Bayesian Generalized Kernel Models

Zhihua Zhang
College of Comp. Sci. and Tech.
Zhejiang University
Zhejiang 310027, China
zhzhang@cs.zju.edu.cn

Guang Dai **Donghui Wang**
College of Comp. Sci. and Tech.
Zhejiang University
Zhejiang 310027, China
{daiguang116, wdh.zju}@gmail.com

Michael I. Jordan
Depts. of EECS and Statistics
University of California, Berkeley
Berkeley, CA 94720, USA
jordan@cs.berkeley.edu

Abstract

We propose a fully Bayesian approach for generalized kernel models (GKMs), which are extensions of generalized linear models in the feature space induced by a reproducing kernel. We place a mixture of a point-mass distribution and Silverman's g-prior on the regression vector of GKMs. This mixture prior allows a fraction of the regression vector to be zero. Thus, it serves for sparse modeling and Bayesian computation. For inference, we exploit data augmentation methodology to develop a Markov chain Monte Carlo (MCMC) algorithm in which the reversible jump method is used for model selection and a Bayesian model averaging method is used for posterior prediction.

1 Introduction

Supervised learning based on reproducing kernel Hilbert spaces (RKHSs) has become increasingly popular since the support vector machine (SVM) and its variants such as penalized kernel logistic regression models (Zhu and Hastie, 2005) have been proposed. Given the high dimensionality generally associated with RKHS methods, sparseness has also emerged as a significant theme. The SVM naturally embodies sparseness due to its use of the hinge loss function. Penalized kernel logistic regression models, on the other hand, are not naturally sparse. Thus, Zhu and Hastie (2005) proposed a methodology that they refer to as the *import vector machine* (IVM), where a fraction of the training data—called *import vectors* by analogy to the support vectors of the SVM, are used to index kernel basis functions. In this paper we employ the terminology *active vector* for this idea.

Kernel supervised learning methods can be unified using the tools of regularization theory. On the one hand, the regularization term is usually defined as the L_1 or L_2 norm of the vector of regression coefficients (as a penalization technique). From the Bayesian standpoint, this term arises by assigning a Gaussian or Laplacian prior to the regression vector. Indeed, using logarithmic scoring rules (Bernardo and Smith, 1994), a loss function can often be viewed as the negative conditional log-likelihood. The duality between "regularization" and "prior" leads to interpreting regularization methods in terms of maximum *a posteriori* (MAP) estimation, and has motivated many Bayesian interpretations of kernel methods (Tipping, 2001; Sollich, 2001; Mallick et al., 2005; Zhang and Jordan, 2006).

Although the use of either the hinge loss function or L_1 regularization is an effective tool for achieving sparsity in the frequentist paradigm, in the Bayesian setting the corresponding prior yields posteriors that cannot be computed in closed form. Unfortunately, the approaches that are usually used for posterior inference do not necessarily retain sparsity. For example, in the Bayesian approach of Mallick et al. (2005), since conjugate priors for the regression vector do not exist due to the non-normal conditional likelihood which is obtained from the hinge loss, a data augmentation methodology was employed to update the regression vector. As a result, the regression vector is no longer sparse.

In this paper we propose *generalized kernel models* (GKMs) as a framework in which sparsity can be given an explicit treatment and in which a fully Bayesian methodology can be carried out. GKMs are derived from generalized linear models (GLMs) in the RKHS. Since active vectors are indexed by the nonzero components of the regression vector in GKMs, we assign to the regression vector a mixture of the point-mass distribution and a prior which is called the Silverman

Appearing in Proceedings of the 13^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, Chia Laguna Resort, Sardinia, Italy. Volume 9 of JMLR: W&CP 9.

g-prior (Silverman, 1985; Zhang et al., 2008). Our point-mass mixture prior naturally possesses sparsity because it allows a fraction of regression coefficients in question to be zero. Thus it provides a Bayesian approach to active vector selection.

In a recent paper, Zhang et al. (2008) provided a theoretical analysis of the posterior consistency of a Bayesian model choice procedure based on the Silverman g-prior. This prior is related to the Zellner g-prior (Zellner, 1986), which has been widely applied to Bayesian variable selection and Bayesian model selection (Smith and Kohn, 1996; George and McCulloch, 1997; Kohn et al., 2001; Nott and Green, 2004; Sha et al., 2004) because of its computational tractability in evaluating marginal likelihoods.

We apply the Silverman g-prior to the problem of developing fully Bayesian GKMs, including Bayesian approaches for parameter estimation, model selection and response prediction, in the setting of classification. In particular, motivated by the use of data augmentation methodology in Bayesian computation for Bayesian GLMs (Albert and Chib, 1993; Holmes and Held, 2006), we exploit this methodology to devise an MCMC algorithm for our Bayesian GKMs which uses the reversible jump procedure (Green, 1995) for the automatic selection of active vectors and the Bayesian model averaging method (Raftery et al., 1997) for the posterior prediction of future observations. Interestingly, the reversible jump procedure with the help of some matrix techniques can make the MCMC algorithm computationally feasible, even for large datasets.

2 A Bayesian Approach for Kernel Supervised Learning

We start with a supervised learning problem based on a set of training data, $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^p$ is an input vector and y_i is a univariate continuous output for the regression problem or binary output for the classification problem. Our current concern is to learn a predictive function $f(\mathbf{x})$ from the training data.

Suppose $f = u + h \in (\{1\} + \mathcal{H}_K)$ where $\mathcal{H}_K$ is an RKHS. Finding $f(\mathbf{x})$ is then formulated as a regularization problem of the form

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

where $L(y, f(\mathbf{x}))$ is a loss function, $\|h\|^2_{\mathcal{H}_K}$ is the RKHS norm and $g > 0$ is the regularization parameter. By the representer theorem (Wahba, 1990), the solution of (1) is of the form

$$f(\mathbf{x}) = u + \sum_{j=1}^n \beta_j K(\mathbf{x}, \mathbf{x}_j), \qquad (2)$$

where u is called an offset term, $K(\cdot, \cdot)$ is the kernel function and the β_j are referred to as regression coefficients. Noticing that $\|h\|^2_{\mathcal{H}_K} = \sum_{i,j=1}^n K(\mathbf{x}_i, \mathbf{x}_j)\beta_i\beta_j$ and substituting (2) into (1), we obtain the minimization problem with respect to (w.r.t.) the β_j as

$$\min_{u, \boldsymbol{\beta}} \left\{ \frac{1}{n} \sum_{i=1}^n L(y_i, u + \mathbf{k}_i'\boldsymbol{\beta}) + \frac{g}{2} \boldsymbol{\beta}'\mathbf{K}\boldsymbol{\beta} \right\}, \qquad (3)$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_n)'$ is the $n \times 1$ regression vector and $\mathbf{K} = [\mathbf{k}_1, \ldots, \mathbf{k}_n]$ is the $n \times n$ kernel matrix with $\mathbf{k}_i = (K(\mathbf{x}_i, \mathbf{x}_1), \ldots, K(\mathbf{x}_i, \mathbf{x}_n))'$.

The predictive function (2) is based on a basis expansion of kernel functions. The predictive function $f(\mathbf{x})$ can also be expressed by a basis expansion of feature functions. Given a Mercer reproducing kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, there exists a corresponding mapping (say $\boldsymbol{\psi}$) from the input space $\mathcal{X}$ to a feature space (say $\mathcal{F} \subset \mathbb{R}^r$). That is, we have a vector-valued function $\boldsymbol{\psi}(\mathbf{x}) = (\psi_1(\mathbf{x}), \ldots, \psi_r(\mathbf{x}))'$, which is called the *feature vector* of $\mathbf{x}$, such that $K(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\psi}(\mathbf{x}_i)'\boldsymbol{\psi}(\mathbf{x}_j)$. By the Mercer-Hilbert-Schmidt Theorem (Wahba, 1990), we know that there exists an orthogonal sequence of continuous eigenfunctions $\{\phi_j\}$ in the square integrable Hilbert functional space $L_2(\mathcal{X})$ and eigenvalues $l_1 \geq l_2 \geq \ldots \geq 0$. Furthermore, we have a definition of the feature functions $\boldsymbol{\psi} : \mathcal{X} \to L_2(\mathcal{X})$ as $\boldsymbol{\psi}(\mathbf{x}) = \left\{\sqrt{l_j}\phi_j(\mathbf{x})\right\}_{j=1}^r$. That is, $\psi_j(\mathbf{x}) = \sqrt{l_j}\phi_j(\mathbf{x})$. Thus the $\psi_j(\mathbf{x})$ constitute a set of basis functions of $L_2(\mathcal{X})$. Consequently, they can be used to express the predictive function as follows:

$$f(\mathbf{x}) = u + \sum_{k=1}^r b_k\psi_k(\mathbf{x}) = u + \boldsymbol{\psi}(\mathbf{x})'\mathbf{b}, \qquad (4)$$

where $\mathbf{b} = (b_1, \ldots, b_r)'$. There are possibly infinitely many basis functions in (4) because r is possibly infinite. In the case that r is infinite, one may use a finite-dimensional approximation to $f(\mathbf{x})$ by keeping the first n $\psi_j(\mathbf{x})$'s and setting the remaining b_j, $j > n$ to zero. Now letting $\mathbf{b} = \boldsymbol{\Psi}'\boldsymbol{\beta}$, we re-derive (2) from (4) due to $\mathbf{K} = \boldsymbol{\Psi}\boldsymbol{\Psi}'$ where $\boldsymbol{\Psi} = [\boldsymbol{\psi}(\mathbf{x}_1), \ldots, \boldsymbol{\psi}(\mathbf{x}_n)]'$.

2.1 Generalized Kernel Models

In terms of the logarithmic scoring rule (Bernardo and Smith, 1994), the loss $L(y, f(\mathbf{x}))$ is viewed as the negative conditional log-likelihood in the Bayesian literature. This motivates us to construct the following model

$$y \sim p(y|\mu) \quad \text{with} \quad \mu = \tau(u + \mathbf{k}'\boldsymbol{\beta}), \qquad (5)$$

where $\tau(\cdot)$ is a given link function and $\mathbf{k} = (K(\mathbf{x}, \mathbf{x}_1), \ldots, K(\mathbf{x}, \mathbf{x}_n))'$. This model can be obtained from the model of

$$y \sim p(y|\mu) \quad \text{with} \quad \mu = \tau(u + \boldsymbol{\psi}(\mathbf{x})'\mathbf{b}) \qquad (6)$$

by using the transformation $\mathbf{b} = \mathbf{\Psi}'\boldsymbol{\beta}$. Since the model in (6) is in fact an extension of GLMs in the feature space, we call model (5) the *generalized kernel model* (GKM).

GKMs provide a unifying framework of kernel models for regression and classification. With different $p(y|\mu)$ and τ, we have different kernel models. In the regression problem, $p(y|\mu)$ is usually normal and τ is the identity function.

In this paper we are mainly concerned with the classification problem where y is encoded as a binary value, i.e., $y \in \{0, 1\}$. We thus model $p(y|\mu)$ as Bernoulli distribution:

$$p(y|\mu) = \mu^y(1-\mu)^{1-y} = [\tau(u+\mathbf{k}'\boldsymbol{\beta})]^y[1-\tau(u+\mathbf{k}'\boldsymbol{\beta})]^{1-y}.$$

Typically, τ is either the logistic link $\tau(z) = \frac{\exp(z)}{1+\exp(z)}$ or the probit link $\tau(z) = \Phi(z)$, the cumulative distribution function of a standard normal variable. The probit link is widely used in Bayesian GLMs due to its tractability in calculating the marginal likelihood. In our fully Bayesian GKMs in Section 3, we will use this link.

2.2 Silverman's g-prior

Assume that the b_k are independent Gaussian variables with $E(b_k) = 0$ and $E(b_k^2) = g^{-1}$, that is, $\mathbf{b} \sim N_r(\mathbf{0},\ g^{-1}\mathbf{I}_r)$. Here and later, we denote by $\mathbf{I}_m$ the $m\times m$ identity matrix, by $\mathbf{1}_m$ the $m\times 1$ vector of ones, and by $\mathbf{0}$ the zero vector or matrix with appropriate size. Because of $\mathbf{b} = \mathbf{\Psi}'\boldsymbol{\beta}$, we have $\boldsymbol{\beta} = \mathbf{K}^{-1}\mathbf{\Psi}\mathbf{b}$. As a result, the prior for $\boldsymbol{\beta}$ is $\boldsymbol{\beta} \sim N_n\left(\mathbf{0},\ g^{-1}\mathbf{K}^{-1}\right)$ due to $\mathbf{K}^{-1}\mathbf{\Psi}\mathbf{\Psi}'\mathbf{K}^{-1} = \mathbf{K}^{-1}$. It is possible that the kernel matrix $\mathbf{K}$ is singular. For such a $\mathbf{K}$, we use its Moore-Penrose inverse $\mathbf{K}^+$ instead and still have $\mathbf{K}^+\mathbf{K}\mathbf{K}^+ = \mathbf{K}^+$. However, the prior distribution of $\boldsymbol{\beta}$ becomes a singular normal distribution. In any case, we always use $\mathbf{K}^{-1}$ for notational simplicity.

The prior $N_n\left(\mathbf{0},\ \mathbf{K}^{-1}\right)$ for $\boldsymbol{\beta}$ was first proposed by Silverman (1985) in his Bayesian formulation of spline smoothing. Subsequently, Zhang et al. (2008) referred to the prior $\boldsymbol{\beta} \sim N_n\left(\mathbf{0},\ g^{-1}\mathbf{K}^{-1}\right)$ as the *Silverman g-prior* by analogy with the Zellner g-prior (Zellner, 1986). When $\mathbf{K}$ is singular, similar to *generalized singular g-prior* (*gsg*-prior) (West, 2003), we call $N_n\left(\mathbf{0},\ g^{-1}\mathbf{K}^{-1}\right)$ a *generalized Silverman g-prior*. It is worth pointing out that Green (1985) argued that the definition of Silverman's prior is implicit. We have presented an explicit derivation of this prior. Since the prior density of $\boldsymbol{\beta}$ is proportional to $\exp(-g\boldsymbol{\beta}'\mathbf{K}\boldsymbol{\beta}/2)$, the Silverman g-prior is design-dependent. Note also that the regularization term $g\boldsymbol{\beta}'\mathbf{K}\boldsymbol{\beta}/2$ in (3) is readily derived from this prior.

2.3 Sparse Models

Recall that the number of active vectors is equal to the number of nonzero components of $\boldsymbol{\beta}$. That is, if $\beta_j = 0$, the jth input vector is excluded from the basis expansion in (2), otherwise the jth input vector is an active vector. We are thus interested in a prior for $\boldsymbol{\beta}$ which allows some components of $\boldsymbol{\beta}$ to be zero. In particular, we assign a point-mass mixture prior to $\boldsymbol{\beta}$ built on the Silverman g-prior.

We introduce an indicator binary vector $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_n)'$ such that $\gamma_j = 1$ if $\mathbf{x}_j$ is an active vector and $\gamma_j = 0$ if it is not. Let $n_\gamma = \sum_{j=1}^n \gamma_j$ be the number of active vectors, and let $\mathbf{K}_\gamma$ be the $n\times n_\gamma$ submatrix of $\mathbf{K}$ consisting of those columns of $\mathbf{K}$ for which $\gamma_j = 1$. We further let $\mathbf{K}_{\gamma\gamma}$ be the $n_\gamma\times n_\gamma$ submatrix of $\mathbf{K}_\gamma$ consisting of those rows of $\mathbf{K}_\gamma$ for which $\gamma_j = 1$, and $\boldsymbol{\beta}_\gamma$ and $\mathbf{k}_\gamma$ be the corresponding $n_\gamma\times 1$ subvectors of $\boldsymbol{\beta}$ and $\mathbf{k}$. Based on GKMs in (5) and the Silverman g-prior, we thus obtain the following sparse model

$$\begin{aligned} y &\sim p(y|\tau(f(\mathbf{x}))) \\ f(\mathbf{x}) &= u + \mathbf{k}_\gamma'\boldsymbol{\beta}_\gamma \ \text{ and } \ \boldsymbol{\beta}_\gamma \sim N_{n_\gamma}(\mathbf{0},\ g^{-1}\mathbf{K}_{\gamma\gamma}^{-1}). \end{aligned} \tag{7}$$

In the existing literature for sparse classification and regression, a typical choice of the prior on $\boldsymbol{\beta}$ is Laplacian prior, also well known as L_1-penalized regularization. In frequentist treatments the corresponding L_1 penalty term is well known to yield sparseness (Tibshirani, 1996). In the Bayesian setting, however, the posterior distribution is not available in closed form, thus approximations are needed; these are based on the expression of the Laplacian prior as scale-mixtures-of-normals or are based on a Laplace approximation. This makes posterior inference tractable, but does not necessarily retain sparseness. Thus, Tipping (2001) and Figueiredo (2003) employed an empirical Bayes approach instead.

In the Bayesian and complete SVMs of Mallick et al. (2005), the prior on $\boldsymbol{\beta}$ is $\boldsymbol{\beta} \sim N_n(\mathbf{0},\ \mathbf{\Lambda}^{-1})$ where $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ is the $n\times n$ diagonal matrix with $\lambda_i > 0$. Note that this prior does not induce the regularization term $g\boldsymbol{\beta}'\mathbf{K}\boldsymbol{\beta}/2$ in (3). Note also that the model selection problem of finding sparse support-vector expansions is not addressed in this approach. In this paper we give a Bayesian method for selecting active vectors based on the mixture of the point-mass prior and the Silverman g-prior.

3 Methodology

In this section we present a fully Bayesian GKM (FBGKM) based on (7). Since $p(y|\tau(f(\mathbf{x})))$ is non-normal for the classification problem, conjugate priors for $\boldsymbol{\beta}$ usually do not exist. In order to facilitate the

implementation of Bayesian inference in this setting, we make use of the data augmentation methodology, which has been used by Albert and Chib (1993) for Bayesian GLMs and by Mallick et al. (2005) for their Bayesian SVMs. The basic idea is to introduce auxiliary variables linking y and the model parameters. We apply this methodology to our FBGKM.

3.1 Hierarchical Models

Let $\mathbf{s} = (s_1, \ldots, s_n)'$ be the auxiliary vector corresponding to the training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$. We in particular define

$$\mathbf{s} = u\mathbf{1}_n + \mathbf{K}_\gamma \boldsymbol{\beta}_\gamma + \boldsymbol{\epsilon} \;\text{ with }\; \boldsymbol{\epsilon} \sim N_n(\mathbf{0},\ \sigma^2 \mathbf{I}_n).$$

Since τ is defined as the probit link in the FBGKM, we have $\sigma^2 = 1$ and

$$y_i = \begin{cases} 1 & \text{if } s_i > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Given $\mathbf{s}$, $\mathbf{y} = (y_1, \ldots, y_n)'$ is thus independent of u, $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$. Consequently, we can assign conjugate priors to these parameters.

First, we assume $u \sim N(0,\ \eta^{-1})$ and $g \sim Ga(a_g/2, b_g/2)$, where $Ga(a, b)$ represents a gamma distribution. Let $\tilde{\boldsymbol{\beta}}_\gamma = (u, \boldsymbol{\beta}_\gamma')'$. We thus have

$$\widetilde{\boldsymbol{\beta}}_\gamma \sim N_{n_\gamma+1}(\mathbf{0},\ \boldsymbol{\Sigma}_\gamma^{-1}) \;\text{ with }\; \boldsymbol{\Sigma}_\gamma = \begin{bmatrix} \eta & \mathbf{0} \\ \mathbf{0} & g\mathbf{K}_{\gamma\gamma} \end{bmatrix}.$$

By integrating out $\widetilde{\boldsymbol{\beta}}_\gamma$, the marginal distribution of $\mathbf{s}$ conditional on $\boldsymbol{\gamma}$ is normal, namely,

$$p(\mathbf{s}|\boldsymbol{\gamma}) = N_n(\mathbf{0},\ \mathbf{Q}_\gamma) \tag{8}$$

with $\mathbf{Q}_\gamma = \mathbf{I}_n + \widetilde{\mathbf{K}}_\gamma \boldsymbol{\Sigma}_\gamma^{-1} \widetilde{\mathbf{K}}_\gamma'$ where $\widetilde{\mathbf{K}}_\gamma = [\mathbf{1}_n, \mathbf{K}_\gamma]$ $(n \times (n_\gamma + 1))$. Bayes' theorem yields the following distribution of $\widetilde{\boldsymbol{\beta}}_\gamma$ conditional on $\mathbf{s}$ and $\boldsymbol{\gamma}$:

$$[\widetilde{\boldsymbol{\beta}}_\gamma | \mathbf{s}, \boldsymbol{\gamma}] \sim N_{n_\gamma+1}(\boldsymbol{\Upsilon}_\gamma^{-1} \widetilde{\mathbf{K}}_\gamma' \mathbf{s},\ \boldsymbol{\Upsilon}_\gamma^{-1}), \tag{9}$$

where $\boldsymbol{\Upsilon}_\gamma = \widetilde{\mathbf{K}}_\gamma' \widetilde{\mathbf{K}}_\gamma + \boldsymbol{\Sigma}_\gamma$.

Second, as in Kohn et al. (2001) and Nott and Green (2004), we assign an independent Bernoulli prior to each component of $\boldsymbol{\gamma}$, namely,

$$p(\boldsymbol{\gamma}|\alpha) = \prod_{j=1}^{n} \alpha^{\gamma_j} (1-\alpha)^{1-\gamma_j} = \alpha^{n_\gamma} (1-\alpha)^{n-n_\gamma},$$

where $\alpha \in (0, 1)$. It is natural to place a Beta prior on α, $\alpha \sim B(a_\alpha, b_\alpha)$. Marginalizing out α results in the following prior on $\boldsymbol{\gamma}$:

$$p(\boldsymbol{\gamma}) = \frac{Be(n_\gamma + a_\alpha, n - n_\gamma + b_\alpha)}{Be(a_\alpha, b_\alpha)}, \tag{10}$$

where $Be(\cdot, \cdot)$ is the Beta function. Kohn et al. (2001) proposed a method of selecting the hyperparameters a_α and b_α by controlling the value of n_γ. In the following experiments, we use the uninformative fixed specification $a_\alpha = 1$ and $b_\alpha = 1$.

Finally, we assume that η follows $Ga(a_\eta/2,\ b_\eta/2)$. As in Zhang and Jordan (2006) and Mallick et al. (2005), where the authors considered the Bayesian estimate of the kernel function K, we can let K be indexed by an unknown hyperparameter $\boldsymbol{\theta}$ to which we assign a prior. For simplicity, however, we shall keep K as well as the hyperparameters a_η, b_η, a_g and b_g fixed in this paper. In summary, we form a hierarchical model in which the joint density of all variables mentioned takes the form

$$\begin{aligned} &p(\mathbf{y}, \mathbf{s}, \boldsymbol{\gamma}, u, \boldsymbol{\beta}, \eta, g) \\ &\quad = p(\eta)p(g)p(\boldsymbol{\gamma})p(u|\eta)p(\boldsymbol{\beta}|g, \boldsymbol{\gamma})p(\mathbf{s}|u, \boldsymbol{\beta}, \boldsymbol{\gamma})p(\mathbf{y}|\mathbf{s}). \end{aligned}$$

3.2 Inference

Our goal is to generate the realizations of parameters from the conditional joint density $p(\mathbf{s}, u, \boldsymbol{\beta}, \boldsymbol{\gamma}, g|\mathbf{y})$ via an MCMC algorithm. In order to speed up mixing of the algorithm, we will use marginal posterior distributions whenever possible. Our MCMC algorithm consists of the following steps.

Start Specify a_η, b_η, a_g and b_g, and initialize $\mathbf{s}$, $\boldsymbol{\gamma}$, g, η, u and $\boldsymbol{\beta}_\gamma$.

Step (a) Impute each s_i from $p(s_i|y_i, u, \boldsymbol{\beta}_\gamma)$.

Step (b) Update η, g and $\widetilde{\boldsymbol{\beta}}_\gamma$ according to $p(\eta|u)$, $p(g|\boldsymbol{\beta}_\gamma)$, and $p(\widetilde{\boldsymbol{\beta}}_\gamma|\mathbf{s}, \boldsymbol{\gamma}, \eta, g)$, respectively.

Step (c) Update $\boldsymbol{\gamma}$ from $p(\boldsymbol{\gamma}|\mathbf{s})$.

Step (a) is to draw $\mathbf{s}$ from $p(\mathbf{s}|\mathbf{y}, u, \boldsymbol{\beta}_\gamma)$. We perform this step by using a technique which was proposed by Holmes and Held (2006) for conventional probit regression. In particular, $\mathbf{s}$ is updated from its marginal distribution having integrated over $\widetilde{\boldsymbol{\beta}}_\gamma$; that is, s_i is generated from $p(s_i|\mathbf{s}_{-i}, y_i, \gamma)$ where $\mathbf{s}_{-i} = (s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n)'$. For details refer to Holmes and Held (2006).

We now consider the updates of $\widetilde{\boldsymbol{\beta}}_\gamma$, η and g. Given $\mathbf{s}$, these parameters are independent of $\mathbf{y}$, so their updates are based on $p(\widetilde{\boldsymbol{\beta}}_\gamma, \eta, g|\mathbf{s}, \boldsymbol{\gamma})$. Hence, we update $\widetilde{\boldsymbol{\beta}}_\gamma$ from $[\widetilde{\boldsymbol{\beta}}_\gamma|\mathbf{s}, \boldsymbol{\gamma}, \eta, g] \sim N_{n_\gamma+1}(\boldsymbol{\Upsilon}_\gamma^{-1}\widetilde{\mathbf{K}}_\gamma'\mathbf{s},\ \boldsymbol{\Upsilon}_\gamma^{-1})$. Since g is dependent only on $\boldsymbol{\beta}_\gamma$ and the prior is conjugate, we use the Gibbs sampler to update g from its conditional distribution, which is given by

$$[g|\boldsymbol{\beta}_\gamma] \sim Ga\Big(\frac{a_g + n_\gamma}{2}, \frac{b_g + \boldsymbol{\beta}_\gamma' \mathbf{K}_{\gamma\gamma} \boldsymbol{\beta}_\gamma}{2}\Big).$$

The update of η is obtained from its conditional distribution as

$$[\eta|u] \sim Ga\Big(\frac{a_\eta+1}{2}, \frac{b_\eta+u^2}{2}\Big).$$

Step (c) is used for the automatic choice of active vectors. To implement this step, we borrow a method devised by Nott and Green (2004). This method was derived from the reversible jump methodology of Green (1995). Specifically, we generate a proposal $\boldsymbol{\gamma}^*$ from the current value of $\boldsymbol{\gamma}$ by one of three possible moves:

Birth move: randomly choose a 0 in $\boldsymbol{\gamma}$ and change it to 1;

Death move: randomly choose a 1 in $\boldsymbol{\gamma}$ and change it to 0;

Swap move: randomly choose a 0 and a 1 in $\boldsymbol{\gamma}$ and switch them.

The acceptance probability for each move is

$$\min\{1, \text{likelihood ratio} \times \text{prior ratio} \times \text{proposal ratio}\}.$$

Letting $k = n_\gamma$, we denote the probabilities of birth, death and swap by b_k, d_k and $1-b_k-d_k$, respectively. For birth, death and swap moves, their acceptance probabilities are

$$\min\left\{1, \frac{p(\mathbf{s}|\boldsymbol{\gamma}^*)p(\boldsymbol{\gamma}^*)d_{k+1}}{p(\mathbf{s}|\boldsymbol{\gamma})p(\boldsymbol{\gamma})b_k}\frac{n-k}{k+1}\right\},$$

$$\min\left\{1, \frac{p(\mathbf{s}|\boldsymbol{\gamma}^*)p(\boldsymbol{\gamma}^*)b_{k-1}}{p(\mathbf{s}|\boldsymbol{\gamma})p(\boldsymbol{\gamma})d_k}\frac{k}{n-k+1}\right\},$$

$$\min\left\{1, \frac{p(\mathbf{s}|\boldsymbol{\gamma}^*)p(\boldsymbol{\gamma}^*)}{p(\mathbf{s}|\boldsymbol{\gamma})p(\boldsymbol{\gamma})}\right\},$$

where $p(\mathbf{s}|\boldsymbol{\gamma})$ and $p(\boldsymbol{\gamma})$ are given in (8) and (10). In our experiments we set $b_0 = 1$ and $d_0 = 0$, $b_k = d_k = 0.3$ for $1 \le k \le k_{\max}-1$, and $d_k = 1$ and $b_k = 0$ for $k_{\max} \le k \le n$. Here, $k_{\max}$ is a specified maximum number of active vectors such that $k_{\max} \le n$.

An alternative to this approach is the stochastic search method of George and McCulloch (1997). This method also employs birth, death and swap moves; it differs from the reversible jump procedure because it does not incorporate the probabilities of birth, death and swap into its acceptance probabilities.

Recall that the main computational burden of our MCMC algorithm comes from the calculations of the determinant and inverse of $\mathbf{Q}_\gamma$ ($\mathbf{Q}_{\gamma^*}$) during the MCMC sweeps. It is worth noting that when n is relatively large, we can reduce the computation burden by giving $k_{\max}$ a value far less than n, i.e., $k_{\max} \ll n$, and then computing:

$$\mathbf{Q}_\gamma^{-1} = \mathbf{I}_n - \widetilde{\mathbf{K}}_\gamma \boldsymbol{\Upsilon}_\gamma^{-1} \widetilde{\mathbf{K}}_\gamma'$$
$$|\mathbf{Q}_\gamma| = |\boldsymbol{\Upsilon}_\gamma||\boldsymbol{\Sigma}_\gamma|^{-1} = \eta^{-1} g^{-n_\gamma} |\mathbf{K}_{\gamma\gamma}|^{-1} |\boldsymbol{\Upsilon}_\gamma|.$$

For example, for both the USPS and NewsGroups datasets used in our experiments, we set $k_{\max} = 200 \ll n$. In this setting, we always have $n_\gamma \le k_{\max} \ll n$. Since $\boldsymbol{\Upsilon}_\gamma$ and $\mathbf{K}_{\gamma\gamma}$ are $(n_\gamma+1)\times(n_\gamma+1)$ and $n_\gamma \times n_\gamma$, these formulae for $\mathbf{Q}_\gamma^{-1}$ and $|\mathbf{Q}_\gamma|$ are feasible computationally. This is an advantage over the stochastic search method of George and McCulloch (1997). Finally, in the reversible jump method, the matrices involved before and after each move only change a column and a row. Thus, it would be possible to exploit low-rank matrix update techniques to make the method more efficient.

3.3 Prediction

Given a new input vector $\mathbf{x}_*$, we now predict its label y_*. The posterior predictive distribution of y_* is

$$p(y_*|\mathbf{x}_*, \mathbf{y}) = \int p(y_*|\mathbf{x}_*, \tilde{\boldsymbol{\beta}}_\gamma, \mathbf{y}) p(\tilde{\boldsymbol{\beta}}_\gamma|\mathbf{y}) d\tilde{\boldsymbol{\beta}}_\gamma.$$

We know that this integral cannot be computed in closed form. Moreover, it is intractable to select the model which is parameterized by $\boldsymbol{\beta}_\gamma$ for prediction. An intuitive approach is to choose a model with a value of $\boldsymbol{\gamma}$ having the highest posterior probability among those $\boldsymbol{\gamma}$ that appear during the MCMC sweeps. However, this is expensive in terms of memory because $\boldsymbol{\gamma}$ takes 2^n possible distinct values. To deal with this problem, we use a Bayesian model averaging method (Raftery et al., 1997) based on the MCMC algorithm. Specifically, we have

$$\frac{1}{T}\sum_{t=1}^{T} p\Big(y_* = 1\Big|\mathbf{y}, \mathbf{x}_*, u^{(t)}, \boldsymbol{\beta}_\gamma^{(t)}\Big).$$

Here $(\cdot)^{(t)}$ is the tth MCMC realization of $(\cdot)$, which is taken at every Mth sweep after the burn-in of the MCMC algorithm. In the following experiments we set $M = 5$.

4 Experimental Evaluations

In this section, we conducted several experiments to evaluate the performance of our proposed Bayesian classification method, called FBGKM. For the sake of clarity, we only considered the binary classification problems and compared FBGKM with several related classification methods. All experiments have been implemented in Matlab on a Pentium 4 with a 2.80GHz CPU and 2.00GB of RAM.

Table 1: Summary of the Benchmark Datasets: n—the size of the training data; m—the size of the test data; p—the dimension of the input vector; $k_{\max}$—the maximum number of active vectors

Datasets	n	m	p	$k_{\max}$
BCI	300	100	117	100
g241d	300	1200	241	200
Digit1	300	1200	241	200
COIL_2	300	1200	241	200
USPS (0 vs.1)	500	1500	256	200
USPS (0 vs.9)	500	1500	256	200
Letters (A vs.B)	300	1255	16	100
Letters (A vs.C)	300	1225	16	100
NewsGroups	500	1485	893	200
Ringnorm	400	7000	20	400
Thyroid	140	75	5	140
Twonorm	400	7000	20	400
Waveform	400	4600	21	400

We performed the experiments on several benchmark datasets: *BCI*, *g241d*, *Digit1*, *COIL*$_2$, *USPS digits* {(*0 vs.1*), (*0 vs.9*)}, *Letters* {(*A vs.B*), (*A vs.C*)}, *NewsGroups corpora*, *Ringnorm*, *Thyroid*, *Twonorm*, and *Waveform*. These datasets are available at `http://www.kyb.tuebingen.mpg.de/ssl-book/` and `http://ida.first.gmd.de/~raetsch/data/benchmarks.htm`.

Table 1 given a summary of these datasets. In our experiments, each dataset was randomly partitioned into two disjoint subsets as training and test datasets. Twenty random partitions were employed for each dataset, and several evaluation criteria were reported, including average classification error rate, standard deviation, and average computational time.

We implemented the methods using the RBF Gaussian kernel with a single parameter, that is, $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\sum_{l=1}^{p}(x_{il}-x_{jl})^2/\theta^2\right)$. The value of θ was set as the mean Euclidean distance between training data points. This setting was empirically found to be effective in real-world applications. In addition, we set the hyperparameters in FBGKM as $a_\eta = 1$, $b_\eta = 0.1$, $a_g = 4$ and $b_g = 0.1$. For all compared Bayesian classification methods, we run each MCMC algorithm for $10,000$ sweeps, discard the first $5,000$ as the burn-in, and retain every 5th realization of parameters after the burn-in for inference and prediction. This implies that the Bayesian model averaging method works with $1,000$ ($T = (10,000-5,000)/5$) active sets.

4.1 Comparison with Bayesian Methods

Recall that the Bayesian SVM (BSVM) (Mallick et al., 2005) and the complete SVM (CSVM) (Mallick et al., 2005) are two existing Bayesian kernel methods closely related to our FBGKM. We thus compared them with FBGKM. Moreover, we only performed BSVM and CSVM in the multiple setting, owing to their effectiveness in experiments presented by (Mallick et al., 2005). For comparison, we also implemented FBGKM without Step (c) of the MCMC algorithm in Section 3.2. That is, we considered an MCMC algorithm consisting of Steps (a)-(b) where we fix $n_\gamma = n$. We denoted the resulting model by BGKM to distinguish it from FBGKM.

We conducted the comparison on the first nine datasets in Table 1. Table 2 reports the experimental results. From this table, we can see that the FBGKM has appealing computational advantages over other three methods due to its sparse properties, and that the FBGKM and BGKM methods based on the Silverman g-prior achieve slightly lower classification error rates than the other two methods on the whole. Moreover, FBGKM and BGKM achieve roughly similar classification error rates on all datasets involved here, while FBGKM is more efficient than BGKM.

In the following experiments, we attempted to analyze the performance of the methods with respect to the change of training size n and of maximum number k_{max} of active vectors. For the sake of simplicity, we only reported the results on the *NewsGroups* dataset.

Table 3 showed the experimental results when changing training size n and fixing the maximum number of active vectors as $k_{max} = 200$. Note that the FBGKM and BGKM methods slightly outperform BSVM and CSVM in both classification error rate and computational cost. Note also that the scaling of the computational cost of the FBGKM with respect to n is relatively favorable.

Table 4 presents experimental results for the FBGKM for different values of the maximum number k_{max} of active vectors and for a fixed training size $n = 800$. We see that the performance of the FBGKM is relatively insensitive to the value of k_{max}. Note also that its computational cost tends to increase slightly when k_{max} increases.

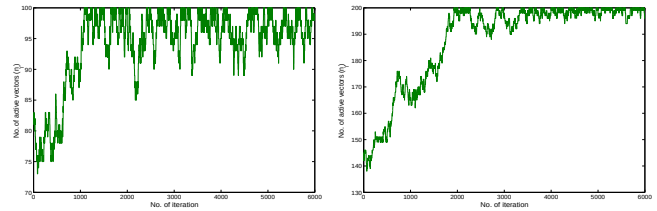
Finally, in order to study the MCMC mixing performance of our FBGKM method, we also reported the numbers of active vectors on different datasets. In Figure 1 we show the value n_γ for to the first 6000 sweeps of MCMC on *BCI*, *Digit1*, *Letters* {(*A vs.B*)} and *NewsGroups* datasets. These results suggest that the model yields reasonably fast mixing, although further study of mixing is needed on a wider range of problems.

Table 2: Experimental results of the five methods on different datasets: $err-$ the test error rates(%); $std-$ the corresponding standard deviation; $time-$ the corresponding computational time (s).

Dataset	BSVM		CSVM		BGKM		FBGKM	
	$err(\pm std)$	$time$	$err(\pm std)$	$time$	$err(\pm std)$	$time$	$err(\pm std)$	$time$
BCI	28.15(±2.15)	2.615×10^3	29.40(±2.58)	2.596×10^3	29.35(±2.82)	1.063×10^3	29.83(±2.36)	0.688×10^3
g241d	17.15(±1.68)	4.339×10^3	17.63(±1.15)	4.365×10^3	16.37(±1.11)	1.819×10^3	16.30(±0.89)	1.451×10^3
Digit1	4.86(±0.74)	5.248×10^3	4.88(±0.75)	5.210×10^3	4.87(±0.65)	2.459×10^3	4.85(±0.67)	2.011×10^3
$COIL_2$	9.71(±0.81)	4.988×10^3	9.86(±0.71)	4.996×10^3	9.16(±0.99)	2.454×10^3	9.797(±0.32)	1.502×10^3
USPS(0 vs.1)	0.40(±0.30)	2.133×10^4	0.35(±0.11)	2.047×10^4	0.28(±0.05)	6.013×10^3	0.28(±0.06)	2.700×10^3
USPS(0 vs.9)	1.36(±0.36)	2.239×10^4	1.40(±0.29)	2.230×10^4	1.36(±0.28)	6.479×10^3	1.37(±0.24)	2.974×10^3
Letters(A vs.B)	0.92(±0.59)	2.009×10^3	0.95(±0.45)	2.007×10^3	0.75(±0.24)	0.914×10^3	0.77(±0.24)	0.593×10^3
Letters(A vs.C)	0.83(±0.15)	2.026×10^3	0.93(±0.27)	2.042×10^3	0.87(±0.15)	0.896×10^3	0.84(±0.15)	0.596×10^3
NewsGroups	5.62(±0.80)	2.286×10^4	5.08(±0.33)	2.291×10^4	4.92(±0.28)	6.270×10^3	4.83(±0.25)	2.910×10^3

Table 3: Experimental results of the five methods correspond to different training sizes n on the NewsGroups dataset with $k_{max} = 200$: $err-$ the test error rates(%); $std-$ the corresponding standard deviation; $time-$ the corresponding computational time (s).

training size n	BSVM		CSVM		BGKM		FBGKM	
	$err(\pm std)$	$time$	$err(\pm std)$	$time$	$err(\pm std)$	$time$	$err(\pm std)$	$time$
n=300	5.99(±1.44)	5.949×10^3	5.84(±0.80)	5.830×10^3	5.37(±0.52)	2.467×10^3	5.08(±0.49)	2.085×10^3
n=400	5.65(±0.98)	1.173×10^4	5.83(±0.93)	1.171×10^4	5.10(±0.35)	4.674×10^3	5.05(±0.39)	2.804×10^3
n=500	5.62(±0.80)	2.286×10^4	5.08(±0.33)	2.291×10^4	4.92(±0.28)	6.270×10^3	4.83(±0.25)	2.910×10^3
n=600	5.77(±0.61)	3.458×10^4	5.13(±0.20)	3.461×10^4	4.92(±0.43)	8.340×10^3	4.74(±0.28)	2.973×10^3
n=700	5.63(±0.82)	5.195×10^4	4.82(±0.21)	5.186×10^4	4.44(±0.36)	1.207×10^4	4.61(±0.52)	3.610×10^3
n=800	5.14(±0.59)	7.754×10^4	5.10(±0.16)	7.757×10^4	4.49(±0.47)	1.673×10^4	4.56(±0.34)	4.327×10^3



(a) $k_{max} = 100$ (b) $k_{max} = 100$

(c) $k_{max} = 200$ (d) $k_{max} = 200$

Figure 1: MCMC output for the numbers n_γ of active vectors of the FBGKM method on four datasets: (a) BCI; (b) Digit1; (c) Letters (A vs.B); (d) NewsGroups.

Table 4: Experimental results of our FBGKM correspond to different maximum numbers k_{max} of active vectors on the NewsGroups dataset with $n = 800$: $err-$ the test error rates(%); $std-$ the corresponding standard deviation; $time-$ the corresponding computational time (s).

training size n	FBGKM	
	err $(\pm std)$	$time$
$k_{max} = 300$	4.55 (±0.46)	6.522×10^3
$k_{max} = 400$	4.62 (±0.45)	7.189×10^3
$k_{max} = 500$	4.64 (±0.37)	8.536×10^3
$k_{max} = 600$	4.75 (±0.48)	1.033×10^4
$k_{max} = 700$	4.72 (±0.28)	1.170×10^4

4.2 Bayesian vs. Frequentist

Since the FBGKM method is a Bayesian alternative to the IVM and SVM, it is useful to compare FBGKM with the conventional IVM and SVM. We have done this on the following datasets: *Ringnorm*, *Thyroid*, *Twonorm* and *Waveform*. These datasets were also used by Zhu and Hastie (2005) and detailed information on the data can be found in (Rätsch et al., 2001). Here each dataset was randomly partitioned into two disjoint subsets as training and test datasets according to the training and test sizes n and m in Table 1. In addition, the maximum number k_{max} of active vectors was set based on Table 1. The results shown in Table 5 were based on the average of these twenty realizations and the results with the conventional IVM and SVM are cited from Zhu and Hastie (2005). From Table 5, we can see that our Bayesian approach slightly out-

Table 5: Classification error rates (%) and corresponding standard deviations of the compared methods on the four datasets.

Datasets	SVM	IVM	FBGKM
Ringnorm	2.03(±0.19)	1.97(±0.29)	1.51(±0.10)
Thyroid	4.80(±2.98)	5.00(±3.02)	4.60(±2.65)
Twonorm	2.90(±0.25)	2.45(±0.15)	2.86(±0.21)
Waveform	9.98(±0.43)	10.13(±0.47)	9.80(±0.31)

performs the frequentist approaches.

5 Conclusion

In this paper we have proposed fully Bayesian kernel methods based on the Silverman g-prior and the Bayesian model averaging method. We have developed an MCMC algorithm for parameter estimation, model selection and posterior prediction. Although our Bayesian methods have been devised for binary classification problems, they can be readily extended to multi-class problems. Moreover, we immediately obtain a fully Bayesian approach to solving the SVM model selection problem by following the treatment of Mallick et al. (2005), who form a conditional likelihood from the hinge loss (see also Sollich (2001)), and assigning the mixture of the point-mass distribution and the Silverman g-prior to the regression vector.

Acknowledgements

Zhihua Zhang acknowledges support from Chinese Universities Scientific Fund and from Doctoral Program of Specialized Research Fund of Chinese Universities.

References

Albert, J. H. and S. Chib (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association 88*(422), 669–679.

Bernardo, J. M. and A. F. M. Smith (1994). *Bayesian Theory*. New York: John Willey and Sons.

Figueiredo, M. (2003). Adaptive sparseness for supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence 25*(9), 1150–1159.

George, E. I. and R. E. McCulloch (1997). Approaches for Bayesian variable selection. *Statistica Sinica 7*, 339–374.

Green, P. J. (1985). Discussion of Dr. Silverman's paper. *Journal of the Royal Statistical Society, Series B 47*(1), 29.

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

Holmes, C. C. and L. Held (2006). Bayesian auxiliary variable models for binary and multinomial regression. *Bayesian Analysis 1*(1), 145–168.

Kohn, R., M. Smith, and D. Chan (2001). Nonparametric regression using linear combinations of basis functions. *Statistics and Computing 11*, 313–322.

Mallick, B. K., D. Ghosh, and M. Ghosh (2005). Bayesian classification of tumours by using gene expression data. *Journal of the Royal Statistical Society Series B 67*, 219–234.

Nott, D. J. and P. J. Green (2004). Bayesian variable selection and the Swendsen-Wang algorithm. *Journal of Computational and Graphical Statistics 13*(1), 1–17.

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

Rätsch, G., T. Onoda, and K. Müller (2001). Soft margins for Adaboost. *Machine Learning 42*, 287–320.

Sha, N., M. Vannucci, M. G. Tadesse, P. J. Brown, I. Dragoni, N. D. T. C. Roberts, A. Contestabile, M. Salmon, C. Buckley, and F. Falciani (2004). Bayesian variable selection in multinomial probit models to identify molecular signatures of disease stage. *Biometrics 60*, 812–819.

Silverman, B. W. (1985). Some aspects of the spline smoothing approach to non-parametric regression curve fitting (with discussion). *Journal of the Royal Statistical Society, Series B 47*(1), 1–52.

Smith, M. and R. Kohn (1996). Nonparametric regression using Bayesian variable selection. *Journal of Econometrics 75*, 317–344.

Sollich, P. (2001). Bayesian methods for support vector machines: evidence and predictive class probabilities. *Machine Learning 46*, 21–52.

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

Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research 1*, 211–244.

Wahba, G. (1990). *Spline Models for Observational Data*. Philadelphia: SIAM.

West, M. (2003). Bayesian factor regression models in the "large p, small n" paradigm. In J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith, and M. West (Eds.), *Bayesian Statistics 7*, pp. 723–732. Oxford University Press.

Zellner, A. (1986). On assessing prior distributions and Bayesian regression analysis with $g-$prior distributions. In P. K. Goel and A. Zellner (Eds.), *Bayesian Inference and Decision Techniques: Essays in Honor of Bruno de Finetti*, pp. 233–243. Amsterdam: North-Holland.

Zhang, Z. and M. I. Jordan (2006). Bayesian multicategory support vector machines. In *the Twenty-Second Conference on Uncertainty in Artificial Intelligence (UAI)*.

Zhang, Z., M. I. Jordan, and D.-Y. Yeung (2008). Posterior consistency of the silverman g-prior in bayesian model choice. In *Neural Information Processing Systems (NIPS) 22*.

Zhu, J. and T. Hastie (2005). Kernel logistic regression and the the import vector machines. *Journal of Computational and Graphical Statistics 14*(1), 185–205.