variables are correlated with each other. One way of taking advantage of such interrelationships between response variables is the reduced-rank regression (RRR) approach that imposes the low-rank restriction on the coefficient matrix. An immediate implication of the RRR method is that the effective number of parameters is substantially reduced so that the estimation efficiency can be improved.

Extensive research has been conducted for various reduced-rank regression problems. To extend the applicability of RRR to big data, it is crucial to break the curse of high-dimensionality. This becomes possible with the use of sparsity constraints in a penalized likelihood estimation framework. In recent years, the penalized likelihood approach to sparse and low-rank matrix estimation has made great advances. However, there remain many challenges for RRR problems. One of the major concerns is how to assess the uncertainty of sparse and low-rank matrix estimation. The focus of this dissertation is to make an important contribution to filling this research gap.

1.1 Sparse reduced-rank multivariate regression

When multiple response variables are available, one may perform separate linear regression analysis with each response by ignoring the interrelationships between the response variables. In practice, the reduced-rank regression model has been used to explain the correlation between multiple response variables (Anderson, 1951; Izenman, 1975). In a RRR model, the low-rank constraint imposed on the coefficient matrix enhances both estimation and prediction by allowing borrowing information across the response variables. See Velu and Reinsel (2013) for a comprehensive overview of theory and applications of RRR.

With the prevalence of high-dimensional data, variable selection becomes a key step to defeat the curse of dimensionality. We have seen a growing number of sparse and low-rank matrix techniques and their wide applications for high-dimensional data problems (Bunea et al., 2011, 2012; Chen et al., 2012, 2013; Chen and Huang, 2012; Ma et al., 2014; Yuan et al., 2007). However, how to assess the uncertainty associated with sparse and low-rank matrix estimation remains an open question.

To account for the uncertainty associated with the low-rank coefficient matrix, a simple yet powerful solution is a Bayesian approach (Alquier, 2013; Babacan et al., 2011; Geweke, 1996; Lim and Teh, 2007). A Bayesian framework enables us to impose the reduced-rank constraint on the coefficient matrix via the prior distribution and draw inferences from the posterior distribution. To incorporate sparsity into Bayesian reduced-rank regression, many attempts have been made through approximate Bayesian approaches including variational Bayesian methods and maximum a posteriori probability (MAP) approaches (Chakraborty et al., 2016; Goh et al., 2017; Marttinen et al., 2014; Zheng, 2014; Zhu et al., 2014). Despite the continuous growth in the development and application of Bayesian methods for sparse and low-rank matrix estimation, a fully Bayesian analysis of sparse reduced-rank regression, which accounts for both rank selection and variable selection uncertainties, is as yet undeveloped. It is worth noting that ignoring the uncertainty associated with model selection leads to over-confident Bayesian inference that results in underestimation of variability of the posterior distribution (Hoeting et al., 1999; Raftery et al., 1997).

In Chapter 2 of this dissertation, we aim to fill this research gap by developing a fully Bayesian framework for sparse reduced-rank regression (SRRR). From a fully Bayesian viewpoint, we treat the low-rank and the relevant predictors as random variables so that it is no longer required to select the true rank and/or the best sub-model, which is necessary in the existing literature. A major challenge with a fully Bayesian SRRR framework is that the rank and the sparsity determine the degree of freedom of the coefficient matrix. Consequently, we need to handle the varying-dimensionality problems that make the traditional Markov chain Monte Carlo (MCMC) computation infeasible. Although the reversible jump MCMC (Green, 1995) generally offers a solution to varying dimensionality problems, its computational cost is too expensive for our SRRR problem. As an alternative, we propose to use the marginal likelihood, which is a key quantity for Bayesian model selection and hypothesis testing (Carlin and Chib, 1995). See Llorente et al. (2020) for a comprehensive review of the marginal likelihood computation. However, in our SRRR framework, the marginal likelihood computation is extremely difficult due to the high-dimensionality of the candidate model space. To address this issue, we develop a new posterior computation procedure using the Laplace method within the collapsed Gibbs sampler.

1.2 A sparse reduced-rank approach to multivariate generalized linear regression

In Chapter 3, we consider an extension of SRRR where response variables are no longer restricted to be continuous. There are several attempts for developing RRR methods for non-continuous data. For example, Yee and Hastie (2003) extended RRR to a wider range of data types by proposing vector-generalized linear models (VGLMs). An application of RRR to survival data with multi-class responses was introduced by Fiocco et al. (2008). The reduced-rank structure was imposed on the proportional hazards model to avoid imprecise estimation in transitions with rare events and facilitate interpretation of the estimates under a competing risk framework. Luo et al. (2018) developed a mixed-outcome RRR (mRRR) method and

established a non-asymptotic performance bound for the proposed mRRR estimator. They also provided a practical modeling strategy and computational implementation for analyzing mixed-type outcomes. However, a major drawback of the existing works is that the rank selection has to be done prior to the estimation procedure so that the model uncertainty associated with the rank selection is not taken into account. In addition, there are limited attempts to tackle high-dimensional problems under the generalized RRR structure.

In Chapter 3, we develop a Bayesian framework for generalized SRRR for non-Gaussian data. Our proposed method aims to address several challenges regarding both estimation and inference problems in the generalized SRRR context. The proposed method enables us to estimate the unknown rank and remove irrelevant predictors simultaneously, in contrast to some existing methods in which the rank is assumed to be known or estimated by the single best model. From a Bayesian perspective, we treat the rank and the relevant predictors as random variables so that it is no longer required to select the true rank and the best sub-model which is necessary in the existing literature. Using a Laplace approximation technique, we develop a unifying Bayesian framework for various non-Gaussian data. The use of the Bayesian paradigm allows us to perform statistical inferences by generating samples from the posterior distribution without the assumption of the asymptotic normality.

1.3 A reduced-rank approach to multivariate support vector machine

Support vector machine (SVM) is a popular machine learning method for classification (Cortes and Vapnik, 1995). For the past two decades, it has been a subject of intense research activity not only in machine learning but also in statistics. SVMs are popular due to their ability to handle both linear and non-linear decision boundaries. For a tutorial introduction of SVM, see Burges (1998); Cristianini et al. (2000); Schølkopf and Burges (1998); Smola and Schölkopf (2004).

With the increasing use of high-dimensional data, many SVM approaches for variable

selection problems have been introduced in the literature. The LASSO penalty (Tibshirani, 1996), for example, was applied to SVM by Bradley and Mangasarian (1998); Song et al. (2002); Zhu et al. (2003). The SVMs with a non-convex penalty were also considered to alleviate biases in estimating nonzero coefficients (Becker et al., 2011; Zhang et al., 2006). Wang et al. (2006) proposed the notion of the double regularization for SVM to encourage the selection of correlated features. Zou and Yuan (2008) suggested the L_∞ penalized SVM to solve a group selection problem. In addition to frequentist approaches, Bayesian SVM methods have also drawn much attention. One important feature of Bayesian approaches is that the uncertainty of model parameters can be well explained by the posterior model probabilities. Marchiori and Sebag (2005) introduced a novel method for improving classification performance of SVM with recursive feature selection. See also Luts and Ormerod (2014); Sun et al. (2018) for examples. However, all the aforementioned methods are suitable for univariate binary outcomes, while in many real world situations, multivariate binary outcomes are very common for classification problems.

In Chapter 4, we propose a new Bayesian SVM approach that permits the use of multiple binary responses in a single SVM framework. Our approach is based on the fact that there are some interrelationships among the different responses due to a low-rank structure in the coefficient matrix. A key idea of the proposed method is to use the data augmentation representation of the hinge loss so that posterior sampling can be performed using the Gibbs sampler. The introduction of the reduced rank structure in a SVM framework leads to the dimension reduction and the prediction accuracy improvement simultaneously. In addtion, the use of the Bayesian paradigm for SVM allows us to make the probabilistic interpretation for prediction.

Chapter 2

A fully Bayesian approach to sparse reduced-rank multivariate regression

High-dimensional variable selection problems have been extensively studied in the Bayesian literature. For example, George and McCulloch (1993) proposed the stochastic search variable selection (SSVS) procedure with a two-component Gaussian mixture prior via Gibbs sampling. Park and Casella (2008) introduced a Bayesian counterpart of the lasso (Tibshirani, 1996) using a hierarchical representation of the Laplace prior. See Bhadra et al. (2019); Kyung et al. (2010); O'Hara et al. (2009) for a comprehensive review of Bayesian variable selection. Despite the large amount of research on high-dimensional variable selection, there have been very few attempts to tackle simultaneous variable selection and rank estimation problems in the Bayesian SRRR framework, which is the focus of this chapter.

In biological research, transcriptional regulatory network modeling is crucial to understanding the relationship between transcription factors and their target genes during a cell cycle. In the recent studies, SRRR has gained great attention as a powerful tool to estimate the regulatory activity of transcription factors (Chen and Huang, 2012; Chun and Keleş, 2010; Goh et al., 2017). The application of the SRRR method of Chen and Huang (2012) to the yeast cell cycle data (Lee et al., 2002; Spellman et al., 1998) selected 81 out of 106 transcription factors as regulators of gene expression. Such a large number makes it difficult

to perform follow up biological experiments. In this chapter, we apply the proposed Bayesian SRRR to the identification of relevant transcriptional regulators in the yeast cell cycle data. As a result, we identified 13 transcription factors as cell cycle regulators, all of which have biological relevance supported by existing bioscience literature.

This chapter is organized as follows. In Section 2.1, we specify the model setting for sparse reduced-rank regression (SRRR) and the priors for unknown parameters. In Section 2.2, we introduce a fully Bayesian approach to SRRR including technical details of posterior computation. The calculation of the full conditional posterior distributions is shown in Appendix. Simulation studies are presented in Section 2.3. The proposed method is applied to transcriptional regulatory network modeling in Section 2.4. Some concluding remarks are given in Section 2.5.

In this chapter, we use the following notations: we use $\mathbf{a}_j^\top$ to denote the j-th row of a generic matrix $\mathbf{A}$. For example, the $n \times p$ design matrix $\mathbf{X}$ can be written as $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\top$. For an index set Ω, $\mathbf{A}_{[\Omega,\Omega]}$ denotes a sub-matrix of $\mathbf{A}$ corresponding to indices in $\Omega \times \Omega$ and $\mathbf{a}_\Omega$ denotes a sub-vector of $\mathbf{a}$ corresponding to indices in Ω.

2.1 Model setup and prior specification

For each subject $i \in \{1, \ldots, n\}$, we observe $(\mathbf{x}_i, \mathbf{y}_i)$, where $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^\top$ denotes a p-dimensional predictor vector and $\mathbf{y}_i = (y_{i1}, \ldots, y_{iq})^\top$ represents a q-dimensional response vector. We consider a multivariate linear regression model,

$$\mathbf{Y} = \mathbf{X}\mathbf{C} + \mathbf{E}, \tag{2.1}$$

where $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_n)^\top$ is the $n \times q$ response matrix, $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\top$ is the $n \times p$ predictor matrix, $\mathbf{C}$ is the $p \times q$ coefficient matrix, and $\mathbf{E} = (\mathbf{e}_1, \ldots, \mathbf{e}_n)^\top$ is the $n \times q$ error matrix with $\mathbf{e}_i \overset{iid}{\sim} \mathcal{N}_q(\mathbf{0}_q, \mathbf{\Sigma})$ for some unknown $q \times q$ positive definite covariance matrix $\mathbf{\Sigma}$. We assume that both $\mathbf{X}$ and $\mathbf{Y}$ are centered so that the intercept is omitted in the regression model (2.1). A main goal of multivariate regression is to perform inferences for

the coefficient matrix $\mathbf{C}$ in order to study the relationship between $\mathbf{X}$ and $\mathbf{Y}$. Due to the fact that the multiple responses share the common predictors, the mean responses are often linearly dependent (Anderson, 1951; Izenman, 1975; Velu and Reinsel, 2013). An effective way to incorporate this linear dependent structure into the regression analysis is to assume that $\mathbf{C}$ does not have full rank, that is, $\text{rank}(\mathbf{C}) < \min(p, q)$.

Suppose that $\text{rank}(\mathbf{C}) = r$. Given r, the coefficient matrix $\mathbf{C}$ can be decomposed as a product of two full rank matrices, $\mathbf{C} = \mathbf{A}\mathbf{B}^\top$, where $\mathbf{A}$ is a $p \times r$ full rank matrix and $\mathbf{B}$ is a $q \times r$ full rank matrix. Therefore, the model (2.1) can be rewritten as

$$\mathbf{Y} = \mathbf{X}\mathbf{A}\mathbf{B}^\top + \mathbf{E}, \tag{2.2}$$

which is called the reduced-rank regression (RRR) model when $r < \min(p, q)$. Note that the decomposition, $\mathbf{C} = \mathbf{A}\mathbf{B}^\top$, is not unique. To verify this, suppose $\mathbf{P}$ is an $r \times r$ orthogonal matrix, i.e., $\mathbf{P}\mathbf{P}^\top = \mathbf{I}_r$. Let $\mathbf{A}_* = \mathbf{A}\mathbf{P}$ and $\mathbf{B}_* = \mathbf{B}\mathbf{P}$. Then, we have $\mathbf{C} = \mathbf{A}\mathbf{B}^\top = \mathbf{A}_*\mathbf{B}_*^\top$ while $\mathbf{A}_* \neq \mathbf{A}$ and $\mathbf{B}_* \neq \mathbf{B}$. To achieve a unique decomposition, following Geweke (1996) and Gilbert and Zemcik (2006), we assume

$$\mathbf{B} = \begin{bmatrix} \mathbf{I}_r \\ \mathbf{F} \end{bmatrix}, \tag{2.3}$$

where $\mathbf{F}$ is a $(q-r) \times r$ full rank matrix. It is worth noting that the number of free parameters of $\mathbf{C}$ in model (2.1) is $p \times q$, but it reduces to $(p + q - r) \times r$ in the RRR model (2.2) with our constraint (2.3).

In high-dimensional regression, a necessary procedure is to eliminate irrelevant predictors from the regression model. In a multivariate regression framework, the variable elimination can be achieved by introducing row-wise sparsity in the coefficient matrix $\mathbf{C}$. Specifically, in the form of $\mathbf{C} = \mathbf{A}\mathbf{B}^\top$, row-wise sparsity in $\mathbf{C}$ can be achieved by corresponding row-wise sparsity in $\mathbf{A}$. For example, if the i-th row of $\mathbf{A}$ is set equal to zero, then the i-th row of $\mathbf{C}$ becomes zero. Define an indicator vector of active predictors as $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_p)$, where

$\gamma_j = 1$ if the j-th predictor is active and $\gamma_j = 0$ otherwise for $j = 1, \ldots, p$. Let $\mathbf{X}_{\boldsymbol{\gamma}}$ be a sub-matrix of $\mathbf{X}$ obtained by deleting the columns of $\mathbf{X}$ corresponding to zero elements of $\boldsymbol{\gamma}$. Similarly, let $\mathbf{A}_{\boldsymbol{\gamma}}$ be a sub-matrix of $\mathbf{A}$ obtained by deleting the rows of $\mathbf{A}$ corresponding to zero elements of $\boldsymbol{\gamma}$. Then, given $\boldsymbol{\gamma}$, the RRR model (2.2) can be further reduced to

$$\mathbf{Y} = \mathbf{X}_{\boldsymbol{\gamma}} \mathbf{A}_{\boldsymbol{\gamma}} \mathbf{B}^\top + \mathbf{E}, \tag{2.4}$$

which is often referred to as the sparse reduced-rank regression (SRRR) model (Chen and Huang, 2012). While many studies including Chen and Huang (2012) and Goh et al. (2017) have been done on SRRR, the existing methods treat $\boldsymbol{\gamma}$ and r as unknown but fixed parameters and ignore the uncertainty involved in estimating $\boldsymbol{\gamma}$ and r. To address this issue, we develop a fully Bayesian approach to SRRR by treating them as random variables in a Bayesian fashion so that the uncertainty associated with r and $\boldsymbol{\gamma}$ can be integrated out in our posterior inference.

To complete our Bayesian modeling, appropriate priors should be assigned for unknown parameters including $\boldsymbol{\gamma}$ and r. For algebraic and computational convenience, we assign the multivariate Gaussian prior for each row of $\mathbf{A}_{\boldsymbol{\gamma}} = [\mathbf{a}_{\boldsymbol{\gamma}1}, \ldots, \mathbf{a}_{\boldsymbol{\gamma}p_{\boldsymbol{\gamma}}}]^\top$ given $\boldsymbol{\gamma}$ and r as follows:

$$\mathbf{a}_{\boldsymbol{\gamma}j} \mid \boldsymbol{\gamma}, r \overset{iid}{\sim} \mathcal{N}_r(\mathbf{0}_r, \nu_1 \mathbf{I}_r), \quad j = 1, \ldots, p_{\boldsymbol{\gamma}},$$

where ν_1 is a hyperparameter and $p_{\boldsymbol{\gamma}}$ denotes the number of ones in $\boldsymbol{\gamma}$, i.e., $p_{\boldsymbol{\gamma}} = \sum_{j=1}^{p} \gamma_j$. To impose a necessary constraint that a reduced model contains less number of predictors than the sample size, as in Chen and Chen (2008), we assume that

$$\pi(\boldsymbol{\gamma}) \propto \frac{1}{\binom{p}{p_{\boldsymbol{\gamma}}}} \mathbb{I}\left(p_{\boldsymbol{\gamma}} < n\right),$$

where $\mathbb{I}(\cdot)$ is an indicator function. To induce the unique decomposition defined in (2.3), we assume that the first r rows in $\mathbf{B}$ is an identity matrix with probability one and then assign

the multivariate Gaussian prior for each row of $\mathbf{F} = [\mathbf{f}_1, \ldots, \mathbf{f}_{q-r}]^\top$, that is,

$$\mathbf{f}_k \mid r \overset{iid}{\sim} \mathcal{N}_r(\mathbf{0}_r, \nu_2 \mathbf{I}_r), \quad k = 1, \ldots, q-r,$$

where ν_2 is a hyperparameter. Given $\boldsymbol{\gamma}$, we assign a discrete uniform prior for rank r, namely, $r \mid \boldsymbol{\gamma} \sim \mathcal{U}\{1, \ldots, \xi_{\boldsymbol{\gamma}}\}$, where $\xi_{\boldsymbol{\gamma}} = \min(p_{\boldsymbol{\gamma}}, q)$. For $\boldsymbol{\Sigma}$, we consider a conjugate prior, the inverse-Wishart distribution, $\mathcal{W}^{-1}(\nu_0, \boldsymbol{\Psi}_0)$ with the probability density function

$$\pi(\boldsymbol{\Sigma}) \quad = \quad \frac{|\boldsymbol{\Psi}_0|^{\nu_0/2}}{(2)^{\nu_0 q/2}\Gamma_q(\nu_0/2)} |\boldsymbol{\Sigma}|^{-(\nu_0+q+1)/2} \exp\left\{-\frac{1}{2}\mathrm{tr}(\boldsymbol{\Psi}_0 \boldsymbol{\Sigma}^{-1})\right\},$$

where ν_0 and $\boldsymbol{\Psi}_0$ are hyperparameters. Note that our priors lead to full conditionals in a closed form. However, Gibbs sampling is infeasible due to the so-called trans-dimensional problems.

Remark 1. *In model (2.4), the coefficient matrix, $\mathbf{C}_{\boldsymbol{\gamma}} = \mathbf{A}_{\boldsymbol{\gamma}}\mathbf{B}^\top$, has $(p_{\boldsymbol{\gamma}} + q - r)r$ unique parameters. Since the dimension of parameter space varies with r and $\boldsymbol{\gamma}$, the joint inference of $(\mathbf{C}_{\boldsymbol{\gamma}}, \boldsymbol{\gamma}, r)$ is subject to a trans-dimensional problem. We will discuss more details of the trans-dimensional problem and provide our solution in the following section.*

2.2 Posterior inference

2.2.1 Fully Bayesian inference via collapsed Gibbs sampler

In the SRRR framework, the pressing challenge of making fully Bayesian inference is that the number of parameters in $\mathbf{C}_{\boldsymbol{\gamma}} = \mathbf{A}_{\boldsymbol{\gamma}}\mathbf{B}^\top$ varies with states of r and $\boldsymbol{\gamma}$. Due to such trans-dimensional problems, posterior inference using the traditional MCMC computation including Gibbs sampling is impracticable. In general, the reversible jump MCMC of Green (1995) offers a way to simulate a varying dimensional Markov chain from the joint posterior distribution, $\pi(\mathbf{A}_{\boldsymbol{\gamma}}, \mathbf{B}, r, \boldsymbol{\gamma} \mid \mathbf{Y})$. Unfortunately, the reversible jump MCMC is impractical under our framework due to the extremely expensive computational cost. As an alternative,

we propose to use the collapsed Gibbs sampling algorithm (Liu, 1994) that generates the joint posterior sample by iterating the following steps until convergence:

1. Generate r from $\pi(r \mid \mathbf{Y}, \boldsymbol{\gamma}, \boldsymbol{\Sigma})$.
2. Generate $\boldsymbol{\gamma}$ from $\pi(\boldsymbol{\gamma} \mid \mathbf{Y}, r, \boldsymbol{\Sigma})$.
3. Generate $\mathbf{F}$ from $\pi(\mathbf{F} \mid \mathbf{Y}, r, \boldsymbol{\gamma}, \mathbf{A}_{\boldsymbol{\gamma}}, \boldsymbol{\Sigma})$ and set $\mathbf{B} = [\mathbf{I}_r, \mathbf{F}^\top]^\top$.
4. Generate $\mathbf{A}_{\boldsymbol{\gamma}}$ from $\pi(\mathbf{A}_{\boldsymbol{\gamma}} \mid \mathbf{Y}, r, \boldsymbol{\gamma}, \mathbf{B}, \boldsymbol{\Sigma})$.
5. Generate $\boldsymbol{\Sigma}$ from $\pi(\boldsymbol{\Sigma} \mid \mathbf{Y}, r, \boldsymbol{\gamma}, \mathbf{A}_{\boldsymbol{\gamma}}, \mathbf{B})$.

Suppose that $\{(r^{(t)}, \boldsymbol{\gamma}^{(t)}, \mathbf{F}^{(t)}, \mathbf{A}^{(t)}_{\boldsymbol{\gamma}^{(t)}}) : t = 1, \ldots, T\}$ is obtained from the above collapsed Gibbs sampler. Then, fully Bayesian inferences for the $p \times q$ coefficient matrix $\mathbf{C}$ in (2.1) can be made by constructing the marginal posterior sample $\{\mathbf{C}^{(t)} : t = 1, \ldots, T\}$ as follows:

- Repeat for $t = 1, \ldots, T$:

  i. Given $(r^{(t)}, \boldsymbol{\gamma}^{(t)}, \mathbf{F}^{(t)}, \mathbf{A}^{(t)}_{\boldsymbol{\gamma}^{(t)}})$, define the reduced coefficient matrix by

$$\mathbf{C}^{(t)}_{\boldsymbol{\gamma}^{(t)}} \;=\; \mathbf{A}^{(t)}_{\boldsymbol{\gamma}^{(t)}} \left[\mathbf{I}_{r^{(t)}}, \mathbf{F}^{(t)\top}\right]. \tag{2.5}$$

  ii. Given $\boldsymbol{\gamma}^{(t)} = (\gamma_1^{(t)}, \ldots, \gamma_p^{(t)})$, define the index set of the active predictors by

$$\mathcal{G}^{(t)} = \left\{ j_1^*, \ldots, j^*_{p_{\boldsymbol{\gamma}^{(t)}}} : \gamma^{(t)}_{j_k^*} = 1, j_1^* < \cdots < j^*_{p_{\boldsymbol{\gamma}^{(t)}}} \right\}.$$

  iii. Compute $\mathbf{C}^{(t)} = [\mathbf{c}_1^{(t)}, \cdots, \mathbf{c}_p^{(t)}]^\top$ with

$$\mathbf{c}_j^{(t)} = \begin{cases} (\mathbf{C}^{(t)}_{\boldsymbol{\gamma}^{(t)}})_{[k,\,]} & \text{if } j = j_k^* \text{ for } j_k^* \in \mathcal{G}^{(t)} \\ (0, \ldots, 0)^\top & \text{otherwise} \end{cases},$$

  where $(\mathbf{C}^{(t)}_{\boldsymbol{\gamma}^{(t)}})_{[k,\,]}$ indicates the k-th row of $\mathbf{C}^{(t)}_{\boldsymbol{\gamma}^{(t)}}$ in (2.5) and j_k^* is the k-th smallest element in $\mathcal{G}^{(t)}$.

2.2.2 Implementation details with asymptotic approximations and stochastic search algorithms

For Steps 3–5, the full conditional posterior distributions are well-known distributions such as Gaussian or inverse-Wishart, so the implementation of the last three steps can be done by generating samples from the known distributions. For Step 3, the full conditional distribution of $\mathbf{F}$ is shown to be

$$\text{vec}(\mathbf{F}^\top) \mid \mathbf{Y}, r, \boldsymbol{\gamma}, \mathbf{A}_{\boldsymbol{\gamma}}, \boldsymbol{\Sigma} \sim \mathcal{N}_{qr-r^2}(\boldsymbol{\mu}^{\mathbf{F}}, \boldsymbol{\Sigma}^{\mathbf{F}}), \tag{2.6}$$

where

$$\begin{aligned}
\boldsymbol{\mu}^{\mathbf{F}} &= \left\{(\boldsymbol{\Sigma}^{-1} \otimes \mathbf{A}_{\boldsymbol{\gamma}}^\top \mathbf{X}_{\boldsymbol{\gamma}}^\top \mathbf{X}_{\boldsymbol{\gamma}} \mathbf{A}_{\boldsymbol{\gamma}})_{[\Omega^*, \Omega^*]} + \nu_2^{-1} \mathbf{I}_{(qr-r^2)}\right\}^{-1} \\
&\quad \times \left\{\text{vec}(\mathbf{A}_{\boldsymbol{\gamma}}^\top \mathbf{X}_{\boldsymbol{\gamma}}^\top \mathbf{Y} \boldsymbol{\Sigma}^{-1})_{\Omega^*} - (\boldsymbol{\Sigma}^{-1} \otimes \mathbf{A}_{\boldsymbol{\gamma}}^\top \mathbf{X}_{\boldsymbol{\gamma}}^\top \mathbf{X}_{\boldsymbol{\gamma}} \mathbf{A}_{\boldsymbol{\gamma}})_{[\Omega^*, -\Omega^*]} \text{vec}(\mathbf{I}_r)\right\}, \\
\boldsymbol{\Sigma}^{\mathbf{F}} &= \left\{(\boldsymbol{\Sigma}^{-1} \otimes \mathbf{A}_{\boldsymbol{\gamma}}^\top \mathbf{X}_{\boldsymbol{\gamma}}^\top \mathbf{X}_{\boldsymbol{\gamma}} \mathbf{A}_{\boldsymbol{\gamma}})_{[\Omega^*, \Omega^*]} + \nu_2^{-1} \mathbf{I}_{(qr-r^2)}\right\}^{-1},
\end{aligned}$$

and $\Omega^* = \{r^2+1, r^2+2, \ldots, q \times r\}$ is an index set indicating the part of $\text{vec}(\mathbf{F}^\top)$ in $\text{vec}(\mathbf{B}^\top)$; see Appendix A. For Step 4, the full conditional distribution of $\mathbf{A}_{\boldsymbol{\gamma}}$ is shown to be

$$\text{vec}(\mathbf{A}_{\boldsymbol{\gamma}}) \mid \mathbf{Y}, r, \boldsymbol{\gamma}, \mathbf{B}, \boldsymbol{\Sigma} \sim \mathcal{N}_{p_{\boldsymbol{\gamma}} r}(\boldsymbol{\mu}^{\mathbf{A}_{\boldsymbol{\gamma}}}, \boldsymbol{\Sigma}^{\mathbf{A}_{\boldsymbol{\gamma}}}), \tag{2.7}$$

where

$$\begin{aligned}
\boldsymbol{\mu}^{\mathbf{A}_{\boldsymbol{\gamma}}} &= \left\{(\boldsymbol{\Sigma}^{-1/2}\mathbf{B} \otimes \mathbf{X}_{\boldsymbol{\gamma}})^\top (\boldsymbol{\Sigma}^{-1/2}\mathbf{B} \otimes \mathbf{X}_{\boldsymbol{\gamma}}) + \nu_1^{-1} \mathbf{I}_{p_{\boldsymbol{\gamma}} r}\right\}^{-1} (\boldsymbol{\Sigma}^{-1}\mathbf{B} \otimes \mathbf{X}_{\boldsymbol{\gamma}})^\top \text{vec}(\mathbf{Y}), \\
\boldsymbol{\Sigma}^{\mathbf{A}_{\boldsymbol{\gamma}}} &= \left\{(\boldsymbol{\Sigma}^{-1/2}\mathbf{B} \otimes \mathbf{X}_{\boldsymbol{\gamma}})^\top (\boldsymbol{\Sigma}^{-1/2}\mathbf{B} \otimes \mathbf{X}_{\boldsymbol{\gamma}}) + \nu_1^{-1} \mathbf{I}_{p_{\boldsymbol{\gamma}} r}\right\}^{-1};
\end{aligned}$$

see Appendix A. For Step 5, the full conditional distribution of $\boldsymbol{\Sigma}$ is shown to be

$$\boldsymbol{\Sigma} \mid \mathbf{Y}, r, \boldsymbol{\gamma}, \mathbf{A}_{\boldsymbol{\gamma}}, \mathbf{B} \sim \mathcal{W}^{-1}(n + \nu_0, \mathbf{S}), \tag{2.8}$$

where $\mathbf{S} = (\mathbf{Y} - \mathbf{X}_{\boldsymbol{\gamma}}\mathbf{A}_{\boldsymbol{\gamma}}\mathbf{B}^\top)^\top(\mathbf{Y} - \mathbf{X}_{\boldsymbol{\gamma}}\mathbf{A}_{\boldsymbol{\gamma}}\mathbf{B}^\top) + \boldsymbol{\Psi}_0$; see Appendix A.

However, there are two serious obstacles to implementing Steps 1 and 2. First, deriving a closed form expression of $\pi(r \mid \mathbf{Y}, \boldsymbol{\gamma}, \boldsymbol{\Sigma})$ and $\pi(\boldsymbol{\gamma} \mid \mathbf{Y}, r, \boldsymbol{\Sigma})$ is challenging due to analytically intractable integrations. Second, when p is large (or even moderate), generating $\boldsymbol{\gamma}$ from $\pi(\boldsymbol{\gamma} \mid \mathbf{Y}, r, \boldsymbol{\Sigma})$ is impractical since it is computationally extremely expensive to explore the 2^p dimensional parameter space of $\boldsymbol{\gamma}$. For example, when $p = 20$, we need to evaluate $\pi(\boldsymbol{\gamma} \mid \mathbf{Y}, r, \boldsymbol{\Sigma})$ for $2^{20} = 1,048,576$ candidates in every iteration.

To overcome the first issue, we employ the Laplace method as in Kass and Raftery (1995); Tierney and Kadane (1986); Tierney et al. (1989). By Bayes' theorem, calculating $\pi(r \mid \mathbf{Y}, \boldsymbol{\gamma}, \boldsymbol{\Sigma})$ can be done by

$$\pi(r \mid \mathbf{Y}, \boldsymbol{\gamma}, \boldsymbol{\Sigma}) = \frac{f(\mathbf{Y} \mid r, \boldsymbol{\gamma}, \boldsymbol{\Sigma})\pi(r \mid \boldsymbol{\gamma})}{\sum_{r'=1}^{\xi_{\boldsymbol{\gamma}}} f(\mathbf{Y} \mid r', \boldsymbol{\gamma}, \boldsymbol{\Sigma})\pi(r' \mid \boldsymbol{\gamma})}, \tag{2.9}$$

where

$$f(\mathbf{Y} \mid r, \boldsymbol{\gamma}, \boldsymbol{\Sigma}) = \iint f(\mathbf{Y} \mid \mathbf{A}_{\boldsymbol{\gamma}}, \mathbf{B}, \boldsymbol{\Sigma}, r, \boldsymbol{\gamma})\pi(\mathbf{A}_{\boldsymbol{\gamma}}, \mathbf{B} \mid r, \boldsymbol{\gamma}, \boldsymbol{\Sigma})\mathrm{d}\mathbf{A}_{\boldsymbol{\gamma}}\mathrm{d}\mathbf{B}. \tag{2.10}$$

Similarly, $\pi(\boldsymbol{\gamma} \mid \mathbf{Y}, r, \boldsymbol{\Sigma})$ can be computed as

$$\pi(\boldsymbol{\gamma} \mid \mathbf{Y}, r, \boldsymbol{\Sigma}) = \frac{f(\mathbf{Y} \mid r, \boldsymbol{\gamma}, \boldsymbol{\Sigma})\pi(\boldsymbol{\gamma})}{\sum_{\boldsymbol{\gamma}' \in \{0,1\}^p} f(\mathbf{Y} \mid r, \boldsymbol{\gamma}', \boldsymbol{\Sigma})\pi(\boldsymbol{\gamma}')}. \tag{2.11}$$

Note that, if $f(\mathbf{Y} \mid r, \boldsymbol{\gamma}, \boldsymbol{\Sigma})$ is given in a closed form, then the calculation of both $\pi(r \mid \mathbf{Y}, \boldsymbol{\gamma}, \boldsymbol{\Sigma})$ and $\pi(\boldsymbol{\gamma} \mid \mathbf{Y}, r, \boldsymbol{\Sigma})$ are done by using (2.9) and (2.11). Hence, it suffices to obtain an approximation of (2.10). Ignoring the constant terms with respect to n (Schwarz, 1978), the Laplace approximation leads to

$$\log f(\mathbf{Y} \mid r, \boldsymbol{\gamma}, \boldsymbol{\Sigma}) \approx \log f(\mathbf{Y} \mid \hat{\mathbf{A}}_{\boldsymbol{\gamma}}, \hat{\mathbf{B}}, \boldsymbol{\Sigma}, r, \boldsymbol{\gamma}) - \frac{1}{2}(p_{\boldsymbol{\gamma}} r + qr - r^2)\log n, \tag{2.12}$$

where $\hat{\mathbf{A}}_{\boldsymbol{\gamma}}$ and $\hat{\mathbf{B}}$ are maximum likelihood estimates (MLEs) of $\mathbf{A}_{\boldsymbol{\gamma}}$ and $\mathbf{B}$ given r and $\boldsymbol{\gamma}$.

Note that the likelihood function of the SRRR model is given as

$$f(\mathbf{Y} \mid \mathbf{A}_{\boldsymbol{\gamma}}, \mathbf{B}, \boldsymbol{\Sigma}, r, \boldsymbol{\gamma}) = \frac{1}{(2\pi)^{nq/2}|\boldsymbol{\Sigma}|^{n/2}} \exp\left\{-\frac{1}{2}\left\|(\mathbf{Y} - \mathbf{X}_{\boldsymbol{\gamma}}\mathbf{C}_{\boldsymbol{\gamma}})\boldsymbol{\Sigma}^{-1/2}\right\|_F^2\right\},$$

where $\mathbf{C}_{\boldsymbol{\gamma}} = \mathbf{A}_{\boldsymbol{\gamma}}\mathbf{B}^\top$ and $\|\cdot\|_F$ denotes the Frobenius norm. Therefore, calculating the MLE of $\mathbf{C}_{\boldsymbol{\gamma}}$ is sufficient to compute $f(\mathbf{Y} \mid \hat{\mathbf{A}}_{\boldsymbol{\gamma}}, \hat{\mathbf{B}}, \boldsymbol{\Sigma}, r, \boldsymbol{\gamma})$ in (2.12) rather than obtaining $\hat{\mathbf{A}}_{\boldsymbol{\gamma}}$ and $\hat{\mathbf{B}}$ individually. Let $\hat{\mathbf{C}}_{\boldsymbol{\gamma}}$ be the MLE of $\mathbf{C}_{\boldsymbol{\gamma}}$. Then, we can obtain $\hat{\mathbf{C}}_{\boldsymbol{\gamma}}$ by solving the following rank constrained optimization problem:

$$\min_{\mathbf{C}_{\boldsymbol{\gamma}}} \left\|(\mathbf{Y} - \mathbf{X}_{\boldsymbol{\gamma}}\mathbf{C}_{\boldsymbol{\gamma}})\boldsymbol{\Sigma}^{-1/2}\right\|_F^2 \quad \text{subject to} \quad \text{rank}(\mathbf{C}_{\boldsymbol{\gamma}}) = r. \tag{2.13}$$

By Velu and Reinsel (2013), a minimizer of (2.13) is given by

$$\hat{\mathbf{C}}_{\boldsymbol{\gamma}} = (\mathbf{X}_{\boldsymbol{\gamma}}^\top\mathbf{X}_{\boldsymbol{\gamma}})^{-1}\mathbf{X}_{\boldsymbol{\gamma}}^\top\mathbf{Y}\boldsymbol{\Sigma}^{-1/2}\mathbf{V}\mathbf{V}^\top\boldsymbol{\Sigma}^{1/2},$$

where $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_r]$ and $\mathbf{v}_j$ is the eigenvector of $(\mathbf{Y}\boldsymbol{\Sigma}^{-1/2})^\top\mathbf{X}_{\boldsymbol{\gamma}}(\mathbf{X}_{\boldsymbol{\gamma}}^\top\mathbf{X}_{\boldsymbol{\gamma}})^{-1}\mathbf{X}_{\boldsymbol{\gamma}}^\top\mathbf{Y}\boldsymbol{\Sigma}^{-1/2}$ corresponding to the j-th largest eigenvalue. Incorporating (2.12) into (2.9), we thus have

$$\pi(r \mid \mathbf{Y}, \boldsymbol{\gamma}, \boldsymbol{\Sigma}) \approx \frac{\tilde{f}(\mathbf{Y} \mid r, \boldsymbol{\gamma}, \boldsymbol{\Sigma})}{\sum_{r'=1}^{\xi_{\boldsymbol{\gamma}}} \tilde{f}(\mathbf{Y} \mid r, \boldsymbol{\gamma}, \boldsymbol{\Sigma})} \equiv \tilde{\pi}(r \mid \mathbf{Y}, \boldsymbol{\gamma}, \boldsymbol{\Sigma}),$$

where $\tilde{f}(\mathbf{Y} \mid r, \boldsymbol{\gamma}, \boldsymbol{\Sigma})$ denotes the Laplace approximation of $f(\mathbf{Y} \mid r, \boldsymbol{\gamma}, \boldsymbol{\Sigma})$ obtained from (2.12). Then, Step 1 can be implemented by generating a value of $r \in \{1, \ldots, \xi_{\boldsymbol{\gamma}}\}$ with probabilities $\tilde{\pi}(r \mid \mathbf{Y}, \boldsymbol{\gamma}, \boldsymbol{\Sigma})$.

Similarly, we have

$$\pi(\boldsymbol{\gamma} \mid \mathbf{Y}, r, \boldsymbol{\Sigma}) \approx \frac{\tilde{f}(\mathbf{Y} \mid r, \boldsymbol{\gamma}, \boldsymbol{\Sigma})\pi(\boldsymbol{\gamma})}{\sum_{\boldsymbol{\gamma}' \in \{0,1\}^p} \tilde{f}(\mathbf{Y} \mid r, \boldsymbol{\gamma}', \boldsymbol{\Sigma})\pi(\boldsymbol{\gamma}')} \equiv \tilde{\pi}(\boldsymbol{\gamma} \mid \mathbf{Y}, r, \boldsymbol{\Sigma}).$$

If p is small, it is straightforward to implement Step 2 by using $\tilde{\pi}(\boldsymbol{\gamma} \mid \mathbf{Y}, r, \boldsymbol{\Sigma})$. However, as mentioned earlier, generating a sample of 2^p dimensional binary vector is computationally difficult in our high-dimensional regression setting. To address this issue, we propose to use

the Metropolized Shotgun Stochastic Search (SSS) algorithm (Hans et al., 2007). To this end, let $\text{nbd}(\boldsymbol{\gamma})$ be a neighborhood of $\boldsymbol{\gamma}$ that includes models with one more predictor or one less predictor than $\boldsymbol{\gamma}$ as well as $\boldsymbol{\gamma}$ itself. For example, if $\boldsymbol{\gamma} = (0, 1, 0)$, then $\text{nbd}(\boldsymbol{\gamma}) = \{(1, 1, 0), (0, 1, 1), (0, 0, 0), (0, 1, 0)\}$. We define a proposal distribution by

$$g(\boldsymbol{\gamma} \mid \boldsymbol{\gamma}^{(t)}) \propto \tilde{\pi}(\boldsymbol{\gamma} \mid \mathbf{Y}, r, \boldsymbol{\Sigma}) \mathbb{I}\left(\boldsymbol{\gamma} \in \text{nbd}(\boldsymbol{\gamma}^{(t)})\right).$$

where $\boldsymbol{\gamma}^{(t)}$ represents the current state of $\boldsymbol{\gamma}$ in the collapsed Gibbs sampling procedure. For Step 2, the Metropolized SSS can be implemented as follows: (i) Generate $\boldsymbol{\gamma}^*$ from $g(\boldsymbol{\gamma} \mid \boldsymbol{\gamma}^{(t)})$. (ii) Accept $\boldsymbol{\gamma}^{(t+1)} = \boldsymbol{\gamma}^*$ with probability

$$\min\left\{1, \frac{\sum_{\boldsymbol{\gamma} \in \text{nbd}(\boldsymbol{\gamma}^{(t)})} \tilde{f}(\mathbf{Y} \mid r, \boldsymbol{\gamma}, \boldsymbol{\Sigma}) \pi(\boldsymbol{\gamma})}{\sum_{\boldsymbol{\gamma}' \in \text{nbd}(\boldsymbol{\gamma}^*)} \tilde{f}(\mathbf{Y} \mid r, \boldsymbol{\gamma}', \boldsymbol{\Sigma}) \pi(\boldsymbol{\gamma}')}\right\},$$

and otherwise, set $\boldsymbol{\gamma}^{(t+1)} = \boldsymbol{\gamma}^{(t)}$.

Our MCMC scheme relies on the idea of the Metropolis-within-Gibbs sampler, which was originally proposed by Tierney (1994) and further developed by Gilks et al. (1995); Martino et al. (2018, 2015). Recent developments of the Metropolis-within-Gibbs algorithms can be incorporated into our framework to improve the computational efficiency of our fully Bayesian inference.

Remark 2. *Although we omit the subscript r in $\mathbf{A}_{\boldsymbol{\gamma}}$, $\mathbf{B}$ and $\mathbf{F}$ for the sake of notational simplicity, the size of the matrices relies on the state of r.*

Remark 3. *Let $\mathbf{C}^{(t)}$ be the current state of the $p \times q$ coefficient matrix defined by $\mathbf{A}^{(t)}_{\boldsymbol{\gamma}^{(t)}}$, $\mathbf{A}^{(t)}_{-\boldsymbol{\gamma}^{(t)}}$ and $\mathbf{B}^{(t)}$. At iteration $t+1$, in Step 3, we can construct the given condition $\mathbf{A}^{(t)}_{\boldsymbol{\gamma}^{(t+1)}}$ from $\mathbf{C}^{(t)}$ by removing the last $q - r^{(t+1)}$ columns and the rows corresponding to zero entries of $\boldsymbol{\gamma}^{(t+1)}$.*

2.3 Simulation study

In this section, we conduct a simulation study to examine the performance of the proposed fully Bayesian method for SRRR. Under different scenarios, data are independently generated from a multivariate regression model, $\mathbf{Y} = \mathbf{XC} + \mathbf{E}$, where $\mathbf{X} = (x_{ij})_{n\times p}$ with $x_{ij} \overset{iid}{\sim} \mathcal{U}(-6, 6)$, $\mathbf{C} = [\mathbf{C}_{\boldsymbol{\gamma}}^\top, \mathbf{0}_{q\times(p-p_0)}]^\top$, $\mathbf{C}_{\boldsymbol{\gamma}}$ is generated by imposing zero constraints on the last $p_0 - r$ singular values of $\tilde{\mathbf{C}}_{\boldsymbol{\gamma}} = (a_{ij}b_{ij})_{p_0\times q}$ such that $a_{ij} \overset{iid}{\sim} \mathcal{U}(0.5, 3)$ and $b_{ij} \sim \mathcal{U}\{-1, 1\}$, and $\mathbf{E} = [\mathbf{e}_1, \ldots, \mathbf{e}_n]^\top$ with $\mathbf{e}_i \sim \mathcal{N}_q(\mathbf{0}, \boldsymbol{\Sigma})$ and $\boldsymbol{\Sigma} = (\rho^{|i-j|})_{q\times q}$.

In Scenario 1, we set $\rho = 0$ and consider the following six cases:

(i) $n = 100$, $p = 80$, $p_0 = 6$, $q = 10$, $r = 1$.

(ii) $n = 100$, $p = 300$, $p_0 = 6$, $q = 10$, $r = 1$.

(iii) $n = 100$, $p = 80$, $p_0 = 6$, $q = 10$, $r = 3$.

(iv) $n = 100$, $p = 300$, $p_0 = 6$, $q = 10$, $r = 3$.

(v) $n = 100$, $p = 80$, $p_0 = 6$, $q = 10$, $r = 5$.

(vi) $n = 100$, $p = 300$, $p_0 = 6$, $q = 10$, $r = 5$.

In Scenario 2, we set $\rho = 0.5$ to create a positive correlation between errors and then consider the above six cases as in Scenario 1.

For each simulated dataset, we employ the proposed fully Bayesian method for estimating the $p \times q$ coefficient matrix $\mathbf{C}$. We consider noninformative or flat priors by setting $\nu_1 = \nu_2 = 1000$, $\nu_0 = 1$ and $\boldsymbol{\Psi}_0 = 0.5\mathbf{I}_q$. In the MCMC algorithm proposed in Section 2.2, we define the necessary initial values by $\mathbf{C}^{(0)} = (\mathbf{X}^\top\mathbf{X} + \nu_1^{-1}\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{Y}$ and $\boldsymbol{\gamma}^{(0)} = [\mathbb{I}(\eta_1 \geq \eta_{(n/4)}), \ldots, \mathbb{I}(\eta_p \geq \eta_{(n/4)})]$, where $\eta_i = \sum_{j=1}^q |c_{ij}^{(0)}|$, $c_{ij}^{(0)}$ denotes the (i, j)-th element of $\mathbf{C}^{(0)}$, and $\eta_{(k)}$ is the k-th largest element of $\{\eta_1, \ldots, \eta_p\}$. We run the MCMC algorithm for $3,000$ iterations after a burn-in period of $2,000$ iterations.

For the sake of comparison, we consider the method of Chen and Huang (2012) (CH-SRRR) that estimates $\mathbf{C} = \mathbf{A}\mathbf{B}^\top$ by minimizing

$$\min_{\mathbf{A},\mathbf{B}} \left\|\mathbf{Y} - \mathbf{X}\mathbf{A}\mathbf{B}^\top\right\|_F^2 + \sum_{i=1}^{p} \lambda \left\|\mathbf{a}_i\right\| \quad \text{such that} \quad \mathbf{B}^\top\mathbf{B} = \mathbf{I}_r, \tag{2.14}$$

where the tuning parameter $\lambda(> 0)$ controls the degree of row-wise sparsity in $\mathbf{A}$. To choose optimal λ and r, we use two-dimensional five-fold cross validation (CV) for $\lambda \in \{0, 0.5, \ldots, 200\}$ and $r \in \{1, \ldots, q\}$. As a benchmark, we consider an oracle estimator that is obtained by minimizing

$$\min_{\mathbf{C}_{\boldsymbol{\gamma}^*}} \|\mathbf{Y} - \mathbf{X}_{\boldsymbol{\gamma}^*}\mathbf{C}_{\boldsymbol{\gamma}^*}\|_F^2 \quad \text{such that} \quad \text{rank}(\mathbf{C}_{\boldsymbol{\gamma}^*}) = r^*, \tag{2.15}$$

where r^* and $\boldsymbol{\gamma}^*$ indicate the true values of r and $\boldsymbol{\gamma}$ used in the data generating process.

The performance of each method is assessed by the following three types of mean squared errors (MSEs):

$$\begin{aligned}
\text{MSE}_1 &= \|\mathbf{Y}_{\text{test}} - \mathbf{X}_{\text{test}}\hat{\mathbf{C}}\|_F^2/(n_{\text{test}}q), \\
\text{MSE}_2 &= \|\mathbf{C} - \hat{\mathbf{C}}\|_F^2/(pq), \\
\text{MSE}_3 &= \|\mathbf{X}_{\text{test}}\mathbf{C} - \mathbf{X}_{\text{test}}\hat{\mathbf{C}}\|_F^2/(n_{\text{test}}q),
\end{aligned}$$

where $\mathbf{X}_{\text{test}}$ is a newly-generated $n_{\text{test}} \times p$ test-sample matrix of predictors and $\mathbf{Y}_{\text{test}}$ is a newly-generated $n_{\text{test}} \times q$ test-sample matrix of responses for $n_{\text{test}} = 1,000$. Note that MSE_1 measures the average squared difference between the fitted values by the regression model and the observed values. MSE_2 quantifies the estimation error by measuring the Frobenius norm distance between the estimated coefficient matrix and the true coefficient matrix. MSE_3, often referred to as the mean squared prediction error, measures the prediction accuracy by using the average squared difference between the fitted values and the true mean function.

Tables 2.1 and 2.2 summarize our simulation results over $1,000$ Monte Carlo experiments. The numbers in the table represent the average MSEs for each method. The result clearly

shows that our fully Bayesian method consistently produces smaller MSEs than that of CH-SRRR. In addition, the performance of the proposed Bayesian method (in terms of all three MSEs) is always closer to that of the oracle method for both moderate-dimensional data ($p = 80$) and high-dimensional data ($p = 300$) for $r = 1, 3, 5$ than the competing SRRR method.By comparing Table 2.1 (Scenario 1) with Table 2.2 (Scenario 2), we also observe that there are small changes in MSEs of the proposed method between Scenarios 1 and 2. This implies that the proposed fully Bayesian method is robust to the change in correlation structure of the error term. This robustness is due to the fact that our fully Bayesian approach accounts for the uncertainty associated with the covariance matrix $\mathbf{\Sigma}$.

Let $\hat{\mathbf{C}}_{\text{Oracle}}$, $\hat{\mathbf{C}}_{\text{Bayes}}$, and $\hat{\mathbf{C}}_{\text{SRRR}}$ be the estimates of $\mathbf{C}$ obtained by the oracle method, the proposed Bayesian method, and CH-SRRR, respectively. To investigate the performance of the proposed Bayesian estimator, we measure the element-wise difference between $\hat{\mathbf{C}}_{\text{Bayes}}$ and $\hat{\mathbf{C}}_{\text{Oracle}}$. In Figure 2.1a, the heatmap visualizes the element-wise average of $\hat{\mathbf{C}}_{\text{Bayes}} - \hat{\mathbf{C}}_{\text{Oracle}}$ over $1,000$ Monte Carlo replicates in Scenario 2(v) (i.e., $\rho = 0.5$, $p = 80$, and $r = 5$). The rectangle in the i-th row and j-th column of the heatmap corresponds to the (i, j)-th entry of the $p \times q$ matrix $\hat{\mathbf{C}}_{\text{Bayes}} - \hat{\mathbf{C}}_{\text{Oracle}}$. The heatmap displays values of 0 as white, positive values as shades of red, and negative values as shades of blue. Similarly, the heatmap in Figure 2.1b shows the average of $\hat{\mathbf{C}}_{\text{SRRR}} - \hat{\mathbf{C}}_{\text{Oracle}}$ over $1,000$ Monte Carlo replicates in Scenario 2(v). By comparing Figure 2.1a with Figure 2.1b, we clearly see that the proposed Bayesian method provides a remarkable improvement in element-wise point estimation for $\mathbf{C}$, especially for zero elements of $\mathbf{C}$. This is consistent throughout all the scenarios; in this chapter, we only show the result of Scenario 2(v) to maintain clarity and avoid unnecessary duplication. To summarize, our simulation demonstrates that the fully Bayesian method not only achieves comparable performance to the oracle estimator but also outperforms the existing SRRR method that ignores the uncertainty associated with variable selection and low-rank reduction.

Table 2.1: Simulation results: average MSEs and standard errors (in parenthesis) over $1,000$ Monte Carlo experiments in Scenario 1. MSE_1: average squared difference between the fitted and observed values. MSE_2: Frobenius norm distance between the estimated and the true coefficient matrix. MSE_3: mean squared prediction error between the fitted values and the true mean function. The performance of Fully Bayes is closer to the Oracle than that of CH-SRRR based on all three MSE measures.

case	r	p	Method	MSE_1	$\text{MSE}_2 \times 10^5$	$\text{MSE}_3 \times 10^2$
(i)	1	80	CH-SRRR	1.0088 (0.0005)	4.1852 (0.0470)	1.0028 (0.0113)
			Oracle	1.0027 (0.0005)	1.6399 (0.0196)	0.3931 (0.0047)
			Fully Bayes	1.0038 (0.0005)	2.1212 (0.0322)	0.5085 (0.0077)
(ii)	1	300	CH-SRRR	1.0136 (0.0005)	1.5962 (0.0178)	1.4336 (0.0161)
			Oracle	1.0030 (0.0005)	0.4280 (0.0051)	0.3845 (0.0045)
			Fully Bayes	1.0049 (0.0005)	0.6321 (0.0156)	0.5684 (0.0141)
(iii)	3	80	CH-SRRR	1.0200 (0.0005)	8.8753 (0.0658)	2.2170 (0.0159)
			Oracle	1.0090 (0.0005)	4.2859 (0.0334)	1.0266 (0.0080)
			Fully Bayes	1.0098 (0.0005)	4.6023 (0.0398)	1.1026 (0.0096)
(iv)	3	300	CH-SRRR	1.0257 (0.0005)	2.9670 (0.0205)	2.6652 (0.0184)
			Oracle	1.0094 (0.0005)	1.1428 (0.0083)	1.0273 (0.0075)
			Fully Bayes	1.0101 (0.0005)	1.2195 (0.0099)	1.0962 (0.0090)
(v)	5	80	CH-SRRR	1.0286 (0.0005)	12.4503 (0.0782)	2.9856 (0.0188)
			Oracle	1.0133 (0.0005)	6.0831 (0.0411)	1.4581 (0.0099)
			Fully Bayes	1.0138 (0.0005)	6.3234 (0.0449)	1.5155 (0.0108)
(vi)	5	300	CH-SRRR	1.0354 (0.0005)	4.0376 (0.0241)	3.6282 (0.0218)
			Oracle	1.0138 (0.0005)	1.6299 (0.0101)	1.4651 (0.0091)
			Fully Bayes	1.0144 (0.0005)	1.6955 (0.0113)	1.5239 (0.0102)

2.4 Transcriptional regulatory network modeling

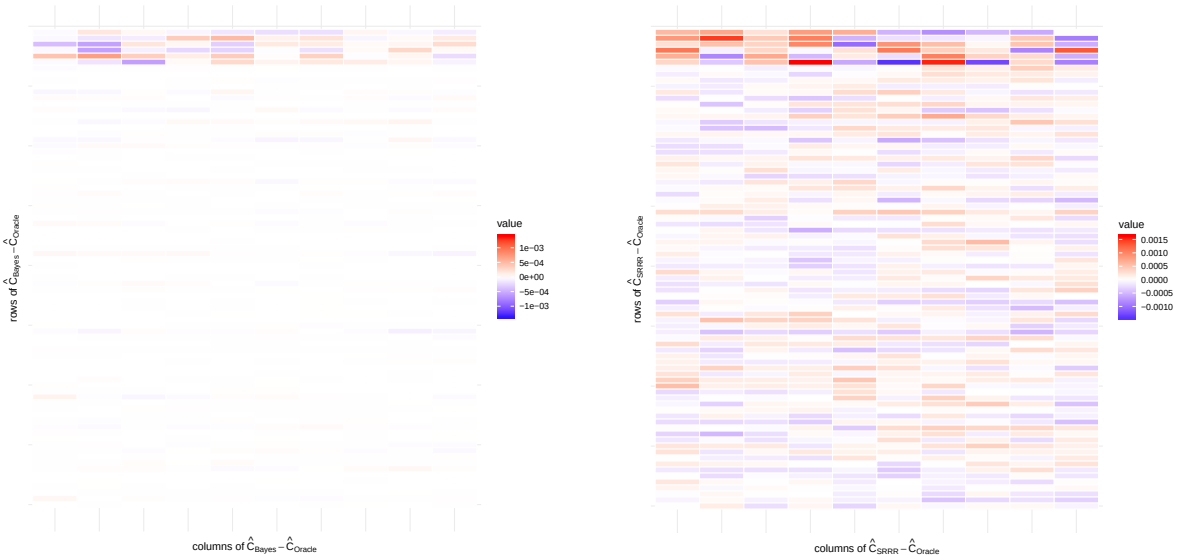
In living organisms, transcription factors (TFs) are DNA-binding proteins that modulate gene expression. Identifying TFs that are relevant regulators of gene expression in a cell cycle is crucial to understanding the transcriptional regulatory network. However, the identification of TFs is a challenging subject of research owing to the lack of technology to directly observe the regulatory activity of TFs (Boulesteix and Strimmer, 2005). In recent years, multivariate linear regression has emerged as a powerful tool for estimating the regulatory role of TFs. The multivariate regression model relates the gene expression levels over time (=the response vector) to the connectivity information between TFs and their target genes (=the predictor vector) so that the activity of TFs can be explained by the coefficient matrix estimate. Since there are many potential regulatory TFs, sparse reduced-rank esti-

Table 2.2: Simulation results: average MSEs and standard errors (in parenthesis) over 1,000 Monte Carlo experiments in Scenario 2. MSE_1: average squared difference between the fitted and observed values. MSE_2: Frobenius norm distance between the estimated and the true coefficient matrix. MSE_3: mean squared prediction error between the fitted values and the true mean function. The performance of Fully Bayes is closer to the Oracle than that of CH-SRRR based on all three MSE measures.

case	r	p	Method	MSE_1	$MSE_2 \times 10^5$	$MSE_3 \times 10^2$
(i)	1	80	CH-SRRR	1.0088 (0.0006)	4.2820 (0.0712)	1.0264 (0.0172)
			Oracle	1.0025 (0.0006)	1.6352 (0.0238)	0.3919 (0.0057)
			Fully Bayes	1.0029 (0.0006)	1.8432 (0.0375)	0.4414 (0.0088)
(ii)	1	300	CH-SRRR	1.0142 (0.0006)	1.6532 (0.0269)	1.4874 (0.0246)
			Oracle	1.0032 (0.0006)	0.4320 (0.0064)	0.3878 (0.0057)
			Fully Bayes	1.0048 (0.0009)	0.6087 (0.0770)	0.5484 (0.0709)
(iii)	3	80	CH-SRRR	1.0205 (0.0006)	9.1819 (0.0873)	2.2009 (0.0210)
			Oracle	1.0088 (0.0006)	4.2964 (0.0400)	1.0295 (0.0096)
			Fully Bayes	1.0094 (0.0006)	4.5365 (0.1086)	1.0874 (0.0262)
(iv)	3	300	CH-SRRR	1.0271 (0.0006)	3.1071 (0.0298)	2.7936 (0.0270)
			Oracle	1.0096 (0.0006)	1.1527 (0.0106)	1.0354 (0.0095)
			Fully Bayes	1.0096 (0.0006)	1.1558 (0.0122)	1.0384 (0.0109)
(v)	5	80	CH-SRRR	1.0296 (0.0006)	12.9995 (0.0978)	3.1178 (0.0235)
			Oracle	1.0130 (0.0006)	6.0934 (0.0504)	1.4609 (0.0122)
			Fully Bayes	1.0137 (0.0006)	6.3587 (0.0643)	1.5246 (0.0155)
(vi)	5	300	CH-SRRR	1.0373 (0.0007)	4.2430 (0.0308)	3.8158 (0.0279)
			Oracle	1.0140 (0.0006)	1.6436 (0.0131)	1.4769 (0.0118)
			Fully Bayes	1.0145 (0.0006)	1.7085 (0.0161)	1.5342 (0.0144)

mation is essential for handling the high-dimensionality of the coefficient problems (Chen and Huang, 2012).

In this section, we apply the proposed fully Bayesian SRRR to the yeast cell cycle data which contain the gene expression data of Spellman et al. (1998) and the Chromatin Immunoprecipitation (ChIP) data of Lee et al. (2002). The gene expression data, served as the response matrix $\mathbf{Y}$, contain mRNA expression levels of 542 yeast cell-cycle-regulated genes measured every 7 minutes during 119 minutes with a total of 18 time points, i.e., $\mathbf{Y}$ is a 542×18 matrix. The ChIP data, served as the design matrix $\mathbf{X}$, contain binding information between 106 transcription factors and the 542 genes, i.e., $\mathbf{X}$ is a 542×106 matrix. The data are publicly available in the `R` package `spls`. We run the proposed MCMC algorithm for 40,000 iterations after a burn-in period of 20,000 iterations.



(a) Fully Bayes vs Oracle (b) CH-SRRR vs Oracle

Figure 2.1: (a) Heatmap of the average element-wise difference between $\hat{\mathbf{C}}_{\text{Bayes}}$ (fully Bayes) and $\hat{\mathbf{C}}_{\text{oracle}}$ (oracle) for $\mathbf{C}$; (b) Heatmap of the average element-wise difference between $\hat{\mathbf{C}}_{\text{SRRR}}$ (CH-SRRR) and $\hat{\mathbf{C}}_{\text{oracle}}$ (oracle) for $\mathbf{C}$. Lighter color in (a) than those in (b) indicates that the performance of the proposed Fully Bayes method is much closer to the oracle than the CH-SRRR.

As a result, we identify 13 TFs (ACE2, MBP1, NDD1, STE12, SWI5, SWI6, GAT3, HIR1, IME4, RME1, ARG81, AZF1, MCM1) whose marginal posterior probabilities of being the relevant TFs are greater than 0.10 (Figure 2.2b). We find that ACE2, MBP1, NDD1, STE12, SWI5, SWI6 and MCM1 have been experimentally verified as yeast cell cycle regulators by Wang et al. (2007). In addition, Lee et al. (2002) have shown that GAT3, HIR1, IME4, RME1, ARG81 and AZF1 are bound to genes encoding other transcriptional regulators. The marginal posterior distribution of the rank, r, is shown in Figure 2.2a. Although the posterior probability is maximized at $r = 4$, the posterior distribution clearly shows that selecting $r = 4$ ignores about 37% chance of $r = 3$ given the data. The dynamic regulation of gene expression by the selected 13 TFs is shown in Figure 2.3.

For comparison with our fully Bayesian approach, we apply the empirical Bayes method that relies on the conditional posterior distribution of $\mathbf{C}$ given the estimates of r and $\boldsymbol{\gamma}$, $\hat{r}$ and $\hat{\boldsymbol{\gamma}}$. By maximizing the marginal posterior distribution of r (Figure 2.2a), we obtain $\hat{r} = 4$. Following the strategy of Barbieri and Berger (2004), we define $\hat{\boldsymbol{\gamma}}$ by using 10 TFs

Table 2.3: Model comparison using LPML and DIC.

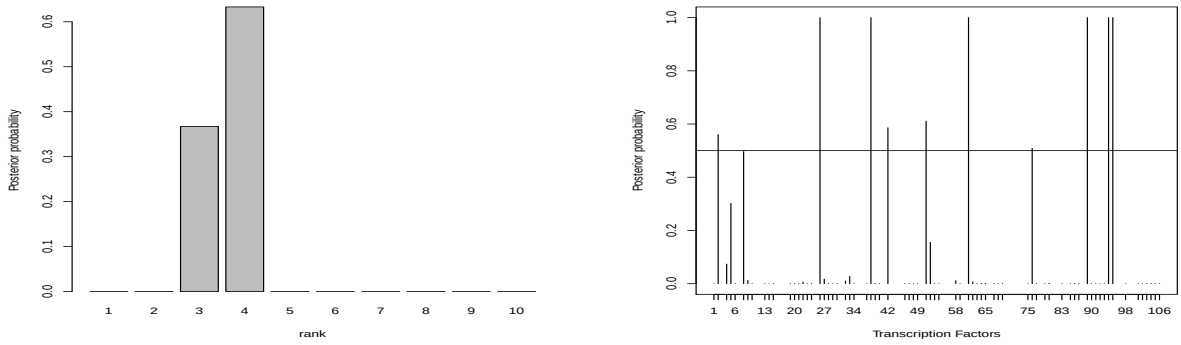
	LPML	DIC
CH-SRRR	-5234.06	9707.14
Empirical Bayes	-854.13	1542.33
Fully Bayes	-846.68	1510.55

(ACE2, MBP1, NDD1, STE12, SWI5, SWI6, GAT3, HIR1, IME4, and RME1) that have marginal posterior inclusion probabilities greater than or equal to 0.5 (Figure 2.2b). Given $\hat{r}$ and $\hat{\boldsymbol{\gamma}}$, the empirical Bayesian inference is performed by iterating Steps 3 – 5 listed in Section 2.2, which is equivalent to Gibbs sampling of Geweke (1996). As a frequentist counterpart of Bayesian methods, CH-SRRR used in Section 2.3 is also applied to the same set. As a result, CH-SRRR selects 81 TFs (out of 106 TFs) as relevant regulators of gene expression and estimates $\hat{r} = 4$ as the optimal rank of the coefficient matrix.

To assess the performance of the three methods (fully Bayes, empirical Bayes, and CH-SRRR), we compute the following log-pseudo marginal likelihood (LPML) which is also known as the Bayesian leave-one-out cross-validation,

$$\text{LPML} = \sum_{i=1}^{n} \log f(\mathbf{y}_i | \mathbf{Y}_{-i}),$$

where $\mathbf{Y}_{-i}$ is a submatrix of $\mathbf{Y}$ obtained by deleting the i-th row, $\mathbf{y}_i$. In addition, we compute the deviance information criterion (DIC), which is the most popular model selection criterion in Bayesian statistics. Note that models with larger LPML and smaller DIC are preferred. Using the formula of Gelfand and Dey (1994), LPML can be computed from MCMC output. Similarly, DIC can be calculated by MCMC samples (Spiegelhalter et al., 2002). Note that CH-SRRR has been developed in a frequentist framework and we compute LPML and DIC of CH-SRRR by applying an empirical Bayes approach under the optimal model selected by CH-SRRR. Table 2.3 shows the results of our model comparison based on LPML and DIC. As the fully Bayesian method has both the largest LPML and the smallest DIC, we conclude that the proposed Bayesian method receives the strongest support from the data for estimating the yeast cell-cycle regulation system.



(a) Estimate of $\mathrm{P}(r \mid \mathbf{Y})$ (b) Estimate of $\mathrm{P}(\gamma_j = 1 \mid \mathbf{Y})$

Figure 2.2: Histogram of MCMC samples over $40,000$ iterations (after $20,000$ burn-in periods).

2.5 Concluding remarks

We have developed a fully-Bayesian approach to sparse and low-rank matrix estimation in a multivariate regression framework. The proposed method offers an effective way to address the uncertainty associated with variable selection and rank selection. As all possible models are integrated out within the proposed MCMC computation, our method is free from the selection of the single best model (or equivalently tuning parameter selection) that is the major challenge in the existing SRRR methods. In addition to the linear dependence due to the low-rank structure, we take the stochastic correlation generated from the error term into account by assigning the inverse-Wishart prior for $\mathbf{\Sigma}$.

Although the proposed MCMC algorithm is relatively efficient and fast, it still merits further research. For example, applying recent developments in Gibbs sampling such as the fast Gibbs sampler (Lucka, 2016) and the recycle Gibbs sampler (Martino et al., 2018) can be a good option to improve the computing speed and accelerate the convergence of MCMC draws. Using a data augmentation approach, many extensions can be achieved under our fully-Bayesian framework. By introducing Gaussian latent variables for multiple binary responses, the proposed method can be adapted to multivariate probit regression (Chib and Greenberg, 1998) and Bayesian support vector machine (Polson and Scott, 2011), which are the topics in Chapters 3 and 4.

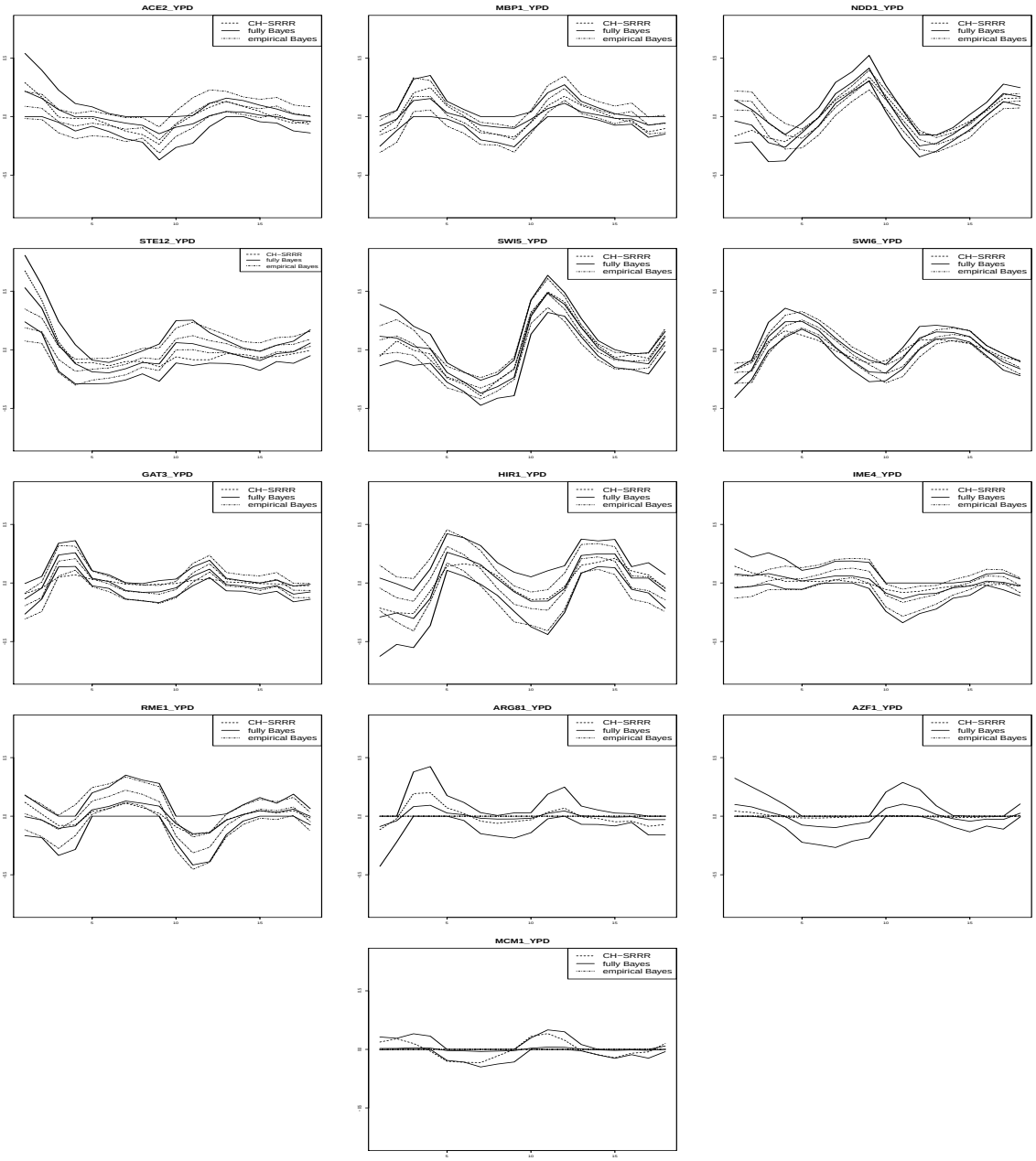


Figure 2.3: Effects of the 13 selected TFs with $\mathrm{P}(\gamma_j = 1|\mathbf{Y}) > 0.1$, where x-axis indicates time points and y-axis indicates coefficient estimates.

Chapter 3

A Bayesian approach to sparse reduced-rank generalized regression models

In this chapter, we extend our Bayesian method developed in Chapter 2 to a general framework that encompasses a wide range of multivariate data sets such as binary, count, or mixed-types. Our motivation is based on the fact that the multiple responses are often correlated and the sparse reduced-rank regression method improves the prediction accuracy. However, due to the use of the nonlinear link function, the computational complexity becomes extremely high. To cope with this computational intensity, we propose to use the Markov chain Monte Carlo model composition approach within the partially collapsed Gibbs sampling scheme. Using a simulation study, we examine the performance of the proposed method for the problem of classification with multivariate binary responses.

This chapter is organized as follows. In Section 3.1, we introduce a general model setting for various types of data and specify the priors for unknown parameters. In Section 3.2, we develop our new Bayesian approach to SRRR under multivariate generalized linear models. In addition, we give technical details of posterior computation. Simulation studies and a real application are presented in Section 3.3 and 3.4. Some remarks are given in Section 3.5.

3.1 Model setup and prior specification

Suppose we observe n independent observations of the response vector $\mathbf{y}_i = (y_{i1}, \cdots, y_{iq})^\top$ and the predictor vector $\mathbf{x}_i = (x_{i1}, \cdots, x_{ip})^\top$ for $i = 1, \cdots, n$. To formulate the relationship between the response and the predictors, the following parametric assumption has been commonly considered:

$$\mathbf{E}(\mathbf{Y} \mid \mathbf{X}, \mathbf{C}, \boldsymbol{\alpha}) = \mathbf{H}(\mathbf{1}\boldsymbol{\alpha}^\top + \mathbf{XC}) = [h(\alpha_k + \mathbf{x}_i^\top \mathbf{c}_k)]_{n \times q}, \tag{3.1}$$

where $\mathbf{Y} = (\mathbf{y}_1, \cdots, \mathbf{y}_n)^\top, \mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_n)^\top, \mathbf{C} = (\mathbf{c}_1, \cdots, \mathbf{c}_q), \mathbf{c}_k = (c_{k1}, \cdots, c_{kp})^\top, \boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_q)^\top$ for $k = 1, \cdots, q$, $\mathbf{1}$ is an $n \times 1$ unit matrix consisting of all 1s, and $h(\cdot)$ is a known non-decreasing inverse-link function. Here, our main goal is to perform inferences about the coefficient matrix $\mathbf{C}$.

In equation (3.1), we assume that the distribution of the outcome belongs to an exponential family and the expectation of the outcome can be given using the link function $h^{-1}(\cdot)$. That is, we assume that the parametric distribution form of $\mathbf{Y}$ given $\mathbf{X}$ and $\mathbf{C}$ is known. For example, when we have binary response variables (i.e., two possible outcomes), we assume that the k-th outcome in subject i, y_{ik} follows the Bernoulli distribution,

$$y_{ik} \overset{ind}{\sim} \mathcal{B}(p_{ik}) \quad \text{with} \quad p_{ik} = P(y_{ik} = 1) = h(\alpha_k + \mathbf{x}_i^\top \mathbf{c}_k), \tag{3.2}$$

where the inverse-link function $h(\cdot)$ is specified by a researcher; e.g., logit, probit, cloglog, Student's t, etc. The likelihood function is given by

$$f(\mathbf{Y} \mid \mathbf{C}, \boldsymbol{\alpha}) = \prod_{k=1}^{q} \prod_{i=1}^{n} \left\{ h(\alpha_k + \mathbf{x}_i^\top \mathbf{c}_k) \right\}^{y_{ik}} \left\{ 1 - h(\alpha_k + \mathbf{x}_i^\top \mathbf{c}_k) \right\}^{1 - y_{ik}}. \tag{3.3}$$

In practice, it is easy to observe that the mean responses are linearly dependent due to the fact that the multiple responses share the common predictors (Anderson, 1951; Izenman, 1975; Velu and Reinsel, 2013). A way of accounting for the linear dependence is to assume that $\mathbf{C}$ does not have full rank, that is, $\text{rank}(\mathbf{C}) = r < \min(p, q)$. Given r, the coefficient

matrix $\mathbf{C}$ can be further decomposed into a product of two full rank matrices, $\mathbf{C} = \mathbf{A}\mathbf{B}^\top$, where $\mathbf{A}$ is a $p \times r$ full rank matrix and $\mathbf{B}$ is a $q \times r$ full rank matrix. Therefore, the model (3.1) can be rewritten as

$$\mathbf{E}(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{\alpha}, \mathbf{A}, \mathbf{B}) = \mathbf{H}(\mathbf{1}\boldsymbol{\alpha}^\top + \mathbf{X}\mathbf{A}\mathbf{B}^\top). \tag{3.4}$$

As discussed in Chapter 2, the decomposition, $\mathbf{C} = \mathbf{A}\mathbf{B}^\top$, is not unique. To construct a unique decomposition, we assume

$$\mathbf{B} = \begin{bmatrix} \mathbf{I}_r \\ \mathbf{F} \end{bmatrix}, \tag{3.5}$$

where $\mathbf{F}$ is a $(q-r) \times r$ full rank matrix. Recall that the number of parameters of $\mathbf{C}$ in model (3.1) is $p \times q$, but it reduces to $(p + q - r) \times r$ in the RRR model (3.4) with our constraint (3.5).

In a high-dimensional regression setting, a necessary procedure is to eliminate irrelevant predictors from the regression model. Under a multivariate regression framework, the variable elimination can be achieved by creating the row-wise sparsity in the coefficient matrix. Recall that row-wise sparsity in $\mathbf{C}$ can be realized by corresponding row-wise sparsity in $\mathbf{A}$. To this end, let $\boldsymbol{\gamma} = (\gamma_1, \cdots, \gamma_p)$ be a $p \times 1$ binary indicator vector that represents active variables, i.e., $\gamma_j = 1$ if the j-th predictor is active and $\gamma_j = 0$ otherwise for $j = 1, \cdots, p$. Let $\mathbf{X}_{\boldsymbol{\gamma}}$ be the $n \times p_{\boldsymbol{\gamma}}$ matrix constructed by selecting columns of $\mathbf{X}$ corresponding to the ones in $\boldsymbol{\gamma}$. Similarly, let $\mathbf{A}_{\boldsymbol{\gamma}}$ be a sub-matrix of $\mathbf{A}$ by selecting rows of $\mathbf{A}$ corresponding to the ones in $\boldsymbol{\gamma}$. Then, given $\boldsymbol{\gamma}$, (3.4) can be reduced to

$$\mathbf{E}(\mathbf{Y} \mid \mathbf{X}_{\boldsymbol{\gamma}}, \boldsymbol{\alpha}, \mathbf{A}_{\boldsymbol{\gamma}}, \mathbf{B}) = \mathbf{H}(\mathbf{1}\boldsymbol{\alpha}^\top + \mathbf{X}_{\boldsymbol{\gamma}}\mathbf{A}_{\boldsymbol{\gamma}}\mathbf{B}^\top). \tag{3.6}$$

While many attempts have been made to extend SRRR, the existing work mainly focuses on continuous data, which is a special case with the identity link function. To fill this research gap, we propose a Bayesian approach to the generalized SRRR framework.

To complete our Bayesian model specification, appropriate priors should be assigned for

the unknown parameters including $\boldsymbol{\gamma}$ and r. In the context of high-dimensional variable selection, it is crucial to impose the assumption that the number of active predictors is less than the sample size. We therefore assign the following prior to $\boldsymbol{\gamma}$:

$$\pi(\boldsymbol{\gamma}) \propto \frac{1}{\binom{p}{p_{\boldsymbol{\gamma}}}} \mathbb{I}(p_{\boldsymbol{\gamma}} < n),$$

where $p_{\boldsymbol{\gamma}} = \sum_{j=1}^{p} \gamma_j$ denotes the number of ones in $\boldsymbol{\gamma}$ and $\mathbb{I}(\cdot)$ is an indicator function. Given $\boldsymbol{\gamma}$, we assume $r \mid \boldsymbol{\gamma} \sim \mathcal{U}\{1, \cdots, \xi_{\boldsymbol{\gamma}}\}$, where $\xi_{\boldsymbol{\gamma}} = \min(p_{\boldsymbol{\gamma}}, q)$. We consider the conjugate Gaussian prior for the intercept,

$$\boldsymbol{\alpha} \sim \mathcal{N}(\mathbf{0}_q, \nu \mathbf{I}_q),$$

where ν is a hyperparameter. Similarly, we assign the conjugate multivariate Gaussian prior for each row of $\mathbf{A}_{\boldsymbol{\gamma}} = [\mathbf{a}_{\boldsymbol{\gamma}1}, \cdots, \mathbf{a}_{\boldsymbol{\gamma}p_{\boldsymbol{\gamma}}}]^\top$ given $\boldsymbol{\gamma}$ and r as follows:

$$\mathbf{a}_{\boldsymbol{\gamma}j} \mid \boldsymbol{\gamma}, r \overset{iid}{\sim} \mathcal{N}_r(\mathbf{0}_r, \nu_1 \mathbf{I}_r), \quad j = 1, \cdots, p_{\boldsymbol{\gamma}},$$

where ν_1 is a hyperparameter. To induce the unique decomposition defined in (3.5), we assume that the first r rows in $\mathbf{B}$ is an identity matrix with probability one. Then, we assign the multivariate Gaussian prior to each row of $\mathbf{F} = [\mathbf{f}_1, \cdots, \mathbf{f}_{q-r}]^\top$:

$$\mathbf{f}_k \mid r \overset{iid}{\sim} \mathcal{N}_r(\mathbf{0}_r, \nu_2 \mathbf{I}_r), \quad k = 1, \cdots, q - r,$$

where ν_2 is a hyperparameter. Note that the use of Gaussian priors leads to a *proper* posterior distribution. Since the dimension of parameter space varies with r and $\boldsymbol{\gamma}$, posterior inference via Gibbs sampling is subject to a trans-dimensional problem. In what follows, we introduce our solution to the trans-dimensional problem under the proposed generalized SRRR framework.

3.2 Posterior inference

Motivated by Yang et al. (2020), we propose to use the partially collapsed Gibbs sampling algorithm (Van Dyk and Park, 2008) that generates the joint posterior sample by iterating the following steps until convergence:

1. Generate r from $\pi(r \mid \mathbf{Y}, \boldsymbol{\gamma}, \boldsymbol{\alpha})$.
2. Generate γ_j and $\mathbf{a}_j$ jointly from $\pi(\gamma_j, \mathbf{a}_j \mid \mathbf{Y}, r, \boldsymbol{\gamma}_{-j}, \mathbf{A}_{-j}, \mathbf{B}, \boldsymbol{\alpha})$.
3. Generate $\mathbf{f}_k$ from $\pi(\mathbf{f}_k \mid \mathbf{y}_k, r, \mathbf{A}, \boldsymbol{\alpha})$ and set $\mathbf{B} = [\mathbf{I}_r, \mathbf{F}^\top]^\top$.
4. Generate $\boldsymbol{\alpha}$ from $\pi(\boldsymbol{\alpha} \mid \mathbf{Y}, \mathbf{A}, \mathbf{B})$.

Although our algorithm is motivated by Yang et al. (2020), their work mainly focused on the Gaussian response so that the calculation can be done in closed-form. However, it does not fit in our situation where a closed-form expression of the full conditional distribution is not available due to analytically intractable integration. To address this problem, we propose to use the Laplace approximation method.

Applying Bayes' theorem, $\pi(r \mid \mathbf{Y}, \boldsymbol{\gamma}, \boldsymbol{\alpha})$ can be computed as

$$\pi(r \mid \mathbf{Y}, \boldsymbol{\gamma}, \boldsymbol{\alpha}) = \frac{f(\mathbf{Y} \mid r, \boldsymbol{\gamma}, \boldsymbol{\alpha})\pi(r)}{\sum_{r'=1}^{\xi_\gamma} f(\mathbf{Y} \mid r', \boldsymbol{\gamma}, \boldsymbol{\alpha})\pi(r')}, \tag{3.7}$$

where

$$f(\mathbf{Y} \mid r, \boldsymbol{\gamma}, \boldsymbol{\alpha}) = \iint f(\mathbf{Y} \mid \mathbf{A}_{\boldsymbol{\gamma}}, \mathbf{B}, r, \boldsymbol{\gamma}, \boldsymbol{\alpha})\pi(\mathbf{A}_{\boldsymbol{\gamma}}, \mathbf{B} \mid r, \boldsymbol{\gamma}, \boldsymbol{\alpha}) d\mathbf{A}_{\boldsymbol{\gamma}} d\mathbf{B}. \tag{3.8}$$

Using the Laplace approximation and ignoring the constant terms, we obtain

$$\log f(\mathbf{Y} \mid r, \boldsymbol{\gamma}, \boldsymbol{\alpha}) \approx \log f(\mathbf{Y} \mid \hat{\mathbf{A}}_{\boldsymbol{\gamma}}, \hat{\mathbf{B}}, r, \boldsymbol{\gamma}, \boldsymbol{\alpha}) - \frac{1}{2}(p_{\boldsymbol{\gamma}} r + qr - r^2) \log n, \tag{3.9}$$

where $\hat{\mathbf{A}}_{\boldsymbol{\gamma}}$ and $\hat{\mathbf{B}}$ are maximum likelihood estimates (MLEs) of $\mathbf{A}_{\boldsymbol{\gamma}}$ and $\mathbf{B}$ with the constraint $\mathbf{B} = [\mathbf{I}_r, \mathbf{F}^\top]^\top$ given r and $\boldsymbol{\gamma}$. Combining (3.7) and (3.9), we have

$$\pi(r \mid \mathbf{Y}, \boldsymbol{\gamma}, \boldsymbol{\alpha}) \propto f(\mathbf{Y} \mid \hat{\mathbf{C}}_{\boldsymbol{\gamma}}^{(r)}, r, \boldsymbol{\gamma}, \boldsymbol{\alpha}) / (n)^{\frac{1}{2}(p_{\boldsymbol{\gamma}} r + qr - r^2)}.$$

Let

$$\text{nbd}(r) = \begin{cases} \{r, r+1\} & \text{if } r = 1 \\ \{r-1, r, r+1\} & \text{if } 1 < r < \xi_{\boldsymbol{\gamma}} \\ \{r-1, r\} & \text{if } r = \xi_{\boldsymbol{\gamma}} \end{cases}.$$

Define the acceptance probability by

$$\min\left\{1, \frac{\#\{\text{nbd}(r)\} f(\mathbf{Y} \mid r', \boldsymbol{\gamma}, \boldsymbol{\alpha})}{\#\{\text{nbd}(r')\} f(\mathbf{Y} \mid r, \boldsymbol{\gamma}, \boldsymbol{\alpha})}\right\}, \tag{3.10}$$

where $f(\mathbf{Y} \mid r, \boldsymbol{\gamma}, \boldsymbol{\alpha})$ and $f(\mathbf{Y} \mid r', \boldsymbol{\gamma}, \boldsymbol{\alpha})$ can be obtained using Laplace approximation in (3.9), and $\#\{\cdot\}$ denotes the number of elements in a set. Then, Step 1 can be implemented as follows: (i) Generate r' from $\text{nbd}(r)$ using a simple random sampling. (ii) Accept $r^{(t+1)} = r'$ with probability (3.10), and otherwise, set $r^{(t+1)} = r^{(t)}$.

Yang et al. (2020) proposed to generate $\boldsymbol{\gamma}$ from $\pi(\boldsymbol{\gamma} \mid \mathbf{Y}, r, \boldsymbol{\alpha})$. However, this is not applicable in our framework due to the computational burden of marginal likelihood calculations. As an alternative, we update each element of $\boldsymbol{\gamma}$ by sampling from the Bernoulli posterior as follows:

$$\gamma_j \mid \mathbf{Y}, r, \boldsymbol{\gamma}_{-j}, \mathbf{A}_{-j}, \mathbf{B}, \boldsymbol{\alpha} \sim \mathcal{B}(\omega_j),$$

where the success probability ω_j is given by

$$\omega_j = \frac{1}{1 + \frac{f(\mathbf{Y} \mid \gamma_j = 0, r, \mathbf{A}_{-j}, \mathbf{B}, \boldsymbol{\gamma}_{-j}, \boldsymbol{\alpha}) \pi(\gamma_j = 0, \boldsymbol{\gamma}_{-j})}{f(\mathbf{Y} \mid \gamma_j = 1, r, \mathbf{A}_{-j}, \mathbf{B}, \boldsymbol{\gamma}_{-j}, \boldsymbol{\alpha}) \pi(\gamma_j = 1, \boldsymbol{\gamma}_{-j})}}.$$

where $\mathbf{A}_{-j}$ is a sub-matrix of $\mathbf{A}$ without the j-th row and $\boldsymbol{\gamma}_{-j}$ is a sub-vector of $\boldsymbol{\gamma}$ without the j-th element. Note that, the calculation of $\pi(\gamma_j \mid \mathbf{Y}, r, \boldsymbol{\gamma}_{-j}, \mathbf{A}_{-j}, \mathbf{B}, \boldsymbol{\alpha})$ heavily relies only on calculations of $f(\mathbf{Y} \mid \gamma_j = 1, r, \mathbf{A}_{-j}, \mathbf{B}, \boldsymbol{\gamma}_{-j}, \boldsymbol{\alpha})$ and $f(\mathbf{Y} \mid \gamma_j = 0, r, \mathbf{A}_{-j}, \mathbf{B}, \boldsymbol{\gamma}_{-j}, \boldsymbol{\alpha})$. When $\gamma_j = 1$, the Laplace approximation leads to

$$\log f(\mathbf{Y} \mid \gamma_j = 1, r, \mathbf{A}_{-j}, \mathbf{B}, \boldsymbol{\gamma}_{-j}, \boldsymbol{\alpha}) \;\approx\; \log f(\mathbf{Y} \mid \hat{\mathbf{A}}_{\gamma_j=1}, \mathbf{B}, r, \boldsymbol{\gamma}_{-j}, \gamma_j = 1, \boldsymbol{\alpha}) - \frac{r \log n}{2},$$

where $\hat{\mathbf{A}}_{\gamma_j=1}$ is the maximum likelihood estimate of $\mathbf{A}$ under the condition $\gamma_j = 1$. Given $\mathbf{A}_{-j}$, the j-th row in $\hat{\mathbf{A}}_{\gamma_j=1}$ is calculated as follows:

$$\hat{\mathbf{a}}_j = \arg\max_{\mathbf{a}_j} f(\mathbf{Y} \mid r, \boldsymbol{\gamma}, \mathbf{A}_{\boldsymbol{\gamma}}, \mathbf{B}, \boldsymbol{\alpha}) \quad \text{subject to} \quad \gamma_j = 1.$$

If $\gamma_j = 0$, that is, the j-th predictor is inactive so that the j-th row of $\mathbf{A}$ is set to 0. The likelihood therefore can be easily computed as

$$\log f(\mathbf{Y} \mid \gamma_j = 0, r, \mathbf{A}_{-j}, \mathbf{B}, \boldsymbol{\gamma}_{-j}, \boldsymbol{\alpha}) \approx \log f(\mathbf{Y} \mid \hat{\mathbf{A}}_{\gamma_j=0}, \mathbf{B}, r, \boldsymbol{\gamma}_{-j}, \gamma_j = 0, \boldsymbol{\alpha}),$$

where $\hat{\mathbf{A}}_{\gamma_j=0}$ is obtained simply by setting the j-th row to zero. According to our constraint (3.5), the coefficient matrix $\mathbf{C}$ can be expressed as $\mathbf{C} = \mathbf{A}\mathbf{B}^\top = \mathbf{A}[\mathbf{I}_r, \mathbf{F}^\top] = [\mathbf{A}, \mathbf{A}\mathbf{F}^\top]$. Hence, $\mathbf{F}^\top$ can be computed as $\mathbf{F}^\top = (\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top\mathbf{C}_{[\ ,(r+1):q]}$, where $\mathbf{C}_{[\ ,(r+1):q]}$ denotes a sub-matrix of $\mathbf{C}$ consisting of the $(r+1)$-th to q-th columns. The matrix $\mathbf{B}$ is then obtained as $\mathbf{B} = [\mathbf{I}_r, \mathbf{F}^\top]^\top$.

In each MCMC iteration, sampling for each row of $\mathbf{A}$ depends on the status of γ_j. If $\gamma_j = 0$ is given, then the j-th row in $\mathbf{A}$ is set to exact zero. If $\gamma_j = 1$ is given, we then draw a sample from $\pi(\mathbf{a}_j \mid \mathbf{Y}, r, \boldsymbol{\gamma}, \mathbf{A}_{-j}, \mathbf{B}, \boldsymbol{\alpha})$, which cannot be expressed in closed form. To address such issue, we propose to generate the j-th row of $\mathbf{A}$ from the following normal approximation:

$$\mathbf{a}_j \sim \mathcal{N}_r(\hat{\mathbf{a}}_j, \hat{\boldsymbol{\Sigma}}_j), \tag{3.11}$$

where

$$\begin{aligned}
\hat{\mathbf{a}}_j &= \arg\max_{\mathbf{a}_j} f(\mathbf{Y} \mid \boldsymbol{\gamma}_{-j}, \gamma_j = 1, r, \mathbf{A}, \mathbf{B}, \boldsymbol{\alpha})\pi(\mathbf{a}_j), \\
\hat{\boldsymbol{\Sigma}}_j &= \left\{ -\frac{\partial^2}{\partial a_{jk}\partial a_{jl}} \log f(\mathbf{Y} \mid \boldsymbol{\gamma}_{-j}, \gamma_j = 1, r, \mathbf{A}, \mathbf{B}, \boldsymbol{\alpha})\pi(\mathbf{a}_j)|_{\mathbf{a}_j=\hat{\mathbf{a}}_j} \right\}^{-1}.
\end{aligned}$$

Note that

$$p(\mathbf{Y} \mid \mathbf{X}\mathbf{C}, \boldsymbol{\alpha}) = \prod_{k=1}^{q} p(\mathbf{y}_k \mid \mathbf{X}\mathbf{c}_k, \boldsymbol{\alpha}), \tag{3.12}$$

where $\mathbf{y}_k$ is the k-th column in $\mathbf{Y}$. With the constraint $\mathbf{B}^\top = [\mathbf{I}_r, \mathbf{F}^\top]^\top$ and $\mathbf{C} = \mathbf{A}\mathbf{B}^\top$, (3.12) can be rewritten as follows:

$$p(\mathbf{Y} \mid \mathbf{X}\mathbf{C}, \boldsymbol{\alpha}) = p(\mathbf{Y} \mid \mathbf{X}\mathbf{A}[\mathbf{I}_r, \mathbf{F}^\top], \boldsymbol{\alpha}) \propto \prod_{k=r+1}^{q} p(\mathbf{y}_k \mid \mathbf{X}^*\mathbf{b}_k, \boldsymbol{\alpha}), \tag{3.13}$$

where $\mathbf{X}^* = \mathbf{X}\mathbf{A}$ and $\mathbf{b}_k$ is the k-th row in $\mathbf{B}$. Equation (3.13) implies that sampling of matrix $\mathbf{B}$ can be done independently for each row. Hence, using the normal approximation, we generate $\mathbf{b}_k$ from

$$\mathbf{b}_k \sim \mathcal{N}_r(\hat{\mathbf{b}}_k, \hat{\boldsymbol{\Sigma}}_k), \quad k = r+1, \cdots, q, \tag{3.14}$$

where

$$\begin{aligned} \hat{\mathbf{b}}_k &= \arg\max_{\mathbf{b}_k} f(\mathbf{y}_k \mid r, \mathbf{A}, \mathbf{B}, \boldsymbol{\alpha})\pi(\mathbf{b}_k), \\ \hat{\boldsymbol{\Sigma}}_k &= \left\{ -\frac{\partial^2}{\partial b_{kl}\partial b_{km}} \log f(\mathbf{y}_k \mid r, \mathbf{A}, \mathbf{B}, \boldsymbol{\alpha})\pi(\mathbf{b}_k)|_{\mathbf{b}_k=\hat{\mathbf{b}}_k} \right\}^{-1}. \end{aligned}$$

Similarly, we generate the intercept term $\boldsymbol{\alpha}$ from

$$\boldsymbol{\alpha} \sim \mathcal{N}_q(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\alpha}}), \tag{3.15}$$

where

$$\begin{aligned} \hat{\boldsymbol{\alpha}} &= \arg\max_{\boldsymbol{\alpha}} f(\mathbf{Y} \mid \mathbf{A}, \mathbf{B})\pi(\boldsymbol{\alpha}), \\ \hat{\boldsymbol{\Sigma}}_{\boldsymbol{\alpha}} &= \left\{ -\frac{\partial^2}{\partial \boldsymbol{\alpha}\partial \boldsymbol{\alpha}^\top} \log f(\mathbf{Y} \mid \mathbf{A}, \mathbf{B})\pi(\boldsymbol{\alpha})|_{\boldsymbol{\alpha}=\hat{\boldsymbol{\alpha}}} \right\}^{-1}. \end{aligned}$$

3.3 Simulation study

In this section, we perform a simulation study to illustrate the proposed method and compare it with six related methods that were proposed in the literature for variable selection in a binary regression framework.

3.3.1 Related methods

The first method is the so-called *Oracle* method in which the rank is fixed at the true value and the true zero coefficients are forced to be zero. The remaining non-zero coefficients are generated as in Steps 2-4 in Section 3.2 using (3.11), (3.14) and (3.15). Consequently, this method is expected to be the best among all the methods.

Another method is to apply separate maximum likelihood estimates using only the active predictors (MLE full rank true sparsity). This method ignores the possible interrelation between the responses and fits each of them separately. This method is considered to show that selecting variables without a low-rank assumption is inefficient.

We also consider four penalized likelihood methods. Each method is implemented for each of the response separately (full rank). The Ridge method uses the L_2 penalty to shrink the magnitude of all coefficients. However, it neither achieves exact zero estimate nor takes care of the low-rank constraint on $\mathbf{C}$. The Lasso (Tibshirani, 1996) method is able to obtain real sparsity by using the L_1 penalty. The SCAD (Fan and Li, 2001) and MCP (Zhang et al., 2010) methods are fitted for each of the response separately using the smoothly clipped absolute deviation penalty and the minimax concave penalty, respectively. Ridge and Lasso are implemented by `R` package `glmnet`. SCAD and MCP are performed by `R` package `ncvreg`. We determine the optimal tuning parameters using 10-fold cross-validations.

3.3.2 Simulation setups

In the simulation, we set the first $p_0 = 4$ rows in $\mathbf{C}$ to be nonzero and therefore $\boldsymbol{\gamma} = \{1, 2, 3, 4\}$. The nonzero part of the coefficient matrix $\mathbf{C}_{\boldsymbol{\gamma}}$ is generated as $\mathbf{C}_{\boldsymbol{\gamma}} = \mathbf{A}_{\boldsymbol{\gamma}}\mathbf{B}^\top$, where $\mathbf{A}_{\boldsymbol{\gamma}} \in \mathbb{R}^{p_0 \times r}$ is an orthogonal matrix from the QR decomposition of a random $p_0 \times r$ matrix filled with $\mathcal{N}(0, 1)$ entries, and all entries in $\mathbf{B} \in \mathbb{R}^{q \times r}$ are independent and identically distributed (iid) random samples from $\mathcal{U}(-1.5, 1.5)$. The rest $p - p_0$ rows in $\mathbf{C}$ are set to be zero. The predictor matrix is conducted by generating its entries as iid random samples from normal distribution $\mathcal{N}(0, 2.5^2)$. The intercept term is defined as $\boldsymbol{\alpha} = (0.5, \cdots, 0.5)^\top$. The latent probability is then modeled with $p_{ik} = h(\mathbf{x}_i^\top \mathbf{c}_k + \alpha_k)$, where $h(x) = 1/\{1 + \exp(-x)\}$.

The response matrix is generated from model (3.2), i.e., $y_{ik} \overset{ind}{\sim} \mathcal{B}(p_{ik})$ for $i = 1, \cdots, n$, and $k = 1, \cdots, q$. The hyperparameters are intentionally set to large values as $\nu = \nu_1 = \nu_2 = 1000$ to reflect the noninformative characteristic of the priors.

We consider the following four cases to cover both low-dimensional and high-dimensional cases:

i. $n = 200, p = 100, q = 10, r = 2$.

ii. $n = 200, p = 300, q = 10, r = 2$.

iii. $n = 200, p = 100, q = 20, r = 2$.

iv. $n = 200, p = 300, q = 20, r = 2$.

The performance is measured by the mean squared error (MSE) via calculating the Frobenius norm as follows:

$$\text{MSE} \quad = \quad \|\mathbf{C} - \hat{\mathbf{C}}\|_F^2 / pq.$$

After obtaining the estimated coefficients for each method, we generate $\mathbf{X}_{\text{test}}$ and $\mathbf{Y}_{\text{test}}$ to measure the prediction accuracy. The test data are independently generated using the same settings with $n_{\text{test}} = 150$. We calculate the expected cross-entropy loss using the test dataset as follows:

$$-\left[\sum_i \sum_k \left(y_{ik} \log p_{ik} + (1 - y_{ik}) \log(1 - p_{ik})\right)\right] / n_{\text{test}} q,$$

where $p_{ik} = h(\mathbf{x}_i^\top \hat{\mathbf{c}}_k)$. To assess the prediction accuracy for classification, we calculate true negative rate (TNR), false negative rate (FNR), false positive rate (FPR), and true positive rate (TPR), representing the percentage of true zeros, false zeros, false non-zeros, and true non-zeros, respectively.

Table 3.1: Simulation results: average MSE, cross-entropy loss and standard errors (in parenthesis) over 100 Monte Carlo experiments.

case	q	p	Method	cross-entropy	MSE $\times 10^4$
(i)	10	100	Proposed	0.3996 (0.0034)	6.2900 (0.3444)
			Oracle Bayes	0.3994 (0.0034)	6.1660 (0.3525)
			MLE full rank true sparsity	0.4049 (0.0035)	7.6458 (0.3803)
			Ridge full rank	0.5757 (0.0019)	102.5263 (2.2433)
			Lasso full rank	0.4434 (0.0033)	37.2917 (1.0017)
			SCAD full rank	0.4229 (0.0036)	15.7505 (0.5626)
			MCP full rank	0.4210 (0.0037)	14.1090 (0.5850)
(ii)	10	300	Proposed	0.3987 (0.0036)	2.3946 (0.2287)
			Oracle Bayes	0.3983 (0.0036)	2.3179 (0.2308)
			MLE full rank true sparsity	0.4035 (0.0037)	2.8424 (0.2499)
			Ridge full rank	0.6604 (0.0006)	46.2929 (1.0166)
			Lasso full rank	0.4572 (0.0031)	15.2124 (0.3876)
			SCAD full rank	0.4301 (0.0037)	6.2646 (0.2872)
			MCP full rank	0.4267 (0.0037)	6.0875 (0.8240)
(iii)	20	100	Proposed	0.3961 (0.0026)	5.5916 (0.2276)
			Oracle Bayes	0.3960 (0.0026)	5.3962 (0.2206)
			MLE full rank true sparsity	0.4017 (0.0027)	7.4524 (0.3088)
			Ridge full rank	0.5774 (0.0013)	103.8881 (1.4992)
			Lasso full rank	0.4425 (0.0022)	37.9412 (0.7064)
			SCAD full rank	0.4212 (0.0025)	15.9453 (0.5856)
			MCP full rank	0.4192 (0.0026)	13.4795 (0.2997)
(iv)	20	300	Proposed	0.3947 (0.0025)	1.8535 (0.0878)
			Oracle Bayes	0.3944 (0.0025)	1.8069 (0.0891)
			MLE full rank true sparsity	0.4003 (0.0025)	2.4900 (0.1056)
			Ridge full rank	0.6604 (0.0005)	46.7077 (0.6545)
			Lasso full rank	0.4535 (0.0023)	15.3951 (0.2561)
			SCAD full rank	0.4280 (0.0026)	7.0262 (0.4399)
			MCP full rank	0.4229 (0.0025)	5.6698 (0.1448)

3.3.3 Simulation results

The simulation results are based on $1,000$ MCMC samples (after $1,000$ burn-in iterations) over 100 Monte Carlo replications. MSEs are summarized in Table 3.1. It shows that our proposed method produces considerably smaller values of the average MSE comparing to the penalized regression methods and MLE with true sparsity. This indicates that our method does a reasonably good job in both variable selection and rank reduction. Note that the Ridge method performs the worst and this demonstrates that variable selection and rank

Table 3.2: Simulation results: average percentage of variable selection accuracy and standard errors (in parenthesis) over 100 Monte Carlo experiments.

				elementwise selection accuracy	
case	q	p	Method	active variables	inactive variables
(i)	10	100	Proposed	0.9800 (0.0068)	0.9996 (0.0002)
			LASSO	0.8315 (0.0078)	0.9044 (0.0025)
			SCAD	0.8250 (0.0081)	0.9277 (0.0016)
			MCP	0.8030 (0.0079)	0.9655 (0.0009)
(ii)	10	300	Proposed	0.9750 (0.0075)	0.9998 (0.0001)
			LASSO	0.8040 (0.0073)	0.9571 (0.0012)
			SCAD	0.8120 (0.0073)	0.9590 (0.0008)
			MCP	0.7812 (0.0076)	0.9838 (0.0004)
(iii)	20	100	Proposed	0.9950 (0.0035)	0.9997 (0.0002)
			LASSO	0.8300 (0.0067)	0.9041 (0.0018)
			SCAD	0.8240 (0.0066)	0.9275 (0.0012)
			MCP	0.7981 (0.0069)	0.9650 (0.0007)
(iv)	20	300	Proposed	0.9900 (0.0049)	0.9998 (0.0001)
			LASSO	0.8002 (0.0072)	0.9577 (0.0008)
			SCAD	0.8042 (0.0070)	0.9600 (0.0006)
			MCP	0.7776 (0.0072)	0.9840 (0.0003)

reduction both are important. Furthermore, it is worth noting that our proposed method is comparable to the Oracle Bayes method while all the full rank methods perform relatively poorly. This demonstrates that implementing rank reduction is essential in the presence of reduced-rank structure and that solely performing the variable selection is not enough.

For the purpose of variable selection comparison, we measure the accuracy of variable selection for the proposed method and three penalized regression methods which can produce exact zero coefficients to induce sparsity. The accuracy is measured separately for both active and inactive predictors. The numbers in Table 3.2 represent the percentages of active/inactive predictors that are correctly selected/deselected. For the proposed method, the predictor is treated as being selected if $P(\gamma_j = 1 \mid \mathbf{Y}) \geq 0.5$. As a result, our proposed method has an outstanding variable selection accuracy. Table 3.3 reports the true positive rate, true negative rate, false positive rate, and false negative rate for each method under each case. The results clearly show that our method is comparable to the Oracle Bayes method and always performs better than other methods regarding true negative rates and

Table 3.3: Simulation results: average true negative rate (TNR), false negative rate (FNR), false positive rate (FPR), true positive rate (TPR), and standard errors (in parenthesis) over 100 Monte Carlo experiments. Methods (1)-(7): (1) Proposed, (2) Oracle Bayes, (3) MLE full rank true sparsity, (4) Ridge full rank, (5) Lasso full rank, (6) SCAD full rank, (7) MCP full rank.

case	Method	TNR	FNR	FPR	TPR
(i)	(1)	0.7503(0.0047)	0.1430(0.0018)	0.2497(0.0047)	0.8570(0.0018)
	(2)	0.7510(0.0048)	0.1432(0.0018)	0.2490(0.0048)	0.8568(0.0018)
	(3)	0.7499(0.0047)	0.1468(0.0019)	0.2501(0.0047)	0.8532(0.0019)
	(4)	0.5242(0.0067)	0.1576(0.0025)	0.4758(0.0067)	0.8424(0.0025)
	(5)	0.7022(0.0056)	0.1376(0.0019)	0.2978(0.0056)	0.8624(0.0019)
	(6)	0.7202(0.0057)	0.1449(0.0020)	0.2798(0.0057)	0.8551(0.0020)
	(7)	0.7235(0.0055)	0.1456(0.0019)	0.2765(0.0055)	0.8544(0.0019)
(ii)	(1)	0.7510(0.0048)	0.1441(0.0017)	0.2490(0.0048)	0.8559(0.0017)
	(2)	0.7515(0.0049)	0.1438(0.0018)	0.2485(0.0049)	0.8562(0.0018)
	(3)	0.7509(0.0048)	0.1472(0.0018)	0.2491(0.0048)	0.8528(0.0018)
	(4)	0.2351(0.0076)	0.0977(0.0044)	0.7649(0.0076)	0.9023(0.0044)
	(5)	0.6855(0.0059)	0.1369(0.0019)	0.3145(0.0059)	0.8631(0.0019)
	(6)	0.7105(0.0059)	0.1458(0.0019)	0.2895(0.0059)	0.8542(0.0019)
	(7)	0.7160(0.0057)	0.1455(0.0019)	0.2840(0.0057)	0.8545(0.0019)
(iii)	(1)	0.7600(0.0032)	0.1461(0.0014)	0.2400(0.0032)	0.8539(0.0014)
	(2)	0.7596(0.0032)	0.1458(0.0014)	0.2404(0.0032)	0.8542(0.0014)
	(3)	0.7583(0.0031)	0.1495(0.0014)	0.2417(0.0031)	0.8505(0.0014)
	(4)	0.5248(0.0047)	0.1620(0.0022)	0.4752(0.0047)	0.8380(0.0022)
	(5)	0.7091(0.0039)	0.1396(0.0015)	0.2909(0.0039)	0.8604(0.0015)
	(6)	0.7285(0.0038)	0.1468(0.0016)	0.2715(0.0038)	0.8532(0.0016)
	(7)	0.7312(0.0037)	0.1471(0.0015)	0.2688(0.0037)	0.8529(0.0015)
(iv)	(1)	0.7590(0.0034)	0.1455(0.0014)	0.2410(0.0034)	0.8545(0.0014)
	(2)	0.7595(0.0034)	0.1457(0.0015)	0.2405(0.0034)	0.8543(0.0015)
	(3)	0.7583(0.0033)	0.1494(0.0014)	0.2417(0.0033)	0.8506(0.0014)
	(4)	0.2360(0.0053)	0.0969(0.0031)	0.7640(0.0053)	0.9031(0.0031)
	(5)	0.6946(0.0044)	0.1373(0.0017)	0.3054(0.0044)	0.8627(0.0017)
	(6)	0.7176(0.0041)	0.1455(0.0017)	0.2824(0.0041)	0.8545(0.0017)
	(7)	0.7231(0.0041)	0.1461(0.0017)	0.2769(0.0041)	0.8539(0.0017)

false positive rates. Note that the Ridge method performs extremely poorly in true negative rates and false positive rates, especially in high-dimensional settings. This coincides with the fact that Ridge regression is not able to do the variable selection so that it over-select the predictors. By re-using the statistics in Table 3.3, we can easily compute the precision, accuracy, as well as F1 score, and they are reported in Table 3.4. In addition, we compute

Table 3.4: Simulation results: average precision, accuracy, F1 score, and standard errors (in parenthesis) over 100 Monte Carlo experiments. Methods (1)-(7): (1) Proposed, (2) Oracle Bayes, (3) MLE full rank true sparsity, (4) Ridge full rank, (5) Lasso full rank, (6) SCAD full rank, (7) MCP full rank.

case	Method	precision	accuracy	F1-score
(i)	(1)	0.8169(0.0027)	0.8106(0.0022)	0.8361(0.0017)
	(2)	0.8173(0.0027)	0.8108(0.0022)	0.8363(0.0017)
	(3)	0.8159(0.0027)	0.8083(0.0023)	0.8339(0.0018)
	(4)	0.6970(0.0024)	0.7039(0.0022)	0.7622(0.0014)
	(5)	0.7904(0.0029)	0.7928(0.0024)	0.8244(0.0017)
	(6)	0.7993(0.0030)	0.7965(0.0024)	0.8258(0.0017)
	(7)	0.8010(0.0029)	0.7976(0.0024)	0.8264(0.0018)
(ii)	(1)	0.8170(0.0027)	0.8103(0.0023)	0.8357(0.0018)
	(2)	0.8173(0.0027)	0.8107(0.0024)	0.8360(0.0018)
	(3)	0.8163(0.0027)	0.8085(0.0024)	0.8339(0.0018)
	(4)	0.6039(0.0018)	0.6112(0.0017)	0.7228(0.0016)
	(5)	0.7811(0.0029)	0.7859(0.0025)	0.8197(0.0018)
	(6)	0.7933(0.0031)	0.7917(0.0025)	0.8222(0.0018)
	(7)	0.7965(0.0030)	0.7943(0.0025)	0.8241(0.0018)
(iii)	(1)	0.8218(0.0018)	0.8130(0.0016)	0.8374(0.0012)
	(2)	0.8216(0.0018)	0.8130(0.0016)	0.8374(0.0012)
	(3)	0.8201(0.0018)	0.8103(0.0016)	0.8349(0.0013)
	(4)	0.6956(0.0016)	0.7015(0.0015)	0.7599(0.0010)
	(5)	0.7933(0.0020)	0.7945(0.0016)	0.8252(0.0012)
	(6)	0.8031(0.0020)	0.7989(0.0016)	0.8272(0.0012)
	(7)	0.8046(0.0020)	0.7999(0.0016)	0.8278(0.0012)
(iv)	(1)	0.8215(0.0019)	0.8130(0.0015)	0.8375(0.0012)
	(2)	0.8217(0.0019)	0.8131(0.0015)	0.8375(0.0012)
	(3)	0.8203(0.0018)	0.8104(0.0015)	0.8350(0.0012)
	(4)	0.6046(0.0013)	0.6122(0.0013)	0.7239(0.0011)
	(5)	0.7859(0.0021)	0.7896(0.0018)	0.8222(0.0013)
	(6)	0.7972(0.0021)	0.7950(0.0017)	0.8246(0.0013)
	(7)	0.8002(0.0021)	0.7970(0.0017)	0.8260(0.0012)

the KS statistic and AUC to further prove the superiority of the proposed method. Under the multivariate regression framework, we calculate the KS statistic and AUC for each response variable and then compute the average. As shown in Table 3.4 and 3.5, our proposed method is slightly worse than the Oracle Bayes and shows a comparable performance with all other penalized regression methods.

Table 3.5: Simulation results: average KS statistic, AUC, and standard errors (in parenthesis) over 100 Monte Carlo experiments.

case	Method	KS statistic	AUC
(i)	Proposed	0.6289(0.0046)	0.8845(0.0026)
	Oracle Bayes	0.6291(0.0046)	0.8847(0.0026)
	MLE full rank true sparsity	0.6233(0.0047)	0.8820(0.0026)
	Ridge full rank	0.4254(0.0047)	0.7696(0.0030)
	Lasso full rank	0.5948(0.0051)	0.8668(0.0029)
	SCAD full rank	0.6007(0.0051)	0.8697(0.0029)
	MCP full rank	0.6024(0.0052)	0.8707(0.0029)
(ii)	Proposed	0.6292(0.0049)	0.8846(0.0027)
	Oracle Bayes	0.6295(0.0049)	0.8849(0.0027)
	MLE full rank true sparsity	0.6257(0.0049)	0.8827(0.0027)
	Ridge full rank	0.2895(0.0036)	0.6772(0.0024)
	Lasso full rank	0.5826(0.0053)	0.8606(0.0031)
	SCAD full rank	0.5903(0.0055)	0.8649(0.0031)
	MCP full rank	0.5925(0.0054)	0.8663(0.0030)
(iii)	Proposed	0.6345(0.0032)	0.8877(0.0017)
	Oracle Bayes	0.6345(0.0032)	0.8877(0.0017)
	MLE full rank true sparsity	0.6286(0.0032)	0.8850(0.0018)
	Ridge full rank	0.4254(0.0031)	0.7687(0.0019)
	Lasso full rank	0.5988(0.0035)	0.8690(0.0020)
	SCAD full rank	0.6064(0.0034)	0.8726(0.0020)
	MCP full rank	0.6072(0.0034)	0.8732(0.0020)
(iv)	Proposed	0.6350(0.0032)	0.8881(0.0018)
	Oracle Bayes	0.6354(0.0032)	0.8882(0.0018)
	MLE full rank true sparsity	0.6302(0.0033)	0.8855(0.0018)
	Ridge full rank	0.2911(0.0029)	0.6777(0.0019)
	Lasso full rank	0.5910(0.0037)	0.8653(0.0021)
	SCAD full rank	0.5985(0.0037)	0.8690(0.0021)
	MCP full rank	0.6018(0.0037)	0.8706(0.0021)

3.4 Case-study: yeast cell cycle data

Transcription factors (TFs) play a vital role for interpreting the mechanism of cell cycle regulation. In this section, we analyze the *yeast cell cycle* data (Chun and Keleş, 2010) as we used in Section 2.4. The response matrix $\mathbf{Y}$ consists of 542 cell-cycle-regulated genes from an α factor arrest method, where the rows and columns correspond to genes and mRNA levels measured at every 7 min during 119 min, respectively, i.e., $n = 542$ and $q = 18$. The 542×106 predictor matrix $\mathbf{X}$ includes the binding information of the target genes for a total

of 106 TFs. Since the data have already been normalized, we choose 0 as the threshold and transform the response matrix $\mathbf{Y}$ to be binary, namely $\mathbf{Y}^* = (y_{ik})_{n\times q}$ such that $y^*_{ik}\mathbb{I}(y_{ik} \geq 0)$.

We randomly select 80% of the data as the training set and the rest 20% as the test set. We use the training set to fit the SRRR model using the proposed Bayesian method. To further demonstrate the importance of the reduced-rank, we also implement our proposed method with a fixed full rank where $q = 18$. In addition, we also consider the MLE as well as the other four penalized regression methods (Ridge, Lasso, SCAD, MCP) which have been mentioned in Section 3.3. The summary of the results based on 100 replications is shown in Tables 3.6 to 3.8. As a result, the proposed method always select the rank to be 2. Our proposed method with full rank selects the least number of TFs (approximately 4 TFs), and the proposed method selects around 23 TFs. For the penalized regression methods which can do variable selection, the lasso selects the largest number of TFs, 100 out of 106. Our proposed method outperforms all other methods with respect to precision, accuracy, F1 score, KS statistic and AUC which indicates that simultaneous variable selection and rank reduction truly increases the prediction performance.

Table 3.6: Average true positive rate (TPR), true negative rate (TNR), false positive rate (FPR), false negative rate (FNR), and standard errors (in parenthesis) over 100 replications.

	TPR	TNR	FPR	FNR
Proposed	0.6135 (0.0017)	0.6779 (0.0018)	0.3221 (0.0018)	0.3865 (0.0017)
Proposed full rank	0.6107 (0.0016)	0.6646 (0.0019)	0.3354 (0.0019)	0.3893 (0.0016)
MLE	0.6016 (0.0018)	0.6324 (0.0017)	0.3676 (0.0017)	0.3984 (0.0018)
Ridge	0.6102 (0.0018)	0.6665 (0.0019)	0.3335 (0.0019)	0.3898 (0.0018)
Lasso	0.6047 (0.0018)	0.6790 (0.0019)	0.3210 (0.0019)	0.3953 (0.0018)
SCAD	0.6030 (0.0022)	0.6737 (0.0020)	0.3263 (0.0020)	0.3970 (0.0022)
MCP	0.5989 (0.0020)	0.6708 (0.0020)	0.3292 (0.0020)	0.4011 (0.0020)

3.5 Concluding remarks

We have developed a Bayesian approach to sparse low-rank matrix estimation in a multivariate generalized regression framework. The proposed method provides an effective way to handle both rank reduction and variable selection simultaneously for various types of

Table 3.7: Average precision, accuracy, F1 score, and standard errors (in parenthesis) over 100 replications.

	Precision	Accuracy	F1 score
Proposed	0.6431 (0.0018)	0.6465 (0.0015)	0.6278 (0.0016)
Proposed full rank	0.6327 (0.0018)	0.6384 (0.0014)	0.6214 (0.0015)
MLE	0.6076 (0.0016)	0.6174 (0.0015)	0.6045 (0.0015)
Ridge	0.6338 (0.0017)	0.6391 (0.0014)	0.6217 (0.0015)
Lasso	0.6405 (0.0018)	0.6428 (0.0014)	0.6219 (0.0015)
SCAD	0.6361 (0.0018)	0.6393 (0.0014)	0.6189 (0.0017)
MCP	0.6326 (0.0018)	0.6358 (0.0014)	0.6151 (0.0016)

Table 3.8: Average KS statistic, AUC, number of predictors, rank, and standard errors (in parenthesis) over 100 replications.

	KS statistic	AUC	number of predictors	rank
Proposed	0.2875 (0.0027)	0.6676 (0.0019)	22.98 (0.2995)	2 (0)
Proposed full rank	0.2758 (0.0027)	0.6570 (0.0019)	4.32 (0.0566)	18 (0)
MLE	0.2502 (0.0028)	0.6359 (0.0019)	106.00 (0.0000)	18 (0)
Ridge	0.0249 (0.0080)	0.6619 (0.0019)	106.00 (0.0000)	18 (0)
Lasso	0.0242 (0.0078)	0.6606 (0.0019)	100.04 (0.2647)	18 (0)
SCAD	0.0238 (0.0076)	0.6562 (0.0020)	87.02 (0.4228)	18 (0)
MCP	0.0225 (0.0072)	0.6491 (0.0019)	63.92 (0.5617)	18 (0)

responses. The number of parameters can be significantly reduced under the reduced-rank structure and the uncertainty of the model can be also well taken care of.

While the proposed method has been applied to binary data with the logit link function in this chapter, it is also applicable to various types of responses with a variety of link functions. One limitation of the proposed framework is that the implementation could be computationally expensive when the number of parameters is extremely large. Note that we assume the first r columns in the coefficient matrix are independent to construct the unique decomposition as in (3.5). If this assumption is not met, one can simply consider rotations of responses to resolve such issue.

Chapter 4

A reduced-rank approach to multivariate support vector machine

Support vector machine (SVM) is one of the popular classification methods in the machine learning literature. In this chapter, we introduce a multivariate extension of SVM using the RRR approach. The use of the reduced rank assumption allows us to take an advantage of utilizing the interrelationship among multiple binary responses. Using data augmentation of Polson and Scott (2011), Bayesian inference can be performed via Gibbs sampling. In addition, the Bayesian framework permits the notion of Occam's window (Madigan and Raftery, 1994) to account for the model uncertainty associated with the rank selection.

The rest of this chapter is arranged in the following way. In Section 4.1, we discuss the assumption and model settings of the univariate Bayesian SVM. Section 4.2 shows the merit of the reduced-rank structure and how we extend the existing method to the multivariate cases. Section 4.3 introduces the latent variable representation. We also derive the conditional distributions which are needed to implement the Gibbs sampling algorithm to make statistical inferences. Simulation studies are conducted in Section 4.4. Section 4.5 illustrates a real data application of our methods with the spider data from the `R` package `mvabund`. Section 4.6 concludes with some discussion.

4.1 Data augmentation for SVM

Suppose that we observe $\mathbf{y} = (y_1, \cdots, y_n)^\top$ with $y_i \in \{-1, 1\}$ and $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_n)^\top$ with $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^\top$. The SVM is performed by minimizing

$$\Theta(\boldsymbol{\beta}) = \sum_{i=1}^{n} \max(1 - y_i \mathbf{x}_i^\top \boldsymbol{\beta}, 0) + p_\lambda(\boldsymbol{\beta}), \tag{4.1}$$

where the first term is often referred to as the hinge loss, and the second term is called the penalty function (e.g., $p_\lambda(\boldsymbol{\beta}) = \lambda \|\boldsymbol{\beta}\|^2$). From a Bayesian perspective, minimizing equation (4.1) is equivalent to finding the mode of the following pseudo-posterior density:

$$\begin{aligned} \pi(\boldsymbol{\beta} \mid \mathbf{y}) &\propto \pi(\boldsymbol{\beta}) \prod_{i=1}^{n} f_i(y_i \mid \boldsymbol{\beta}) \\ &\propto \pi(\boldsymbol{\beta}) \exp\left\{ -2 \sum_{i=1}^{n} \max(1 - y_i \mathbf{x}_i^\top \boldsymbol{\beta}, 0) \right\}, \end{aligned}$$

where $f_i(y_i \mid \boldsymbol{\beta}) = \exp(-2\max(1 - y_i \mathbf{x}_i^\top \boldsymbol{\beta}, 0))$ and $\pi(\boldsymbol{\beta}) \propto \exp\{-p_\lambda(\boldsymbol{\beta})\}$. The univariate Bayesian SVM framework is originally proposed by Polson and Scott (2011). Their main contribution is to show that

$$\begin{aligned} f_i(y_i \mid \boldsymbol{\beta}) &= \exp\left\{-2\max(1 - y_i \mathbf{x}_i^\top \boldsymbol{\beta}, 0)\right\} \\ &= \int_0^\infty \frac{1}{\sqrt{2\pi\lambda_i}} \exp\left\{ -\frac{1}{2} \frac{(1 + \lambda_i - y_i \mathbf{x}_i^\top \boldsymbol{\beta})^2}{\lambda_i} \right\} d\lambda_i. \end{aligned} \tag{4.2}$$

In practice, it is common to observe that the response variable consists of multivariate outcomes rather than a single value. However, the existing Bayesian SVM is limited to the univariate case. In what follows, we develop a multivariate Bayesian SVM using the reduced-rank approach.

4.2 Reduced-rank SVM

Suppose that we observe $\mathbf{y}_i = (y_{i1}, \cdots, y_{iq})^\top$ which contains q different binary outcomes with $y_{ik} \in \{-1, 1\}$ for $k = 1, \cdots, q$. Let $\mathbf{Y} = (\mathbf{y}_1, \cdots, \mathbf{y}_n)^\top$. Also assume that we observe q design matrices $\mathbf{X}_k = (\mathbf{x}_{1k}, \cdots, \mathbf{x}_{nk})^\top$. Then, the SVM objective function in (4.1) can be extended to

$$\Theta(\mathbf{C}) = \sum_{i=1}^{n} \sum_{q=1}^{k} \max(1 - y_{ik}\mathbf{x}_{ik}^\top \mathbf{c}_k, 0) + p_\lambda(\mathbf{c}_k), \tag{4.3}$$

where $\mathbf{c}_k$ is the k-th column in the $p \times q$ coefficient matrix $\mathbf{C} = (\mathbf{c}_1, \cdots, \mathbf{c}_q)$. Minimizing equation (4.3) is equivalent to finding the mode of the following pseudo-posterior density

$$\begin{aligned} \pi(\mathbf{C} \mid \mathbf{Y}, \mathbf{X}_1, \cdots, \mathbf{X}_q) &\propto \pi(\mathbf{C}) f(\mathbf{Y} \mid \mathbf{X}_1, \cdots, \mathbf{X}_q, \mathbf{C}) \\ &\propto \pi(\mathbf{C}) \exp\left\{ -2 \sum_{i=1}^{n} \sum_{k=1}^{q} \max(1 - y_{ik}\mathbf{x}_{ik}^\top \mathbf{c}_k, 0) \right\}. \end{aligned}$$

Note that the main goal of classification is to make predictions for the future data rather than the observed data.

Suppose that $\text{rank}(\mathbf{C}) = r$. Given r, the coefficient matrix can be decomposed into a product of two full rank matrices, $\mathbf{C} = \mathbf{A}\mathbf{B}^\top$, where $\mathbf{A}$ is a $p \times r$ full rank matrix and $\mathbf{B}$ is a $q \times r$ full rank matrix. For a unique decomposition, we assume that

$$\mathbf{B} = \begin{bmatrix} \mathbf{I}_r \\ \mathbf{F} \end{bmatrix}, \tag{4.4}$$

where $\mathbf{F}$ is a $(q - r) \times r$ full rank matrix.

By following Polson and Scott (2011), we introduce a latent variable λ_{ik} in order to express the pseudo-likelihood as a mixture of normals. For computational convenience, we express the pseudo-likelihood of y_{ik} as a proportional form of normal distribution as follows:

$$\begin{aligned} f(y_{ik} \mid \mathbf{c}_k, \lambda_{ik}) &\propto \lambda_{ik}^{-1/2} \exp\left\{ -\frac{1}{2} \frac{(1 + \lambda_{ik} - y_{ik}\mathbf{x}_{ik}^\top \mathbf{c}_k)^2}{\lambda_{ik}} \right\}, \\ \pi(\lambda_{ik}) &\propto \mathbb{I}(\lambda_{ik} > 0). \end{aligned}$$

We use the noninformative Gaussian prior with large variance for each column in $\mathbf{A}$ and each row in $\mathbf{F}$ as follows:

$$\mathbf{a}_k \sim \mathcal{N}(\mathbf{0}, \nu_1 \mathbf{I}_p), \quad k = 1, \cdots, r,$$
$$\mathbf{f}_k \sim \mathcal{N}(\mathbf{0}, \nu_2 \mathbf{I}_r) \quad k = 1, \cdots, q - r,$$

where ν_1 and ν_2 are prespecified hyperparameters.

4.3 Posterior inference

The greatest merit of Bayesian SVM is that posterior inference can be performed via Gibbs sampling. In this section, we derive the full conditionals and discuss the implementation of the Gibbs sampler under the reduced rank constraint.

4.3.1 Conditional distributions

Full conditional distribution of $\mathbf{f}_k$: Let $\mathbf{x}_{ik}$ be the i-th row in the k-th design matrix. Define $\mathbf{x}^*_{ik} = \mathbf{A}^\top \mathbf{x}_{ik}$, $\mathbf{Z}^*_k = (\mathbf{z}^*_{1k}, \cdots, \mathbf{z}^*_{nk})$ with $\mathbf{z}^*_{ik} = y_{ik}\mathbf{x}^*_{ik}/\sqrt{\lambda_{ik}}$, and $\mathbf{d}_k = (d_{1k}, \cdots, d_{nk})^\top$ with $d_{ik} = (1 + \lambda_{ik})/\lambda_{ik}$. For $k' = k + r$, the full conditional distribution of $\mathbf{f}_k$ is obtained as follows:

$$\begin{aligned}
\pi(\mathbf{f}_k \mid -) &\propto \prod_{i=1}^{n} \exp\left\{ -\frac{1}{2} \frac{(1 + \lambda_{ik'} - y_{ik'} \mathbf{x}_{ik'}^\top \mathbf{c}'_k)^2}{\lambda_{ik'}} \right\} \pi(\mathbf{f}_k) \\
&\propto \prod_{i=1}^{n} \exp\left\{ -\frac{1}{2} \frac{(1 + \lambda_{ik'} - y_{ik'} \mathbf{x}_{ik'}^\top \mathbf{A}\mathbf{f}_k)^2}{\lambda_{ik'}} \right\} \pi(\mathbf{f}_k) \\
&\propto \prod_{i=1}^{n} \exp\left\{ -\frac{\mathbf{f}_k^\top \mathbf{A}^\top \mathbf{x}_{ik'} y_{ik'}^2 \mathbf{x}_{ik'}^\top \mathbf{A}\mathbf{f}_k - 2 y_{ik'} \mathbf{x}_{ik'}^\top \mathbf{A}\mathbf{f}_k (1 + \lambda_{ik'})}{2\lambda_{ik'}} \right\} \pi(\mathbf{f}_k) \\
&= \prod_{i=1}^{n} \exp\left\{ -\frac{\mathbf{f}_k^\top \mathbf{z}^*_{ik'} \mathbf{z}^{*\top}_{ik'} \mathbf{f}_k - 2 \mathbf{z}^{*\top}_{ik'} d_{ik'} \mathbf{f}_k}{2} \right\} \exp\left\{ -\frac{1}{2} \mathbf{f}_k^\top (\nu_2^{-1} \mathbf{I}_r) \mathbf{f}_k \right\} \\
&\propto \exp\left(-\frac{1}{2} (\mathbf{f}_k - \mu_k)^\top \Sigma_k^{-1} (\mathbf{f}_k - \mu_k) \right),
\end{aligned}$$

where

$$\begin{aligned}\mu_k &= \left(\mathbf{Z}_{k'}^{*\top}\mathbf{Z}_{k'}^{*} + \nu_2^{-1}\mathbf{I}_r\right)^{-1}\mathbf{Z}_{k'}^{*\top}\mathbf{d}_{k'},\\ \Sigma_k &= \left(\mathbf{Z}_{k'}^{*\top}\mathbf{Z}_{k'}^{*} + \nu_2^{-1}\mathbf{I}_r\right)^{-1}.\end{aligned}$$

This implies

$$\mathbf{f}_k \mid - \sim \mathcal{N}\left(\left(\mathbf{Z}_{k'}^{*\top}\mathbf{Z}_{k'}^{*} + \nu_2^{-1}\mathbf{I}_r\right)^{-1}\mathbf{Z}_{k'}^{*\top}\mathbf{d}_{k'}, \left(\mathbf{Z}_{k'}^{*\top}\mathbf{Z}_{k'}^{*} + \nu_2^{-1}\mathbf{I}_r\right)^{-1}\right).$$

Full conditional distribution of λ_{ik}: Since we assign the constant prior to the latent variable, the conditional distribution of λ_{ik} is only proportional to the pseudo-likelihood. Here, we show that the conditional distribution of λ_{ik} becomes the inverse Gaussian distribution. A random variable has the inverse Gaussian distribution $\mathcal{IG}(\mu, \lambda)$ with mean and variance $E(x) = \mu$ and $Var(x) = \mu^3/\lambda$ if the density function is given by

$$f(x \mid \mu, \lambda) = \sqrt{\frac{\lambda}{2\pi x^3}}\exp\left\{-\frac{\lambda(x-\mu)^2}{2\mu^2 x}\right\}, \quad x \in (0, \infty).$$

Let $\eta = \frac{1}{\lambda_{ik}}$, and $\lambda_{ik} = \frac{1}{\eta}$, $\frac{d\lambda_{ik}}{d\eta} = -\eta^{-2}$. We now derive the full conditional distribution of λ_{ik} using the Jacobian transformation.

$$\begin{aligned}\pi(\eta \mid -) &\propto f(y_{ik} \mid \mathbf{c}_k, \lambda_{ik})\pi(\eta)\\ &\propto \frac{1}{\sqrt{\frac{1}{\eta}}}\exp\left\{-\frac{1}{2}\frac{(1+\frac{1}{\eta} - y_{ik}\mathbf{x}_{ik}^{\top}\mathbf{c}_k)^2}{\frac{1}{\eta}}\right\}\eta^{-2}\\ &\propto \frac{1}{\sqrt{\eta^3}}\exp\left\{-\frac{\eta^2\left((y_{ik}\mathbf{x}_{ik}^{\top}\mathbf{c}_k)^2 - 2y_{ik}\mathbf{x}_{ik}^{\top}\mathbf{c}_k + 1\right) - 2\eta(y_{ik}\mathbf{x}_{ik}^{\top}\mathbf{c}_k - 1) + 1}{2\eta}\right\}\\ &\propto \frac{1}{\sqrt{\eta^3}}\exp\left\{-\frac{\eta^2 - 2\eta(y_{ik}\mathbf{x}_{ik}^{\top}\mathbf{c}_k - 1)^{-1} + (y_{ik}\mathbf{x}_{ik}^{\top}\mathbf{c}_k)^{-2}}{2\eta(y_{ik}\mathbf{x}_{ik}^{\top}\mathbf{c}_k - 1)^{-2}}\right\}\\ &\propto \frac{1}{\sqrt{\eta^3}}\exp\left\{-\frac{(\eta - |y_{ik}\mathbf{x}_{ik}^{\top}\mathbf{c}_k - 1|^{-1})^2}{2\eta(y_{ik}\mathbf{x}_{ik}^{\top}\mathbf{c}_k - 1)^{-2}}\right\},\end{aligned}$$

which implies $\frac{1}{\lambda_{ik}} \mid \mathbf{c}_k, y_{ik} \overset{ind}{\sim} \mathcal{IG}(|1 - y_{ik}\mathbf{x}_{ik}^{\top}\mathbf{c}_k|^{-1}, 1)$.

Full conditional distribution for vec(A): Define $\mathbf{Z}_k = (\mathbf{z}_{1k}, \cdots, \mathbf{z}_{nk})^\top$ with $\mathbf{z}_{ik} = y_{ik}\mathbf{x}_{ik}/\sqrt{\lambda_{ik}}$, $\mathbf{d}_k = (d_{1k}, \cdots, d_{nk})^\top$ with $d_{ik} = (1+\lambda_{ik})/\sqrt{\lambda_{ik}}$, $\mathbf{d} = (\mathbf{d}_1^\top, \cdots, \mathbf{d}_q^\top)$, $\mathbf{Z} = bdiag(\mathbf{Z}_1, \cdots, \mathbf{Z}_q)$, where $bdiag(\cdot)$ stands for the block diagonal matrix and $\otimes$ denotes the Kronecker product between two matrices. To develop the full conditional distribution $\pi(\text{vec}(\mathbf{A}) \mid -)$, we show that the pseudo-likelihood can be expressed as a form of multivariate Gaussian linear model:

$$
\begin{aligned}
f(\mathbf{Y} \mid \mathbf{C}) &= \prod_{k=1}^{q}\prod_{i=1}^{n} f(y_{ik} \mid \mathbf{c}_k, \lambda_{ik}) \\
&\propto \prod_{k=1}^{q}\prod_{i=1}^{n} \frac{1}{\lambda_{ik}^{1/2}} \exp\left\{-\frac{1}{2}\frac{(1+\lambda_{ik} - y_{ik}\mathbf{x}_{ik}^\top \mathbf{c}_k)^2}{\lambda_{ik}}\right\} \\
&\propto \prod_{k=1}^{q}\prod_{i=1}^{n} \exp\left\{-\frac{1}{2}\frac{\mathbf{c}_k^\top \mathbf{x}_{ik} y_{ik}^2 \mathbf{x}_{ik}^\top \mathbf{c}_k - 2(1+\lambda_{ik}) y_{ik}\mathbf{x}_{ik}^\top \mathbf{c}_k}{\lambda_{ik}}\right\} \\
&\propto \prod_{k=1}^{q}\prod_{i=1}^{n} \exp\left\{-\frac{1}{2}(\mathbf{c}_k^\top \mathbf{z}_{ik}^\top \mathbf{z}_{ik}\mathbf{c}_k - 2\mathbf{z}_{ik}^\top d_{ik}\mathbf{c}_k)\right\} \\
&\propto \prod_{k=1}^{q}\prod_{i=1}^{n} \exp\left\{-\frac{1}{2}(d_{ik} - \mathbf{z}_{ik}^\top \mathbf{c}_k)^2\right\} \\
&\propto \prod_{k=1}^{q} \exp\left\{-\frac{1}{2}\|\mathbf{d}_k - \mathbf{Z}_k\mathbf{c}_k\|^2\right\} \\
&\propto \exp\left\{-\frac{1}{2}\|\mathbf{d} - \mathbf{Z}\text{vec}(\mathbf{C})\|^2\right\}. \qquad (4.5)
\end{aligned}
$$

The last equation (4.5) has the same form of a multivariate linear model. We can write it as $\mathbf{d} \sim \mathcal{N}(\mathbf{Z}\text{vec}(\mathbf{C}), \mathbf{I}_{nq})$.

Lemma 1. *$vec(\mathbf{AB}^\top) = (\mathbf{B} \otimes \mathbf{I}_p)vec(\mathbf{A})$, where $\otimes$ denotes Kronecker product.*

Using Lemma 1, we can show that

$$\begin{aligned}
\pi(\text{vec}(\mathbf{A}) \mid -) &\propto \exp\left(-\frac{1}{2}\|\mathbf{d} - \mathbf{Z}(\mathbf{B}\otimes\mathbf{I}_p)\text{vec}(\mathbf{A})\|^2\right)\pi(\mathbf{A}) \\
&\propto \exp\left\{-\frac{1}{2}\|\mathbf{d} - \mathbf{Z}(\mathbf{B}\otimes\mathbf{I}_p)\text{vec}(\mathbf{A})\|^2\right\}\exp\left\{-\frac{1}{2\nu_1}(\text{vec}(\mathbf{A}))^\top\text{vec}(\mathbf{A})\right\} \\
&\propto \exp\left\{-\frac{1}{2}\left(\|\mathbf{d} - \mathbf{Z}(\mathbf{B}\otimes\mathbf{I}_p)\text{vec}(\mathbf{A})\|^2 + (\text{vec}(\mathbf{A}))^\top\nu_1^{-1}\mathbf{I}_{pr}\text{vec}(\mathbf{A})\right)\right\} \\
&\propto \exp\left\{-\frac{1}{2}(\text{vec}(\mathbf{A}) - \boldsymbol{\mu}^{\mathbf{A}})^\top\left(\mathbf{Z}^{*\top}\mathbf{Z}^* + \nu_1^{-1}\mathbf{I}_{pr}\right)(\text{vec}(\mathbf{A}) - \boldsymbol{\mu}^{\mathbf{A}})\right\},
\end{aligned}$$

where $\boldsymbol{\mu}^{\mathbf{A}} = (\mathbf{Z}^{*\top}\mathbf{Z}^* + \nu_1^{-1}\mathbf{I}_{pr})^{-1}\mathbf{Z}^{*\top}\mathbf{d}$, with $\mathbf{Z}^* = \mathbf{Z}(\mathbf{B}\otimes\mathbf{I}_p)$. Therefore, we have

$$\text{vec}(\mathbf{A}) \mid - \sim \mathcal{N}\left((\mathbf{Z}^{*\top}\mathbf{Z}^* + \nu_1^{-1}\mathbf{I}_{pr})^{-1}\mathbf{Z}^{*\top}\mathbf{d}, (\mathbf{Z}^{*\top}\mathbf{Z}^* + \nu_1^{-1}\mathbf{I}_{pr})^{-1}\right).$$

4.3.2 Rank selection

Under reduced-rank framework, an important procedure is to select rank using data from the validation set. One typically selects a single rank from a class of candidate values and then proceeds as if the selected rank is the true. However, this ignores the model uncertainty in model selection and leads to the problem of over-confident inferences (Hoeting et al., 1999; Raftery et al., 1997).

To get rid of the ambiguity and account for the model uncertainty associated with rank selection, we propose to use the Bayesian model averaging method. We show more details of the rank selection problem under two different scenarios as follows:

Scenario 1: When there is a rank which dominates other ranks, i.e., the posterior probability at the single rank is extremely higher than others. Define a set which contains all possible candidate rank values as $\mathcal{A} = \{1, 2, \cdots, \min(p, q)\}$. Suppose that

$$\frac{p(r^* \mid \mathbf{Y})}{p(r' \mid \mathbf{Y})} > \epsilon \quad \forall\, r' \in \mathcal{A}\backslash r^*,$$

for some constant ϵ. Here, we set $\epsilon = 20$, following Madigan and Raftery (1994). In this case, it is clear that we choose r^* as our final decision since the other models have negligibly small posterior probabilities.

Scenario 2: Suppose that some ranks have similar magnitudes of posterior probabilities in the sense that

$$\exists\, r' \in \mathcal{A} \quad s.t. \quad \frac{\max_{r \in \mathcal{A}}\{p(r \mid \mathbf{Y})\}}{p(r' \mid \mathbf{Y})} \leq \epsilon.$$

Then, the posterior inferences for the $p \times q$ coefficient matrix $\mathbf{C}$ can be made as follows:

1. Repeat for $l = 1, 2, \cdots, c$:

   i. Given $r = r_l$, define N_b as the burn-in period sample size and $T^{(r_l)}$ as the sample size after the burn-in period where

   $$\frac{T^{(r_l)}}{\sum_{l=1}^{c} T^{(r_l)}} = \frac{p(r_l \mid \mathbf{Y})}{\sum_{l=1}^{c} p(r_l \mid \mathbf{Y})}.$$

   ii. Conduct the entire Gibbs sampling and then save the generated posterior samples after the burn-in period $\mathbf{C}_{(t)}^{(r_l)}, t = 1, \cdots, T^{(r_l)}$.

2. Compute the average using all posterior samples from Step 1.

$$\hat{\mathbf{C}} = \sum_{l=1}^{c} \sum_{t=1}^{T^{(r_l)}} \mathbf{C}_{(t)}^{(r_l)} / \sum_{l=1}^{c} T^{(r_l)}.$$

It is important to note that calculation of $p(r \mid \mathbf{Y})$ is infeasible under SVM. To address this issue, we propose to use the Laplace approximation. However, there is a pressing challenge that it is impossible to obtain the first-order and second-order derivatives of the hinge loss in (4.3). To overcome this difficulty, we propose to use the smoothed hinge loss (Chapelle, 2007) whose value is almost the same with the classic hinge loss except for the

non-differentiable point at $\xi = 1$.

$$\phi(\xi) = \begin{cases} 1 - \xi, & \xi < 1 - h, \\ -\frac{(1-\xi)^4}{16h^3} + \frac{3(1-\xi)^2}{8h} + \frac{1-\xi}{2} + \frac{3h}{16}, & |1 - \xi| \leq h, \\ 0, & \xi > 1 + h, \end{cases} \tag{4.6}$$

where h is a constant controlling the smoothness. As recommended in Chapelle (2007), we use $h = 0.1$ to meet the balance between numerical stability and performance. Given $\text{rank}(\mathbf{C}) = r$, we then obtain the maximum likelihood estimates (MLEs) of $\mathbf{A}$ and $\mathbf{F}$ by minimizing the smoothed hinge loss as follows:

$$\left(\text{vec}(\hat{\mathbf{A}}), \text{vec}(\hat{\mathbf{F}})\right) = \arg\min_{\mathbf{A},\mathbf{F}} \sum_{i=1}^{n} \sum_{k=1}^{q} \phi(\xi_{ik}),$$

where $\xi_{ik} = y_{ik}(\mathbf{x}_{ik}^\top \mathbf{c}_k)$. Combining with the constraint (4.4), the estimated coefficient matrix is obtained as $\hat{\mathbf{C}} = [\hat{\mathbf{A}}, \hat{\mathbf{A}}\hat{\mathbf{F}}]$. Using the smoothed hinge loss (4.6) and ignoring the constant terms with respect to n, the Laplace approximation leads to

$$\log p(r \mid \mathbf{Y}) \approx -2 \sum_{i=1}^{n} \sum_{q=1}^{k} \phi(\xi_{ik}) - \frac{1}{2}(pr + qr - r^2) \log n.$$

4.4 Simulation study

In this section, we study the performance of our proposed methods through simulated datasets. We set all entries in the coefficient matrix equal to 0.5 including the intercept as $\mathbf{C} = (0.5)_{p \times q}$. Consequently, the true rank of $\mathbf{C}$ is 1. The q predictor matrices are conducted by independently generating all their entries as iid random samples from uniform distribution $\mathcal{U}(-3, 3)$, and their first columns are set to be all 1s. The true model is defined by the probit link. The success probability p_{ik} is defined by $p_{ik} = \Phi(\mathbf{x}_{ik}^\top \mathbf{c}_k)$, where $\Phi(\cdot)$ is the cumulative density function of standard normal distribution. The elements of the response matrix is generated from Bernoulli distribution with the success probabilities p_{ik} as

$y_{ik} \overset{ind}{\sim} \mathcal{B}(p_{ik})$ for $i = 1, \cdots, n$, and $k = 1, \cdots, q$.

To justify that more outcomes can provide more interrelations to improve the prediction accuracy, we consider the following three cases with various q values:

i. $n = 100, p = 6, q = 10$.

ii. $n = 100, p = 6, q = 20$.

iii. $n = 100, p = 6, q = 30$.

We assume that there is no preferred model by setting $\nu_1 = \nu_2 = 100$. For the implementation of MCMC, we use $2,000$ samples (after $1,000$ burn-in iterations) from the posterior distribution using the Gibbs sampling algorithm. For the sake of comparison, we consider the data augmentation method of Polson and Scott (2011) that obtains the posterior inferences for each column in $\mathbf{C}$ separately. This method is equivalent with running our algorithm under the full rank where we assume that each column in the coefficient matrix is independent with one another. Under the multivariate framework, the data augmentation method can be expressed as

$$\begin{aligned} f(y_{ik} \mid \mathbf{c}_k, \lambda_{ik}) &\propto \frac{1}{\sqrt{2\pi\lambda_{ik}}} \exp\left\{ -\frac{1}{2} \frac{(1 + \lambda_{ik} - y_{ik}\mathbf{x}_{ik}^\top \mathbf{c}_k)^2}{\lambda_{ik}} \right\}, \\ \pi(\lambda_{ik}) &\propto 1, \\ \mathbf{c}_k &\sim \mathcal{N}(\mathbf{0}, \tau^2 \mathbf{I}_p), \end{aligned}$$

where τ is a prespecified hyperparameter and we set $\tau^2 = 100$. The posterior inference can be made by drawing samples from the following full conditionals iteratively for $k = 1, \cdots, q$:

$$\begin{aligned} \mathbf{c}_k \mid - &\sim \mathcal{N}\left((\mathbf{Z}_k^\top \mathbf{Z}_k + \tau^{-2}\mathbf{I}_p)^{-1} \mathbf{Z}_k^\top \mathbf{d}_k, (\mathbf{Z}_k^\top \mathbf{Z}_k + \tau^{-2}\mathbf{I}_p)^{-1} \right) \\ \frac{1}{\lambda_{ik}} \mid - &\overset{ind}{\sim} \mathcal{IG}(|1 - y_{ik}\mathbf{x}_{ik}^\top \mathbf{c}_k|^{-1}, 1). \end{aligned}$$

As an alternative, we also consider the logistic regression model which is commonly used to model binary dependent variables, to serve as a baseline algorithm. We employ the logistic

model q times, one for each response. The design matrices and response matrix in the test set are generated with the same settings as in the training set for $n_{\text{test}} = 100$.

To assess the prediction accuracy of each method, we calculate true positive rate (TPR), true negative rate (TNR), false positive rate (FPR), and false negative rate (FNR) where they denote the percentage of true positive ones, true negative ones, false positive ones, and false negative ones, respectively. In addition, we also compute accuracy, precision, as well as F1 score to give a more comprehensive measure for the prediction accuracy. All measurements are estimated by the Monte Carlo method with 100 replications. Table 4.1 reports the true positive rate, true negative rate, false positive rate, false negative rate, precision, accuracy, as well as F1 score for each method under each case. The results show that our proposed method outperforms both the logistic model and the data augmentation method. This clearly implies that the proposed method is better with respect to the prediction accuracy in all aspects and takes good care of the reduced-rank structure among the multiple response variables. In addition, as shown in Figure 4.1, the increment of prediction performance between the proposed method and the other two methods increases as the number of response variables gets larger in that more of the interrelationships can be utilized to increase the prediction accuracy. Therefore, we argue that our proposed method truly captures the dependency among the responses and the reduced-rank structure is fairly needed.

4.5 Real data analysis

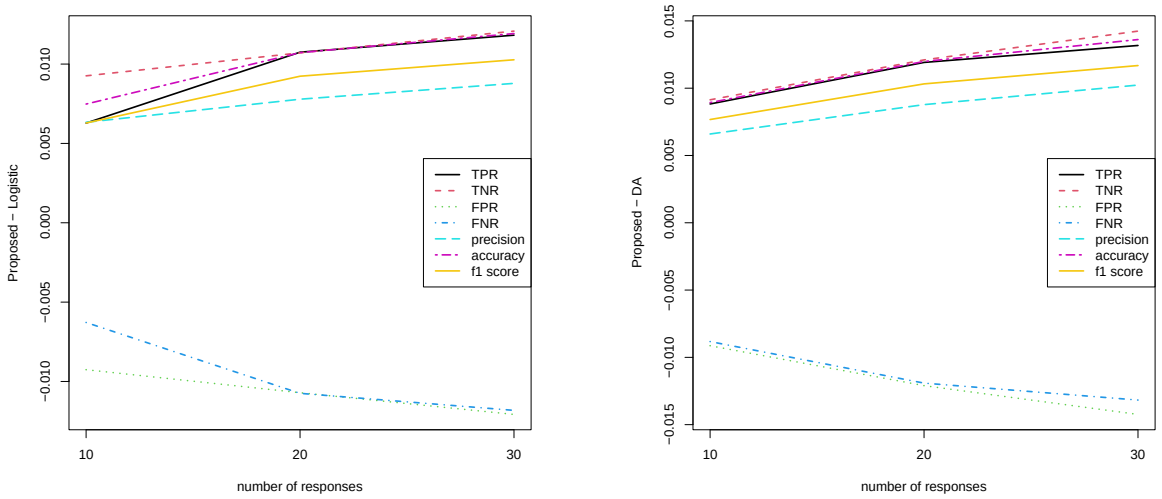
In this section, we study the performance of our proposed method for multivariate classification problem using *spider* data; the data set is available in `R` package `mvabund`. In the spider data, the design matrix $\mathbf{X} \in \mathbb{R}^{28\times 6}$ contains the information of 6 environmental features. The response matrix $\mathbf{Y} \in \mathbb{R}^{28\times 12}$ is a data frame with 28 observations of abundance of 12 hunting spider species. Since the response matrix $\mathbf{Y}$ is count data showing the number of each spider species, we have to transform them to binary outcomes. For the i-th observation, if $y_{ik} = 0$, indicating there is no such species surviving in certain environment, we define $y^*_{ik} = -1$. Otherwise, let $y^*_{ik} = 1$. In sum, we have a total of 154 negative ones and 182

Table 4.1: Simulation results: average true positive rate (TPR), true negative rate (TNR), false positive rate (FPR), false negative rate (FNR), precision, accuracy, F1 score, and standard errors (in parenthesis) over 100 Monte Carlo experiments. q is the number of responses.

Method	Logistic	DA	Proposed
		$q = 10$	
TPR	0.8761 (0.0016)	0.8736 (0.0016)	0.8824 (0.0016)
TNR	0.8000 (0.0025)	0.8001 (0.0026)	0.8092 (0.0025)
FPR	0.2000 (0.0025)	0.1999 (0.0026)	0.1908 (0.0025)
FNR	0.1239 (0.0016)	0.1264 (0.0016)	0.1176 (0.0016)
precision	0.8627 (0.0015)	0.8624 (0.0016)	0.8690 (0.0016)
accuracy	0.8448 (0.0012)	0.8434 (0.0013)	0.8523 (0.0012)
F1 score	0.8692 (0.0010)	0.8678 (0.0011)	0.8755 (0.0010)
		$q = 20$	
TPR	0.8738 (0.0012)	0.8726 (0.0012)	0.8845 (0.0012)
TNR	0.8002 (0.0016)	0.7988 (0.0017)	0.8109 (0.0017)
FPR	0.1998 (0.0016)	0.2012 (0.0017)	0.1891 (0.0017)
FNR	0.1262 (0.0012)	0.1274 (0.0012)	0.1155 (0.0012)
precision	0.8628 (0.0010)	0.8618 (0.0011)	0.8705 (0.0011)
accuracy	0.8436 (0.0008)	0.8423 (0.0008)	0.8543 (0.0008)
F1 score	0.8682 (0.0007)	0.8671 (0.0007)	0.8774 (0.0007)
		$q = 30$	
TPR	0.8728 (0.0010)	0.8714 (0.0011)	0.8846 (0.0010)
TNR	0.7959 (0.0013)	0.7937 (0.0014)	0.8080 (0.0014)
FPR	0.2041 (0.0013)	0.2063 (0.0014)	0.1920 (0.0014)
FNR	0.1272 (0.0010)	0.1286 (0.0011)	0.1154 (0.0010)
precision	0.8594 (0.0008)	0.8580 (0.0008)	0.8682 (0.0009)
accuracy	0.8411 (0.0007)	0.8395 (0.0007)	0.8531 (0.0007)
F1 score	0.8660 (0.0006)	0.8646 (0.0006)	0.8763 (0.0006)

positive ones.

For comparison of prediction accuracy, we randomly split the data into a training set (of size 23) and a test set (of size 5). For each random partition of the data, we train the model using our proposed method, logistic regression model, as well as the data augmentation method. After the model training, we measure the prediction accuracy based on the test data set using the same classification metrics as we used in Section 4.4. Note that in rank selection process, we compute the marginal likelihood using Laplace approximation and employ the Occam's window criterion as in Section 4.3.2. We repeat the above procedure 100 times.



(a) Difference between Proposed and Logistic (b) Difference between Proposed and DA

Figure 4.1: Prediction increment as the number of responses increases.

As a result, 74% of them, we select one single rank and 26% of them more than one ranks are selected. This justifies the fact that the Bayesian model averaging approach works well in model comparison and covers the uncertainty associated with the rank selection. The summary of the results is shown in Table 4.2 based on the 100 replications. As a result, our proposed method outperforms both the logistic model and the existing data augmentation method.

Table 4.2: Prediction accuracy and standard errors (in parenthesis) using *spider* data over 100 replications.

Method	Logistic	DA	Proposed
TPR	0.7969 (0.0095)	0.8256 (0.0091)	0.8237 (0.0095)
TNR	0.8031 (0.0088)	0.8054 (0.0081)	0.8397 (0.0074)
FPR	0.1969 (0.0088)	0.1946 (0.0081)	0.1603 (0.0074)
FNR	0.2031 (0.0095)	0.1744 (0.0091)	0.1763 (0.0095)
precision	0.8397 (0.0074)	0.8476 (0.0067)	0.8703 (0.0064)
accuracy	0.7983 (0.0059)	0.8162 (0.0055)	0.8295 (0.0057)
F1 score	0.8130 (0.0062)	0.8321 (0.0056)	0.8419 (0.0057)

4.6 Concluding remarks

The hinge loss objective function for SVM seems to be a challenge for traditional Bayesian analysis, especially under the multivariate framework with multiple response variables. However, the pseudo-likelihood for SVM can be expressed as a mixture of normal distributions which allow SVM to be analyzed using Gaussian linear models. We have developed an extended data augmentation method for multivariate SVM under the reduced-rank structure. Our method is based on the fact that multiple responses might have some interrelationships so that the prediction accuracy can be improved by borrowing information across the outcomes. The MCMC algorithm leads to closed-form full conditional distributions.

While the proposed method successfully extends the data augmentation method to multivariate reduced-rank framework, there is a lack of a solution to the problem of high-dimensional variable selection. Extending our methods with high-dimensional feature selection could be a good direction for future research. Another interesting area is to extend our methods to nonlinear generalizations using kernel-based approaches.

Chapter 5

Conclusion

In this dissertation, Bayesian methods for sparse reduced-rank regression are proposed to achieve both rank reduction and variable selection simultaneously in the presence of multiple response variables and possible interrelationships between the responses.

In Chapter 2, we have developed a fully-Bayesian approach to sparse and low-rank matrix estimation in a multivariate regression framework. The proposed method provides a solution to accounting for the model uncertainty associated with variable selection and rank selection. One advantage of our method is free of selecting the single best model, which is a major problem in existing SRRR methods , by marginalization via the proposed MCMC method. In addition, our proposed method is robust to the change in correlation structure of the error term by assigning the inverse-Wishart prior to the covariance matrix.

In Chapter 3, we have developed a Bayesian approach to sparse low-rank matrix estimation in a multivariate generalized regression framework. The proposed method provides an efficient way to handle both rank selection and variable selection at the same time. Under the reduced-rank structure, the number of parameters to be estimated is greatly reduced and the model uncertainty is also taken into account. Furthermore, our proposed method can simply be extended to mixed-type outcomes by using different link functions and likelihoods.

In Chapter 4, we have proposed a multivariate latent variable representation of Bayesian support vector machine (SVM) under the reduced-rank structure. The use of the hierarchical

representation for SVM allows us to perform Bayesian inference via the Gibbs sampling algorithm so that we can make probabilistic interpretation of class probabilities rather than deterministic class label predictions.

A major limitation of our proposed methods is the computational cost, especially in the high-dimensional settings where p is large. Moreover, our Bayesian approaches would be extremely inefficient when both n and q are huge due to the high-dimensional inverse matrix computation. Another potential issue is that the estimate of the parameters might be biased if the reduced-rank structure is not present because the major assumption of our methods is that the coefficient matrix has reduced-rank.

In this dissertation, we mainly focus on developing the methodologies to solve various challenges of the reduced-rank approach. Therefore, some theoretical properties including asymptotic theory and computational complexity have been left for further investigation. Owing to the generality of our proposed methods, many extensions can be easily done on the basis of the proposed framework. For instance, our approaches can be extended to various data types, such as binary, count, continuous and time-to-event (survival) data. More applications of our method in biological fields such as gene expression and image recognition are greatly welcomed. Finally, extensions of the proposed multivariate SVM to the problem of high-dimensional feature selection and nonlinear generalizations using kernel-based methods would be good directions for future research work.

Bibliography

Alquier, P. (2013). Bayesian methods for low-rank matrix estimation: Short survey and theoretical study. In S. Jain, R. Munos, F. Stephan, and T. Zeugmann (Eds.), *Algorithmic Learning Theory*, Berlin, Heidelberg, pp. 309–323. Springer Berlin Heidelberg.

Anderson, T. W. (1951). Estimating linear restrictions on regression coefficients for multivariate normal distributions. *The Annals of Mathematical Statistics 22*(3), 327–351.

Babacan, S. D., M. Luessi, R. Molina, and A. K. Katsaggelos (2011). Low-rank matrix completion by variational sparse Bayesian learning. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2188–2191. IEEE.

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

Becker, N., G. Toedt, P. Lichter, and A. Benner (2011). Elastic scad as a novel penalization method for svm classification tasks in high-dimensional data. *BMC Bioinformatics 12*(1), 1–13.

Bhadra, A., J. Datta, N. G. Polson, and B. Willard (2019). Lasso meets horseshoe: A survey. *Statistical Science 34*(3), 405–427.

Boulesteix, A.-L. and K. Strimmer (2005). Predicting transcription factor activities from combined analysis of microarray and ChIP data: a partial least squares approach. *Theoretical Biology and Medical Modelling 2*(1), 23.

Bradley, P. S. and O. L. Mangasarian (1998). Feature selection via concave minimization and support vector machines. In *ICML*, Volume 98, pp. 82–90. Citeseer.

Bunea, F., Y. She, and M. H. Wegkamp (2011). Optimal selection of reduced rank estimators of high-dimensional matrices. *The Annals of Statistics 39*(2), 1282–1309.

Bunea, F., Y. She, and M. H. Wegkamp (2012). Joint variable and rank selection for parsimonious estimation of high-dimensional matrices. *The Annals of Statistics 40*(5), 2359–2388.

Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery 2*(2), 121–167.

Carlin, B. P. and S. Chib (1995). Bayesian model choice via Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Methodological) 57*(3), 473–484.

Chakraborty, A., A. Bhattacharya, and B. K. Mallick (2016). Bayesian sparse multiple regression for simultaneous rank reduction and variable selection. *arXiv preprint arXiv:1612.00877*.

Chapelle, O. (2007). Training a support vector machine in the primal. *Neural Computation 19*(5), 1155–1178.

Chen, J. and Z. Chen (2008). Extended Bayesian information criteria for model selection with large model spaces. *Biometrika 95*(3), 759–771.

Chen, K., K.-S. Chan, and N. C. Stenseth (2012). Reduced rank stochastic regression with a sparse singular value decomposition. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 74*(2), 203–221.

Chen, K., H. Dong, and K.-S. Chan (2013). Reduced rank regression via adaptive nuclear norm penalization. *Biometrika 100*(4), 901–920.

Chen, L. and J. Z. Huang (2012). Sparse reduced-rank regression for simultaneous dimension reduction and variable selection. *Journal of the American Statistical Association 107*(500), 1533–1545.

Chib, S. and E. Greenberg (1998). Analysis of multivariate probit models. *Biometrika 85*(2), 347–361.

Chun, H. and S. Keleş (2010). Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 72*(1), 3–25.

Cortes, C. and V. Vapnik (1995). Support-vector networks. *Machine Learning 20*(3), 273–297.

Cristianini, N., J. Shawe-Taylor, et al. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.

Fan, J. and R. Li (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association 96*(456), 1348–1360.

Fiocco, M., H. Putter, and H. C. van Houwelingen (2008). Reduced-rank proportional hazards regression and simulation-based prediction for multi-state models. *Statistics in Medicine 27*(21), 4340–4358.

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

George, E. I. and R. E. McCulloch (1993). Variable selection via Gibbs sampling. *Journal of the American Statistical Association 88*(423), 881–889.

Geweke, J. (1996). Bayesian reduced rank regression in econometrics. *Journal of Econometrics 75*(1), 121–146.

Gilbert, S. and P. Zemcik (2006). Who's afraid of reduced-rank parameterizations of multivariate models? theory and example. *Journal of Multivariate Analysis 97*(4), 925–945.

Gilks, W. R., N. G. Best, and K. K. C. Tan (1995). Adaptive rejection Metropolis sampling within Gibbs sampling. *Journal of the Royal Statistical Society: Series C (Applied Statistics) 44*(4), 455–472.

Goh, G., D. K. Dey, and K. Chen (2017). Bayesian sparse reduced rank multivariate regression. *Journal of Multivariate Analysis 157*, 14–28.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika 82*(4), 711–732.

Hans, C., A. Dobra, and M. West (2007). Shotgun stochastic search for "large p" regression. *Journal of the American Statistical Association 102*(478), 507–516.

Hoeting, J. A., D. Madigan, A. E. Raftery, and C. T. Volinsky (1999). Bayesian model averaging: a tutorial. *Statistical Science 14*(4), 382–401.

Izenman, A. J. (1975). Reduced-rank regression for the multivariate linear model. *Journal of Multivariate Analysis 5*(2), 248–264.

Kass, R. E. and A. E. Raftery (1995). Bayes factors. *Journal of the American Statistical Association 90*(430), 773–795.

Kyung, M., J. Gill, M. Ghosh, and G. Casella (2010). Penalized regression, standard errors, and Bayesian lassos. *Bayesian Analysis 5*(2), 369–411.

Lee, T. I., N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thomson, I. Simon, J. Zeitlinger, E. G. Jennings, H. L. Murray, D. B. Gordon, B. Ren, J. J. Wyrick, J.-B. Tagne, T. L. Volkert, E. Fraenkel, D. K. Gifford, and R. A. Young (2002). Transcriptional regulatory networks in Saccharmomyces cerevisiae. *Science 298*, 799–804.

Lim, Y. J. and Y. W. Teh (2007). Variational Bayesian approach to movie rating prediction. In *Proceedings of KDD cup and workshop*, Volume 7, pp. 15–21.

Liu, J. S. (1994). The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem. *Journal of the American Statistical Association 89*(427), 958–966.

Llorente, F., L. Martino, D. Delgado, and J. Lopez-Santiago (2020). Marginal likelihood computation for model selection and hypothesis testing: an extensive review. *arXiv preprint arXiv:2005.08334*.

Lucka, F. (2016). Fast Gibbs sampling for high-dimensional Bayesian inversion. *Inverse Problems 32*(11), 115019.

Luo, C., J. Liang, G. Li, F. Wang, C. Zhang, D. K. Dey, and K. Chen (2018). Leveraging mixed and incomplete outcomes via reduced-rank modeling. *Journal of Multivariate Analysis 167*, 378–394.

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

Ma, Z., Z. Ma, and T. Sun (2014). Adaptive estimation in two-way sparse reduced-rank regression. *arXiv preprint arXiv:1403.1922*.

Madigan, D. and A. E. Raftery (1994). Model selection and accounting for model uncertainty in graphical models using occam's window. *Journal of the American Statistical Association 89*(428), 1535–1546.

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

Martino, L., R. Casarin, F. Leisen, and D. Luengo (2018). Adaptive independent sticky MCMC algorithms. *EURASIP Journal on Advances in Signal Processing 2018*(1), 5.

Martino, L., V. Elvira, and G. Camps-Valls (2018). The recycling Gibbs sampler for efficient learning. *Digital Signal Processing 74*, 1–13.

Martino, L., J. Read, and D. Luengo (2015). Independent doubly adaptive rejection Metropolis sampling within Gibbs sampling. *IEEE Transactions on Signal Processing 63*(12), 3123–3138.

Marttinen, P., M. Pirinen, A.-P. Sarin, J. Gillberg, J. Kettunen, I. Surakka, A. J. Kangas, P. Soininen, P. O'Reilly, M. Kaakinen, M. Kähönen, T. Lehtimäki, M. Ala-Korpela, O. T. Raitakari, V. Salomaa, M.-R. Järvelin, S. Ripatti, and S. Kaski (2014). Assessing multivariate gene-metabolome associations with rare variants using Bayesian reduced rank regression. *Bioinformatics 30*(14), 2026–2034.

O'Hara, R. B., M. J. Sillanpää, et al. (2009). A review of Bayesian variable selection methods: what, how and which. *Bayesian analysis 4*(1), 85–117.

Park, T. and G. Casella (2008). The Bayesian lasso. *Journal of the American Statistical Association 103*(482), 681–686.

Polson, N. G. and S. L. Scott (2011). Data augmentation for support vector machines. *Bayesian Analysis 6*(1), 1–23.

Raftery, A. E., D. Madigan, and J. A. Hoeting (1997). Bayesian model averaging for linear regression models. *Journal of the American Statistical Association 92*(437), 179–191.

Schølkopf, B. and C. J. Burges (1998). Alexander j. smola advances in kernel methods: Support vector learning.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics 6*(2), 461–464.

Smola, A. J. and B. Schölkopf (2004). A tutorial on support vector regression. *Statistics and Computing 14*(3), 199–222.

Song, M., C. M. Breneman, J. Bi, N. Sukumar, K. P. Bennett, S. Cramer, and N. Tugcu (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.

Spellman, P. T., G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, B. D, and B. Futcher (1998). Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell 9*, 3273–3297.

Spiegelhalter, D. J., N. G. Best, B. P. Carlin, and A. Van Der Linde (2002). Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 64*(4), 583–639.

Sun, W., C. Chang, Y. Zhao, and Q. Long (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)*, pp. 1484–1493. IEEE.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological) 58*(1), 267–288.

Tierney, L. (1994). Markov chains for exploring posterior distributions. *The Annals of Statistics 22*(4), 1701–1728.

Tierney, L. and J. B. Kadane (1986). Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association 81*(393), 82–86.

Tierney, L., R. E. Kass, and J. B. Kadane (1989). Fully exponential Laplace approximations to expectations and variances of nonpositive functions. *Journal of the American Statistical Association 84*(407), 710–716.

Van Dyk, D. A. and T. Park (2008). Partially collapsed gibbs samplers: Theory and methods. *Journal of the American Statistical Association 103*(482), 790–796.

Velu, R. and G. C. Reinsel (2013). *Multivariate reduced-rank regression: theory and applications*, Volume 136. Springer Science & Business Media.

Wang, L., G. Chen, and H. Li (2007). Group SCAD regression analysis for microarray time course gene expression data. *Bioinformatics 23*(12), 1486–1494.

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

Yang, D., G. Goh, and H. Wang (2020). A fully bayesian approach to sparse reduced-rank multivariate regression. *Statistical Modelling*, 1471082X20948697.

Yee, T. W. and T. J. Hastie (2003). Reduced-rank vector generalized linear models. *Statistical Modelling 3*(1), 15–41.

Yuan, M., A. Ekici, Z. Lu, and R. Monteiro (2007). Dimension reduction and coefficient estimation in multivariate linear regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 69*(3), 329–346.

Zhang, C.-H. et al. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics 38*(2), 894–942.

Zhang, H. H., J. Ahn, X. Lin, and C. Park (2006). Gene selection using support vector machines with non-convex penalty. *Bioinformatics 22*(1), 88–95.

Zheng, W. (2014). Multi-view facial expression recognition based on group sparse reduced-rank regression. *IEEE Transactions on Affective Computing 5*(1), 71–85.

Zhu, H., Z. Khondker, Z. Lu, and J. G. Ibrahim (2014). Bayesian generalized low rank regression models for neuroimaging phenotypes and genetic markers. *Journal of the American Statistical Association 109*(507), 977–990.

Zhu, J., S. Rosset, R. Tibshirani, and T. Hastie (2003). 1-norm support vector machines. *Advances in Neural Information Processing Systems 16*.

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

Appendix A

Calculation of full conditionals

A.1 Derivation of (2.6)

Note that $\text{vec}(\mathbf{B}^\top) = (\text{vec}(\mathbf{I}_r)^\top, \text{vec}(\mathbf{F}^\top)^\top)^\top$ since $\mathbf{B} = [\mathbf{I}_r, \mathbf{F}^\top]^\top$. For $\mathbf{F}$, the likelihood is proportional to

$$
\begin{aligned}
f(\mathbf{Y} \mid \mathbf{A}_{\boldsymbol{\gamma}}, \mathbf{B}, \boldsymbol{\Sigma}, r, \boldsymbol{\gamma}) &\propto \exp\left[-\frac{1}{2}\text{tr}\{(\mathbf{Y} - \mathbf{X}_{\boldsymbol{\gamma}}\mathbf{A}_{\boldsymbol{\gamma}}\mathbf{B}^\top)\boldsymbol{\Sigma}^{-1}(\mathbf{Y} - \mathbf{X}_{\boldsymbol{\gamma}}\mathbf{A}_{\boldsymbol{\gamma}}\mathbf{B}^\top)^\top\}\right] \\
&\propto \exp\left[-\frac{1}{2}\left\{\text{tr}(\mathbf{B}\mathbf{A}_{\boldsymbol{\gamma}}^\top\mathbf{X}_{\boldsymbol{\gamma}}^\top\mathbf{X}_{\boldsymbol{\gamma}}\mathbf{A}_{\boldsymbol{\gamma}}\mathbf{B}^\top\boldsymbol{\Sigma}^{-1}) - 2\text{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{Y}^\top\mathbf{X}_{\boldsymbol{\gamma}}\mathbf{A}_{\boldsymbol{\gamma}}\mathbf{B}^\top)\right\}\right] \\
&\propto \exp\left[-\frac{1}{2}\left\{\text{vec}(\mathbf{B}^\top)^\top\mathbf{G}\text{vec}(\mathbf{B}^\top) - 2\mathbf{m}^\top\text{vec}(\mathbf{B}^\top)\right\}\right] \\
&\propto \exp\left[-\frac{1}{2}\left\{\text{vec}(\mathbf{F}^\top)^\top\mathbf{G}\text{vec}(\mathbf{F}^\top) - 2\text{vec}(\mathbf{F}^\top)^\top(\mathbf{m}_{\Omega^*} - \mathbf{G}_{[\Omega^*,-\Omega^*]}\text{vec}(\mathbf{I}_r))\right\}\right],
\end{aligned}
$$

where $\mathbf{G} = \boldsymbol{\Sigma}^{-1} \otimes \mathbf{A}_{\boldsymbol{\gamma}}^\top\mathbf{X}_{\boldsymbol{\gamma}}^\top\mathbf{X}_{\boldsymbol{\gamma}}\mathbf{A}_{\boldsymbol{\gamma}}$ and $\mathbf{m} = \text{vec}(\mathbf{A}_{\boldsymbol{\gamma}}^\top\mathbf{X}_{\boldsymbol{\gamma}}^\top\mathbf{Y}\boldsymbol{\Sigma}^{-1})$. From the above result, we have

$$
\begin{aligned}
&\pi(\mathbf{F} \mid \mathbf{Y}, r, \boldsymbol{\gamma}, \mathbf{A}_{\boldsymbol{\gamma}}, \boldsymbol{\Sigma}) \propto f(\mathbf{Y} \mid \mathbf{A}_{\boldsymbol{\gamma}}, \mathbf{B}, \boldsymbol{\Sigma}, r, \boldsymbol{\gamma})\pi(\mathbf{F} \mid r) \\
&\propto \exp\left(-\frac{1}{2}\left\|(\mathbf{Y} - \mathbf{X}_{\boldsymbol{\gamma}}\mathbf{A}_{\boldsymbol{\gamma}}\mathbf{B}^\top)\boldsymbol{\Sigma}^{-\frac{1}{2}}\right\|_F^2\right)\exp\left(-\frac{1}{2\nu_2}\text{vec}(\mathbf{F}^\top)^\top\text{vec}(\mathbf{F}^\top)\right) \\
&\propto \exp\left\{-\frac{1}{2}(\text{vec}(\mathbf{F}^\top) - \boldsymbol{\mu}^{\mathbf{F}})^\top\boldsymbol{\Sigma}^{\mathbf{F}-1}(\text{vec}(\mathbf{F}^\top) - \boldsymbol{\mu}^{\mathbf{F}})\right\},
\end{aligned}
$$

where

$$\begin{aligned}
\boldsymbol{\mu}^{\mathbf{F}} &= \left\{(\boldsymbol{\Sigma}^{-1} \otimes \mathbf{A}_{\boldsymbol{\gamma}}^{\top}\mathbf{X}_{\boldsymbol{\gamma}}^{\top}\mathbf{X}_{\boldsymbol{\gamma}}\mathbf{A}_{\boldsymbol{\gamma}})_{[\Omega^*,\Omega^*]} + \nu_2^{-1}\mathbf{I}_{(qr-r^2)}\right\}^{-1} \\
&\quad \times \left\{\mathrm{vec}(\mathbf{A}_{\boldsymbol{\gamma}}^{\top}\mathbf{X}_{\boldsymbol{\gamma}}^{\top}\mathbf{Y}\boldsymbol{\Sigma}^{-1})_{\Omega^*} - (\boldsymbol{\Sigma}^{-1} \otimes \mathbf{A}_{\boldsymbol{\gamma}}^{\top}\mathbf{X}_{\boldsymbol{\gamma}}^{\top}\mathbf{X}_{\boldsymbol{\gamma}}\mathbf{A}_{\boldsymbol{\gamma}})_{[\Omega^*,-\Omega^*]}\mathrm{vec}(\mathbf{I}_r)\right\}, \\
\boldsymbol{\Sigma}^{\mathbf{F}} &= \left\{(\boldsymbol{\Sigma}^{-1} \otimes \mathbf{A}_{\boldsymbol{\gamma}}^{\top}\mathbf{X}_{\boldsymbol{\gamma}}^{\top}\mathbf{X}_{\boldsymbol{\gamma}}\mathbf{A}_{\boldsymbol{\gamma}})_{[\Omega^*,\Omega^*]} + \nu_2^{-1}\mathbf{I}_{(qr-r^2)}\right\}^{-1}.
\end{aligned}$$

Hence, this implies that $\mathrm{vec}(\mathbf{F}^{\top}) \mid \mathbf{Y}, r, \boldsymbol{\gamma}, \mathbf{A}_{\boldsymbol{\gamma}}, \boldsymbol{\Sigma} \sim \mathcal{N}(\boldsymbol{\mu}^{\mathbf{F}}, \boldsymbol{\Sigma}^{\mathbf{F}})$.

A.2 Derivation of (2.7)

It follows from a direct calculation that

$$\begin{aligned}
&\pi(\mathbf{A}_{\boldsymbol{\gamma}} \mid \mathbf{Y}, r, \boldsymbol{\gamma}, \mathbf{B}, \boldsymbol{\Sigma}) \propto f(\mathbf{Y} \mid \mathbf{A}_{\boldsymbol{\gamma}}, \mathbf{B}, \boldsymbol{\Sigma}, r, \boldsymbol{\gamma})\pi(\mathbf{A}_{\boldsymbol{\gamma}} \mid \boldsymbol{\gamma}, r) \\
&\propto \exp\left(-\frac{1}{2}\left\|(\mathbf{Y} - \mathbf{X}_{\boldsymbol{\gamma}}\mathbf{A}_{\boldsymbol{\gamma}}\mathbf{B}^{\top})\boldsymbol{\Sigma}^{-\frac{1}{2}}\right\|_F^2\right)\exp\left(-\frac{1}{2\nu_1}\left\|\mathbf{A}_{\boldsymbol{\gamma}}\right\|_F^2\right) \\
&\propto \exp\left\{-\frac{1}{2}\mathrm{tr}\left(\mathbf{X}_{\boldsymbol{\gamma}}\mathbf{A}_{\boldsymbol{\gamma}}\mathbf{B}^{\top}\boldsymbol{\Sigma}^{-1}\mathbf{B}\mathbf{A}_{\boldsymbol{\gamma}}^{\top}\mathbf{X}_{\boldsymbol{\gamma}}^{\top} - 2\mathbf{Y}^{\top}\mathbf{X}_{\boldsymbol{\gamma}}\mathbf{A}_{\boldsymbol{\gamma}}\mathbf{B}^{\top}\boldsymbol{\Sigma}^{-1} + \frac{1}{\nu_1}\mathbf{A}_{\boldsymbol{\gamma}}\mathbf{A}_{\boldsymbol{\gamma}}^{\top}\right)\right\} \\
&\propto \exp\left[-\frac{1}{2}\left\{\mathrm{vec}(\mathbf{A}_{\boldsymbol{\gamma}})^{\top}\mathbf{W}\mathrm{vec}(\mathbf{A}_{\boldsymbol{\gamma}}) - 2\mathbf{v}^{\top}\mathrm{vec}(\mathbf{A}_{\boldsymbol{\gamma}})\right\}\right],
\end{aligned}$$

where $\mathbf{W} = (\boldsymbol{\Sigma}^{-1/2}\mathbf{B} \otimes \mathbf{X}_{\boldsymbol{\gamma}})^{\top}(\boldsymbol{\Sigma}^{-1/2}\mathbf{B} \otimes \mathbf{X}_{\boldsymbol{\gamma}}) + \nu_1^{-1}\mathbf{I}_{p_{\boldsymbol{\gamma}}r}$ and $\mathbf{v} = (\boldsymbol{\Sigma}^{-1}\mathbf{B} \otimes \mathbf{X}_{\boldsymbol{\gamma}})^{\top}\mathrm{vec}(\mathbf{Y})$. This leads to $\mathrm{vec}(\mathbf{A}_{\boldsymbol{\gamma}}) \mid \mathbf{Y}, r, \boldsymbol{\gamma}, \mathbf{B}, \boldsymbol{\Sigma} \sim \mathcal{N}_{p_{\boldsymbol{\gamma}}r}(\boldsymbol{\mu}^{\mathbf{A}_{\boldsymbol{\gamma}}}, \boldsymbol{\Sigma}^{\mathbf{A}_{\boldsymbol{\gamma}}})$, where

$$\begin{aligned}
\boldsymbol{\mu}^{\mathbf{A}_{\boldsymbol{\gamma}}} &= \left\{(\boldsymbol{\Sigma}^{-1/2}\mathbf{B} \otimes \mathbf{X}_{\boldsymbol{\gamma}})^{\top}(\boldsymbol{\Sigma}^{-1/2}\mathbf{B} \otimes \mathbf{X}_{\boldsymbol{\gamma}}) + \nu_1^{-1}\mathbf{I}_{p_{\boldsymbol{\gamma}}r}\right\}^{-1}(\boldsymbol{\Sigma}^{-1}\mathbf{B} \otimes \mathbf{X}_{\boldsymbol{\gamma}})^{\top}\mathrm{vec}(\mathbf{Y}), \\
\boldsymbol{\Sigma}^{\mathbf{A}_{\boldsymbol{\gamma}}} &= \left\{(\boldsymbol{\Sigma}^{-1/2}\mathbf{B} \otimes \mathbf{X}_{\boldsymbol{\gamma}})^{\top}(\boldsymbol{\Sigma}^{-1/2}\mathbf{B} \otimes \mathbf{X}_{\boldsymbol{\gamma}}) + \nu_1^{-1}\mathbf{I}_{p_{\boldsymbol{\gamma}}r}\right\}^{-1}.
\end{aligned}$$

A.3 Derivation of (2.8)

A direct calculation shows that

$$\begin{aligned}
&\pi(\boldsymbol{\Sigma} \mid \mathbf{Y}, r, \boldsymbol{\gamma}, \mathbf{A}_{\boldsymbol{\gamma}}, \mathbf{B}) \propto f(\mathbf{Y} \mid \mathbf{A}_{\boldsymbol{\gamma}}, \mathbf{B}, \boldsymbol{\Sigma}, r, \boldsymbol{\gamma}) \pi(\boldsymbol{\Sigma}) \\
&\propto |\boldsymbol{\Sigma}|^{-\frac{n}{2}} \exp\left\{ -\frac{1}{2} \left\| (\mathbf{Y} - \mathbf{X}_{\boldsymbol{\gamma}} \mathbf{A}_{\boldsymbol{\gamma}} \mathbf{B}^\top) \boldsymbol{\Sigma}^{-\frac{1}{2}} \right\|_F^2 \right\} |\boldsymbol{\Sigma}|^{-\frac{\nu_0 + q + 1}{2}} \exp\left\{ -\frac{1}{2} \mathrm{tr}(\boldsymbol{\Psi}_0 \boldsymbol{\Sigma}^{-1}) \right\} \\
&\propto |\boldsymbol{\Sigma}|^{-\frac{\nu_0 + q + 1 + n}{2}} \exp\left\{ -\frac{1}{2} \mathrm{tr}\left(\mathbf{S} \boldsymbol{\Sigma}^{-1} \right) \right\},
\end{aligned}$$

where $\mathbf{S} = (\mathbf{Y} - \mathbf{X}_{\boldsymbol{\gamma}} \mathbf{A}_{\boldsymbol{\gamma}} \mathbf{B}^\top)^\top (\mathbf{Y} - \mathbf{X}_{\boldsymbol{\gamma}} \mathbf{A}_{\boldsymbol{\gamma}} \mathbf{B}^\top) + \boldsymbol{\Psi}_0$. Therefore, we have

$$\boldsymbol{\Sigma} \mid \mathbf{Y}, \mathbf{A}_{\boldsymbol{\gamma}}, \mathbf{B}, \boldsymbol{\mu}, r, \boldsymbol{\gamma} \sim \mathcal{W}^{-1}(n + \nu_0, \mathbf{S}).$$