Sparse Bayesian kernel learning for high-dimensional regression and classification

by

Weikang Duan

M.S., University of Minnesota, Twin Cities, 2016

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 the past decades, statistical learning has been an increasingly popular topic that has drawn a significant amount of attention from researchers. Kernel based nonlinear models, in particular, are powerful tools due to their flexibility to extract information from complex datasets. A major challenge with the kernel modeling in the current big data era is the curse of dimensionality. Although an abundance of variable selection methods have been proposed, the developments in high-dimensional Bayesian kernel models is still in its infancy. In addition to the variable selection, the innate nature of kernel based models induces heavy computational costs, which further prohibit the application of related methods. The goal of this dissertation is to develop new, fast variable selection and prediction procedures in order to address the problem of high-dimensional nonlinear regression and classification from the Bayesian perspective. To reduce the computational cost, we propose a novel hybrid search algorithm and the Bayesian doubly-sparse frameworks to the kernel based models.

In Chapter 1, we discuss the background, existing methods and their limitations. We also give the motivation for our study. In Chapter 2, we propose a Bayesian model hybrid search algorithm for Gaussian process (GP) regression models, which quickly scan through the model space to search for a set of models with high posterior probabilities. In addition, we address the massive and high-dimensional data problem for GP by proposing an approach which combines quantile subsample hybrid search with a nearest neighbor GP scheme. In Chapter 3, we propose a novel Bayesian doubly-sparse framework to the reproducing kernel Hilbert space (RKHS) regression models. The proposed doubly-sparse frame work performs both variable selection and sparse kernel matrix estimation. In Chapter 4, we extend our proposed Bayesian doubly-sparse framework to the nonlinear Bayesian support vector machine.

Sparse Bayesian kernel learning for high-dimensional regression and classification

by

Weikang Duan

M.S., University of Minnesota, Twin Cities

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:

Major Professor
Dr. Gyuhyeong Goh

Copyright

Abstract

In the past decades, statistical learning has been an increasingly popular topic that has drawn a significant amount of attention from researchers. Kernel based nonlinear models, in particular, are powerful tools due to their flexibility to extract information from complex datasets. A major challenge with the kernel modeling in the current big data era is the curse of dimensionality. Although an abundance of variable selection methods have been proposed, the developments in high-dimensional Bayesian kernel models is still in its infancy. In addition to the variable selection, the innate nature of kernel based models induces heavy computational costs, which further prohibit the application of related methods. The goal of this dissertation is to develop new, fast variable selection and prediction procedures in order to address the problem of high-dimensional nonlinear regression and classification from the Bayesian perspective. To reduce the computational cost, we propose a novel hybrid search algorithm and the Bayesian doubly-sparse frameworks to the kernel based models.

In Chapter 1, we discuss the background, existing methods and their limitations. We also give the motivation for our study. In Chapter 2, we propose a Bayesian model hybrid search algorithm for Gaussian process (GP) regression models, which quickly scan through the model space to search for a set of models with high posterior probabilities. In addition, we address the massive and high-dimensional data problem for GP by proposing an approach which combines quantile subsample hybrid search with a nearest neighbor GP scheme. In Chapter 3, we propose a novel Bayesian doubly-sparse framework to the reproducing kernel Hilbert space (RKHS) regression models. The proposed doubly-sparse frame work performs both variable selection and sparse kernel matrix estimation. In Chapter 4, we extend our proposed Bayesian doubly-sparse framework to the nonlinear Bayesian support vector machine.

Contents

List of Figures

List of Tables

Acknowledgments

Looking back, my journey at graduate school has been full of ups and downs. I would like to express my great gratitude to the many people in my life who had a tremendous impact on me.

First of all, I would like to give my greatest thanks to my advisor, Dr. Gyuhyeong Goh. Throughout my graduate years at Kansas state university, he was always there to help me, instruct me, give me guidance and encourage me. Under his guidance not only through research work, but also through his classes, I was able to lay a solid foundation for statistics. In addition to technical knowledge and research works, his work ethics, patience and humility have also carved a deep mark on me. Without him, I would not be the same person that I am today.

I would also like to express my gratitude to Dr. Weixing Song, Dr. Jingru Mu and Dr. Jisang Yu for their support and time to serve on my committee. Their great insights and advice have helped me significantly in revising my research works and in forming my dissertation.

To Dr. Wei-wen Hsu, who helped me and guided me a lot through his theory class. To Dr. Christopher Vahl, Dr. Trevor Hefley, Dr. Abby Jager, Dr. Michael Higgins, Dr. Jieun Lee for their help and guidance throughout the past years. To Bonnie Messmer and Jo Blackburn for their help on all those administrative and paper works. In addition, I would like to thank my fellow graduate students, Dr. Shiqiang Jin, Dr. Jia Liang, Dr. Weijia Jia, Linruo Guo, Rigele Te, Dunfu Yang, Shengnan Chen for their friendship and support. Great thanks to the Department of Statistics for giving me this precious chance and full financial support, which enabled me to go through this long journey.

I would like to say thank you to my great friend and mentor Ryan Curtis Jackson, who brought me to church and shared the gospel with me. Also, great thanks to Dr. Jianxiong

Li for his fellowship and guidance. I would also like to say thank you to brothers and sisters at Ichthus Fellowship house church and MCCF.

I would like to express my greats thanks to my families. To my parents and my brother for their unconditional love and support. I would also like to express my special gratitude to my wife, Candace, one of the best gifts from God. For her love, for our tears and laughter together, for her delicious egg noodles, Brussels sprouts, Korean side dishes and for making me hot tea every early morning.

At last, I would like thank God, Jesus Christ, who never gave up on me and opened this door for me, brought me to Kansas and gave me a chance to know Him and redeem myself.

Chapter 1

Introduction

With the development of technology in the big data era, scientists are able to collect massive complex datasets with high dimensions. To extract useful information from those datasets, many methods have been developed in machine learning and statistics (Hastie et al., 2009). Among those methods, the kernel based nonlinear models (Smola and Schölkopf, 1998) have been a very popular tool in particular.

Even though the kernel based method can be very useful, its performance can also suffer from the curse of dimensionality. In particular, as the number of irrelevant variables increases, more noises are added into the model and so it results in erroneous estimated kernels and poor predictions. Fan and Lv (2008) discussed that including all variables for prediction is essentially as random guessing for binary classification. Hence, the problem of variable selection is one of the biggest concerns on kernel learning methods with high-dimensional data. Other than variable selection, another aspect that hinders the application of kernel learning models is the heavy burden of the computational cost.

Even though many methods have been proposed to address the variable selection problem or deal with the large sample size, challenges still remain. One in particular is that the existing methods for performing variable selection still suffer from heavy computational costs. Another problem is how to conduct variable selection under the large sample size for kernel based models.

1.1 Variable selection for high dimensional Gaussian process (GP) regression

The Gaussian process (GP) has been one of the most popular Bayesian tools for statistics and machine learning research. Within the field of machine learning, GP has been widely used for supervised learning tasks. In the statistics literature, GP based models are popular for a variety of nonlinear modeling problems. Even though GP has drawn much attention from researchers, the problem of variable selection in the GP regression framework is less addressed. When the size of potential predictors increases, the elimination of irrelevant variables can greatly improve the model fits as well as the prediction accuracy.

There are some attempts to tackle variable selection problems for GP regression modeling. For example, Linkletter et al. (2006) proposed a Bayesian variable selection method for GP regression by employing a mixture prior such that spikes at zero on the kernel bandwidth parameters which corresponds to the irrelevant predictors. Savitsky et al. (2011) then formulated a unified approach of GP models for exponential dispersion family data and survival data, encompassing Linkletter et al. (2006)'s variable selection method as a special case. They proposed a general Metropolis-Hastings algorithm within a Gibbs sampling scheme. However, the proposed algorithm requires to go through each variable for every iteration. As a result, the algorithm suffers from many computational burdens including reconstructing the kernel matrix, conducting the inversion computation, and calculating the determinant of the kernel matrix. Those computational operations require the computational cost of $O(n^3)$, where n is the sample size of the data. Hence, for one iteration of the algorithm, the computational cost is $O(pn^3)$ where p is the total number of predictors. This greatly increases the computational cost as p increases. Note that when p is large, this method is nearly infeasible.

From the frequentist point of view, Yi et al. (2011) and Yan and Qi (2010) proposed penalized approaches by asserting penalties on the bandwidth parameters. However, choosing tuning parameters can greatly increases the computational cost. In addition, the penal-

ized likelihood approach relies on only one single best model and thus it ignores the model uncertainties.

Another aspect that hinders the application of GP based models is the high-dimensional massive data setting where both p and n are large. To address the large n problem, many ideas have been proposed in the literature (e.g. Liu et al., 2020; Williams and Rasmussen, 2006). One popular approach is to approximate the large kernel matrix by a low-rank matrix. For instance, using the Nyström approximation Williams and Seeger (2001) achieved the computational cost reduction from $O(n^3)$ to $O(nm^2)$. Some researchers have paid attention to the subset of data approach (SoD) that uses a smaller set of representative m samples instead of the original n samples. The SoD method leads to a reduced cost of $O(m^3)$. In addition, Seeger et al. (2003); Snelson and Ghahramani (2006a, 2007) proposed to approximate the likelihood by assuming conditional independence of training points and testing points given m *inducing points* such that $m \ll n$. The computational cost of those approaches are generally $O(nm^2)$. Furthermore, Datta et al. (2016a,b); Gramacy et al. (2016); Gramacy and Apley (2015); Kim et al. (2005); Gramacy and Haaland (2016) proposed the localized regression approach or local krigging approach based on the fact that points far away play little role in prediction. However, despite many advances in large n problems, the variable selection problem in large n settings has not been explored yet.

1.2 Variable selection for high dimensional reproducing kernel Hilbert space (RKHS) regression

For variable selection in RKHS regression models, several approaches have been proposed in the literature. For example, Gao et al. (2010); Allen (2013) proposed the penalized approach from a frequentist perspective. From a Bayesian perspective, Liang et al. (2007); Chakraborty (2009) proposed sparsity inducing prior approaches by assuming point mass priors on the kernel bandwidth parameters. In addition, Crawford et al. (2018) proposed a projection method in which variable selection can be conducted using a thresholding ap-

proach. One of the limitations of the existing methods is that model uncertainties have been ignored. As discussed in Barbieri and Berger (2004); Hoeting et al. (1999), prediction with a single model could lead to poor performance.

Even though many approaches for variable selection in the RKHS method have been proposed, vector selection or sparse matrix estimation were not considered in the existing works. As suggested by Zhang et al. (2016), the use of all points for representing the whole function can be less optimal if the underlying function has a sparse representation. To address this issue, Zhang et al. (2016) proposed a data sparsity constraint approach. In a Bayesian framework, Tipping (2001) suggested to employ a subset of vectors to represent the whole function so that it leads to the reduction in the computational cost. Zhang et al. (2008) showed the posterior consistency of the so-called Silver g-prior, which is commonly used for sparse Bayesian kernel regression models. With the theoretical work, Zhang et al. (2011) proposed using sparse priors on vector extraction to reduce the computational cost. However, those works were developed under the nonexistence of many irrelevant predictors.

To address both variable selection and sparse kernel matrix estimation, Chen et al. (2018) proposed a double sparse kernel learning (DoSK) by appending double L_1-penalties to the cost function. With the double L_1-penalties, one can achieve double sparsity on both variable selection and vector selection. Some asymptotic theoretical properties have also been discussed in their work. However, the penalty likelihood approach has limitations. One is that the choice of the penalty weights, which is usually conducted via cross validation. The cross validation procedure for tuning two penalty weights results in the massive computational cost. In addition, the double L_1-penalty approach did not address the uncertainties associated with both variable selection and active vector selection. Furthermore, the existing approach is not applicable to the large sample size problem.

1.3 Variable selection for high dimensional nonlinear support vector machine

The support vector machine (SVM) (Cortes and Vapnik, 1995) has been one of the most popular methods in machine learning research since its introduction. It has drawn a lot of attention from researchers and been widely used in many fields, such as image classification, speech recognition, etc. With the successful application and popularity, the SVM was then extended to a Bayesian framework. For instance, Mallick et al. (2005) proposed a Bayesian SVM by appending a Gaussian like error to the linear predictors to get a tractable likelihood. The greatest breakthrough in Bayesian SVM came with the invention of the data augmentation approach in Polson and Scott (2011). In the data augmentation SVM method, the hinge loss is represented as a form of the Bayesian hierarchical model, so that MCMC and EM algorithms can be used for posterior inference. The Bayesian SVM approach provides several advantages over the frequentist approaches, including automatic hyperparameter tuning and predictive uncertainty quantification. In addition to the Bayesian linear SVM, the extension to nonlinear SVM was considered by Henao et al. (2014) in which the GP prior was further assumed for an unknown underlying function.

However, SVM can perform poorly when there exist many irrelevant predictors. Hence, many works have focused on addressing variable selection problems in SVM. For instance, Bradley and Mangasarian (1998); Zhu et al. (2003); Wang et al. (2006); Zhang et al. (2006); Zou and Yuan (2008); Becker et al. (2011) proposed penalization methods for linear SVM from a frequentist perspective. For variable selection of nonlinear SVM, Zhang (2006) proposed a smoothing spline framework to perform feature selection and classification simultaneously. In addition, Mangasarian and Kou (2007) proposed an approach for variable selection by inserting a diagonal indicator matrix into the kernel matrix. Many works for variable selection from a Bayesian perspective have also been proposed. For instance, Marchiori and Sebag (2005); Luts and Ormerod (2014); Sun et al. (2018) have proposed the Bayesian methods for linearly separable data. For a nonlinear framework, Sun et al. (2019)

proposed an approach by incorporating graph information on features.

Even though many works for variable selection has been proposed as discussed above, many limitations still exist for variable selection of nonlinear Bayesian SVM. One limitation is that the existing work has not addressed the sparse kernel matrix estimation. As suggested in Zhang et al. (2016), including all data points for training the kernel based model can lead to suboptimal results in some cases. In addition, as proposed by Tipping (2001); Zhang et al. (2008, 2011), the use of a subset of active vectors for model fitting can greatly reduce the computational cost and obtain similar or even better prediction accuracy. However, those existing methods for sparse kernel matrix estimation have ignored the variable selection issue. Hence, they would suffer from the curse of dimensionality in the presence of many noisy variables.

To address both sparse kernel estimation and variable selection problems together, Chen et al. (2018) proposed the doubly sparse kernel learning by appending double L_1-penalty to the cost function. However, the limitations of this work are the choice of the tuning parameters and prediction uncertainty quantification.

1.4 Motivation and dissertation outline

To deal with limitations of the existing methods, this dissertation aims to develop novel, fast variable selection and prediction procedures for kernel learning models from a Bayesian perspective. In particular, we propose a novel hybrid model search algorithm for high dimensional GP regression. In addition, we propose a novel Bayesian doubly-sparse framework for regression and classification problems using the RKHS approach. The rest of the dissertation is organized as follows.

In Chapter 2, we address challenges in high dimensional GP regression modeling, which is the most popular Bayesian approach to nonlinear regression. We develop a novel Bayesian model hybrid search algorithm to quickly scan through the model space to search for a set of models having high posterior probabilities. Prediction is then conducted via the notion of Bayesian model averaging. In addition to the variable selection problem, another challenge

is how to deal with variable selection under the case of massive sample size. To address the massive data situation, we propose an approach which incorporates a quantile-based subsample selection idea into the nearest neighbor GP framework.

In Chapter 3, we focus on the variable selection problem for RKHS models under the nonlinear regression framework. For RKHS models, simultaneous variable selection and sparse kernel estimation, often referred to as the doubly sparse estimation problem, are needed. To address this problem, we propose a Bayesian doubly sparse RKHS regression method via double spike and slab priors. A key merit of our proposed approach is to factor the sparse kernel matrix estimation into the variable selection procedures, which allows a fast Markov Chain Monte Carlo (MCMC) implementation. In addition, our proposed method does not require to select a single best model and a single best sparse kernel representation. In the proposed framework, all candidate models and sparse kernel representations are automatically integrated thorough the MCMC integration.

In Chapter 4, we extend our propose doubly-sparse framework to the nonlinear Bayesian support vector machine via the data augmentation method of Polson and Scott (2011). The performance of proposed method is examined by real data applications.

Chapter 5 concludes the dissertation with some remarks and future directions. Some technical details about our full conditional derivations are given in Appendix.

Chapter 2

Bayesian hybrid model search for sparse Gaussian process regression

In this chapter, we develop a novel hybrid model search algorithm for sparse Gaussian process (GP) regression. In Section 2.1, we briefly present the model setup and an illustrative example for showing the need of variable selection in GP regression modeling. In Section 2.2, we present our proposed hybrid search algorithm under the GP regression framework and also give details of our prediction procedure. In Section 2.3, we address high-dimensional massive data problems by developing a new approach using the quantile-based subsample hybrid model search and the nearest neighbor GP method. We report the results of simulation experiments and real data analysis in Section 2.4 and Section 2.5 respectively. We conclude this chapter in Section 2.6.

2.1 Basic set-up and motivation

2.1.1 The model

Consider a nonlinear regression problem:

$$Y = f(X) + \epsilon,$$

where $Y \in \mathbb{R}$ is the response variable, $X \in \mathbb{R}^p$ is the vector-valued input variable, ϵ is the random noise, and $f : \mathbb{R}^p \to \mathbb{R}$ is an unknown function, which is of our interest. We assume that $f \sim \text{GP}$ and $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

Suppose that our data set $\{(x_i, y_i) : i = 1, \cdots, n\}$ consists of n independent realizations of (X, Y). Let $\boldsymbol{y} = (y_1, \ldots, y_n)^\top$ be the response vector and $\boldsymbol{x} = [x_1, x_2, \cdots, x_n]^\top$ be the matrix of input values with $x_i = (x_{i1}, x_{i2}, \cdots, x_{ip})^\top$. With this given data, let $\boldsymbol{f} = (f(x_1), f(x_2), \cdots, f(x_n))^\top$ be a vector of the unknown function evaluated at $\boldsymbol{x}$. Then, a GP prior would lead $\boldsymbol{f}$ to a multivariate Gaussian distribution, that is,

$$\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{m}(\boldsymbol{x}), \boldsymbol{K}_{\boldsymbol{\theta}}(\boldsymbol{x})),$$

where $\boldsymbol{m}(\boldsymbol{x})$ is the n by 1 mean vector and $\boldsymbol{K}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \{K(x_i, x_j|\boldsymbol{\theta})\}_{n\times n}$ is the n by n kernel matrix governed by the hyperparameters $\boldsymbol{\theta}$. For convenience, we assume zero mean GP for $\boldsymbol{f}$, that is, $\boldsymbol{m}(\boldsymbol{x}) = \boldsymbol{0}$.

Note it is well known that by the conjugacy of our problem, we can get the marginal likelihood as

$$\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}, \sigma^2 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{K}_{\boldsymbol{\theta}}(\boldsymbol{x}) + \sigma^2\mathbb{I}) \tag{2.1}$$

by integrating out the function $\boldsymbol{f}$. Then this essentially becomes a regression model where response $\boldsymbol{y}$ depends on the kernel matrix $\boldsymbol{K}_{\boldsymbol{\theta}}(\boldsymbol{x})$, which is a function of input data $\boldsymbol{x}$ and hyperparameters $\boldsymbol{\theta}$.

The key component for the regression problem discussed above is the kernel matrix $\boldsymbol{K}_{\boldsymbol{\theta}}(\boldsymbol{x})$. There are many forms to model it. Among those forms, one popular choice is the Gaussian form. For variable selection purpose, we introduce an index set $\boldsymbol{\gamma} \subset \{1, \ldots, p\}$ to the kernel matrix. Let x_i and x_j denote the i^{th} and j^{th} observations of training data $\boldsymbol{x}$. Then the $(i, j)^{th}$ term of the kernel matrix is defined as

$$K(x_i, x_j|\boldsymbol{\theta}, \boldsymbol{\gamma}) = \lambda \exp\left\{-\frac{1}{\tau}\sum_{k=1}^{p} I(k \in \boldsymbol{\gamma})(x_{ik} - x_{jk})^2\right\} \tag{2.2}$$

where the hyperparameter $\boldsymbol{\theta} = (\lambda, \tau)^\top$. (Note we borrow the kernel form in Quinonero-Candela et al. (2007) such that we assume a single bandwidth parameter τ.) Among $\boldsymbol{\theta}$, λ controls the magnitude of the covariance and τ controls the smoothness of the function. In addition, if $k \notin \boldsymbol{\gamma}$, the k^{th} predictor is excluded from constructing the kernel matrix. Otherwise, the k^{th} predictor is included. This formulation can be easily extended to other popular kernel forms, including the matérn kernel form.

Let $\boldsymbol{x}^* = [x_1^*, \ldots, x_{n^*}^*]^\top$ be an $n^* \times p$ matrix of new points at which we are interested in making prediction. Letting $\boldsymbol{f}^* = \boldsymbol{f}(\boldsymbol{x}^*)$, the predictive distribution is $\boldsymbol{f}^*|\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{K}^*)$ with

$$\boldsymbol{\mu}^* = \boldsymbol{K}(\boldsymbol{x}^*, \boldsymbol{x})\{\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}) + \sigma^2\mathbb{I}\}^{-1}\boldsymbol{y}$$
$$\boldsymbol{K}^* = \boldsymbol{K}(\boldsymbol{x}^*, \boldsymbol{x}^*) - \boldsymbol{K}(\boldsymbol{x}^*, \boldsymbol{x})\{\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}) + \sigma^2\mathbb{I}\}^{-1}\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}^*),$$

where $K(\boldsymbol{x}^*, \boldsymbol{x}) = \{K(x_i^*, x_j)\}_{n^*,n}$. To account for the random noise ϵ, we can use $\boldsymbol{y}^*|\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}^*, \boldsymbol{K}^* + \sigma^2\mathbb{I})$.

2.1.2 Why variable selection

With the model defined and discussed above, if $\gamma_k = 0$, then the k^{th} variable is excluded from constructing the kernel matrix, i.e., the k^{th} variable is not associated with the response $\boldsymbol{y}$. What will happen if we include those nuisance features into our models? We present a 'toy' example to show that as the number of those nuisance features increases, the prediction accuracy and the model fit of the GP regression decrease.

Let $y = 3sin(x_1) + \epsilon$ be the true model, where $\epsilon \sim \mathcal{N}(0, 1)$. Note we abuse the notation a bit here by letting x_k denote the k^{th} column of $\boldsymbol{x}$. With this 'true model', we generate samples for each x_k by $x_k \sim \mathcal{U}(-\pi, \pi)$ with $p = 100$ and $n = 200$. Given the 200 samples, we randomly divide them into 100 samples for training and another 100 for testing. With this setup, we fit three GP regression models with $p = 1, 20, 100$ respectively. That is, we fit

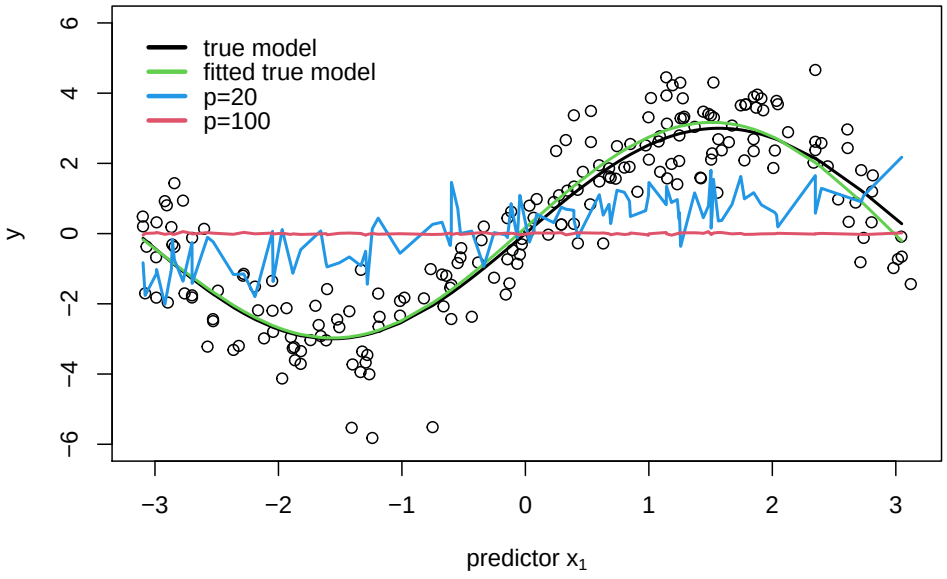


Figure 2.1: Fitted curve for GP regression models with different dimension

three GP regression models with the first feature, first 20 features and all 100 features:

$$\text{Model 1: } y = f(x_1) + \epsilon,$$

$$\text{Model 2: } y = f(x_1, x_2, \cdots, x_{20}) + \epsilon,$$

$$\text{Model 3: } y = f(x_1, x_2, \cdots, x_{100}) + \epsilon.$$

We train each regression model with the Gaussian kernel matrix defined in equation (2.2) via the Laplace approximation method discussed in Section 2.2.4. Note that only the first one, x_1, is truly associated with the response, y. Hence, the rest of the variables are considered as irrelevant. Figure 2.1 shows the true curve f and other three fitted curves for model 1, model 2, and model 3. The plot suggests that as the number of noisy variables increases, the fitted curve deviates further from the true curve. Adding irrelevant variables distorts the kernel matrix and furthermore it leads to poor prediction.

Other than the visualization, to check the prediction results, we repeat this experiment

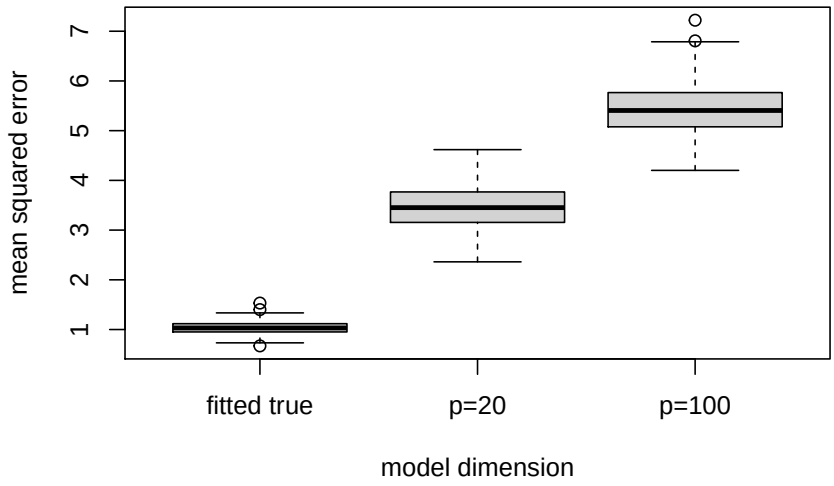


Figure 2.2: MSE for different model dimensions

100 times and record the prediction Mean Squared Error (MSE) as well as the mean squared bias on the testing set for each experiment. We define the mean squared bias as $\frac{1}{n_t}\sum_{i=1}^{n_t}(\hat{f}_i - f_i)^2$ and MSE as $\frac{1}{n_t}\sum_{i=1}^{n_t}(\hat{y}_i - y_i)^2$ where n_t denotes the testing set size, $\hat{f}_i$ denotes predicted value for the i^{th} point, and f_i and y_i are the actual i^{th} values of f and y respectively. We summarize the experimental average bias and prediction MSE in Figures 2.3 and 2.2. Both plots suggest that the bias and the MSE increase as the number of nuisance variables increases. This demonstrates that variable selection for GP is necessary.

2.2 Model adaptation, variable selection, and prediction

Given the necessity of variable selection, we present our approach for Bayesian model fitting and prediction in this section.

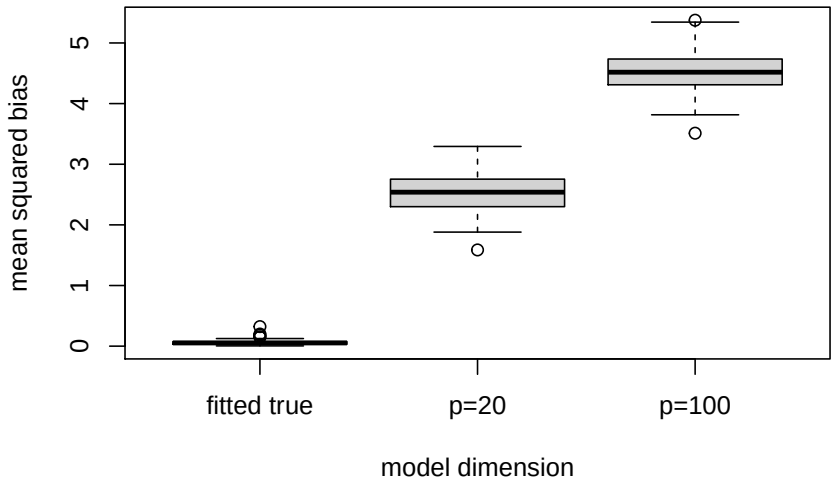


Figure 2.3: Bias for different model dimensions

2.2.1 The posterior

Recall that $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \cdots, \gamma_p)^\top$ is the binary indicator vector that governs the feature selection result and $\boldsymbol{\theta}$ consists of the kernel matrix hyperparameters. For instance, if we model kernel matrix $\boldsymbol{K}$ with the Gaussian form, then $\boldsymbol{\theta} = (\lambda, \tau)^\top$. In addition, we assume a prior, $\pi(\boldsymbol{\theta}, \sigma^2|\boldsymbol{\gamma})\pi(\boldsymbol{\gamma})$. The posterior is then given as

$$\pi(\boldsymbol{\theta}, \sigma^2, \boldsymbol{\gamma}|\boldsymbol{y}) \propto \pi(\boldsymbol{y}|\boldsymbol{\theta}, \sigma^2, \boldsymbol{\gamma})\pi(\boldsymbol{\theta}, \sigma^2|\boldsymbol{\gamma})\pi(\boldsymbol{\gamma}).$$

To perform posterior inference, as discussed in Savitsky et al. (2011), a Metropolis-Hastings algorithm within the Gibbs sampler can be considered to draw samples from $\pi(\boldsymbol{\gamma}|\boldsymbol{\theta}, \sigma^2, \boldsymbol{y})$, $\pi(\boldsymbol{\theta}|\boldsymbol{\gamma}, \sigma^2, \boldsymbol{y})$, and $\pi(\sigma^2|\boldsymbol{\gamma}, \boldsymbol{\theta}, \boldsymbol{y})$, iteratively. However, sampling from those three conditional marginal distributions can be computationally expensive. In particular, the computational cost for sampling from $\pi(\boldsymbol{\gamma}|\boldsymbol{\theta}, \sigma^2, \boldsymbol{y})$ can be extremely high especially when p is large. This is because for each γ_i, the algorithm needs to reform the kernel matrix and compute the marginal likelihood, whose computational cost is $O(n^3)$. For sampling from $\pi(\boldsymbol{\theta}|\boldsymbol{\gamma}, \sigma^2, \boldsymbol{y})$

and $\pi(\sigma^2|\boldsymbol{\gamma}, \boldsymbol{\theta}, \boldsymbol{y})$, we also need to reform the kernel and compute the determinant as well as the inverse of the kernel matrix for each iteration, and this also charges a computational cost of $O(n^3)$.

To alleviate the computational burden, we borrow the idea of the collapsed Gibbs sampler approach (Liu, 1994) and Bayesian model averaging (Hoeting et al., 1999). Specifically, we propose to draw samples from $\pi(\boldsymbol{\gamma}|\boldsymbol{y})$ and $\pi(\boldsymbol{\theta}, \sigma^2|\gamma, \boldsymbol{y})$ respectively. However, like the classical Gibbs sampler, sampling from those posterior distributions are still computationally expensive. To be more specific, sampling from $\pi(\boldsymbol{\gamma}|\boldsymbol{y})$ still needs to go through each $\gamma_i \in \boldsymbol{\gamma}$. In addition, sampling from $\pi(\boldsymbol{\theta}, \sigma^2|\boldsymbol{\gamma}, \boldsymbol{y})$ via the Metropolis-Hastings algorithm can be costly in each MCMC iteration. Other than that, the determination of a good proposal distribution is a challenging issue.

Our solution to the aforementioned obstacles is as follows. For sampling from the parameter posterior $\pi(\boldsymbol{\theta}, \sigma^2|\boldsymbol{\gamma}, \boldsymbol{y})$, we use the Laplace approximation (See Section 2.2.3). More importantly, for making inference for $\pi(\boldsymbol{\gamma}|\boldsymbol{y})$, instead of the sampling based methods, we propose a similar hybrid search algorithm (Jin and Goh, 2021) under the GP regression framework that quickly converges to models with high posterior probabilities. After identifying acceptable models in set $\mathcal{A}$, using the Bayesian model averaging techniques (Madigan and Raftery, 1994; Hoeting et al., 1999), we construct the posterior distribution of $\boldsymbol{\theta}$ and σ^2 as follow:

$$\pi(\boldsymbol{\theta}, \sigma^2|\boldsymbol{y}) = \sum_{\boldsymbol{\gamma}\in\mathcal{A}} \pi(\boldsymbol{\theta}, \sigma^2|\boldsymbol{\gamma}, \boldsymbol{y})\pi(\boldsymbol{\gamma}|\boldsymbol{y}, \mathcal{A}),$$

where

$$\pi(\boldsymbol{\gamma}|\boldsymbol{y}, \mathcal{A}) = \frac{\pi(\boldsymbol{\gamma}|\boldsymbol{y})}{\sum_{\boldsymbol{\gamma}'\in\mathcal{A}} P(\boldsymbol{\gamma}'|\boldsymbol{y})}. \tag{2.3}$$

Note that for Bayesian model averaging, there are 2^p possible candidates for $\boldsymbol{\gamma}$. However, as discussed in (Madigan and Raftery, 1994), we should exclude some $\boldsymbol{\gamma}$ with less evidence. Our hybrid search algorithm can automatically discard those models and seize $\boldsymbol{\gamma}$ with high probabilities. One may consider a sampling based approach since $\boldsymbol{\gamma}$ with low posteriors would be less likely to be generated. However, our hybrid search algorithm allows us to do

the same task with a much smaller computational cost.

2.2.2 Hybrid algorithm for variable search

As discussed above, with the idea of the collapsed Gibbs sampler (Liu, 1994), we aim to draw samples from $\pi(\boldsymbol{\theta}, \sigma^2|\boldsymbol{\gamma}, \boldsymbol{y})$ and $\pi(\boldsymbol{\gamma}|\boldsymbol{y})$ respectively. For sampling from $\pi(\boldsymbol{\gamma}|\boldsymbol{y})$, we propose a hybrid search algorithm which is motivated by Jin and Goh (2021). To be more specific, we borrow the idea from the iterative conditional modes (ICM) (Besag, 1986) and stochastic shotgun search (SSS) (Hans et al., 2007) to develop our hybrid search algorithm. Using the ICM algorithm, we first identify a local maximum. If we have a convex optimization problem, ICM itself is sufficient. However, $\pi(\boldsymbol{\gamma}|\boldsymbol{y})$ has generally a multimodal distribution. To ensure we reach the global maximum, we combine the ICM with the neighborhood search idea used in SSS. By combining those two steps, we develop a hybrid search algorithm that converges to the global maximum faster than a stochastic search algorithm.

We present our proposed hybrid model search algorithm in Algorithm 1. Within Algorithm 1, we define the neighborhood of γ as

$$\mathcal{N}(\boldsymbol{\gamma}) = \{\boldsymbol{\gamma} \cup \{j\} : j \notin \boldsymbol{\gamma}\} \cup \{\boldsymbol{\gamma} \setminus \{j\} : j \in \boldsymbol{\gamma}\}. \tag{2.4}$$

Also note that the stochastic search steps in step 6 and step 8 in our proposed algorithm, we have

$$\tilde{\pi}(\tilde{\boldsymbol{\gamma}}|\boldsymbol{y}) \propto \tilde{\pi}(\boldsymbol{y}|\tilde{\boldsymbol{\gamma}})\pi(\tilde{\boldsymbol{\gamma}}),$$

where

$$\log \tilde{\pi}(\boldsymbol{y}|\tilde{\boldsymbol{\gamma}}) \approx \log \pi(\boldsymbol{y}|\tilde{\boldsymbol{\theta}}, \tilde{\sigma}^2, \tilde{\boldsymbol{\gamma}}) - \frac{q+1}{2} \log n.$$

We set q as the number of hyperparameters for the kernel matrix. In addition, we let $(\tilde{\boldsymbol{\theta}}, \tilde{\sigma}^2) = \text{argmax}_{\boldsymbol{\theta}, \sigma^2} \log p(\boldsymbol{y}|\boldsymbol{\theta}, \sigma^2, \boldsymbol{\gamma}^*) p(\boldsymbol{\theta}, \sigma^2|\boldsymbol{\gamma}^*)$, where $\boldsymbol{\gamma}^*$ is the best model returned from the deterministic search.

Our proposed hybrid search algorithm aims to maximize the posterior of $\boldsymbol{\gamma}$ and find comparable models with high posterior probabilities. The step 1 to step 4 perform the

Algorithm 1 Hybrid Search Algorithm

Step 1. Set an initial model $\boldsymbol{\gamma}^*$ and define $\mathcal{A} = \{\boldsymbol{\gamma}^*\}$.

Step 2. Set $\mathcal{A}' = \mathcal{A}$.

Step 3. Repeat for $j = 1, \ldots, p$;

a. Update

$$\gamma^{'} = \begin{cases} \boldsymbol{\gamma}^* \cup \{j\} & \text{if } j \notin \boldsymbol{\gamma}^*, \\ \boldsymbol{\gamma}^* \setminus \{j\} & \text{otherwise.} \end{cases}$$

b. Define $\mathcal{A}^* = \mathcal{A} \cup \{\boldsymbol{\gamma}^{'}\}$ and update

$$\boldsymbol{\gamma}^* = \boldsymbol{\gamma}^{'} \quad \text{if} \quad \pi(\boldsymbol{\gamma}^{'}|\boldsymbol{y}) \geq \pi(\boldsymbol{\gamma}^*|\boldsymbol{y})$$

c. Update

$$\mathcal{A} = \{\boldsymbol{\gamma} \in \mathcal{A}^* : \pi(\boldsymbol{\gamma}^*|\boldsymbol{y})/\pi(\boldsymbol{\gamma}|\boldsymbol{y}) \leq c\}.$$

Step 4. If $\mathcal{A} \neq \mathcal{A}'$, go to Step 2, and otherwise move to Step 5.

Step 5. Set $\boldsymbol{\gamma}^I = \boldsymbol{\gamma}^*$, $r = 1$.

Step 6. Compute $\mathcal{N}(\boldsymbol{\gamma}^I)$ and $\tilde{\pi}(\gamma|\boldsymbol{y})$ for $\boldsymbol{\gamma} \in \mathcal{N}(\boldsymbol{\gamma}^I)$. Let $\tilde{\boldsymbol{\gamma}} = \arg\max_{\boldsymbol{\gamma} \in \mathcal{N}(\boldsymbol{\gamma}^I)} \tilde{\pi}(\boldsymbol{\gamma}|\boldsymbol{y})$.

Step 7. For $\tilde{\boldsymbol{\gamma}}$:

a. Compute $\pi(\tilde{\boldsymbol{\gamma}}|\boldsymbol{y})$.

b. If $\pi(\boldsymbol{y}|\boldsymbol{\gamma}^*)/\pi(\boldsymbol{y}|\tilde{\boldsymbol{\gamma}}) < c$ and $\tilde{\boldsymbol{\gamma}} \notin \mathcal{A}$, add $\tilde{\boldsymbol{\gamma}}$ to $\mathcal{A}$.

c. If $\pi(\tilde{\boldsymbol{\gamma}}|\boldsymbol{y}) > \pi(\boldsymbol{\gamma}^*|\boldsymbol{y})$, set $\boldsymbol{\gamma}^* = \tilde{\boldsymbol{\gamma}}$, $\mathcal{A} = \mathcal{A} \cup \{\tilde{\boldsymbol{\gamma}}\}$, update

$$\mathcal{A} = \{\boldsymbol{\gamma} \in \mathcal{A} : \pi(\boldsymbol{\gamma}^*|\boldsymbol{y})/\pi(\boldsymbol{\gamma}|\boldsymbol{y}) \leq c\}$$

and then go to Step 2. Otherwise, continue.

Step 8. Sample new $\boldsymbol{\gamma}^I$ from $\mathcal{N}(\boldsymbol{\gamma}^I)$ with probability $\frac{\tilde{\pi}(\boldsymbol{\gamma}|\boldsymbol{y})}{\sum \tilde{\pi}(\boldsymbol{\gamma}|\boldsymbol{y})}$ for $\boldsymbol{\gamma} \in \mathcal{N}(\boldsymbol{\gamma}^I)$ and set $r = r + 1$.

Step 9. If $r < \tilde{r}$, go to step 6. Else, return $\mathcal{A}$ as the set of selected models.

deterministic search based on ICM (Besag, 1986). The stochastic neighbor search (step 5 to step 8) is based on SSS of Hans et al. (2007).

The algorithm implementation can be summarized as follows. The proposed algorithm starts with a model $\boldsymbol{\gamma}^*$. (Note in practice, we can start the algorithm with a GP regression model with a single feature having the largest correlation with the response). Then the algorithm step-wisely goes through the model space and updates to a model with a larger posterior. To be more specific, in each step, the algorithm update the current best model by adding one feature to or deleting one from.

With the best model $\boldsymbol{\gamma}^*$ found by the deterministic search, we define $\boldsymbol{\gamma}^I = \boldsymbol{\gamma}^*$ and computes $\mathcal{N}(\boldsymbol{\gamma}^I)$ defined in equation (2.4). Then, for each $\boldsymbol{\gamma}$ in $\mathcal{N}(\boldsymbol{\gamma}^I)$, we compute the approximate posterior $\tilde{\pi}(\boldsymbol{\gamma}|\boldsymbol{y})$ using the hyper-parameters posterior mode of fitting model $\boldsymbol{\gamma}^*$. We then find $\tilde{\boldsymbol{\gamma}}$ by maximizing $\tilde{\pi}(\boldsymbol{\gamma}|\boldsymbol{y})$ over $\boldsymbol{\gamma} \in \mathcal{N}(\boldsymbol{\gamma}^I)$ and compute $\pi(\tilde{\boldsymbol{\gamma}}|\boldsymbol{y})$.

The algorithm then compares the posteriors of $\tilde{\boldsymbol{\gamma}}$ with $\boldsymbol{\gamma}^*$, the current best. If $\tilde{\boldsymbol{\gamma}}$ has a larger posterior than $\boldsymbol{\gamma}^*$, the algorithm goes back to the deterministic search procedure in step 2 with this 'better' $\tilde{\boldsymbol{\gamma}}$ as the starting point. Otherwise, the algorithm stochastically jumps to a new starter model $\boldsymbol{\gamma}^I$ such that $\boldsymbol{\gamma}^I = \boldsymbol{\gamma}$ with probability $\frac{\tilde{\pi}(\boldsymbol{\gamma}|\boldsymbol{y})}{\sum \tilde{\pi}(\boldsymbol{\gamma}|\boldsymbol{y})}$ for $\boldsymbol{\gamma} \in \mathcal{N}(\boldsymbol{\gamma}^I)$, where $\mathcal{N}(\boldsymbol{\gamma}^I)$ denotes the current neighbor set. Then the algorithm constructs a new neighbor set of this starting model $\boldsymbol{\gamma}^I$ and conducts search on this new set of $\mathcal{N}(\boldsymbol{\gamma}^I)$.

In the stochastic search procedure (from step 5 to step 8), if it failed to update the 'best' model, $\boldsymbol{\gamma}^*$, within a preset number of iterations (say, $\tilde{r}$), the algorithm then stops and returns the selected model set $\mathcal{A}$. At this point, it is most likely that the algorithm has reached the global maximum of the whole model space.

To account for model uncertainty, our proposed algorithm computes the posterior ratio and adds the candidate model to the selected set $\mathcal{A}$ such that $\pi(\boldsymbol{\gamma}^*|\boldsymbol{y})/\pi(\boldsymbol{\gamma}|\boldsymbol{y}) < c$ for $\boldsymbol{\gamma} \in \mathcal{A}$, where $\boldsymbol{\gamma}^*$ is the current best model. For the choice of c, following Madigan and Raftery (1994), we set $c = 20$.

2.2.3 Computation of $\pi(\gamma|y)$

The hybrid search algorithm aims to maximize the posterior $\pi(\boldsymbol{\gamma}|\boldsymbol{y})$, where

$$\pi(\boldsymbol{\gamma}|\boldsymbol{y}) \propto \pi(\boldsymbol{y}|\boldsymbol{\gamma})\pi(\boldsymbol{\gamma}).$$

If we assume a uniform prior on $\boldsymbol{\gamma}$, we can simplify the posterior as $\pi(\boldsymbol{\gamma}|\boldsymbol{y}) \propto \pi(\boldsymbol{y}|\boldsymbol{\gamma})$. However, a uniform can cause troubles when the size of the variables is large since the dimension p increases, the larger model receives higher weights from the prior. This may lead to the inflation of false discovery rate (Scott and Berger, 2010; Chen and Chen, 2008).

To solve this issue, following Chen and Chen (2008), we use the prior

$$\pi(\boldsymbol{\gamma}) = \frac{1}{\binom{p}{|\gamma|}}\mathbb{I}\{|\gamma| \leq k\}. \tag{2.5}$$

The use of the indicator function in (2.5) allows us to only consider models with a size smaller than k. Given the prior, we also need to compute the marginal likelihood $\pi(\boldsymbol{y}|\boldsymbol{\gamma})$ in order to obtain the posterior.

To compute the marginal likelihood, we use the Laplace method (Tierney and Kadane, 1986). We first assume the prior on $(\boldsymbol{\theta}, \sigma^2)^\top$, $\pi(\boldsymbol{\theta}, \sigma^2|\boldsymbol{\gamma})$. Then using the Laplace method (Tierney and Kadane, 1986), the marginal likelihood can be computed by

$$\begin{aligned}\pi(\boldsymbol{y}|\boldsymbol{\gamma}) &= \int_\Theta \int_0^\infty \pi(\boldsymbol{y}|\boldsymbol{\theta}, \sigma^2, \boldsymbol{\gamma})\pi(\boldsymbol{\theta}, \sigma^2|\boldsymbol{\gamma}) d\boldsymbol{\theta} d\sigma^2 \\ &\approx (2\pi)^{\frac{q+1}{2}}[\det(\tilde{\boldsymbol{\Sigma}})]^{\frac{1}{2}}\pi(\boldsymbol{y}|\tilde{\boldsymbol{\theta}}, \tilde{\sigma^2}, \boldsymbol{\gamma})\pi(\tilde{\boldsymbol{\theta}}, \tilde{\sigma^2}|\boldsymbol{\gamma})\end{aligned} \tag{2.6}$$

where Θ denote the support of $\boldsymbol{\theta}$, and $\tilde{\boldsymbol{\theta}}$, $\tilde{\sigma}^2$ and, $\tilde{\boldsymbol{\Sigma}}$ are obtained as follows:

$$(\tilde{\boldsymbol{\theta}}, \tilde{\sigma^2}) = \text{argmax}_{\boldsymbol{\theta}, \sigma^2} \log \pi(\boldsymbol{y}|\boldsymbol{\theta}, \sigma^2, \boldsymbol{\gamma}) p(\boldsymbol{\theta}, \sigma^2|\boldsymbol{\gamma}),$$

$$\tilde{\boldsymbol{\Sigma}} = \left[-\frac{\partial^2}{\partial \boldsymbol{\theta} \partial \sigma^2} \log \pi(\boldsymbol{y}|\boldsymbol{\theta}, \sigma^2, \boldsymbol{\gamma})\pi(\boldsymbol{\theta}, \sigma^2|\boldsymbol{\gamma})|_{\boldsymbol{\theta}=\tilde{\boldsymbol{\theta}}, \sigma^2=\tilde{\sigma^2}} \right]^{-1}$$

Recall that the likelihood $\pi(\boldsymbol{y}|\boldsymbol{\theta}, \sigma^2, \boldsymbol{\gamma})$ is given in equation (2.1).

We can further simplify the computation of the marginal likelihood. By ignoring constant terms, as $n \to \infty$, we have

$$-2\log\pi(\boldsymbol{y}|\boldsymbol{\gamma}) \approx -2\log\pi(\boldsymbol{y}|\tilde{\boldsymbol{\theta}}, \tilde{\sigma^2}, \boldsymbol{\gamma}) + (q+1)\log n. \tag{2.7}$$

where q denotes the dimension of $\boldsymbol{\theta}$. This can be used for computing posterior probability in the algorithm to improve the speed of calculating the marginal likelihood by avoiding the computation of the Hessian matrix.

To further speed up, we can use a hash table to store the posterior probabilities of the visited models since the algorithm may visit the same $\boldsymbol{\gamma}$ several times. In practice, if the support of the hyper-parameters Θ does not fill the whole real space, one can consider a transformation to improve the Laplace approximation. In this paper, we assume an inverse-Gamma prior for λ, τ and σ^2 respectively and consider the log transformation for λ, τ, and σ^2.

2.2.4 Predictions

Once the hybrid algorithm returns the set of selected models $\mathcal{A}$, we make prediction using $\mathcal{A}$. With the Bayesian model averaging technique, (Hoeting et al., 1999; Madigan and Raftery, 1994; Wasserman et al., 2000), we can obtain the posterior predictive distribution for $\boldsymbol{f}^*$ as

$$\begin{aligned}\pi(\boldsymbol{f}(\boldsymbol{x}^*)|\boldsymbol{y}) &\approx \sum_{\boldsymbol{\gamma}\in\mathcal{A}} \pi(\boldsymbol{f}^*|\boldsymbol{\gamma}, \boldsymbol{y})\pi(\boldsymbol{\gamma}|\mathcal{A}, \boldsymbol{y}) \\ &\approx \sum_{\boldsymbol{\gamma}\in\mathcal{A}} \left[\int\int \pi(\boldsymbol{f}^*|\boldsymbol{\theta}, \sigma^2, \boldsymbol{\gamma}, \boldsymbol{y})\pi(\boldsymbol{\theta}, \sigma^2|\boldsymbol{\gamma}, \boldsymbol{y}) d\boldsymbol{\theta} d\sigma^2\right] \pi(\boldsymbol{\gamma}|\mathcal{A}, \boldsymbol{y}).\end{aligned}$$

Similarly, the predictive distribution of $\boldsymbol{y}^*$ can be computed as

$$\begin{aligned}\pi(\boldsymbol{y}^*|\boldsymbol{y}) &\approx \sum_{\boldsymbol{\gamma}\in\mathcal{A}} \pi(\boldsymbol{y}^*|\boldsymbol{\gamma},\boldsymbol{y})\pi(\boldsymbol{\gamma}|\mathcal{A},\boldsymbol{y}) \\ &\approx \sum_{\boldsymbol{\gamma}\in\mathcal{A}} \left[\int\int \pi(\boldsymbol{y}^*|\boldsymbol{\theta},\sigma^2,\boldsymbol{\gamma},\boldsymbol{y})\pi(\boldsymbol{\theta},\sigma^2|\boldsymbol{\gamma},\boldsymbol{y})d\boldsymbol{\theta}d\sigma^2\right]\pi(\boldsymbol{\gamma}|\mathcal{A},\boldsymbol{y}).\end{aligned}$$

Note that $\pi(\boldsymbol{\gamma}|\mathcal{A},\boldsymbol{y})$ can be easily computed by (2.3) and (2.6). However, the double integration of $\pi(\boldsymbol{f}^*|\boldsymbol{\theta},\sigma^2,\boldsymbol{\gamma},\boldsymbol{y})\pi(\boldsymbol{\theta},\sigma^2|\boldsymbol{\gamma},\boldsymbol{y})$ can be intractable in general. To solve this issue, we use the sampling approach to address this integration. Our prediction algorithm is as follows:

Step 1: Compute $\pi(\boldsymbol{\gamma}|\mathcal{A},\boldsymbol{y})$ for each $\boldsymbol{\gamma}\in\mathcal{A}$.

Step 2: Repeat for $\boldsymbol{\gamma}\in\mathcal{A}$,

(a) Generate $\{(\boldsymbol{\theta}^{(t)},\sigma^{2(t)}), t=1,\cdots,T\}$ from $\pi(\boldsymbol{\theta},\sigma^2|\boldsymbol{\gamma},\boldsymbol{y})$ via the Laplace approximation, where the posterior $\pi(\boldsymbol{\theta},\sigma^2|\boldsymbol{\gamma},\boldsymbol{y})$ is given as

$$\pi(\boldsymbol{\theta},\sigma^2|\boldsymbol{\gamma},\boldsymbol{y}) \propto \pi(\boldsymbol{y}|\boldsymbol{\theta},\sigma^2,\boldsymbol{\gamma})\pi(\boldsymbol{\theta},\sigma^2|\boldsymbol{\gamma}).$$

(b) Given $\{(\boldsymbol{\theta}^{(t)},\sigma^{2(t)}), t=1,\cdots,T\}$, generate predictive samples $\{(\boldsymbol{y}^{*(t)},\boldsymbol{f}^{*(t)}),$ $t=1,\cdots,T\}$ using $\pi(\boldsymbol{f}^*|\boldsymbol{\theta},\sigma^2,\boldsymbol{\gamma},\boldsymbol{y})$ and $\pi(\boldsymbol{y}^*|\boldsymbol{\theta},\sigma^2,\boldsymbol{\gamma},\boldsymbol{y})$ (given at the end of Section 2.1.1).

(c) Compute the predicted mean for $\boldsymbol{f}^*$ and $\boldsymbol{y}^*$ as $\hat{\boldsymbol{f}}^*_{\boldsymbol{\gamma}} = \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{f}^{*(t)}$ and $\hat{\boldsymbol{y}}^*_{\boldsymbol{\gamma}} = \frac{1}{T}\sum_{t=1}^{T}\boldsymbol{y}^{*(t)}$.

Step 3: Using the Bayesian model averaging techniques, the final prediction for $\boldsymbol{y}^*$ and $\boldsymbol{f}^*$ are given as

$$\hat{\boldsymbol{f}}^* = \sum_{\boldsymbol{\gamma}\in\mathcal{A}} \hat{\boldsymbol{f}}^*_{\boldsymbol{\gamma}}\pi(\boldsymbol{\gamma}|\mathcal{A},\boldsymbol{y}),$$
$$\hat{\boldsymbol{y}}^* = \sum_{\boldsymbol{\gamma}\in\mathcal{A}} \hat{\boldsymbol{y}}^*_{\boldsymbol{\gamma}}\pi(\boldsymbol{\gamma}|\mathcal{A},\boldsymbol{y}).$$

2.3 Dealing with massive data

In addition to variable selection, one aspect that hinders the application of GP based models is the heavy computational cost with a large sample size. To address this problem, many methods have been proposed by researchers (Liu et al., 2020; Williams and Rasmussen, 2006; Quiñonero-Candela and Rasmussen, 2005). However, the problem of variable selection under the large n setting has not been explored yet. In this section, we propose a new approach to fill this void.

Our strategy can be summarized as follows. We tackle the problem of large p first by our proposed hybrid search algorithm combined with a subset of data (SoD) approach. Adopting the SoD approach for variable selection leads to low computational cost. Our proposed idea is based on the fact that a well-selected subsample retains sufficient information for model selection. For prediction, we use the nearest neighbor GP regression approach (Datta et al., 2016a,b; Gramacy et al., 2016; Gramacy and Apley, 2015; Gramacy and Haaland, 2016) in the framework of Bayesian model averaging.

2.3.1 Variable selection with QSoD

When n is large, solving $\pi(\boldsymbol{\gamma}|\boldsymbol{y})$ with full data points is nearly impossible since the computational cost is $O(n^3)$. Even though the SoD approach works poorly for prediction due to its large variance, a well-chosen subset of data can retain good enough information for variable selection. Hence, in order to fast pinpoint those important models, we use a subsample $\tilde{\boldsymbol{y}}$ rather than the full data set, $\boldsymbol{y}$. To achieve good performance, we want this subset of data $\tilde{\boldsymbol{y}}$ to uniformly cover the full data set $\boldsymbol{y}$ so that it can well represent the variation of the original data set. With this idea, we propose a quantile-based subset of data (QSoD) approach. We present the approach below.

Let $\boldsymbol{y}_Q = [y_{q_1}, y_{q_2}, \cdots, y_{q_m}]^\top$ be a vector of m empirical quantiles of $\boldsymbol{y}$. Note that we let $y_{q1} < y_{q2} < \cdots < y_{qm}$ so that the elements of empirical quantile vector is in a nondecreasing order. In addition, we keep the grid distances between each y_{q_i} and each $y_{q_{i+1}}$ to be same and fixed in $\boldsymbol{y}_Q$ to ensure a uniform coverage of the original data $\boldsymbol{y}$. With this setup, we

choose the subset of data such that

$$\tilde{\boldsymbol{y}} = \cup_{i=1}^{m} \tilde{\boldsymbol{y}_{qi}} \tag{2.8}$$

where each $\tilde{\boldsymbol{y}_{qi}}$ is a vector that contains c closest observations (measured by absolute distance) to the i^{th} empirical quantile y_{qi} for $i = 1, \cdots, m$. Treating $\tilde{\boldsymbol{y}}$ as the observed data, we conduct model search via the hybrid feature search algorithm in Algorithm 1.

Note that the size of subsample data $\boldsymbol{y}_Q$ is $c \times m$, which is chosen by a researcher. Also note that there is a trade-off between the size of subsamples and the accuracy of the estimation for the marginal likelihood of the model $\pi(\boldsymbol{y}|\boldsymbol{\gamma})$. That is, a larger subsample results in a more accurate estimation of the marginal likelihood $\pi(\boldsymbol{y}|\boldsymbol{\gamma})$ but charges a higher computational cost.

2.3.2 Prediction with nearest neighbor GP

After obtaining $\mathcal{A}$ using the hybrid search algorithm with the QSoD approach, we employ a localized regression approach for prediction in a Bayesian model averaging framework. In particular, we borrow the idea of nearest neighbor (NN) based GP approach as in Datta et al. (2016a,b); Gramacy et al. (2016); Gramacy and Apley (2015); Gramacy and Haaland (2016). The localized regression approach is based on the fact that the association decreases as the distance between two points increases. In other words, such faraway points retain less or no information for prediction. In this paper, we define a fixed nearest neighbor set for each unobserved location and train a local GP expert for this particular point. We then make prediction for each point with its local model.

Let $\boldsymbol{x}^* = [x_1^*, x_2^*, \cdots, x_{n^*}^*]^\top$ be a finite set of n^* new points at which we aim to make prediction. We also define $\mathcal{D} = \{(x_i, y_i) : i = 1, \cdots, n\}$ as the whole set of the training data. Given $\mathcal{A}$ returned by algorithm 1 with QSoD subsamples, the prediction procedure is performed as follows:

Step 1: Compute $\pi(\boldsymbol{\gamma}|\mathcal{A}, \tilde{\boldsymbol{y}})$ for each $\boldsymbol{\gamma} \in \mathcal{A}$, where $\tilde{\boldsymbol{y}}$ is the QSoD subsample defined

in equation (2.8).

Step 2: For each x_i^* in $\boldsymbol{x}^*$

- Repeat for $\boldsymbol{\gamma} \in \mathcal{A}$,

  1. Construct a nearest neighbor set $\mathcal{D}_k(x_i^*; \boldsymbol{\gamma}) = \{(x_i, y_i), i = 1, \cdots, k\}$ such that $\mathcal{D}_k(x_i^*; \boldsymbol{\gamma}) \in \mathcal{D}$ contains the k closest pairs of (x_i, y_i)'s to x_i^* in terms of the Euclidean distance defined as

$$l(x_i, x_j; \boldsymbol{\gamma}) = \sqrt{\sum_{g=1}^{p} \gamma_g (x_{ig} - x_{jg})^2}.$$

  2. Train a local GP expert and generate posterior samples $\{(\boldsymbol{\theta}^{(t)}, \sigma^{2(t)}), t = 1, \cdots, T\}$ from $\pi(\boldsymbol{\theta}, \sigma^2 | \boldsymbol{\gamma}, \mathcal{D}_k(x_i^*; \boldsymbol{\gamma}))$ via the Laplace approximation.
  3. Given $\{(\boldsymbol{\theta}^{(t)}, \sigma^{2(t)}), t = 1, \cdots, T\}$, generate predictive samples $\{(y_i^{*(t)}, f_i^{*(t)}), t = 1, \cdots, T\}$ from the predictive distributions $\pi(f_i^* | \boldsymbol{\theta}, \sigma^2, \boldsymbol{\gamma}, \mathcal{D}_k(x_i^*; \boldsymbol{\gamma}))$ and $\pi(y_i^* | \theta, \sigma^2, \boldsymbol{\gamma}, \mathcal{D}_k(x_i^*; \boldsymbol{\gamma}))$.
  4. Compute $\hat{f}_{i\boldsymbol{\gamma}}^* = \frac{1}{T} \sum_{t=1}^{T} f_i^{*(t)}$ and $\hat{y}_{i\boldsymbol{\gamma}}^* = \frac{1}{T} \sum_{t=1}^{T} y_i^{*(t)}$.

- The final prediction for y_i^* and f_i^* are computed as

$$\hat{f}_i^* = \sum_{\boldsymbol{\gamma} \in \mathcal{A}} \hat{f}_{i\boldsymbol{\gamma}}^* \pi(\boldsymbol{\gamma} | \mathcal{A}, \tilde{\boldsymbol{y}}),$$
$$\hat{y}_i^* = \sum_{\boldsymbol{\gamma} \in \mathcal{A}} \hat{y}_{i\boldsymbol{\gamma}}^* \pi(\boldsymbol{\gamma} | \mathcal{A}, \tilde{\boldsymbol{y}}).$$

In practice, one could use different divergence measures instead of the Euclidean distance used in this paper. The size of the nearest neighbor set, k, is a subjective parameter. Note that there is a trade-off between the prediction accuracy and computational cost associated with k.

The approach discussed above requires a computational cost of $O(n^* k^3)$. Note that, in this paper, we used a fixed NN. In practice, to improve the prediction accuracy, one could

also use the greedy search approaches as in Gramacy et al. (2016); Gramacy and Apley (2015); Gramacy and Haaland (2016). In addition, the NN localized regression approach can be considered when the size of predictions n^* is small. When the size n^* is huge, one can consider alternative approaches discussed in (Liu et al., 2020; Williams and Rasmussen, 2006; Quiñonero-Candela and Rasmussen, 2005).

2.4 Simulation studies

In this section, we conduct simulation studies to validate our proposed methods. We consider two cases: 1) moderate sample size data and 2) massive sample size data.

2.4.1 The moderate sample size case

Following Savitsky et al. (2011), we generate the simulated data from the following 'true' data generating process:

$$y_i = 2\sin(x_{i1}) + \frac{x_{i2}^2}{2} + \frac{\exp(x_{i3})}{5} + x_{i4} + \epsilon_i,$$

where $\{i1, i2, i3, i4\}$ are randomly selected from $\{1, \cdots, p\}$, $x_{ij} \overset{iid}{\sim} \mathcal{U}(-\pi, \pi)$, and $\epsilon_i \overset{iid}{\sim} \mathcal{N}(0, 1)$. Hence, the randomly selected four features are truly associated with the response and the rest features are irrelevant. We set the sample size to be $n = 100$ with different feature sizes $p = 20, 200, 1000$. We also generate another set of observations with size $n^* = 100$ from the true model for validation.

The aim of this simulation study is to compare our proposed approach to the scheme 2 adaptive MCMC algorithm proposed in Savitsky et al. (2011). We also include the true model and full model trained with the Laplace approximation (LA) for reference.

To compare the performance, we compute the following measures: the computational cost measured by the number of iterations as well as the CPU time (measured for variable selection part of algorithm); variable selection accuracy measured by true positive rate (TPR) and true negative rate (TNR); and the mean squared prediction error (MSPE) on the validation

set defined as

$$\text{MSPE} = \frac{1}{n^*}\sum_{i=1}^{n^*}(\hat{y}_i - y_i)^2. \tag{2.9}$$

For the number of iterations, we count how many times our algorithm scans through the whole feature space (i.e., from x_1 to x_p). For our proposed hybrid search algorithm, we set the maximum number of the stochastic search iterations to $\tilde{r} = 100$. In addition, we set the predictive sample size to $1,500$ to match the number of the iterations for MCMC. For the MCMC approaches, we run the GP MCMC twice. For the first one (called GP MCMC 1), we set the time cost approximately the same as the hybrid search. For the second one (called GP MCMC 2), we run $1,500$ iterations and use the first 500 iterations as a burn-in period. In addition to the number of iterations, we measure the CPU time (in seconds). The experiments of both methods are conducted on the same hardware configuration. In particular, the Monte Carlo experiments are conducted with R on the Beocat Linux based server with a CPU MHz of 2533.414.

In terms of the variable selection accuracy, we record TPR and TNR for both methods. To compute those two metrics, we set $\gamma_j = 1$ if $\pi(\gamma_j = 1|\boldsymbol{y}) > 0.5$ and $\gamma_j = 0$ otherwise. For our proposed hybrid search algorithm, we compute $\pi(\gamma_j|\boldsymbol{y})$ by

$$\pi(\gamma_j = 1|\boldsymbol{y}) = \sum_{\boldsymbol{\gamma}\in\mathcal{A}} \pi(\boldsymbol{\gamma}|\boldsymbol{y})\mathbb{I}\{\gamma_j = 1\}.$$

For the MCMC algorithm, $\pi(\gamma_j = 1|\boldsymbol{y})$ can be computed easily by the MCMC sample mean.

For both methods, we consider a single width Gaussian form. For the prior specification, we assume $\lambda \sim$ inv-Gamma$(1, 1)$, $\tau \sim$ inv-Gamma$(1, 1)$, and $\sigma^2 \sim$ inv-Gamma$(1, 1)$.

For both hybrid search and MCMC methods, we set the starting point for γ as the model with one feature having the strongest marginal correlation with the response. For the initial values of the kernel matrix hyperparameters, we use the maximum likelihood estimation approach. We repeat the Monte Carlo experiments for $1,000$ times. The results are summarized in Table 2.1. We also plot the MSPE in Figure 2.4, where the error bars are computed by 2 times the estimated standard error.

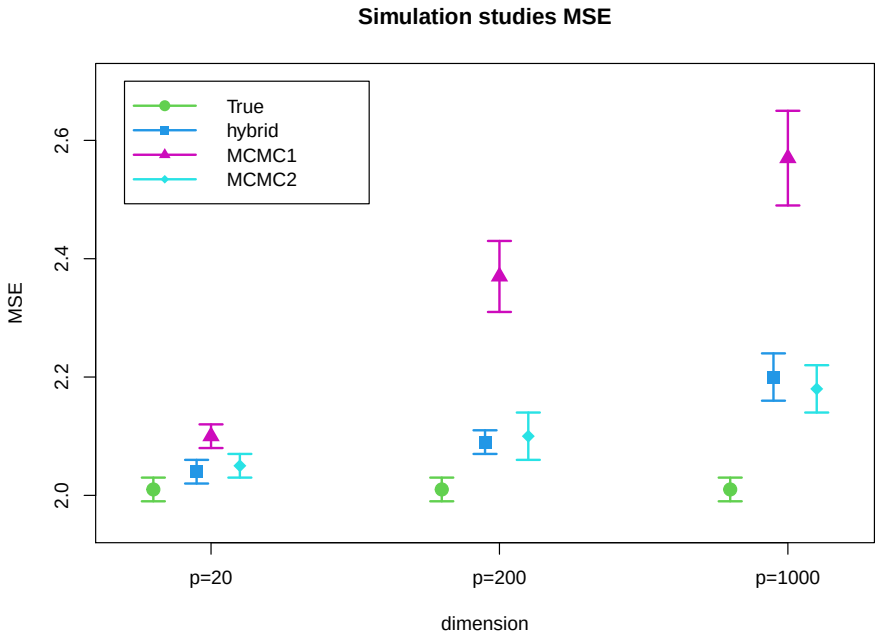


Figure 2.4: MSPE of simulation studies for moderate sample size case

In Table 2.1, we can see that GP without model selection provides poor prediction. Our proposed method outperforms the GP MCMC 1 approach in terms of both feature selection and prediction accuracy. When we compare with GP MCMC 2, we observe that our method has approximately the same performance in terms of prediction accuracy as well as the variable selection results. In particular, both methods provide similar prediction results close to the true model. However, the time costs of our proposed method are 3 to 10 times smaller than that of GP MCMC 2. This demonstrates that the proposed method is computationally more efficient than the GP MCMC method.

Table 2.1: Moderate sample size simulation results

Dimensions	MSPE	TPR	TNR	Iterations	time
		Full model (LA)			
p=20	5.64(0.03)	1(0)	0(0)	(NA)	(NA)
p=200	15.44(0.06)	1(0)	0(0)	(NA)	(NA)
p=1000	15.35(0.07)	1(0)	0(0)	(NA)	(NA)
		True model (LA)			
p=20	2.01(0.01)	1(0)	1(0)	(NA)	(NA)
p=200	2.01(0.01)	1(0)	1(0)	(NA)	(NA)
p=1000	2.01(0.01)	1(0)	1(0)	(NA)	(NA)
		Hybrid search			
p=20	2.04(0.01)	0.991(0.001)	0.999(0.0002)	103.24(0.01)	4.71(0.04)
p=200	2.09(0.01)	0.978(0.002)	0.999(0.0002)	103.44(0.02)	63.42(0.37)
p=1000	2.20(0.02)	0.955(0.004)	0.999(0.0001)	104.07(0.15)	711.42(5.08)
		GP MCMC 1			
p=20	2.10(0.01)	0.986(0.002)	0.999(0.0003)	150(0)	6.72(0.04)
p=200	2.37(0.03)	0.951(0.004)	0.999(0.0001)	150(0)	62.66(0.16)
p=1000	2.57(0.04)	0.914(0.006)	0.999(0.0002)	350(0)	713.78(4.25)
		GP MCMC 2			
p=20	2.05(0.01)	0.992(0.001)	0.999(0.0002)	1500(0)	65.33(0.36)
p=200	2.10(0.02)	0.980(0.002)	0.999(0.0002)	1500(0)	616.77(1.36)
p=1000	2.18(0.02)	0.961(0.003)	0.999(0.0001)	1500(0)	2295.93(34.02)

The estimated standard errors are given in the parenthesis.

2.4.2 The massive data case

For the massive data case, we consider the following data generating process: :

$$y_i = 2\sin(x_{i1}) + \frac{x_{i2}^2}{2} + \frac{\exp(x_{i3})}{6} + \epsilon_i \tag{2.10}$$

where $\{i1, i2, i3\}$ are randomly selected from $\{1, \cdots, p\}$, $x_{ij} \overset{iid}{\sim} \mathcal{U}(-\pi, \pi)$, and $\epsilon_i \overset{iid}{\sim} \mathcal{N}(0, 1)$. We set the sample size to $n = 10,000$ with different feature sizes $p = 100, 1000$. To validate the prediction performance, we also generate another 100 testing observations.

We use our proposed QSoD approach with hybrid search for model selection and the NN localized regression for prediction. For the QSoD approach, we select 100 representative subsamples from the original 10^4 training samples. For NN GP, we select 50 nearest neighbor samples of each testing point to train a local GP for prediction.

We compare our proposed method to the hybrid search algorithm implemented with a random subsample combined with NNGP, which is commonly considered in practice. In addition, we also record the prediction performance of NNGP based on the true model and the full model for reference. For each method, we record MSPE, TPR, TNR, and true model coverage probability (denoted by $\pi(\gamma_{\text{true}} \in \mathcal{A})$). To ensure the fairness of the comparison, we consider the same size of subsamples and nearest neighbor sets for both methods. The remaining settings including priors are the same as in Section 2.4.1.

We repeated the experiment $1,000$ times and tabulated the simulation results in Table 2.2. The MSPEs for each method are plotted in Figure 2.5. The error bars are also drawn as 2 times the estimated standard error.

From Table 2.2, we can see that our proposed QSoD approach provides the closest prediction accuracy to the true model. In addition, both the prediction accuracy and model selection performances of our QSoD approach significantly outperforms the random subsample approach. Furthermore, the standard error of our proposed approach on both MSPE and variable selection metrics is smaller. Hence, we conclude that our proposed method provides a more stable performance than the random subsample approach.

Table 2.2: Massive data simulation results

Dimenions	MSPE	TPR	TNR	$\pi(\gamma_{\text{true}} \in \mathcal{A})$
		NNGP full		
p=100	5.18(0.02)	1(0)	0(0)	(NA)
p=1000	11.01(0.05)	1(0)	0(0)	(NA)
		NNGP true		
p=100	1.05(0.005)	1(0)	1(0)	(NA)
p=1000	1.05(0.005)	1(0)	1(0)	(NA)
		NNGP QSoD hybrid search		
p=100	1.06(0.005)	0.995(0.001)	0.999(0.0002)	0.998
p=1000	1.07(0.006)	0.988(0.002)	0.999(0.0002)	0.990
		NNGP random hybrid search		
p=100	1.17(0.01)	0.941(0.004)	0.999(0.0002)	0.929
p=1000	1.23(0.012)	0.913(0.005)	0.999(0.0002)	0.878

The estimated standard errors are given in the parenthesis.

2.5 Real data application

To demonstrate the applicability of our proposed method, we also apply our methods to two real data sets: 1) *meatspec* data available at the R package *faraway* (Faraway, 2004) and 2) *online news population* data available at the UCI machine learning repository (https://archive.ics.uci.edu/ml/datasets/Online+News+Popularity).

2.5.1 The meatspec data

The *meatspec* dataset (available with R package *faraway*) consists of measurements of a 100 channel spectrum of absorbances and the fat content for 215 finely chopped pure meat

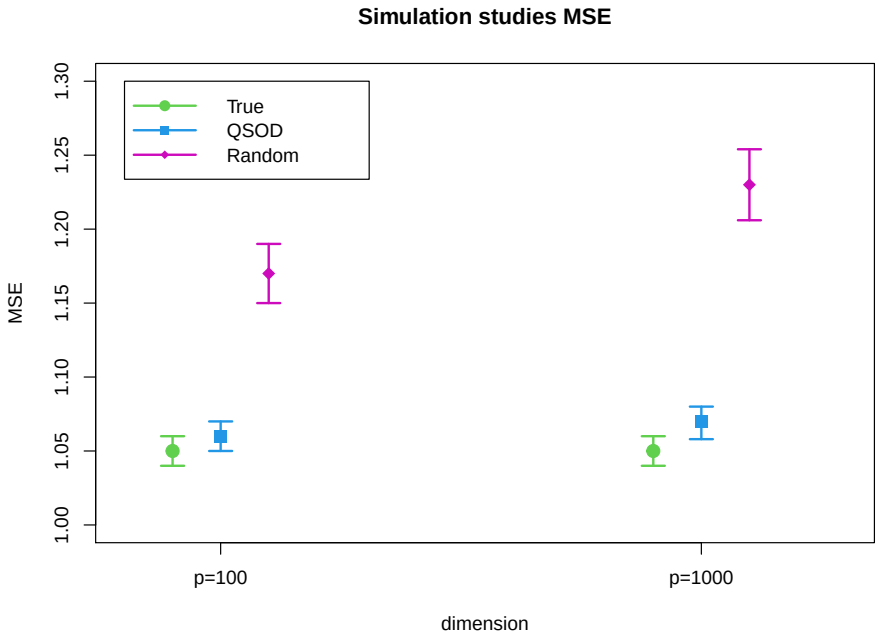


Figure 2.5: MSPE of simulation studies for massive data case

samples. Since directly measuring the fat content can be costly, people want to build a statistical model to predict the fat content based on the 100 absorbances that are much easier to obtain. As discussed in Borggaard and Thodberg (1992) and Yi et al. (2011), the response fat content has a nonlinear relationship with absorbance measurements. Hence, training a nonlinear regression model would give the best prediction performance.

To analyze the data, we first remove two outliers based on studentized residuals and take a log transformation of the fat content following Yi et al. (2011). We then conduct Monte Carlo cross validation: randomly select 149 samples for training and use the remaining 64 for testing. We repeat this experiment $5,000$ times.

For each replication, we compare our proposed GP hybrid search method to the full feature GP (trained with the Laplace approximation), the lasso linear regression model, the GP trained with lasso selected features, and the GP with variable selection trained by the MCMC. In particular, the GP MCMC algorithm is the scheme 2 adaptive MCMC proposed by Savitsky et al. (2011) (denote by "GP MCMC" in Table 2.3), where the single bandwidth Gaussian kernel is employed. For the lasso linear regression model, we use the

cv.glmnet function from the *glmnet* (Friedman et al., 2009) package to select the best tuning parameters λ. To evaluate the prediction accuracy, we measure the MSPE as defined in equation (2.9). In addition, we record the CPU time (in seconds) for both our hybrid search method and the MCMC approach.

For our hybrid search algorithm, we set the iteration bounds to 100 and the size of predictive samples to $5,000$. For the MCMC approach, we run the algorithm $5,000$ times and use the first $1,000$ iterations as the burn-in period. In addition, we tune the proposal distribution so that the average acceptance rates for each parameter σ^2, λ, and τ are 0.78, 0.9, 0.68, respectively. Furthermore, as in Section 2.4.1, we train all GP models with the single width Gaussian kernel matrix $\boldsymbol{K}$ and assume the noninformative inverse gamma priors for the hyperparameters. The analysis results are shown in Table 2.3.

Table 2.3: Meatspec data analysis results

Method	MSPE
Full feature GP (LA)	0.075(0.0003)
Linear regression lasso	0.145(0.0005)
GP lasso	0.082(0.0006)
GP-MCMC	0.056(0.0004)
Hybrid search	**0.054(0.0006)**

The estimated standard errors are given in the parenthesis.

From the results in Table 2.3, we see that the GP model with our proposed hybrid search method has the smallest MSPE. In addition, the MSPE of all GP models are smaller than the linear regression model. Note that the MCMC GP method is slightly worse than the hybrid search. This may be due to the fact that our proposed hybrid search eliminate many redundant models using the notion of the "Occam window" (Madigan and Raftery, 1994) while the MCMC algorithm includes all visited models. The CPU time of our proposed hybrid search on average is 331.93 seconds while the MCMC approach with $5,000$ iterations

takes $2,673.153$ seconds. This demonstrates that our proposed method is computationally efficient.

2.5.2 Online news popularity data

In addition to the meatspec data, we also apply our proposed method to the online news popularity data (Fernandes et al., 2015). The data contains $39,797$ observations and 59 features. The number of shares is used as a measure of the popularity of the news. We use the logarithm transformation of the share numbers.

We conduct Monte Carlo cross validation by randomly retaining 100 for testing and using the rest for training. We repeat this experiment $5,000$ times. Note that the size of the data is enormous such that training a full-size kernel GP is infeasible. Hence, we apply the nearest neighbor approach combined with the QSoD hybrid search approach. To show the advantage of our proposed method, using MPSE defined in equation (2.9), we compare the prediction performance of our proposed method to the NNGP model with the full feature model and the NNGP model with random subsamples hybrid search.

Table 2.4: Online new popularity data analysis results

Method	MSPE
Full-NNGP	0.857(0.0024)
Random sample hybrid search-NNGP	0.815(0.0024)
QSoD hybrid search-NNGP	**0.794(0.0023)**

The estimated standard errors are given in the parenthesis.

For each replication of the experiment, we selected roughly 250 representative subsamples from the training set using the QSoD approach. In addition, we set the iteration bounds of the hybrid search algorithm to be 50. For the size of the nearest neighbors, we set it to 50. As in Section 2.4.1, we set the same size of subsamples and nearest neighbor sets for both methods. In addition, we also trained both GPs with the single width Gaussian kernel with the inverse-Gamma priors as in Section 2.4.

We display the final results in Table 2.4. The result clearly shows that our proposed method outperforms both the full feature NNGP and the NNGP trained with the random sample hybrid search.

2.6 Concluding remarks

We have developed a fast hybrid search algorithm for GP regression models in this chapter. The proposed method provides a significantly faster and effective way to address both variable selection and model uncertainties problems. As shown in Section 2.4.1, while our proposed method provides a comparable performance to the existing MCMC approach, the computational cost can be significantly reduced by our proposed method. In addition, we have addressed the variable selection problem under massive data settings. Note that the proposed method can be incorporated with the big data scalable GP technique of Liu et al. (2020). Our future research directions include, but are no limited to, employing lower rank matrix approximation techniques to further reduce the computational costs and extensions to the generalized GP models for analyzing a variety of data types.

Chapter 3

Bayesian doubly-sparse reproducing kernel Hilbert space regression

In the reproducing kernel Hilbert space (RKHS) modeling, simultaneous variable selection and sparse kernel matrix estimation are both needed to achieve the optimal results. This is known as the "doubly-sparse" problem. In this chapter, we develop a novel Bayesian doubly-sparse approach to RKHS regression modeling.

We first give a brief presentation of our model set-up and the prior specifications in Section 3.1. Then in Section 3.2, we develop a collapsed Gibbs sampler that allows the selection of the active vectors to be incorporated into the variable selection sampling procedures. In Section 3.3, we extend our proposed method to the large sample cases, where a variation of the collapsed Gibbs sampler algorithm is developed. In Section 3.4, we describe the procedure of making prediction. In Sections 3.5 and 3.6, we examine our proposed methods through simulation studies and real data analysis. We conclude this chapter in Section 3.7.

3.1 Model set-up and prior specification for 'double-sparsity'

3.1.1 Model set-up and likelihood

Suppose that we observe a dataset consists n pairs of $\{(x_i, y_i)\}$ for $i = 1, \cdots, n$. For each pair, $x_i = (x_{i1}, x_{i2}, \cdots, x_{ip})^\top \in \mathbb{R}^p$ is defined as the p-dimensional input vector and the $y_i \in \mathbb{R}$ is defined as the uni-variate continuous response. We assume that the observed data are generated from

$$y_i = f(x_i) + \epsilon_i,$$

where $\epsilon_i \overset{i.i.d}{\sim} N(0, \sigma^2)$ and $f(x_i) : \mathbb{R}^p \to \mathbb{R}$ is an unknown regression function.

Note that our goal is to estimate the unknown regression function f based on the given dataset. In this paper, we employ the RKHS approach for estimating f so that we assume $f = u + h \in (\{\mathbf{1}\} + \mathcal{H}_K)$, where $\mathcal{H}_K$ is a RKHS. Then, the estimation problem of f is turned into the following optimization problem:

$$\min_{f \in \mathcal{H}_K} \left\{ \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(y_i, f(x_i)) + \frac{g}{2} ||h||^2_{\mathcal{H}_K} \right\}, \tag{3.1}$$

where $\mathcal{L}(y_i, f(x_i))$ is the loss function, $||h||^2_{\mathcal{H}_K}$ is the RKHS norm, and g is a tuning parameter.

By the representer theorem (Kimeldorf and Wahba, 1970), the solution for the optimization problem (3.1) can be given as

$$f(x_i) \approx \beta_0 + \sum_{j=1}^{n} \beta_j K(x_i, x_j | \boldsymbol{\theta}),$$

where $K(x_i, x_j|\boldsymbol{\theta})$ corresponds to the $(i,j)^{th}$ element of the n by n kernel matrix $\boldsymbol{K_\theta} = \{K(x_i, x_j|\boldsymbol{\theta})\}_{n \times n}$. With this solution, the inference towards the unknown regression function f is turned into the estimation of a linear regression function with the kernel matrix $\boldsymbol{K_\theta}$

defined as the design matrix. That is, our model can be rewritten as

$$y_i \approx \beta_0 + \sum_{j=1}^{n} \beta_j K(x_i, x_j|\boldsymbol{\theta}) + \epsilon_i. \tag{3.2}$$

To model the kernel matrix $\boldsymbol{K_\theta}$, following Linkletter et al. (2006); Savitsky et al. (2011), we define

$$K(x_i, x_j|\boldsymbol{\rho}) = \exp\left\{-\sum_{k=1}^{p} -\log(\rho_k)(x_{ik} - x_{jk})^2\right\}$$

where $\boldsymbol{\rho} = (\rho_1, \cdots, \rho_p)^\top$ are the kernel bandwidth. As proposed by Savitsky et al. (2011), we set $\rho_k \in (0, 1]$. Note that if $\rho_k = 1$, the k^{th} feature is totally excluded from the constructing the kernel matrix. This formulation can be easily extended to other kernel forms, such as Laplacian and Mercer kernel. For notational simplicity, we abuse notations related to $\boldsymbol{K}$ in the rest of this dissertation. In particular, we let it to denote the kernel matrix itself plus a column of ones that corresponds to the intercept. Letting $\boldsymbol{y} = (y_1, \cdots, y_n)^\top$, the model (3.2) can be equivalently written as

$$\boldsymbol{y}|\boldsymbol{\rho}, \boldsymbol{\beta}, \sigma^2 \sim \mathcal{N}(\boldsymbol{K_\rho}\boldsymbol{\beta}, \sigma^2\mathbb{I}). \tag{3.3}$$

3.1.2 Prior specification

As discussed in Fan and Lv (2008), variable selection is of key importance for both the model training and prediction. With this purpose, we append a binary variable selection index vector $\boldsymbol{\gamma} = \{\gamma_1, \cdots, \gamma_p\}$ to our model. Note that variable selection can be accomplished by manipulating the kernel bandwidth, and in this spirit, Linkletter et al. (2006); Savitsky et al. (2011) proposed the following spike and slab prior:

$$\pi(\boldsymbol{\rho}|\boldsymbol{\gamma_k}) = \prod_{k=1}^{p} \left\{\gamma_k \mathbb{I}[0 < \rho_k < 1] + (1 - \gamma_k)\delta_1(\rho_k)\right\},$$

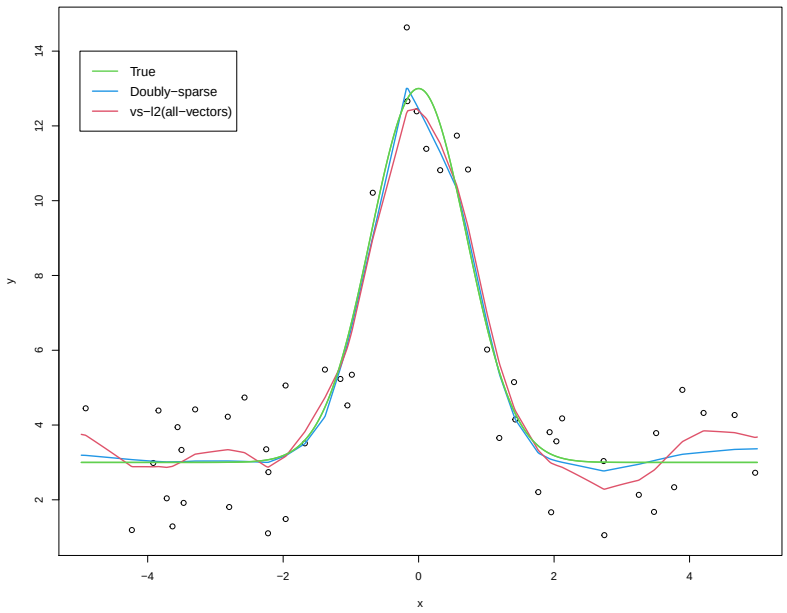


Figure 3.1: Fitted curve for simulation studies Case 1, section 3.5

which is also assumed in this paper, where $\delta_1(\rho_k)$ is a point mass distribution on 1. Note that if $\gamma_k = 0$, the k^{th} variable are discarded for computing the kernel matrix $\boldsymbol{K}$. If $\gamma_k = 1$, the i^{th} feature is included in the kernel matrix construction.

For $\boldsymbol{\gamma}$, as in Narisetty and He (2014), we assume

$$\pi(\boldsymbol{\gamma}) \propto \prod_{k=1}^{p} \left(\frac{1}{p}\right)^{\gamma_k} \left(1 - \frac{1}{p}\right)^{1-\gamma_k},$$

where p is the number of variables.

Other variable selection, as suggested in Zhang et al. (2016); Tipping (2001); Zhang et al. (2011), learning a relevant set of active vectors can either help reducing the computational cost or help both reducing the computational cost and improving the prediction accuracy. To show the merit, we give an illustrative example in Figure 3.1 motivated by Zhang et al. (2016). In the figure, we plot the true curve, the curve estimated by sparse kernels, and the curve estimated by full kernels. This figure clearly shows that including all vectors would lead to over-fitting problems.

Note that the sparse kernel estimation can be achieved by sparse estimation of the linear coefficients $\boldsymbol{\beta}$. A larger value of β_j would put more weight on j^{th} vector, indicating its relatively stronger significance. To achieve the sparse estimation of $\boldsymbol{\beta}$, we also employ a spike and slab prior proposed by George and McCulloch (1993) originally for the variable selection of linear regression models. As with the variable selection index vector $\boldsymbol{\gamma}$, we append another binary index vector $\boldsymbol{g} = (g_j, \cdots, g_n)^\top$ to our model. The spike and slab prior for $\boldsymbol{\beta}$ is then defined as

$$\begin{aligned}\pi(\boldsymbol{\beta}|\sigma^2, \boldsymbol{g}, \nu_1) = \prod_{j=1}^{n} \left\{(1-g_j)\mathcal{N}(\beta_j, 0, \sigma^2\nu_0) + g_j\mathcal{N}(\beta_j, 0, \sigma^2\nu_1)\right\} \\ \times \mathcal{N}(\beta_0, 0, \lambda_0\sigma^2),\end{aligned} \tag{3.4}$$

where ν_0 and λ_0 are pre-specified hyperparameters such that $\nu_0 \approx 0$ and $\lambda_0 \approx \infty$. Note that if $g_j = 1$, then a flat normal prior is assigned to the j^{th} vector. If $g_j = 0$, the j^{th} vector is approximately excluded due to the assignment of a spike prior. For ν_1 and σ^2, we assume the inverse-Gamma priors:

$$\begin{aligned}\pi(\nu_1) &\sim \text{inv-Gamma}(a_{\nu_1}, b_{\nu_1}), \\ \pi(\sigma^2) &\sim \text{inverse-Gamma}(a_\sigma, b_\sigma),\end{aligned} \tag{3.5}$$

where a_{ν_1}, b_{ν_1}, a_σ and b_σ are hyper-parameters that need to be pre-specified.

For the binary index vector $\boldsymbol{g}$, we assume

$$\pi(\boldsymbol{g}) \propto \prod_{j=1}^{n} \left(\frac{c_n}{n}\right)^{g_j} \left(1 - \frac{c_n}{n}\right)^{1-g_j}$$

as proposed by Narisetty and He (2014). For the choice of the coefficient c_n, we choose it such that $\Phi((g_{\max} - c_n)/\sqrt{c_n(1-c_n/n)}) \approx 1$ as suggested by Narisetty and He (2014), where $g_{\max}$ is the upper bound size for number of the active vectors. With the motivation of learning a parsimonious sparse kernel, we set $g_{\max}$ to be half of the sample size.

3.2 Posterior inference

3.2.1 Posterior sampling via the collapsed Gibbs sampler

With the prior set-up in Section 3.1.2 and the model defined in (3.3), the posterior is obtained as

$$\begin{aligned}\pi(\boldsymbol{\gamma},\boldsymbol{\rho},\boldsymbol{g},\boldsymbol{\beta},\sigma^2,\nu_1|\boldsymbol{y}) \propto & \frac{1}{\sqrt{|\sigma^2\mathbb{I}|}}\exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{y}-\boldsymbol{K}^\top\boldsymbol{\beta})^\top(\boldsymbol{y}-\boldsymbol{K}^\top\boldsymbol{\beta})\right\} \\ & \times \frac{1}{\sqrt{|\sigma^2\boldsymbol{V}|}}\exp\left\{-\frac{1}{2\sigma^2}\boldsymbol{\beta}^\top\boldsymbol{V}^{-1}\boldsymbol{\beta}\right\} \\ & \times (\frac{1}{\sigma^2})^{a_\sigma+1}\exp\left\{-\frac{b_\sigma}{\sigma^2}\right\}\times(\frac{1}{\nu_1})^{a_{\nu_1}+1}\exp\left\{-\frac{b_{\nu_1}}{\nu_1}\right\} \\ & \times \prod_{k=1}^{p}\gamma_k\mathbb{I}[0<\rho_k<1]+(1-\gamma_k)\delta_1(\rho_k) \\ & \times \prod_{k=1}^{p}\left(\frac{1}{p}\right)^{\gamma_k}\left(1-\frac{1}{p}\right)^{1-\gamma_k}\times\prod_{j=1}^{n}\left(\frac{c_n}{n}\right)^{g_j}\left(1-\frac{c_n}{n}\right)^{1-g_j},\end{aligned} \tag{3.6}$$

where $\boldsymbol{V}$ is defined as a $n+1$ by $n+1$ diagonal matrix with diagonal elements, ν_0, ν_1 and λ_0.

The equation (3.6) is complex and so making direct inference from this distribution is impossible. Specifically, directly sampling from this distribution is infeasible due to the complex structure. One popular way for sampling from the posterior is the Gibbs sampler (Gelfand and Smith, 1990). However, the traditional Gibbs sampler may fail in this situation. In particular, the sampling from the full conditionals of $\boldsymbol{\gamma}$ and $\boldsymbol{\rho}$ are highly sensitive to the value of $\boldsymbol{\beta}$ and σ^2. As a result, directly sampling from the full conditionals of $\boldsymbol{\gamma}$ and $\boldsymbol{\rho}$ would lead to the poor mixing as well as the slow convergence of the Markov chain.

To address this issue, we propose a collapsed Gibbs sampler (Liu, 1994). We iteratively generate the joint posterior samples from the following conditional distributions until convergence:

Step 1. Jointly generate $(\boldsymbol{\gamma},\boldsymbol{\rho})$ from $\pi(\boldsymbol{\gamma},\boldsymbol{\rho}|\boldsymbol{g},\nu_1,\boldsymbol{y})$.

Step 2. Generate $\boldsymbol{\beta}$ from $\pi(\boldsymbol{\beta}|\boldsymbol{\gamma},\boldsymbol{\rho},\sigma^2,\boldsymbol{g},\nu_1,\boldsymbol{y})$.

Step 3. Generate σ^2 from $\pi(\sigma^2|\boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{g}, \boldsymbol{\beta}, \nu_1, \boldsymbol{y})$.

Step 4. Generate ν_1 from $\pi(\nu_1|\boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{g}, \boldsymbol{\beta}, \sigma^2, \boldsymbol{y})$.

Step 5. Generate each g_i for $j = 1, \cdots, n$ from $\pi(g_j|\boldsymbol{g}_{-j}, \boldsymbol{\gamma}, \boldsymbol{\rho}, \sigma^2, \boldsymbol{\beta}, \nu_1, \boldsymbol{y})$.

Let $\{(\boldsymbol{\gamma}^{(t)}, \boldsymbol{\rho}^{(t)}, \boldsymbol{\beta}^{(t)}, \sigma^{2(t)}, \nu_1^{(t)}, \boldsymbol{g}^{(t)}) : t = 1, \cdots, T\}$ be the Markov chain generated by the above collapsed Gibbs sampler. Using the MCMC computation theory, it can be easily shown that the strationary distribution of the Markov chain is $\pi(\boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\beta}, \sigma^2, \nu_1, \boldsymbol{g}|\boldsymbol{y})$.

3.2.2 Conditional distribution and implementation details

In this section, we discuss the implementation of the proposed collapsed Gibbs sampler. First, we give the sampling procedure for updating $\boldsymbol{\gamma}$ and $\boldsymbol{\rho}$. We revise the sampling scheme of Savitsky et al. (2011) using the reversible jump MCMC idea of Green (1995) and Gottardo and Raftery (2008). Specifically, we jointly update the samples for $(\boldsymbol{\gamma}, \boldsymbol{\rho})$ as follows: Generate samples from $p(\boldsymbol{\rho}, \boldsymbol{\gamma}|\boldsymbol{g}, \nu_1, \boldsymbol{y})$ via the Metropolis-Hastings algorithm. The update for $(\rho_k, \gamma_k)^\top$ is performed with two moves conducted in successions for $k = 1, \cdots, p$:

1 *Between-models move*: Jointly propose a new model such that if $\gamma_k = 1$, propose $\gamma_k' = 0$ and set $\rho_k' = 1$. If $\gamma_k = 0$, propose $\gamma_k' = 1$ and randomly draw $\rho_k' \sim \mathcal{U}(0, 1)$. Accept the proposed value of $(\gamma_k', \rho_k')^\top$ with the probability of

$$\alpha = \min\left\{1, \frac{\pi(\gamma_k', \rho_k'|\boldsymbol{\gamma}_{-k}, \boldsymbol{\rho}_{-k}, \boldsymbol{g}, \nu_1, \boldsymbol{y})}{\pi(\gamma_k, \rho_k|\boldsymbol{\gamma}_{-k}, \boldsymbol{\rho}_{-k}, \boldsymbol{g}, \nu_1, \boldsymbol{y})}\right\}.$$

As suggested by Savitsky et al. (2011), the proposal ratio reduces to 1 since we employ a uniform proposal for ρ_k and a symmetric Dirac measure proposal for γ_k.

2 *With-in model move*: This move is performed only for those $\gamma_k = 1$ resulted from the previous between-model move. We set $\gamma_k' = 1$ first. Instead of using the uniform proposal for ρ proposed by Savitsky et al. (2011), we use an adaptive random walk proposal for better mixing (Andrieu and Thoms, 2008; Roberts and Rosenthal, 2009). We draw $\rho_k' \sim \mathcal{U}(\rho_k - \frac{s_{\rho_k}}{2}, \rho_k + \frac{s_{\rho_k}}{2})$, where s_{ρ_k} is the standard deviation for ρ_k computed

with the generated MCMC samples. We accept the joint proposal for $(\gamma_k', \rho_k')^\top$ with the probability of

$$\alpha = \min\left\{1, \frac{\pi(\gamma_k', \rho_k' | \boldsymbol{\gamma}_{-k}, \boldsymbol{\rho}_{-k}, \boldsymbol{g}, \nu_1, \boldsymbol{y})}{\pi(\gamma_k, \rho_k | \boldsymbol{\gamma}_{-k}, \boldsymbol{\rho}_{-k}, \boldsymbol{g}, \nu_1, \boldsymbol{y})}\right\}.$$

Again, the proposal ratio reduces to 1 as in the between model move. One could also use Laplace approximation (Tierney and Kadane, 1986) for this step. However, a potential drawback of the Laplace method is the computational cost associated with the optimization step.

Note that $\pi(\boldsymbol{\gamma}, \boldsymbol{\rho} | \boldsymbol{g}, \nu_1, \boldsymbol{y}) \propto \pi(\boldsymbol{y} | \boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{g}, \nu_1)\pi(\boldsymbol{\rho}, \boldsymbol{\gamma})$. Due to the conjugacy, we can compute the marginal likelihood of $\pi(\boldsymbol{y} | \boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{g}, \nu_1)$ by integrating out $\boldsymbol{\beta}$ and σ^2 from the full likelihood as follows:

$$\pi(\boldsymbol{y} | \boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{g}, \nu_1) \propto |\boldsymbol{K}^\top \boldsymbol{K} + \boldsymbol{V}^{-1}|^{-\frac{1}{2}} \frac{1}{b^{*a^*}} \tag{3.7}$$

where $a^* = \frac{n+2a_\sigma}{2}$ and $b^* = \frac{1}{2}(\boldsymbol{y}^\top(\mathbb{I} - \boldsymbol{K}(\boldsymbol{K}^\top \boldsymbol{K} + \boldsymbol{V}^{-1})^{-1}\boldsymbol{K}^\top)\boldsymbol{y}) + b_\sigma$. The derivation of equation (3.7) is given in appendix A.1.

To speed up computing the marginal likelihood, we divide $\boldsymbol{\beta}$ into two partitions $[\boldsymbol{\beta}_{\boldsymbol{g}}, \boldsymbol{\beta}_I]$ and $\boldsymbol{K}$ into $[\boldsymbol{K}_{\boldsymbol{g}}, \boldsymbol{K}_I]$. As proposed by Narisetty et al. (2018), we make this division by the vector selection index $\boldsymbol{g}$. To be more specific, we let $\boldsymbol{\beta}_{\boldsymbol{g}}$ and $\boldsymbol{\beta}_I$ to contain the elements of $\boldsymbol{\beta}$ corresponding to $g_j = 1$ and $g_j = 0$ respectively. Similarly, we let $\boldsymbol{K}_{\boldsymbol{g}}$ and $\boldsymbol{K}_I$ contain the columns of $\boldsymbol{K}$ corresponding to $g_j = 1$ and $g_j = 0$. Due to the spike shrinkage prior, we have the approximately zero coefficients in $\boldsymbol{\beta}_I$. Hence, we can rewrite the model as

$$\boldsymbol{y} = \boldsymbol{K}_{\boldsymbol{g}}\boldsymbol{\beta}_{\boldsymbol{g}} + \boldsymbol{K}_I\boldsymbol{\beta}_I + \epsilon \approx \boldsymbol{K}_{\boldsymbol{g}}\boldsymbol{\beta}_{\boldsymbol{g}} + \epsilon.$$

Similar to the equation (3.7), we integrate out $\boldsymbol{\beta}_{\boldsymbol{g}}$ and σ^2 and then obtain the marginal likelihood as

$$\pi(\boldsymbol{y} | \boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{g}, \nu_1) \approx c|\boldsymbol{K}_{\boldsymbol{g}}^\top \boldsymbol{K}_{\boldsymbol{g}} + \boldsymbol{V}_{\boldsymbol{g}}^{-1}|^{-\frac{1}{2}} \frac{1}{\tilde{b^*}^{a^*}},$$

where c is constant, $a^* = \frac{n+2a_\sigma}{2}$, and

$$\tilde{b^*} = \frac{1}{2}(\boldsymbol{y}^\top(\mathbb{I} - \boldsymbol{K_g}(\boldsymbol{K_g}^\top \boldsymbol{K_g} + \boldsymbol{V_g}^{-1})^{-1}\boldsymbol{K_g}^\top)\boldsymbol{y}) + b_\sigma.$$

In addition, we define $\boldsymbol{V_g}$ to be a $(n_{\boldsymbol{g}}+1) \times (n_{\boldsymbol{g}}+1)$ diagonal matrix with ν_1 and λ_0 as its diagonal elements, where $n_{\boldsymbol{g}}$ is defined as the number of active vectors, i.e., $n_{\boldsymbol{g}} = \sum_{j=1}^{n} g_j$.

It is easy to show that the full conditional distributions of the rest model parameter are as follows:

$$\begin{aligned}
&\boldsymbol{\beta}|\boldsymbol{\gamma}, \boldsymbol{\rho}, \sigma^2, \boldsymbol{g}, \nu_1, \boldsymbol{y} \sim \mathcal{N}\left(\left[\boldsymbol{K}^\top\boldsymbol{K} + \boldsymbol{V}^{-1}\right]^{-1}\boldsymbol{K}^\top\boldsymbol{y}, \sigma^2\left[\boldsymbol{K}^\top\boldsymbol{K} + \boldsymbol{V}^{-1}\right]^{-1}\right)\\
&\sigma^2|\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{g}, \nu_1, \boldsymbol{y} \sim \text{Inverse-Gamma}\left(\frac{2n+2a_\sigma+1}{2}, \frac{||\boldsymbol{y}-\boldsymbol{K}\boldsymbol{\beta}||^2 + \boldsymbol{\beta}^\top\boldsymbol{V}^{-1}\boldsymbol{\beta}}{2} + b_\sigma\right)\\
&\nu_1|\boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\beta}, \sigma^2, \boldsymbol{g}, \boldsymbol{y} \sim \text{inverse-Gamma}\left(\frac{|\boldsymbol{g}|}{2} + a_{\nu_1}, \frac{||\boldsymbol{\beta_g}||^2}{2\sigma^2} + b_{\nu_1}\right)\\
&g_j|\boldsymbol{g}_{-j}, \boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\beta}, \sigma^2, \nu_1, \boldsymbol{y} \sim \text{Bernoulli}\left(\frac{\frac{c_n}{n}\phi(\beta_j; 0, \sigma^2\nu_1)}{\frac{c_n}{n}\phi(\beta_j; 0, \sigma^2\nu_1) + (1-\frac{c_n}{n})\phi(\beta_j; 0, \sigma^2\nu_0)}\right)
\end{aligned}$$

for $j = 1, \cdots, n$.

3.3 Extension for dealing with large sample size

In addition to variable selection, a challenge of the RKHS approach is the computational burden when the sample size is large. To handle the large n problem, many approaches have been developed under the framework of Gaussian process (GP) based models, which is the most popular way to learn an unknown nonlinear function from the Bayesian perspective (Hensman et al., 2013; Liu et al., 2020). Among those approaches, one popular way is to select fewer representative points of $m \ll n$ via some criteria and learn the function with those fewer induced points, for instance, Snelson and Ghahramani (2006b), Seeger et al. (2003).

Under the RKHS framework, learning a sparse representation with fewer supporting vectors can achieve similar results as learning with full data (Tipping, 2001; Zhang et al.,

2011, 2016). In particular, Zhang et al. (2011) showed that learning the function with fewer selected vectors can be equivalent to the sparse GP with the subset of a regressors (Williams and Rasmussen, 2006). The perk of learning the sparse kernel representation is that it can significantly reduce the computational cost and so the method is scalable to the large sample cases.

However, the question of how to perform variable selection given the large data is seldom addressed. For instance, the DoSK method proposed by Chen et al. (2018) only addressed selecting vectors for better model fit under the smaller sample settings. In this section, we modify our proposed collapsed Gibbs sampler to fill this gap.

3.3.1 The modified Collapsed Gibbs sampler

The key idea of our modification is to train a model based on the 'active' subset of data corresponding to $g_j = 1$. With this 'active' subset of data, we rewrite our model as

$$\boldsymbol{y_g} = \tilde{\boldsymbol{K}}\boldsymbol{\beta_g} + \epsilon, \tag{3.8}$$

where $\boldsymbol{y_g}$ is a subset of the original $\boldsymbol{y}$ that contains the components that corresponding to $g_j = 1$ and $\tilde{\boldsymbol{K}}$ is a submatrix of $\boldsymbol{K_g}$ consisting only its rows that corresponding to $g_j = 1$.

Based on model (3.8), we present our modified collapsed Gibbs sampler. For $(\boldsymbol{\gamma}, \boldsymbol{\rho})^\top$, we still generate samples via the same Metropolis-Hastings algorithm introduced in the Section 3.2.2. Here, the marginal likelihood is given by

$$\pi(\boldsymbol{\gamma}, \boldsymbol{\rho}|\boldsymbol{g}, \nu_1, \boldsymbol{y_g}) \propto |\tilde{\boldsymbol{K}}^\top\tilde{\boldsymbol{K}} + \boldsymbol{V_g}^{-1}|^{-\frac{1}{2}} \frac{1}{{b^*}^{a^*}},$$

where $a^* = \frac{n_{\boldsymbol{g}}+2a_\sigma}{2}$ and

$$b^* = \frac{1}{2}(\boldsymbol{y_g}^\top(\mathbb{I} - \tilde{\boldsymbol{K}}(\tilde{\boldsymbol{K}}^\top\tilde{\boldsymbol{K}} + \boldsymbol{V_g}^{-1})^{-1}\tilde{\boldsymbol{K}}^\top)\boldsymbol{y_g}) + b_\sigma.$$

Then we employ the idea of the "skinny Gibbs" proposed by Narisetty et al. (2018) for

sampling from the full conditionals of $\boldsymbol{\beta}$ and $\boldsymbol{g}$. For $\boldsymbol{\beta_g}$, we draw samples from

$$\boldsymbol{\beta_g}|\boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\beta}_I, \sigma^2, \boldsymbol{g}, \nu_1, \boldsymbol{y_g} \sim \mathcal{N}\left(\left[\tilde{\boldsymbol{K}}^\top \tilde{\boldsymbol{K}} + \boldsymbol{V_g}^{-1}\right]^{-1} \tilde{\boldsymbol{K}}^\top \boldsymbol{y_g}, \sigma^2 \left[\tilde{\boldsymbol{K}}^\top \tilde{\boldsymbol{K}} + \boldsymbol{V_g}^{-1}\right]^{-1}\right).$$

For $\boldsymbol{\beta}_I$, we generate samples from

$$\boldsymbol{\beta_I}|\boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\beta_g}, \sigma^2, \boldsymbol{g}, \nu_1, \boldsymbol{y} \sim \mathcal{N}\left(0, \boldsymbol{S_I}^{-1}\right),$$

where $\boldsymbol{S_I} = \text{Diag}\left(\boldsymbol{K_I}^\top \boldsymbol{K_I} + (\sigma^2\nu_0)^{-1}\mathbb{I}_{n-|\boldsymbol{g}|}\right)$. σ^2 and ν_1 are updated via

$$\sigma^2|\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{g}, \nu_1, \boldsymbol{y_g} \sim \text{Inverse-Gamma}\left(\frac{2|\boldsymbol{g}| + 2a_\sigma + 1}{2}, \frac{||\boldsymbol{y_g} - \tilde{\boldsymbol{K}}\boldsymbol{\beta_g}||^2 + \boldsymbol{\beta_g}^\top \boldsymbol{V_g}^{-1}\boldsymbol{\beta_g}}{2} + b_\sigma\right)$$

$$\nu_1|\boldsymbol{\gamma}, \boldsymbol{\beta}, \sigma^2, \boldsymbol{g}, \boldsymbol{y_g} \sim \text{inverse-Gamma}\left(\frac{|\boldsymbol{g}|}{2} + a_{\nu_1}, \frac{||\boldsymbol{\beta_g}||^2}{2\sigma^2} + b_{\nu_1}\right).$$

The last parameter is the vector extraction index $\boldsymbol{g}$, which governs the selection of the active subset of data in each iteration. We update $\boldsymbol{g}$ via

$$g_j|\boldsymbol{g}_{-j}, \boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\beta}, \sigma^2, \nu_1, \boldsymbol{y} \sim \text{Bernoulli}(\omega_j^*)$$

for $j \in 1, \cdots, n$, where

$$\omega_j^* = \left[1 + \frac{(1 - \frac{c_n}{n})\phi(\beta_j; 0, \nu_0\sigma^2)}{\frac{c_n}{n}\phi(\beta_j; 0, \nu_1\sigma^2)} \exp\left\{-\frac{2\beta_j \tilde{\boldsymbol{K}}_j^\top(\boldsymbol{y_g} - \tilde{\boldsymbol{K}}_{C_j}\boldsymbol{\beta}_{C_j}) + (\sigma^2 - 1)\tilde{\boldsymbol{K}}_j^\top \tilde{\boldsymbol{K}}_j \beta_j^2}{2\sigma^2}\right\}\right]^{-1}$$

where $C_j = \{k : k \neq j, g_k = 1\}$, $\tilde{\boldsymbol{K}}_{C_j}$ is a submatrix of $\tilde{\boldsymbol{K}}$ with columns indexed by the C_j, and $\tilde{\boldsymbol{K}}_j$ is the j^{th} column of $\tilde{\boldsymbol{K}}$.

For the choice the c_n, we use the similar idea presented in the Section 3.2. The modification is that $g_{\max}$ is chosen now by the statistician instead of setting it to be $\frac{n}{2}$. The trade-off is that a large number $g_{\max}$ would lead to more accurate predictions but also increases computational costs. For the starting point of $\boldsymbol{g}$, we divide the data $\boldsymbol{y}$ into several grid and randomly generate subsamples from each grid with a total size of $m < g_{\max}$ to ensure an

approximate uniform coverage of the data $\boldsymbol{y}$.

3.4 Prediction

One major objective of our proposed method is to make predictions for unseen observations. Let $\boldsymbol{x}_{\text{new}}$ be the input data of the unobserved points and $\mathcal{D} = \{(x_i, y_i), i = 1, \cdots, n\}$ be the observed data set. For the simplicity of notations, we denote the set of all model parameters by $\boldsymbol{\Theta} = (\boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\beta}, \boldsymbol{g}, \sigma^2, \nu_1)$. Then the posterior predictive distribution is given as

$$\pi(\boldsymbol{y}_{\text{new}}|\boldsymbol{x}_{\text{new}}, \mathcal{D}) = \int_{\boldsymbol{\Theta}} \pi(\boldsymbol{y}_{\text{new}}|\boldsymbol{\Theta}, \mathcal{D})\pi(\boldsymbol{\Theta}|\mathcal{D})d\boldsymbol{\Theta},$$

where $\pi(\boldsymbol{\Theta}|\mathcal{D})$ is the posterior distribution of model parameters. Using the MCMC technique, we use the following steps to make prediction:

step 1. Generate T samples of model parameters from its posterior distribution $\pi(\boldsymbol{\Theta}|\mathcal{D})$ via the collapsed Gibbs sampler given in Section 3.2 or Section 3.3.

step 2. Generate predictive samples $\{\boldsymbol{y}_{\text{new}}^{(t)}, t = 1, \cdots, T\}$ for $\boldsymbol{y}_{\text{new}}$ from the predictive distribution $\pi(\boldsymbol{y}_{\text{new}}|\mathcal{D}, \boldsymbol{\Theta})$.

step 3. Compute $\hat{\boldsymbol{y}}_{\text{new}} = \frac{1}{T}\sum_{i=1}^{T} \boldsymbol{y}_{\text{new}}^{(t)}$.

Note that $\pi(\boldsymbol{y}_{\text{new}}|\boldsymbol{\Theta}, \mathcal{D}) \sim \mathcal{N}(\boldsymbol{K}_{\text{new}}\boldsymbol{\beta}_{\boldsymbol{g}}, \sigma^2)$, where $\boldsymbol{K}_{\text{new}}$ is the $n_{\text{new}} \times (n_{\boldsymbol{g}} + 1)$ prediction kernel matrix computed based on the unseen points and the active vectors denoted by index vector $\boldsymbol{g}$. By using those active vectors only can we significantly reduce the computational cost. This can be most effective when the size of prediction set is large.

3.5 Simulation studies

In this section, we conduct simulation studies to examine our proposed method. We consider three cases for data generation. We consider a small data size for Cases 1 and 2, and the large data size for Case 3. We give the details of the data generating procedures as follows.

Case 1. $y_i = 10\exp\left(-x_{ik_1}^2\right) + \epsilon_i$, where k_1 is randomly selected from $\{1, \cdots, p\}$ and $\epsilon_i \overset{i.i.d}{\sim} \mathcal{U}(-2, 2)$. Each x_k is simulated from $\mathcal{U}(-5, 5)$ with $n = 50$ and $p = 20, 200, 500$.

Case 2. $y_i = 2\cos(x_{ik_1}) + \frac{x_{ik_2}^2}{2} + \frac{\exp(x_{ik_3})}{5} + \epsilon_i$, where k_1, k_2, k_3 are randomly selected from $\{1, \cdots, p\}$ and $\epsilon_i \overset{i.i.d}{\sim} \mathcal{N}(0, 1)$. and $\epsilon \sim \mathcal{N}(0, 1)$. Each x_k is simulated from $\mathcal{U}(-\pi, \pi)$ with $n = 100$ and $p = 20, 200, 500$.

Case 3. $y_i = 2\cos(x_{ik_1}) + \frac{x_{ik_2}^2}{2} + \frac{\exp(x_{ik_3})}{5} + x_{ik_4} + \epsilon_i$, where k_1, k_2, k_3, k_4 are randomly selected from $\{1, \cdots, p\}$ and $\epsilon_i \overset{i.i.d}{\sim} \mathcal{N}(0, 1)$. Each x_k are simulated from $\mathcal{U}(-\pi, \pi)$ with $n = 1,000$ and $p = 20, 200$.

For the simulation studies, we compare our proposed Bayesian doubly-sparse regression model to several existing Bayesian RKHS regression models tabulated in Table 3.1. For each experiment, we measure the performance of each models with the measurements tabulated in Table 3.2.

For the choice of the hyperparameters, we set $a_\sigma = b_\sigma = a_{\nu_1} = b_{\nu_1} = a_\lambda = b_\lambda = 10^{-3}$. For the spike and slab hyperparameters, we set $\nu_0 = 10^{-3}$ and $\lambda_0 = 10^3$. For c_n, we set $c_n = 8$ for Case 1 and $c_n = 30$ for Case 2. For Case 3, we set $g_{\max} = 200$ and results in $c_n = 60$. For the choice of the starting point, we assume $\boldsymbol{\gamma}$ to be the vector with one feature selected such that the selected feature maximizes the marginal likelihood with the response. In addition, we choose the Laplacian kernel for Case 1 and Gaussian kernel for Cases 2, 3.

Table 3.1: The Bayesian RKHS regression models to be compared

Abbreviation	Method	Remarks
True-ssvs	Bayesian RKHS regression with true variables and spike and slab prior on $\boldsymbol{\beta}$	Trained with true models, sparse kernel estimated
Full-ssvs	Bayesian RKHS regression with full variables and spike and slab prior on $\boldsymbol{\beta}$	Variable selection is not addressed, sparse kernel estimated, regression version of Zhang et al. (2011)
VS-L_2	Bayesian RKHS regression with variable selection and ridge prior on $\boldsymbol{\beta}$	Variable selection is conducted, no sparse kernel matrix estimation, regression version of Chakraborty (2009), details are given in the Appendix B.1

Table 3.2: Measurements for model performance

Abbreviation	Explanation	Formula
MSPE	Mean squared prediction error	$\text{MSPE} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} (\hat{y}_i^{\text{new}} - y_i)^2$
TP	Number of true positive variables	$\lvert\{k : \gamma_k = 1 \cap \hat{\gamma}_k = 1\}\rvert$
TF	Number of true negative variables	$\lvert\{k : \gamma_k = 0 \cap \hat{\gamma}_k = 0\}\rvert$
Train time	CPU time for training the model	
Pred time	CPU time for prediction	

$\boldsymbol{\gamma}$ and $\boldsymbol{\gamma}^*$ denote binary index for the true model and estimated model. We set $\hat{\gamma}_j = 1$ if $\pi(\gamma_k = 1|\boldsymbol{y}) > 0.5$ and $\hat{\gamma}_k = 0$ otherwise. The CPU time is measured in seconds.

Table 3.3: Case 1 simulation results

Dimensions	Method	MSPE	TP	TN	Train time	Pred time
p=20	True-ssvs	1.707(0.008)	1(0)	19(0)	18.64	4.526
	Full-ssvs	9.125(0.12)	1(0)	0(0)	180.046	15.589
	VS-L_2	1.815(0.01)	1(0)	18.987(0.004)	528.129	14.405
	Doubly-sparse	1.724(0.009)	1(0)	18.994(0.002)	126.167	4.1
p=200	True-ssvs	1.701(0.009)	1(0)	199(0)	19.447	4.326
	Full-ssvs	12.199(0.046)	1(0)	0(0)	1643.766	165.10
	VS-L_2	1.861(0.02)	0.997(0.002)	198.983(0.004)	4889.586	14.012
	Doubly-sparse	1.755(0.019)	0.997(0.002)	198.983(0.005)	1078.374	4.101
p=500	True-ssvs	1.701(0.009)	1(0)	499(0)	17.217	3.6
	Full-ssvs	12.085(0.043)	1(0)	0(0)	3754.667	435.944
	VS-L_2	1.873(0.024)	0.994(0.002)	498.979 (0.005)	10865.85	12.153
	Doubly-sparse	1.781(0.024)	0.994(0.002)	498.983 (0.004)	2385.619	3.755

The estimated standard errors are given in the parenthesis.

We repeat the Monte Carlo experiments $1,000$ times for Cases 1 and 2, and 50 times for Case 3. For each method in Cases 1 and 2, we let the MCMC run for $10,000$ iterations with the first $5,000$ iterations as burn-in samples. For Case 3, we set MCMC to run $5,000$ times and set the first $2,000$ iterations as burn-in. We tabulate the results in Tables 3.3, 3.4 and 3.5 respectively. In addition, we also plot the MSPE for each method in Figure 3.2. The error bar is plotted to be equal to 2 times the estimated standard error. Furthermore, we plotted the CPU time for each method in Cases 1 and 2 in Figure 3.3.

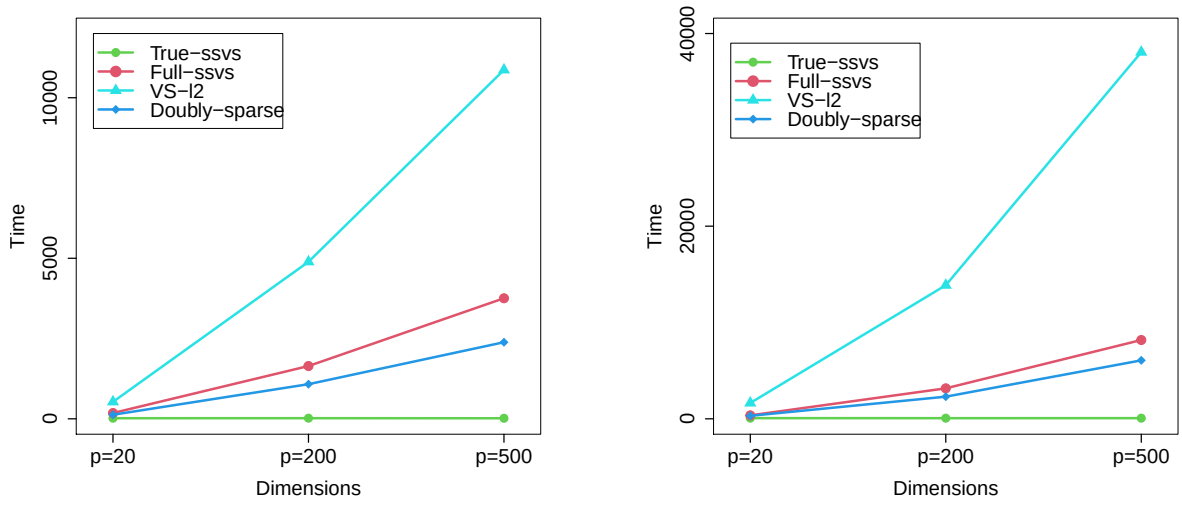
From the experiment results of Cases 1 and 2, we observe that our proposed method provides smaller or equal prediction errors compared to the VS-L_2. With the simulation results of Case 1, we can see that including all vectors would lead to the problem of overfitting, which is corroborated by Zhang et al. (2016). In terms of variable selection, VS-L_2 and the proposed method have similar performances. In addition, including all variables would lead

to poor prediction if there exist many irrelevant variables. This suggest that the sparse kernel methods proposed by Zhang et al. (2011, 2016) would fail when many noisy variables exist. Other than the prediction accuracy and variable selection, another benefit of our proposed method is the reduction of the computational cost. In Figure 3.2, for both Cases 1 and 2, the doubly-sparse model are 4 to 5 times faster than the model trained with all data points. For the large sample scenarios in Case 3, we observe that our proposed method performs significantly better than the sparse model on prediction performance. Furthermore, our proposed method also requires smaller computational cost.

Table 3.4: Case 2 simulation results

Dimensions	Method	MSPE	TP	TN	Train Time	Pred Time
p=20	True-ssvs	1.273(0.003)	3(0)	17(0)	69.62	13.79
	Full-ssvs	1.968(0.01)	3(0)	0(0)	358.105	36.14
	VS-L_2	1.276(0.0035)	3(0)	16.992(0.003)	1642.141	30.37
	Doubly-sparse	1.276(0.0035)	3(0)	16.991(0.003)	309.634	13.38
p=200	True-ssvs	1.279(0.004)	3(0)	197(0)	65.934	12.89
	Full-ssvs	6.401(0.013)	3(0)	0(0)	3164.186	374.28
	VS-L_2	1.284(0.0035)	3(0)	196.993 (0.002)	13868.27	29.2
	Doubly-sparse	1.284(0.0035)	3(0)	196.992 (0.003)	2299.298	12.34
p=500	True-ssvs	1.279(0.003)	3(0)	497(0)	67.969	13.33
	Full-ssvs	6.392(0.012)	3(0)	0(0)	8179.819	1048.23
	VS-L_2	1.284(0.004)	2.998(0.0014)	496.991(0.003)	38064.95	36.22
	Doubly-sparse	1.285(0.004)	2.997(0.0017)	496.994(0.002)	6068.945	14.17

The estimated standard errors are given in the parenthesis.



(a) Case 1

(b) Case 2

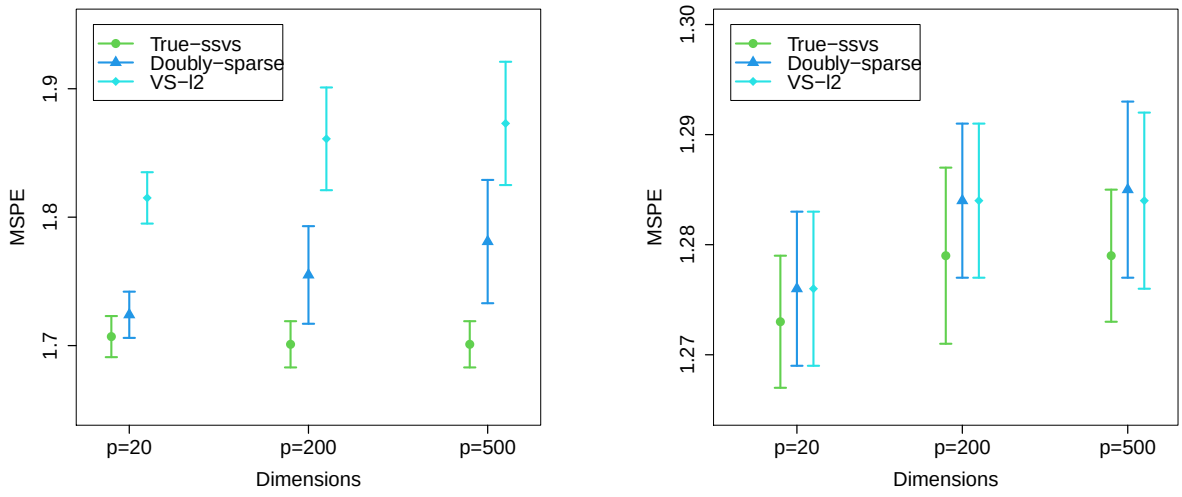
Figure 3.2: Training time for each methods

3.6 Real data analysis

In this section, we apply our proposed Bayesian doubly-sparse RKHS method to real datasets to demonstrate its advantages.

3.6.1 The Bardet-Biedl syndrome Gene expression data

The Bardet-Biedl syndrome Gene expression (Trim32) data were first introduced by Scheetz et al. (2006) and also analyzed in many works (Fan et al., 2011; Huang et al., 2010). As discussed in Fan et al. (2011); Huang et al. (2010), the goal is to build a statistical model to predict the expressions of the TRIM32 gene, which leads to the Bardet-Biedl Sydrome. The micro-array data were gathered from tissues of eyes from 120 twelve-week-old rats. We use the *TRIM32* dataset available at the **abess** R package with sample size of $n = 120$ and pre-selected 500 genes expressions.



(a) Case 1 (b) Case 2

Figure 3.3: MSPE for each methods

Table 3.5: Case 3 simulation results

Dimensions	Method	MSPE	TP	TN	Train time	Pred time
p=20	Doubly-sparse	1.264(0.0126)	4(0)	16(0)	11080.88	41.889
	Full-ssvs	1.4524(0.0138)	4(0)	0(0)	11237.04	140.564
p=200	Doubly-sparse	1.278(0.01)	4(0)	196(0)	46489.95	33.255
	Full-ssvs	9.62(0.078)	4(0)	0(0)	132038.1	2036.434

The estimated standard errors are given in the parenthesis.

Since the data are not pre-spitted, we conduct Monte Carlo cross validation (repeated 1000 times) to compare and evaluate the performance of our proposed method to other existing methods. In each experiment, we split the data into two sets with 100 observations as the training set and the remaining 20 observations as the testing set. We record the MSPE computed on the testing set for each method. To evaluate the computational cost, we record the CPU time for training measured by seconds.

As in Section 3.5, we compare our proposed Bayesian doubly-sparse method (denoted by Doubly-sparse) to the methods tabulated in Table 3.1. In addition, we adopt the same

setting as in the Case 2 simulation studies. The results are tabulated in Table 3.6.

Table 3.6: Trim32 data analysis results

Method	MSPE	Time
Full-ssvs	0.01893(0.0006)	7521.816
VS-L_2	0.00855(0.000155)	27557.3
Doubly-sparse	**0.00849(0.00015)**	**6016.71**

The estimated standard errors are given in the parenthesis.

From the table 3.6, we can see that the full model has larger prediction error compared to methods with variable selections. This indicates the necessity of variable selection for analyzing this dataset. In addition, our proposed doubly-sparse method not only has the smallest prediction error but also has the smallest computational cost. To be more specific, the doubly-sparse method is more than 4 times faster than VS-L_2.

3.6.2 The breast invasive carcinoma (BRCA) data

The breast invasive carcinoma (BRCA) data are available through the Cancer Genome Atlas (TCGA) Research Network: http://cancergenome.nih.gov. The original data contains 526 observations and $17,814$ gene expressions recorded on the log scale. The objective of the analysis is to identify the most important genes and train a model to make predictions on the expressions of BRCA1. The BRCA1 gene produces proteins that help repair the damaged DNA. The risk of developing breast cancer increases tremendously if one inherits a harmful variant of BRCA1 gene (Kuchenbaecker et al., 2017).

To analyze the data, we first screen the variables by training a single variable Gaussian process regression model and computing its marginal likelihood via the Laplace method Tierney and Kadane (1986). We retain the first $1,000$ gene expressions with largest marginal likelihood. Then we conduct Monte Carlo cross validation by randomly spiting the data into 500 observations for training and the rest for testing. For each experiment, we record

the mean squared prediction error computed on the testing set and record the CPU time measured by seconds.

To train models for predictions, we apply our proposed updated collapsed Gibbs sampler (denoted by "Doubly-sparse") due to the large sample size. We also apply the sparse kernel learning method with full variable (denoted by Full-ssvs, tabulated in Table 3.1). For both methods, we set the $g_{\max} = 200$ and $c_n = 60$. We let the MCMC method run for $5,000$ iterations with the first $2,000$ as the burn-in period. For the prior set-up and the kernel form, we employ the same specifications as in the simulation experiments Case 3.

We repeat this Monte Carlo cross validation for 500 times and record the results in table 3.7. From the Table 3.7, we can see that our proposed doubly-sparse method provides a significant smaller prediction error compared to the sparse kernel method without variable selection. This indicates the importance of variable selection in the kernel based regression models. The sparse kernel methods like Zhang et al. (2011) provides a pathway for training the kernel nonlinear models with the data of large size. However, such methods would fail when there existing many irrelevant noisy variables. In addition, our proposed method also requires a smaller computational cost. This is because training the model with full features imputes large noises to the model and so it leads to the slow convergence.

Table 3.7: BRCA data analysis results

	MSPE	Training Time
Full-ssvs	1.031(0.015)	129175.8
Doubly-sparse	**0.157(0.0035)**	**103638.3**

The estimated standard errors are given in the parenthesis.

3.7 Concluding remarks

In this chapter, we have developed a Bayesian doubly-sparse RKHS model. Our proposed method performs variable selection and active vector selection simultaneously. The un-

certainties of variable selection and vector extractions have been addressed in a Bayesian fashion. In addition, the method is free from selecting a single best model or tuning penalty parameters. The Bayesian model averaging for both variable selection and sparse kernel matrix estimation (Hoeting et al., 1999; Raftery et al., 1997) are achieved automatically through the MCMC integration.

For future research directions, we can extend our Bayesian doubly-sparse framework to the Bayesian kernel probit models or Bayesian support vector machine models (Chakraborty, 2009; Mallick et al., 2005; Albert and Chib, 1993; Polson and Scott, 2011) by introducing a latent variable, which is the topic of the next chapter. In addition, we can also extend our method to the robust regression framework by assuming Laplace or student t-distributed errors (Jylänki et al., 2011).

Chapter 4

Bayesian doubly-sparse kernel support vector machine

In this chapter, we extend our proposed doubly-sparse framework to the nonlinear Bayesian support vector machine. In Section 4.1, we briefly introduce the model set-up. We present the hierarchical representation of the nonlinear Bayesian support vector machine via the data augmentation (Polson and Scott, 2011). The prior specifications are also given in Section 4.1. With the model set-up and the prior assumptions, we address the posterior inference in Section 4.2. We develop a collapsed Gibbs sampler that the selection of active vectors can be factored into the variable selection procedures. The procedure for making prediction are given in Section 4.3. In Section 4.4, we examine our proposed methods via the analysis of leukemia cancer data (Golub et al., 1999). We conclude this chapter in Section 4.5.

4.1 Model set-up and prior specification

4.1.1 The Bayesian nonlinear SVM

Suppose we observe n pairs of $\{x_i, y_i\}$ for $i = 1, \cdots, n$. For each pair, $y_i \in \{-1, +1\}$ is defined as the binary outcome and $x_i = (x_{i1}, \cdots, x_{ip})^\top \in \mathbb{R}^p$ is the p-dimensional input vector. With the observed data, our objective to learn a hyper-plane based on the unknown smooth function $f : \mathbb{R}^p \to \mathbb{R}$ to classify the response $\boldsymbol{y} = (y_1, \cdots, y_n)$ based on the input data $\boldsymbol{x} = [x_1, \cdots, x_n]^\top$. To train this classifier, one can minimize the following cost function,

$$d(f) = \sum_{i=1}^{n} \max(1 - y_i f(x_i), 0) + \tau \mathcal{J}(f), \tag{4.1}$$

where $\max(1 - y_i f(x_i), 0)$ is the hinge loss and τ is defined as the tuning parameter. In addition, $\mathcal{J}$ is defined as the regularization function, which controls the complexity of f. By optimizing the cost function (4.1), we can obtain a classification hyper-plane such that if $f(x_i) > 0$, one classifies the i^{th} observation as $+1$. If $f(x_i) < 0$, one classifies the i^{th} observation as -1.

In this paper, we employ the approach of the reproducing kernel Hilbert space (RKHS) for the inference of the unknown function f. With the RKHS approach, we assume $f = u + h \in (\{\mathbf{1}\} + \mathcal{H}_K)$, where $\mathcal{H}_K$ is a reproducing kernel Hilbert space and u is the intercept. Then, the minimization of cost (4.1) can be rewritten as

$$\min_{u, h \in \mathcal{H}_K} \left\{ \frac{1}{n} \sum_{i=1}^{n} \max(1 - y_i f(x_i), 0) + \frac{\tau}{2} ||h||^2_{\mathcal{H}_K} \right\}, \tag{4.2}$$

where $||h||^2_{\mathcal{H}_K}$ is defined as the RKHS norm and τ is the tuning parameter.

By the representer theorem (Kimeldorf and Wahba, 1970), the solution for the optimization problem (4.2) can be given as

$$f(x_i) \approx \beta_0 + \sum_{j=1}^{n} \beta_j K(x_i, x_j | \boldsymbol{\theta}), \tag{4.3}$$

where $K(x_i, x_j|\boldsymbol{\theta})$ is the $(i,j)^{th}$ component of the kernel matrix $\boldsymbol{K}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \{K(x_i, x_j|\boldsymbol{\theta})\}_{n\times n}$ governed by the hyper-paramters $\boldsymbol{\theta}$. The complex optimization problem (4.2) is turned into an estimation problem of the linear coefficients $\boldsymbol{\beta}$ and the kernel $\boldsymbol{K}_{\boldsymbol{\theta}}(\boldsymbol{x})$.

To estimate the $\boldsymbol{\beta}$ and $\boldsymbol{K}_{\boldsymbol{\theta}}(\boldsymbol{x})$, with equation (4.3) and the hinge loss in equation (4.1), we define the pseudo-likelihood for the nonlinear SVM as

$$\mathcal{L}(\boldsymbol{y}|\boldsymbol{\beta}) \propto \exp\left\{-2\sum_{i=1}^{n}\max(1-y_i k_i^\top \boldsymbol{\beta}, 0)\right\},$$

where k_i is the i^{th} row of the kernel matrix $\boldsymbol{K}_{\boldsymbol{\theta}}(\boldsymbol{x})$. As proposed by Polson and Scott (2011), we can introduce a latent variable $\boldsymbol{\lambda} = (\lambda_1, \cdots, \lambda_n)^\top$ such that

$$\begin{aligned}\mathcal{L}(y_i|\boldsymbol{\beta}) &\propto \exp\left\{-2\max(1-y_i k_i^\top \boldsymbol{\beta}, 0)\right\} \\ &\propto \int_0^\infty \frac{1}{\sqrt{2\pi\lambda_i}} \exp\left\{-\frac{1}{2}\frac{(1+\lambda_i - y_i k_i^\top \boldsymbol{\beta})^2}{\lambda_i}\right\} d\lambda_i.\end{aligned}$$

With this data augmentation technique, we successfully transform the hinge loss into the following Gaussian shape likelihood:

$$\mathcal{L}(\boldsymbol{y}|\boldsymbol{\beta}, \boldsymbol{K}, \boldsymbol{\lambda}) \propto \prod_{i=1}^{n} \frac{1}{\sqrt{\lambda_i}} \exp\left\{-\frac{1}{2}\frac{(1+\lambda_i - y_i k_i^\top \boldsymbol{\beta})^2}{\lambda_i}\right\}. \tag{4.4}$$

With the model likelihood defined above, we model the kernel matrix $\boldsymbol{K}_{\boldsymbol{\theta}}$ following Linkletter et al. (2006); Savitsky et al. (2011). We define the $(i,j)^{th}$ term of the kernel matrix as

$$K(x_i, x_j|\boldsymbol{\rho}) = \exp\left\{-\sum_{k=1}^{p} -\log(\rho_k)(x_{ik} - x_{jk})^2\right\},$$

where $\boldsymbol{\rho} = (\rho_1, \cdots, \rho_p)^\top$ is the kernel bandwidth with each $\rho_k \in (0, 1]$. Note if $\rho_k = 1$, the k^{th} variable is completely excluded from the constructing the kernel matrix. This formulation can be easily extended to other forms, such as Laplacian.

4.1.2 The prior specification

As commented by Fan and Lv (2008), variable selection plays a key important role for model prediction. With this objective, we assume the spike and slab prior proposed by Linkletter et al. (2006); Savitsky et al. (2011) to the kernel bandwidth $\boldsymbol{\rho}$ on the basis that variable selection can be accomplished by manipulating the kernel bandwidth. In addition, we append a variable selection binary index vector $\boldsymbol{\gamma} = (\gamma_1, \cdots, \gamma_p)^\top$, which we assume the prior proposed by Narisetty and He (2014). With the above set-up, the prior for $\boldsymbol{\rho}$ and $\boldsymbol{\gamma}$ is assumed as:

$$\begin{aligned}\pi(\boldsymbol{\rho}, \boldsymbol{\gamma}) &= \prod_{k=1}^{p} \pi(\rho_k|\gamma_k)\pi(\boldsymbol{\gamma}) \\ &\propto \prod_{k=1}^{p} \{\gamma_k \mathbb{I}[0 < \rho_k < 1] + (1-\gamma_k)\delta_1(\rho_k)\} \\ &\quad \times \prod_{k=1}^{p} \left\{ \frac{1}{p}\mathbb{I}(\gamma_k = 1) + (1 - \frac{1}{p})\mathbb{I}(\gamma_k = 0) \right\}.\end{aligned} \tag{4.5}$$

Within the equation (4.5), $\delta_1(\rho_k)$ is defined as a point mass distribution on 1. Note if $\gamma_k = 0$, a point mass on the bandwidth ρ_k would lead to the exclusion of the k^{th} feature for computing the kernel matrix. Contrarily, a uniform prior is assigned to ρ_k if $\gamma_k = 1$.

Other than the variable selection, learning a sparse kernel representation can lead to better or same prediction results with reduced computational costs (Zhang et al., 2016, 2008, 2011; Tipping, 2001). Note the learning of the sparse kernel can be accomplished by the sparse estimation of the vector weights $\boldsymbol{\beta}$. With this aim, we also employ a spike and slab prior proposed by George and McCulloch (1993). Similar to $\boldsymbol{\gamma}$, we supplement another binary index vector $\boldsymbol{g} = (g_j, \cdots, g_n)^\top$ for the purpose of the active vector selection. Then we assign the spike and slab prior for $\boldsymbol{\beta}$ as

$$\pi(\boldsymbol{\beta}|\boldsymbol{g}) = \prod_{j=1}^{n} \left\{ (1-g_j)\mathcal{N}\left(\beta_j, 0, \nu_0^2\right) + g_j \mathcal{N}\left(\beta_j, 0, \nu_1^2\right) \right\} \times \mathcal{N}\left(\beta_0, 0, \nu_1^2\right) \tag{4.6}$$

Within the equation (4.6), the prior variance ν_1^2 and ν_0^2 are pre-defined hyperparameters such that $\nu_1^2 \approx \infty$ and $\nu_0^2 \approx 0$. With this specification, the slab priors are assigned to the coefficients of active vectors. Contrarily, the j^{th} vector are approximately discarded due to the allocation of the spike priors.

As proposed by Narisetty and He (2014), we assume

$$\boldsymbol{g} \propto \prod_{j=1}^{n} \left(\frac{c_n}{n}\right)^{g_j} \left(1 - \frac{c_n}{n}\right)^{1-g_j}.$$

for the index vector $\boldsymbol{g}$. Also as suggested by Narisetty and He (2014), we choose the tuning parameter c_n by $\Phi((g_{\max} - c_n)/(\sqrt{c_n(1 - c_n/n)})) \approx 1$. The $g_{\max}$ is set as the upper bound of the size of active vectors. Since we aim to learn a sparse representation of the kernel, we set $g_{\max}$ to be half of the sample size.

One last parameter is the latent variable $\boldsymbol{\lambda} = (\lambda_1, \cdots, \lambda_n)$, for which we assume a uniform prior such that each $\lambda_i \propto 1$.

4.2 Posterior inference

4.2.1 Posterior distribution and the collapsed Gibbs sampler

With the prior specification in Section 4.1.2 and model likelihood defined in equation (4.4), we acquire the posterior as

$$\begin{aligned}
\pi(\boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{g}|\boldsymbol{y}) &\propto \mathcal{L}(\boldsymbol{y}|\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{K})\pi(\boldsymbol{\rho}, \boldsymbol{\gamma})\pi(\boldsymbol{\beta}|\boldsymbol{g})\pi(\boldsymbol{g})\pi(\lambda) \\
&\propto \prod_{i=1}^{n} \frac{1}{\lambda_i} \exp\left\{-\frac{1}{2}\frac{(1+\lambda_i - y_i k_i^\top \boldsymbol{\beta})^2}{\lambda_i}\right\} \\
&\quad \times \frac{1}{\sqrt{|\boldsymbol{V}|}} \exp\left\{-\frac{1}{2}\boldsymbol{\beta}^\top \boldsymbol{V}^{-1}\boldsymbol{\beta}\right\} \times \prod_{j=1}^{n} \left(\frac{c_n}{n}\right)^{g_j} \left(1 - \frac{c_n}{n}\right)^{1-g_j} \\
&\quad \times \prod_{k=1}^{p} \{\gamma_k \mathbb{I}[0 < \rho_k < 1] + (1-\gamma_k)\delta_1(\rho_k)\} \\
&\quad \times \prod_{k=1}^{p} \left\{\frac{1}{p}\mathbb{I}(\gamma_k = 1) + (1 - \frac{1}{p})\mathbb{I}(\gamma_k = 0)\right\}
\end{aligned} \tag{4.7}$$

The V is defined as a diagonal matrix with ν_1^2 and ν_0^2 as its diagonal elements.

The distribution in the equation (4.7) is so complex that direct inference is almost impossible. One common approach for posterior sampling is the Gibbs sampler. However, the sampling of the full conditionals of $(\boldsymbol{\rho}, \boldsymbol{\gamma})$ is highly sensitive to $\boldsymbol{\beta}$, which leads to poor mixing and slow convergence of the Markov chain. To solve this issue, we propose a collapsed Gibbs sampler (Liu, 1994). We iteratively generate samples from the joint posterior with the following conditional distributions until convergence:

Step 1. Jointly generate $(\boldsymbol{\gamma}, \boldsymbol{\rho})$ from $\pi(\boldsymbol{\gamma}, \boldsymbol{\rho}|\boldsymbol{g}, \boldsymbol{\lambda}, \boldsymbol{y})$.

Step 2. Generate each g_j for $j = 1, \cdots, n$ from $\pi(g_j|\boldsymbol{g}_{-j}, \boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{y})$.

Step 3. Generate $\boldsymbol{\beta}$ from $\pi(\boldsymbol{\beta}|\boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\lambda}, \boldsymbol{g}, \boldsymbol{y})$.

Step 4. Generate $\boldsymbol{\lambda}$ from $\pi(\boldsymbol{\lambda}|\boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{g}, \boldsymbol{\beta}, \boldsymbol{y})$.

4.2.2 Conditional distributions and implementation details

With the collapsed Gibbs sampler proposed above, we give the implementation details in this section. The sampling steps from 2 to 4 are straightforward because their full conditionals are all well-known distributions. The marginal conditional of $(\boldsymbol{\gamma}, \boldsymbol{\rho})^\top$, however, is the complex one, which we have to generate samples via the Metropolis-Hastings algorithm. In particular, we revise the sampling scheme proposed by Savitsky et al. (2011).

We update the samples of $(\boldsymbol{\gamma}, \boldsymbol{\rho})^\top$ as follows. For $k = 1, \cdots, p$:

1 *Between-models move*: Jointly propose a new model such that if $\gamma_k = 1$, propose $\gamma_k' = 0$ and set $\rho_k' = 1$. If $\gamma_k = 0$, then propose $\gamma_k' = 1$ and randomly draw $\rho_k' \sim \mathcal{U}(0, 1)$. Accept the proposed value of $(\gamma_k', \rho_k')^\top$ with probability

$$\alpha = \min\left\{1, \frac{\pi(\gamma_k', \rho_k' | \boldsymbol{\gamma}_{-k}, \boldsymbol{\rho}_{-k}, \boldsymbol{g}, \boldsymbol{\lambda}, \boldsymbol{y})}{\pi(\gamma_k, \rho_k | \boldsymbol{\gamma}_{-k}, \boldsymbol{\rho}_{-k}, \boldsymbol{g}, \boldsymbol{\lambda}, \boldsymbol{y})}\right\}$$

The proposal ratio reduces to 1 given that we employ a uniform proposal for ρ_k and a symmetric Dirac measure proposal for γ_k.

2 *Within-models move*: This move is performed only when sampling $\gamma_k = 1$ from the previous step. The aim of this step is to further refine the bandwidth parameter ρ_k. We first propose $\gamma_k' = 1$. Then we use an adaptive random walk approach (Andrieu and Thoms, 2008; Roberts and Rosenthal, 2009) for the proposal of ρ_k. In particular, we draw $\rho_k' \sim \mathcal{U}(\rho_k - \frac{s_{\rho_k}}{2}, \rho_k + \frac{s_{\rho_k}}{2})$ where s_{ρ_k} is the sample standard deviation for ρ_k computed with the generated MCMC samples. We accept the joint proposal for $(\gamma_k', \rho_k')^\top$ with the probability of

$$\alpha = \min\left\{1, \frac{\pi(\gamma_k', \rho_k' | \boldsymbol{\gamma}_{-k}, \boldsymbol{\rho}_{-k}, \boldsymbol{g}, \boldsymbol{\lambda}, \boldsymbol{y})}{\pi(\gamma_k, \rho_k | \boldsymbol{\gamma}_{-k}, \boldsymbol{\rho}_{-k}, \boldsymbol{g}, \boldsymbol{\lambda}, \boldsymbol{y})}\right\}$$

Again, the proposal ratio reduces to 1 just like the between-model move.

For the computation of $\pi(\boldsymbol{\gamma}, \boldsymbol{\rho} | \boldsymbol{g}, \boldsymbol{\lambda}, \boldsymbol{y}) \propto \pi(\boldsymbol{y} | \boldsymbol{\rho}, \boldsymbol{g}, \boldsymbol{\lambda}) \pi(\boldsymbol{\gamma}, \boldsymbol{\rho})$, we need to integrate out $\boldsymbol{\beta}$

from the full likelihood. The marginal likelihood is then given as

$$\pi(\boldsymbol{y}|\boldsymbol{\rho},\boldsymbol{g},\boldsymbol{\lambda}) \propto |\boldsymbol{A}|^{-\frac{1}{2}} \exp\left\{\frac{1}{2}\tilde{\boldsymbol{\lambda}}^{\top}\boldsymbol{\lambda}^{*-1}\tilde{\boldsymbol{K}}\boldsymbol{A}^{-1}\tilde{\boldsymbol{K}}^{\top}\boldsymbol{\lambda}^{*-1}\tilde{\boldsymbol{\lambda}}\right\} \tag{4.8}$$

where $\tilde{\boldsymbol{\lambda}} = \boldsymbol{\lambda} + 1$ and $\tilde{\boldsymbol{K}} = \boldsymbol{y}^{\top}\boldsymbol{K}$. The derivation of the (4.8) is given in Appendix A.2.

To speed up the computation of the equation (4.8), as proposed by Narisetty and He (2014), we divide the $\boldsymbol{\beta}$ into $[\boldsymbol{\beta}_{\boldsymbol{g}}, \boldsymbol{\beta}_I]$ such that $\boldsymbol{\beta}_{\boldsymbol{g}}$ and $\boldsymbol{\beta}_I$ contains the components of $\boldsymbol{\beta}$ corresponding to $g_j = 1$ and $g_j = 0$. Similarly, we divide $\boldsymbol{K}$ into $[\boldsymbol{K}_{\boldsymbol{g}}, \boldsymbol{K}_I]$ that they consist columns of $\boldsymbol{K}$ corresponding to $g_j = 1$ and $g_j = 0$. With the spike priors, the components within the $\boldsymbol{\beta}_I$ are shrink to approximately zero. Then by discarding the $\boldsymbol{K}_I$, the approximate marginal likelihood is computed as

$$\pi(\boldsymbol{y}|\boldsymbol{\rho},\boldsymbol{g},\boldsymbol{\lambda}) = c|\boldsymbol{A}_{\boldsymbol{g}}|^{-\frac{1}{2}} \exp\left\{\frac{1}{2}\tilde{\boldsymbol{\lambda}}^{\top}\boldsymbol{\lambda}^{*-1}\tilde{\boldsymbol{K}}_{\boldsymbol{g}}\boldsymbol{A}_{\boldsymbol{g}}^{-1}\tilde{\boldsymbol{K}}_{\boldsymbol{g}}^{\top}\boldsymbol{\lambda}^{*-1}\tilde{\boldsymbol{\lambda}}\right\}$$

where $\tilde{\boldsymbol{\lambda}} = \boldsymbol{\lambda}+1$ and $\tilde{\boldsymbol{K}}_{\boldsymbol{g}} = \boldsymbol{y}^{\top}\boldsymbol{K}_{\boldsymbol{g}}$. The c is defined as the normalizing constant. In addition, we define $\boldsymbol{A}_{\boldsymbol{g}}$ as $\boldsymbol{A}_{\boldsymbol{g}} = (\tilde{\boldsymbol{K}}_{\boldsymbol{g}}\boldsymbol{\lambda}^{*-1}\tilde{\boldsymbol{K}}_{\boldsymbol{g}} + \boldsymbol{V}_{\boldsymbol{g}}^{-1})$, where $\boldsymbol{V}_{\boldsymbol{g}}$ is the $(n_{\boldsymbol{g}}+1)$ by $(n_{\boldsymbol{g}}+1)$ diagonal matrix with ν_1^2 as its diagonal elements. The $n_{\boldsymbol{g}}$ is defined as the count of active vectors, i.e, $n_{\boldsymbol{g}} = \sum_{j=1}^{n} g_j$.

With the generated sample of $(\boldsymbol{\gamma},\boldsymbol{\rho})^{\top}$, the samples of the rest of model parameters are updated as follows. For index $\boldsymbol{g}$, we have

$$g_j|\boldsymbol{g}_{-j},\boldsymbol{\gamma},\boldsymbol{\rho},\boldsymbol{\beta},\boldsymbol{\lambda},\boldsymbol{y} \sim \text{Bernoulli}\left(\frac{\frac{c_n}{n}\phi(\beta_j;0,\nu_1^2)}{\frac{c_n}{n}\phi(\beta_j;0,\nu_1^2)+(1-\frac{c_n}{n})\phi(\beta_j;0,\nu_0^2)}\right).$$

for $j = 1,\cdots,n$. For vector weights $\boldsymbol{\beta}$, we have

$$\boldsymbol{\beta}|\boldsymbol{\gamma},\boldsymbol{\rho},\boldsymbol{g},\boldsymbol{\lambda},\boldsymbol{y} \sim \mathcal{N}\left((\boldsymbol{Z}^{\top}\boldsymbol{Z}+\boldsymbol{V})^{-1}\boldsymbol{Z}^{\top}\boldsymbol{w}, (\boldsymbol{Z}^{\top}\boldsymbol{Z}+\boldsymbol{V})^{-1}\right)$$

where $\boldsymbol{Z} = (z_1, z_2, \cdots, z_n)^{\top}$ with $z_i = \frac{y_i k_i}{\sqrt{\lambda_i}}$ and $\boldsymbol{w}$ is defined as $\boldsymbol{w} = (w_1, w_2, \cdots, w_n)^{\top}$ with

$w_i = \frac{1+\lambda_1}{\sqrt{\lambda_i}}$. The last parameter is the latent variable $\boldsymbol{\lambda}$, which we generate samples from

$$\frac{1}{\lambda_i}|\boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\beta}, \boldsymbol{g}, \boldsymbol{y} \overset{ind.}{\sim} \text{Inverse-Gaussian}\left(|1 - y_i k_i^\top \boldsymbol{\beta}|, 1\right).$$

4.3 Prediction

With the collapsed Gibbs sampler algorithm developed in Section 4.1, we give the procedure for making prediction in this section. Let $\boldsymbol{x}_{\text{new}}$ be the input data for the unseen points and $\mathcal{D} = \{(x_i, y_i), i = 1, \cdots, n\}$ denotes the observed data. In addition, we denote $\boldsymbol{\Theta} = (\boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\beta}, \boldsymbol{g}, \boldsymbol{\lambda})$ as the set of model parameters. The procedure for prediction is given as:

step 1. Generate T samples for model parameters from its posterior distribution $\pi(\boldsymbol{\Theta}|\mathcal{D})$ via the collapsed Gibbs sampler given in Section 4.3.

step 2. Set

$$\hat{\boldsymbol{y}}_{\text{new}} = \begin{cases} +1 & \text{if } \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{K}_{\text{new}}^{(t)} \boldsymbol{\beta}_{\boldsymbol{g}}^{(t)} > 0 \\ \\ -1 & \text{if } \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{K}_{\text{new}}^{(t)} \boldsymbol{\beta}_{\boldsymbol{g}}^{(t)} < 0 \end{cases}$$

Note prediction kernel $\boldsymbol{K}_{\text{new}}$ is a n_{new} by $n_{\boldsymbol{g}} + 1$ matrix computed based on the input data of unseen points and the active vectors indexed by $\boldsymbol{g}$. This formulation can greatly reduce the computational cost of predictions, especially when the size of prediction is large.

4.4 Application

In this section, we examine our proposed method via the analysis of the leukemia cancer data. The leukemia data (Golub et al., 1999) is a benchmark high dimensional binary classification dataset. The goal of the analysis is to to classify two types of leukemia cancer using the microarray gene expressions. The dataset is available with the R SIS package (Saldana and Feng, 2018; Fan et al., 2015), which contains $7,129$ gene expressions and only 72 samples

To analyze the data, we first conduct the variable screening by training uni-variate generalized additive models (Hastie and Tibshirani, 2017) via the R package mgcv (Wood and Wood, 2015) and compute its BIC (Schwarz et al., 1978). We retain the first 200 gene expressions based on the BIC. Then we conduct the Monte carlo cross validation such that for each experiment, we randomly split half as the training set and the other half as the testing set. To avoid imbalanced classification, we retain the class distribution in each split. We compare the performance of our proposed doubly-sparse method to the Bayesian SVM models tabulated in Table 4.1. Both methods uses the Gaussian kernel. The hyperparameters are set as $\nu_0^2 = 10^{-3}$, $\nu_1^2 = 10^3$ and $c_n = 7$.

Table 4.1: The Bayesian RKHS SVM models to be compared

Abbreviation	Method	Remarks
BSVM-full-ridge	Bayesian RKHS SVM with full variables and ridge priors on $\boldsymbol{\beta}$	Variable selection is not addressed, non-sparse kernel model.
BSVM-full-ssvs	Bayesian RKHS SVM with full variables and spike and slab prior on $\boldsymbol{\beta}$	Variable selection is not addressed, sparse kernel estimated, SVM version of Zhang et al. (2011).
BSVM-vs-L_2	Bayesian RKHS SVM with variable selection and ridge priors on $\boldsymbol{\beta}$	Variable selection is addressed, non-sparse kernel model, SVM version of Chakraborty (2009), details given in Appendix B.2.

To evaluate each method, we compute the misclassification rate defined in equation (4.9).

$$\text{Classfication error} = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \mathbb{I}(y_i \neq \hat{y}_i) \tag{4.9}$$

In addition to the prediction accuracy, the CPU time in seconds are recorded. For both methods, we run the MCMC for $10,000$ iterations with the first $5,000$ iterations set as

burn-in. We repeat the experiment 1,000 times and tabulate the results in Table 4.2.

Table 4.2: Leukemia data analysis results

Method	Classification error	Training time
BSVM-full-L_2	0.333(0)	2229.85
BSVM-full-ssvs	0.321 (0.0021)	1572.45
BSVM-vs-L_2	0.062(0.00139)	1607.801
BSVM-ds	**0.058(0.00137)**	**979.98**

The estimated standard errors are given in the parenthesis.

From the experiment results in Table 4.2, predictions made without variable selection show little improvement upon classification via random guess. Other than the necessity of the variable selection, we also observe that training a Bayesian SVM with all vectors (BSVM-VS-L_2) is less optimal than the proposed doubly-sparse method, which is corroborated by the Zhang et al. (2016); Chen et al. (2018). The proposed doubly-sparse model not only improves the prediction accuracy but also reduces the computational cost.

4.5 Discussion and future work

In this chapter, we have developed a Bayesian doubly-sparse nonlinear support vector machine. For posterior inference, a collapsed Gibbs sampler is developed such that the sparse kernel can be incorporated into the variable selection sampling steps. Through the analysis of the Leukemia data (Golub et al., 1999), we showed the benefits of our proposed method on both the aspects of computational cost and prediction accuracy. Notably our proposed method is not restricted to the Bayesian SVM framework; it can be easily extended to other type of classification models, such as Bayesian logistic regression (Polson et al., 2013), probit regression (Albert and Chib, 1993). In addition, the framework could also be extended to model other types of data, such as count data, survival data, etc.

Even though our proposed method showed advantages over existing Bayesian SVM mod-

els, room for improvements still exists. One potential future research direction would be developing faster procedures for the latent variable sampling. Extensions to the large sample size would be another interesting future research direction.

Chapter 5

Conclusion

In this dissertation, we have developed several strategies for performing fast variable selection for both Gaussian process and reproducing kernel Hilbert space models. Through various simulation studies and real data analyses, we have shown the advantages of our proposed method.

In Chapter 2, we developed a novel Bayesian model hybrid search algorithm for Gaussian process regression. The proposed method is able to quickly scan through the large model space and collect those models with high posterior probabilities. To compute the marginal likelihood, we used the Laplace approximation. Prediction was then conducted via Bayesian model averaging. To address the model selection problem under the massive data case, we proposed a hybrid model search algorithm based on the quantile subset of data. Predictions are made through the nearest neighbor Gaussian process within the Bayesian model averaging framework.

In Chapter 3, a novel Bayesian approach for the reproducing kernel Hilbert space regression models is developed to address the doubly-sparse estimation problem. We assumed the double spike and slab priors on both the kernel bandwidths and the vector weights. To address the posterior inference, a collapsed Gibbs sampler is developed such that the sparse kernel estimation can be factored into the variable selection procedures. To address the large sample size cases, we proposed a data-driven subset of data approach and modified the

collapsed Gibbs sampler by revising the 'skinny Gibbs' method.

In Chapter 4, we extended our Bayesian doubly-sparse framework to the nonlinear kernel Bayesian support vector machine. Through the analysis of the Leukemia cancer data, the merits of our proposed method are shown.

For future work, extensions of the model hybrid search algorithm and Bayesian doubly-sparse framework to other types of data, such as, count data, time to event data, etc, would be beneficial. In addition, the proposed method can be further speed up through the implementation with high-performance computation R packages (e.g. Rcpp) or parallel computations.

Bibliography

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

Alex J Smola and Bernhard Schölkopf. *Learning with kernels*, volume 4. Citeseer, 1998.

Jianqing Fan and Jinchi Lv. Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5): 849–911, 2008.

Crystal Linkletter, Derek Bingham, Nicholas Hengartner, David Higdon, and Kenny Q Ye. Variable selection for gaussian process models in computer experiments. *Technometrics*, 48(4):478–490, 2006.

Terrance Savitsky, Marina Vannucci, and Naijun Sha. Variable selection for nonparametric gaussian process priors: Models and computational strategies. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 26(1):130, 2011.

G Yi, JQ Shi, and T Choi. Penalized gaussian process regression and classification for high-dimensional nonlinear data. *Biometrics*, 67(4):1285–1294, 2011.

Feng Yan and Yuan Alan Qi. Sparse gaussian process regression via l1 penalization. In *ICML*, 2010.

Haitao Liu, Yew-Soon Ong, Xiaobo Shen, and Jianfei Cai. When gaussian process meets big data: A review of scalable gps. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

Christopher KI Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *Advances in neural information processing systems*, pages 682–688, 2001.

Matthias Seeger, Christopher Williams, and Neil Lawrence. Fast forward selection to speed up sparse gaussian process regression. Technical report, 2003.

Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pages 1257–1264, 2006a.

Edward Snelson and Zoubin Ghahramani. Local and global sparse gaussian process approximations. In *Artificial Intelligence and Statistics*, pages 524–531, 2007.

Abhirup Datta, Sudipto Banerjee, Andrew O Finley, and Alan E Gelfand. Hierarchical nearest-neighbor gaussian process models for large geostatistical datasets. *Journal of the American Statistical Association*, 111(514):800–812, 2016a.

Abhirup Datta, Sudipto Banerjee, Andrew O Finley, and Alan E Gelfand. On nearest-neighbor gaussian process models for massive spatial data. *Wiley Interdisciplinary Reviews: Computational Statistics*, 8(5):162–171, 2016b.

Robert B Gramacy et al. lagp: large-scale spatial modeling via local approximate gaussian processes in r. *Journal of Statistical Software*, 72(1):1–46, 2016.

Robert B Gramacy and Daniel W Apley. Local gaussian process approximation for large computer experiments. *Journal of Computational and Graphical Statistics*, 24(2):561–578, 2015.

Hyoung-Moon Kim, Bani K Mallick, and CC Holmes. Analyzing nonstationary spatial data using piecewise gaussian processes. *Journal of the American Statistical Association*, 100 (470):653–668, 2005.

Robert B Gramacy and Benjamin Haaland. Speeding up neighborhood search in local gaussian process prediction. *Technometrics*, 58(3):294–303, 2016.

Junbin Gao, Paul W Kwan, and Daming Shi. Sparse kernel learning with lasso and bayesian inference algorithm. *Neural networks*, 23(2):257–264, 2010.

Genevera I Allen. Automatic feature selection via weighted kernels and regularization. *Journal of Computational and Graphical Statistics*, 22(2):284–299, 2013.

Feng Liang, Kai Mao, Ming Liao, Sayan Mukherjee, and Mike West. Nonparametric bayesian kernel models. *Department of Statistical Science, Duke University, Discussion Paper*, pages 07–10, 2007.

Sounak Chakraborty. Bayesian binary kernel probit model for microarray based cancer classification and gene selection. *Computational Statistics & Data Analysis*, 53(12):4198–4209, 2009.

Lorin Crawford, Kris C Wood, Xiang Zhou, and Sayan Mukherjee. Bayesian approximate kernel regression with variable selection. *Journal of the American Statistical Association*, 113(524):1710–1721, 2018.

Maria Maddalena Barbieri and James O Berger. Optimal predictive model selection. *The annals of statistics*, 32(3):870–897, 2004.

Jennifer A Hoeting, David Madigan, Adrian E Raftery, and Chris T Volinsky. Bayesian model averaging: a tutorial. *Statistical science*, pages 382–401, 1999.

Chong Zhang, Yufeng Liu, and Yichao Wu. On quantile regression in reproducing kernel hilbert spaces with the data sparsity constraint. *The Journal of Machine Learning Research*, 17(1):1374–1418, 2016.

Michael E Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of machine learning research*, 1(Jun):211–244, 2001.

Zhihua Zhang, Michael Jordan, and Dit-Yan Yeung. Posterior consistency of the silverman g-prior in bayesian model choice. *Advances in Neural Information Processing Systems*, 21: 1969–1976, 2008.

Zhihua Zhang, Guang Dai, and Michael I Jordan. Bayesian generalized kernel mixed models. *The Journal of Machine Learning Research*, 12:111–139, 2011.

Jingxiang Chen, Chong Zhang, Michael R Kosorok, and Yufeng Liu. Double sparsity kernel learning with automatic variable selection and data extraction. *Statistics and its interface*, 11(3):401, 2018.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3): 273–297, 1995.

Bani K Mallick, Debashis Ghosh, and Malay Ghosh. Bayesian classification of tumours by using gene expression data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):219–234, 2005.

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

Ricardo Henao, Xin Yuan, and Lawrence Carin. Bayesian nonlinear support vector machines and discriminative factor modeling. *Advances in neural information processing systems*, 27, 2014.

Paul S Bradley and Olvi L Mangasarian. Feature selection via concave minimization and support vector machines. In *ICML*, volume 98, pages 82–90. Citeseer, 1998.

Ji Zhu, Saharon Rosset, Robert Tibshirani, and Trevor Hastie. 1-norm support vector machines. *Advances in neural information processing systems*, 16, 2003.

Li Wang, Ji Zhu, and Hui Zou. The doubly regularized support vector machine. *Statistica Sinica*, pages 589–615, 2006.

Hao Helen Zhang, Jeongyoun Ahn, Xiaodong Lin, and Cheolwoo Park. Gene selection using support vector machines with non-convex penalty. *bioinformatics*, 22(1):88–95, 2006.

Hui Zou and Ming Yuan. The f∞-norm support vector machine. *Statistica Sinica*, pages 379–398, 2008.

Natalia Becker, Grischa Toedt, Peter Lichter, and Axel Benner. Elastic scad as a novel penalization method for svm classification tasks in high-dimensional data. *BMC bioinformatics*, 12(1):1–13, 2011.

Hao Helen Zhang. Variable selection for support vector machines via smoothing spline anova. *Statistica Sinica*, pages 659–674, 2006.

Olvi L Mangasarian and Gang Kou. Feature selection for nonlinear kernel support vector machines. In *Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007)*, pages 231–236. IEEE, 2007.

Elena Marchiori and Michele Sebag. Bayesian learning with local support vector machines for cancer classification with gene expression data. In *Workshops on Applications of Evolutionary Computation*, pages 74–83. Springer, 2005.

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

Wenli Sun, Changgee Chang, Yize Zhao, and Qi Long. 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)*, pages 1484–1493. IEEE, 2018.

Wenli Sun, Changgee Chang, and Qi Long. Bayesian non-linear support vector machine for high-dimensional data with incorporation of graph information on features. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 4874–4882. IEEE, 2019.

Joaquin Quinonero-Candela, Carl Edward Rasmussen, and Christopher KI Williams. Approximation methods for gaussian process regression. In *Large-scale kernel machines*, pages 203–223. MIT Press, 2007.

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

Shiqiang Jin and Gyuhyeong Goh. Bayesian selection of best subsets via hybrid search. *Computational Statistics*, 36(3):1991–2007, 2021.

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

Julian Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society: Series B (Methodological)*, 48(3):259–279, 1986.

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

James G Scott and James O Berger. Bayes and empirical-bayes multiplicity adjustment in the variable-selection problem. *The Annals of Statistics*, pages 2587–2619, 2010.

Jiahua Chen and Zehua Chen. Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3):759–771, 2008.

Luke Tierney and Joseph B Kadane. Accurate approximations for posterior moments and marginal densities. *Journal of the american statistical association*, 81(393):82–86, 1986.

Larry Wasserman et al. Bayesian model selection and model averaging. *Journal of mathematical psychology*, 44(1):92–107, 2000.

Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6(Dec): 1939–1959, 2005.

Julian J Faraway. *Linear models with R*. Chapman and Hall/CRC, 2004.

Claus Borggaard and Hans Henrik Thodberg. Optimal minimal neural interpretation of spectra. *Analytical chemistry*, 64(5):545–551, 1992.

Jerome Friedman, Trevor Hastie, and Rob Tibshirani. glmnet: Lasso and elastic-net regularized generalized linear models. *R package version*, 1(4):1–24, 2009.

Kelwin Fernandes, Pedro Vinagre, and P. Cortez. A proactive intelligent decision support system for predicting the popularity of online news. In *EPIA*, 2015.

George S Kimeldorf and Grace Wahba. A correspondence between bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, 41 (2):495–502, 1970.

Naveen Naidu Narisetty and Xuming He. Bayesian variable selection with shrinking and diffusing priors. *The Annals of Statistics*, 42(2):789–817, 2014.

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

Alan E Gelfand and Adrian FM Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410):398–409, 1990.

Peter J Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

Raphael Gottardo and Adrian E Raftery. Markov chain monte carlo with mixtures of mutually singular distributions. *Journal of Computational and Graphical Statistics*, 17(4): 949–975, 2008.

Christophe Andrieu and Johannes Thoms. A tutorial on adaptive mcmc. *Statistics and computing*, 18(4):343–373, 2008.

Gareth O Roberts and Jeffrey S Rosenthal. Examples of adaptive mcmc. *Journal of computational and graphical statistics*, 18(2):349–367, 2009.

Naveen N Narisetty, Juan Shen, and Xuming He. Skinny gibbs: A consistent and scalable gibbs sampler for model selection. *Journal of the American Statistical Association*, 2018.

James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013.

Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18. MIT Press, 2006b. URL https://proceedings.neurips.cc/paper/2005/file/4491777b1aa8b5b32c2e8666dbe1a495-Paper.pdf.

Todd E Scheetz, Kwang-Youn A Kim, Ruth E Swiderski, Alisdair R Philp, Terry A Braun, Kevin L Knudtson, Anne M Dorrance, Gerald F DiBona, Jian Huang, Thomas L Casavant, et al. Regulation of gene expression in the mammalian eye and its relevance to eye disease. *Proceedings of the National Academy of Sciences*, 103(39):14429–14434, 2006.

Jianqing Fan, Yang Feng, and Rui Song. Nonparametric independence screening in sparse ultra-high-dimensional additive models. *Journal of the American Statistical Association*, 106(494):544–557, 2011.

Jian Huang, Joel L Horowitz, and Fengrong Wei. Variable selection in nonparametric additive models. *Annals of statistics*, 38(4):2282, 2010.

Karoline B Kuchenbaecker, John L Hopper, Daniel R Barnes, Kelly-Anne Phillips, Thea M Mooij, Marie-José Roos-Blom, Sarah Jervis, Flora E Van Leeuwen, Roger L Milne, Nadine Andrieu, et al. Risks of breast, ovarian, and contralateral breast cancer for brca1 and brca2 mutation carriers. *Jama*, 317(23):2402–2416, 2017.

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

James H Albert and Siddhartha Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American statistical Association*, 88(422):669–679, 1993.

Pasi Jylänki, Jarno Vanhatalo, and Aki Vehtari. Robust gaussian process regression with a student-t likelihood. *Journal of Machine Learning Research*, 12(11), 2011.

Todd R Golub, Donna K Slonim, Pablo Tamayo, Christine Huard, Michelle Gaasenbeek, Jill P Mesirov, Hilary Coller, Mignon L Loh, James R Downing, Mark A Caligiuri, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, 286(5439):531–537, 1999.

Diego Franco Saldana and Yang Feng. Sis: An r package for sure independence screening in ultrahigh-dimensional statistical models. *Journal of Statistical Software*, 83:1–25, 2018.

Jianqing Fan, Yang Feng, Diego Franco Saldana, Richard Samworth, Yichao Wu, and Maintainer Diego Franco Saldana. Package 'sis'. *CRAN, https://cran. r-project. org/web/packages/SIS/index. html*, 2015.

Trevor J Hastie and Robert J Tibshirani. *Generalized additive models*. Routledge, 2017.

Simon Wood and Maintainer Simon Wood. Package 'mgcv'. *R package version*, 1(29):729, 2015.

Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2): 461–464, 1978.

Nicholas G Polson, James G Scott, and Jesse Windle. Bayesian inference for logistic models using pólya–gamma latent variables. *Journal of the American statistical Association*, 108 (504):1339–1349, 2013.

Appendix A

Calculation of marginal likelihood

A.1 Derivation of equation (3.7)

From equation (3.3), the model is defined as $\boldsymbol{y}|\boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\beta}, \sigma^2 \sim \mathcal{N}(\boldsymbol{K}\boldsymbol{\beta}, \sigma^2\mathbb{I})$. With the spike and slab prior $\pi(\boldsymbol{\beta}|\sigma^2, \boldsymbol{g}, \nu_1)$ defined in equation (3.4) and the inverse-Gamma prior for σ^2 in (3.5), we have

$$
\begin{aligned}
\pi(\boldsymbol{y}|\boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{g}, \nu_1) &\propto \int\int \pi(\boldsymbol{y}|\boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\beta}, \sigma^2)\pi(\boldsymbol{\beta}|\sigma^2, \boldsymbol{g}, \nu_1)\pi(\sigma^2)d\boldsymbol{\beta}d\sigma^2 \\
&\propto \int\int \frac{1}{\sqrt{|\sigma^2\mathbb{I}_n|}} \exp\left\{-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{K}\boldsymbol{\beta})^\top(\sigma^2\mathbb{I})^{-1}(\boldsymbol{y}-\boldsymbol{K}\boldsymbol{\beta})\right\} \\
&\quad \times \frac{1}{\sqrt{|\sigma^2\boldsymbol{V}|}} \exp\left\{-\frac{1}{2}\boldsymbol{\beta}^\top(\sigma^2\boldsymbol{V})^{-1}\boldsymbol{\beta}\right\} \\
&\quad \times \frac{b_1^{a_\sigma}}{\Gamma(a_\sigma)}(\sigma^2)^{-a_\sigma-1}\exp\left\{-\frac{b_\sigma}{\sigma^2}\right\} d\boldsymbol{\beta}d\sigma^2 \\
&\propto \int\int (\sigma^2)^{-\frac{2n+2a_\sigma+3}{2}} \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{y}^\top\boldsymbol{y} - 2\boldsymbol{\beta}^\top\boldsymbol{K}^\top\boldsymbol{y}\right. \\
&\quad \left. +\boldsymbol{\beta}^\top\boldsymbol{K}^\top\boldsymbol{K}\boldsymbol{\beta} + \boldsymbol{\beta}^\top\boldsymbol{V}^{-1}\boldsymbol{\beta} + 2b_\sigma)\right\} d\boldsymbol{\beta}d\sigma^2
\end{aligned}
\tag{A.1}
$$

The $\boldsymbol{V}$ is a diagonal matrix with ν_1 and ν_0 as its diagonal elements. Inside the exponential term, we have

$$\begin{aligned}
&\boldsymbol{y}^\top \boldsymbol{y} - 2\boldsymbol{\beta}^\top \boldsymbol{K}\boldsymbol{y} + \boldsymbol{\beta}^T \boldsymbol{K}^T \boldsymbol{K}\boldsymbol{\beta} + \boldsymbol{\beta}^\top \boldsymbol{V}^{-1}\boldsymbol{\beta} + 2b_\sigma \\
&= \boldsymbol{\beta}^\top (\boldsymbol{K}^\top \boldsymbol{K} + V^{-1})\boldsymbol{\beta} - 2\boldsymbol{\beta}^\top (\boldsymbol{K}^\top \boldsymbol{K} + V^{-1})^{-1}(\boldsymbol{K}^\top \boldsymbol{K} + V^{-1})\boldsymbol{K}^\top \boldsymbol{y} \\
&\quad + \boldsymbol{y}^\top \boldsymbol{y} + 2b_\sigma
\end{aligned} \tag{A.2}$$

Let $\boldsymbol{A} = (\boldsymbol{K}^\top \boldsymbol{K} + \boldsymbol{V}^{-1})$ and $\tilde{\boldsymbol{\beta}} = \boldsymbol{A}^{-1}\boldsymbol{K}^\top \boldsymbol{y}$, then the equation (A.2) is rewritten as

$$\begin{aligned}
&\boldsymbol{y}^\top \boldsymbol{y} - 2\boldsymbol{\beta}^\top \boldsymbol{K} y + \boldsymbol{\beta}^\top \boldsymbol{K}^\top \boldsymbol{K}\boldsymbol{\beta} + \boldsymbol{\beta}^\top \boldsymbol{V}^{-1}\boldsymbol{\beta} + 2b_\sigma \\
&= (\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}})^\top \boldsymbol{A}(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}) + \boldsymbol{y}^\top \boldsymbol{y} - \tilde{\boldsymbol{\beta}}^T \boldsymbol{A}\tilde{\boldsymbol{\beta}} + 2b_\sigma \\
&= (\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}})^\top \boldsymbol{A}(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}) + \boldsymbol{y}^\top \boldsymbol{y} - \boldsymbol{y}^\top \boldsymbol{K}\boldsymbol{A}^{-1}\boldsymbol{A}\boldsymbol{A}^{-1}\boldsymbol{K}^\top \boldsymbol{y} + 2b_\sigma \\
&= (\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}})^\top \boldsymbol{A}(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}) + \boldsymbol{y}^\top \boldsymbol{y} - \boldsymbol{y}^\top \boldsymbol{K}(\boldsymbol{K}^\top \boldsymbol{K} + \boldsymbol{V}^{-1})^{-1}\boldsymbol{K}^\top \boldsymbol{y} + 2b_\sigma \\
&= (\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}})^\top \boldsymbol{A}(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}) + \boldsymbol{y}^\top (\mathbb{I} - \boldsymbol{K}(\boldsymbol{K}^\top \boldsymbol{K} + \boldsymbol{V}^{-1})^{-1}\boldsymbol{K}^\top)\boldsymbol{y} + 2b_\sigma \quad (*)
\end{aligned}$$

Plug $(*)$ back in the equation (A.1),

$$
\begin{aligned}
\pi(\boldsymbol{y}|\boldsymbol{\gamma},\boldsymbol{\rho},\boldsymbol{g},\nu_1) \propto & \int\int (\sigma^2)^{-\frac{2n+2a_\sigma+3}{2}}|\boldsymbol{V}|^{-\frac{1}{2}} \\
& \times \exp\left\{-\frac{1}{2\sigma^2}((\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}})^\top \boldsymbol{A}(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}}) + \boldsymbol{y}^\top(\mathbb{I}-\boldsymbol{K}\boldsymbol{A}^{-1}\boldsymbol{K}^\top)\boldsymbol{y} + 2b_\sigma)\right\} d\boldsymbol{\beta} d\sigma^2 \\
= & V|^{-\frac{1}{2}} \int\int (\sigma^2)^{-\frac{2n+2a_\sigma+3}{2}} \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}})^\top \boldsymbol{A}(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}})\right. \\
& \left. + \boldsymbol{y}^\top(\mathbb{I}-\boldsymbol{K}\boldsymbol{A}^{-1}\boldsymbol{K}^\top)\boldsymbol{y} + 2b_\sigma\right\} d\boldsymbol{\beta} d\sigma^2 \\
= & |\boldsymbol{V}|^{-\frac{1}{2}} \int (\sigma^2)^{-\frac{2n+2a_\sigma+3}{2}} \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{y}^\top(\mathbb{I}-\boldsymbol{K}\boldsymbol{A}^{-1}\boldsymbol{K}^\top)\boldsymbol{y} + 2b_\sigma)\right\} \\
& \int \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}})^\top \boldsymbol{A}(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}})\right\} d\boldsymbol{\beta} d\sigma^2 \\
= & |V|^{-\frac{1}{2}} \int (\sigma^2)^{-\frac{2n+2a_\sigma+3}{2}} \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{y}^\top(\mathbb{I}-\boldsymbol{K}\boldsymbol{A}^{-1}\boldsymbol{K}^\top)\boldsymbol{y} + 2b_\sigma)\right\} \\
& \times (2\pi)^{\frac{n+1}{2}}(\sigma^2)^{\frac{n+1}{2}}|\boldsymbol{A}^{-1}|^{\frac{1}{2}} \int (2\pi)^{-\frac{n+1}{2}}|\sigma^2\boldsymbol{A}^{-1}|^{-\frac{1}{2}} \\
& \times \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}})^\top \boldsymbol{A}(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}})\right\} d\boldsymbol{\beta} d\sigma^2 \\
\propto & |\boldsymbol{V}|^{-\frac{1}{2}}|\boldsymbol{A}^{-1}|^{\frac{1}{2}} \int (\sigma^2)^{-\frac{n+2a_\sigma+2}{2}} \\
& \times \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{y}^\top(\mathbb{I}-\boldsymbol{K}\boldsymbol{A}^{-1}\boldsymbol{K}^\top)\boldsymbol{y} + 2b_\sigma)\right\} d\sigma^2
\end{aligned}
\tag{A.3}
$$

Let $a^* = \frac{n+2a_\sigma}{2}$ and $b^* = \frac{1}{2}(\boldsymbol{y}^\top(\mathbb{I}-\boldsymbol{K}\boldsymbol{A}^{-1}\boldsymbol{K}^\top)\boldsymbol{y}) + b_1$ and plug a^* and b^* back in the equation (A.3),

$$
\begin{aligned}
\pi(\boldsymbol{y}|\boldsymbol{\gamma},\boldsymbol{\rho},\boldsymbol{g},\nu_1) \propto & |\boldsymbol{V}|^{-\frac{1}{2}}|\boldsymbol{A}|^{-\frac{1}{2}} \frac{\Gamma(a^*)}{{b^*}^{a^*}} \\
& \times \int \frac{{b^*}^{a^*}}{\Gamma(a^*)}(\sigma^2)^{-(a^*+1)} \exp\left\{-\frac{b^*}{\sigma^2}\right\} d\sigma^2 \\
= & |\boldsymbol{V}|^{-\frac{1}{2}}|\boldsymbol{A}|^{-\frac{1}{2}} \frac{\Gamma(a^*)}{{b^*}^{a^*}} \\
\propto & |\boldsymbol{A}|^{-\frac{1}{2}} \frac{1}{{b^*}^{a^*}}
\end{aligned}
$$

A.2 Derivation of equation (4.8)

The model likelihood is defined in equation (4.4). With the spike and slab prior, we have

$$\begin{aligned}\pi(\boldsymbol{y}|\boldsymbol{\rho},\boldsymbol{g},\boldsymbol{\lambda}) &\propto \int \mathcal{L}(\boldsymbol{y}|\boldsymbol{\rho},\boldsymbol{\beta},\boldsymbol{\lambda})\pi(\boldsymbol{\beta}|\sigma^2,\boldsymbol{g})d\boldsymbol{\beta} \\ &\propto \int \frac{1}{\sqrt{|\boldsymbol{\lambda}\mathbb{I}_n|}} \exp\left\{-\frac{1}{2}\left(1-\boldsymbol{y}^\top\boldsymbol{K}\boldsymbol{\beta}+\boldsymbol{\lambda}\right)^\top(\boldsymbol{\lambda}\mathbb{I}_n)^{-1}\left(1-\boldsymbol{y}^\top\boldsymbol{K}\boldsymbol{\beta}+\boldsymbol{\lambda}\right)\right\} \\ &\quad \times \exp\left\{-\frac{1}{2}\boldsymbol{\beta}^\top(\boldsymbol{V})^{-1}\boldsymbol{\beta}\right\}d\boldsymbol{\beta}\end{aligned} \tag{A.4}$$

where $\boldsymbol{V}$ is a diagonal matrix with ν_1 and ν_0 as its diagonal elements. Let $\tilde{\boldsymbol{\lambda}} = 1+\boldsymbol{\lambda}$, $\boldsymbol{\lambda}^* = \boldsymbol{\lambda}\mathbb{I}_n$ and $\tilde{\boldsymbol{K}} = \boldsymbol{y}^\top\boldsymbol{K}$, then the equation (A.4) is given as

$$\begin{aligned}\pi(\boldsymbol{y}|\boldsymbol{\rho},\boldsymbol{g},\boldsymbol{\lambda}) &\propto \int \exp\left\{-\frac{1}{2}\left(\tilde{\boldsymbol{\lambda}}-\tilde{\boldsymbol{K}}\boldsymbol{\beta}\right)^\top(\boldsymbol{\lambda}\mathbb{I}_n)^{-1}\left(\tilde{\boldsymbol{\lambda}}-\tilde{\boldsymbol{K}}\boldsymbol{\beta}\right)\right\}\exp\left\{-\frac{1}{2}\boldsymbol{\beta}^\top(\boldsymbol{V})^{-1}\boldsymbol{\beta}\right\}d\boldsymbol{\beta} \\ &= \int \exp\left\{-\frac{1}{2}\left(\tilde{\boldsymbol{\lambda}}^\top\boldsymbol{\lambda}^*\tilde{\boldsymbol{\lambda}} - 2\boldsymbol{\beta}^\top\tilde{\boldsymbol{K}}^\top\boldsymbol{\lambda}^{*-1}\tilde{\boldsymbol{\lambda}} + \boldsymbol{\beta}^\top\tilde{\boldsymbol{K}}^\top\boldsymbol{\lambda}^{*-1}\tilde{\boldsymbol{K}}\boldsymbol{\beta} + \boldsymbol{\beta}^T\boldsymbol{V}^{-1}\boldsymbol{\beta}\right)\right\}d\boldsymbol{\beta}\end{aligned}$$

Inside the exponential term, we have

$$\begin{aligned}&\tilde{\boldsymbol{\lambda}}^\top\boldsymbol{\lambda}^*\tilde{\boldsymbol{\lambda}} - 2\boldsymbol{\beta}^\top\tilde{\boldsymbol{K}}^\top\boldsymbol{\lambda}^{*-1}\tilde{\boldsymbol{\lambda}} + \boldsymbol{\beta}^\top\tilde{\boldsymbol{K}}^\top\boldsymbol{\lambda}^{*-1}\tilde{\boldsymbol{K}}\boldsymbol{\beta} + \boldsymbol{\beta}^T\boldsymbol{V}^{-1}\boldsymbol{\beta} \\ &= \boldsymbol{\beta}^\top\left(\tilde{\boldsymbol{K}}^\top\boldsymbol{\lambda}^{*-1}\tilde{\boldsymbol{K}}\right)\boldsymbol{\beta} - 2\boldsymbol{\beta}^\top\left(\tilde{\boldsymbol{K}}^\top\boldsymbol{\lambda}^{*-1}\tilde{\boldsymbol{K}}+\boldsymbol{V}^{-1}\right)^{-1} \\ &\quad \times\left(\tilde{\boldsymbol{K}}^\top\boldsymbol{\lambda}^{*-1}\tilde{\boldsymbol{K}}+\boldsymbol{V}^{-1}\right)\tilde{\boldsymbol{K}}^\top\boldsymbol{\lambda}^{*-1}\tilde{\boldsymbol{\lambda}} + \tilde{\boldsymbol{\lambda}}^\top\boldsymbol{\lambda}^{*-1}\tilde{\boldsymbol{\lambda}}\end{aligned} \tag{A.5}$$

Let $\boldsymbol{A} = \tilde{\boldsymbol{K}}^\top\boldsymbol{\lambda}^{*-1}\tilde{\boldsymbol{K}} + \boldsymbol{V}^{-1}$ and $\tilde{\boldsymbol{\beta}} = \boldsymbol{A}^{-1}\tilde{\boldsymbol{K}}^\top\boldsymbol{\lambda}^{*-1}\tilde{\boldsymbol{\lambda}}$, the equation (A.5) is updated to

$$\begin{aligned}&\tilde{\boldsymbol{\lambda}}^\top\boldsymbol{\lambda}^*\tilde{\boldsymbol{\lambda}} - 2\boldsymbol{\beta}^\top\tilde{\boldsymbol{K}}^\top\boldsymbol{\lambda}^{*-1}\tilde{\boldsymbol{\lambda}} + \boldsymbol{\beta}^\top\tilde{\boldsymbol{K}}^\top\boldsymbol{\lambda}^{*-1}\tilde{\boldsymbol{K}}\boldsymbol{\beta} + \boldsymbol{\beta}^T\boldsymbol{V}^{-1}\boldsymbol{\beta} \\ &= (\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}})^\top\boldsymbol{A}(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}}) - \tilde{\boldsymbol{\beta}}^T\boldsymbol{A}\tilde{\boldsymbol{\beta}} + \tilde{\boldsymbol{\lambda}}^\top\boldsymbol{\lambda}^{*-1}\tilde{\boldsymbol{\lambda}} \\ &= (\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}})^\top\boldsymbol{A}(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}}) + \tilde{\boldsymbol{\lambda}}^\top\boldsymbol{\lambda}^{*-1}\tilde{\boldsymbol{\lambda}} \\ &\quad - \tilde{\boldsymbol{\lambda}}^\top\boldsymbol{\lambda}^{*-1}\tilde{\boldsymbol{K}}\boldsymbol{A}^{-1}\boldsymbol{A}\boldsymbol{A}^{-1}\tilde{\boldsymbol{K}}^\top\boldsymbol{\lambda}^{*-1}\tilde{\boldsymbol{\lambda}} \\ &= (\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}})^\top\boldsymbol{A}(\boldsymbol{\beta}-\tilde{\boldsymbol{\beta}}) + \tilde{\boldsymbol{\lambda}}^\top\boldsymbol{\lambda}^{*-1}\tilde{\boldsymbol{\lambda}}\end{aligned}$$

$$- \tilde{\boldsymbol{\lambda}}^\top \boldsymbol{\lambda}^{*-1} \tilde{\boldsymbol{K}} \left(\tilde{\boldsymbol{K}}^\top \boldsymbol{\lambda}^{*-1} \tilde{\boldsymbol{K}} + \boldsymbol{V}^{-1} \right)^{-1} \tilde{\boldsymbol{K}}^\top \boldsymbol{\lambda}^{*-1} \tilde{\boldsymbol{\lambda}} \quad (*)$$

Plug $(*)$ back in equation (A.4),

$$\begin{aligned}
\pi(\boldsymbol{y}|\boldsymbol{\rho}, \boldsymbol{g}, \boldsymbol{\lambda}) &\propto \int \exp\left\{ -\frac{1}{2} \left(\tilde{\boldsymbol{\lambda}}^\top \boldsymbol{\lambda}^* \tilde{\boldsymbol{\lambda}} - 2\boldsymbol{\beta}^\top \tilde{\boldsymbol{K}}^\top \boldsymbol{\lambda}^{*-1} \tilde{\boldsymbol{\lambda}} + \boldsymbol{\beta}^\top \tilde{\boldsymbol{K}}^\top \boldsymbol{\lambda}^{*-1} \tilde{\boldsymbol{K}} \boldsymbol{\beta} + \boldsymbol{\beta}^T \boldsymbol{V}^{-1} \boldsymbol{\beta} \right) \right\} d\boldsymbol{\beta} \\
&= \int \exp\left\{ -\frac{1}{2} \Big[(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}})^\top \boldsymbol{A} (\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}) + \tilde{\boldsymbol{\lambda}}^\top \boldsymbol{\lambda}^{*-1} \tilde{\boldsymbol{\lambda}} \right. \\
&\qquad \left. - \tilde{\boldsymbol{\lambda}}^\top \boldsymbol{\lambda}^{*-1} \tilde{\boldsymbol{K}} \left(\tilde{\boldsymbol{K}}^\top \boldsymbol{\lambda}^{*-1} \tilde{\boldsymbol{K}} + \boldsymbol{V}^{-1} \right)^{-1} \tilde{\boldsymbol{K}}^\top \boldsymbol{\lambda}^{*-1} \tilde{\boldsymbol{\lambda}} \Big] \right\} d\boldsymbol{\beta} \\
&\propto \exp\left\{ \frac{1}{2} \tilde{\boldsymbol{\lambda}}^\top \boldsymbol{\lambda}^{*-1} \tilde{\boldsymbol{K}} \left(\tilde{\boldsymbol{K}}^\top \boldsymbol{\lambda}^{*-1} \tilde{\boldsymbol{K}} + \boldsymbol{V}^{-1} \right)^{-1} \tilde{\boldsymbol{K}}^\top \boldsymbol{\lambda}^{*-1} \tilde{\boldsymbol{\lambda}} \right\} \\
&\qquad \times \int \exp\{ -\frac{1}{2} \Big[(\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}})^\top \boldsymbol{A} (\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}) \Big] \} d\boldsymbol{\beta} \\
&\propto |\boldsymbol{A}|^{-\frac{1}{2}} \exp\left\{ \frac{1}{2} \tilde{\boldsymbol{\lambda}}^\top \boldsymbol{\lambda}^{*-1} \tilde{\boldsymbol{K}} \left(\tilde{\boldsymbol{K}}^\top \boldsymbol{\lambda}^{*-1} \tilde{\boldsymbol{K}} + \boldsymbol{V}^{-1} \right)^{-1} \tilde{\boldsymbol{K}}^\top \boldsymbol{\lambda}^{*-1} \tilde{\boldsymbol{\lambda}} \right\}
\end{aligned}$$

Appendix B

Bayesian ridge kernel models for high dimensional regression and SVM

B.1 Bayesian ridge penalized RKHS regression

Note in both Section 3.5 and Section 3.6, we have compared our proposed Bayesian doubly-sparse RKHS regression to the Bayesian RKHS ridge penalized regression. In this section, we give an introduction to the prior assumptions and implementations details of the method. For the Bayesian RKHS ridge penalized regression, we assume a prior on the linear coefficients which are equivalent to appending an L_2 penalty to the likelihood from the frequentist perspective.

With the model defined in equation (3.3), the same prior assumptions are employed as Mallick et al. (2005). To be more specific, we assume

$$\boldsymbol{\beta}, \sigma^2 \sim \mathcal{N}_{n+1}(\mathbf{0}, \sigma^2 \boldsymbol{D}_*) \times \text{Inverse-Gamma}(a_\sigma, b_\sigma)$$

where $\boldsymbol{D}_* \equiv \text{diag}(\lambda_0, \lambda, \cdots, \lambda)$ is a $(n+1) \times (n+1)$ diagonal matrix. We fix the λ_0 to be a small number and assume

$$\lambda \sim \text{Gamma}(a_\lambda, b_\lambda).$$

Then similar as our proposed collapsed Gibbs sampler for the doubly-sparse method, we update the samples of $\boldsymbol{\beta}$ via

$$\boldsymbol{\beta}|\text{others} \sim \mathcal{N}\left(\left[\boldsymbol{K}^\top\boldsymbol{K}+\boldsymbol{D}_*^{-1}\right]^{-1}\boldsymbol{K}^\top\boldsymbol{y}, \sigma^2\left[\boldsymbol{K}^\top\boldsymbol{K}+\boldsymbol{D}_*^{-1}\right]^{-1}\right).$$

For the random error variance σ^2, its samples are generated through

$$\sigma^2|\text{others} \sim \text{Inverse-Gamma}\left(\frac{2n+2a_\sigma+1}{2}, \frac{||\boldsymbol{y}-\boldsymbol{K}\boldsymbol{\beta}||^2+\boldsymbol{\beta}^\top\boldsymbol{D}_*^{-1}\boldsymbol{\beta}}{2}+b_\sigma\right).$$

For λ, we generate its samples by

$$\lambda|\text{others} \sim \text{Gamma}\left(\frac{n}{2}+a_\lambda, \frac{||\boldsymbol{\beta}||^2}{2\sigma^2}+b_\lambda\right).$$

For the purpose of variable selection, we update $(\boldsymbol{\gamma}, \boldsymbol{\rho})$ via the same Metropolis-Hastings algorithm that is developed for the doubly-sparse method. The marginal likelihood is computed with

$$\pi(\boldsymbol{y}|\boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{g}, \lambda) \propto |\boldsymbol{K}^\top\boldsymbol{K}+\boldsymbol{D}_*^{-1}|^{-\frac{1}{2}}\frac{1}{{b^*}^{a^*}}$$

where $a^* = \frac{n+2a_\sigma}{2}$ and $b^* = \frac{1}{2}(\boldsymbol{y}^\top(\mathbb{I}-\boldsymbol{K}(\boldsymbol{K}^\top\boldsymbol{K}+\boldsymbol{D}_*^{-1})^{-1}\boldsymbol{K}^\top)\boldsymbol{y})+b_\sigma$.

B.2 Bayesian nonlinear SVM with variable selection

In this section, we give an introduction to the prior set-up and the implementation details of the Bayesian nonlinear kernel SVM with variable selection. We assume a flat prior on $\boldsymbol{\beta}$, which is equivalent of appending an L_2 penalty to the likelihood function. Variable selection is conducted with point mass priors within the kernel. The method can be seen as the Bayesian SVM version of Chakraborty (2009).

With model likelihood defined in equation (4.4), the same prior proposed by Mallick et al.

(2005) is adopted. In particular, we assume

$$\boldsymbol{\beta} \sim \mathcal{N}_{n+1}(\mathbf{0}, \sigma^2 \boldsymbol{D}_*)$$

where $\boldsymbol{D}_* \equiv \text{diag}(\tau)$ is a $(n+1) \times (n+1)$ diagonal matrix. In practice, τ is fixed to be a small number, say $\tau = 0.001$. One can also assume a Gamma prior over τ as suggested in Polson and Scott (2011), but which could lead to suboptimal results.

Similar to the doubly-sparse SVM, we update the samples of $\boldsymbol{\beta}$ via

$$\boldsymbol{\beta}|\boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\lambda}, \boldsymbol{y} \sim \mathcal{N}\left((\boldsymbol{Z}^\top \boldsymbol{Z} + \boldsymbol{D}_*)^{-1} \boldsymbol{Z}^\top \boldsymbol{w}, (\boldsymbol{Z}^\top \boldsymbol{Z} + \boldsymbol{D}_*)^{-1}\right)$$

and the samples of $\boldsymbol{\lambda}$ via

$$\frac{1}{\lambda_i}|\boldsymbol{\gamma}, \boldsymbol{\rho}, \boldsymbol{\beta}, \boldsymbol{y} \overset{ind.}{\sim} \text{Inverse-Gaussian}\left(|1 - y_i \tilde{k_i}^\top \boldsymbol{\beta}|, 1\right)$$

The last remaining work is the sampling from the variable selection index $\boldsymbol{\gamma}$ and bandwidth $\boldsymbol{\rho}$, which we update their samples by making use of the same Metropolis-Hastings algorithm developed for the doubly-sparse models. In particular, the marginal likelihood is computed with

$$\pi(\boldsymbol{y}|\boldsymbol{\rho}, \boldsymbol{\lambda}) \propto |\boldsymbol{A}|^{-\frac{1}{2}} \exp\left\{\frac{1}{2} \tilde{\boldsymbol{\lambda}}^\top \boldsymbol{\lambda}^{*-1} \tilde{\boldsymbol{K}} \boldsymbol{A}^{-1} \tilde{\boldsymbol{K}}^\top \boldsymbol{\lambda}^{*-1} \tilde{\boldsymbol{\lambda}}\right\}$$

where $\boldsymbol{A} = (\tilde{\boldsymbol{K}} \boldsymbol{\lambda}^{*-1} \tilde{\boldsymbol{K}} + \boldsymbol{D}_*^{-1})$.