
Nonparametric Bayesian Kernel Models

Anonymous Author(s)

Affiliation

Address

email

Abstract

Kernel models for classification and regression have emerged as widely applied tools in statistics and machine learning. We discuss a Bayesian framework and theory for kernel methods, providing a new rationalization of kernel regression based on nonparametric Bayesian models. Functional analytic results ensure that such a nonparametric prior specification induces a class of functions that span the reproducing kernel Hilbert space corresponding to the selected kernel. Bayesian analysis of the model allows for direct and formal inference on the uncertain regression or classification functions. Augmenting the model with Bayesian variable selection priors over kernel bandwidth parameters extends the framework to automatically address the key practical questions of kernel feature selection. Novel, customized MCMC methods are detailed and used in example analysis. The practical benefits and modelling flexibility of the Bayesian kernel framework are illustrated in both simulated and real data examples that address prediction and classification inference with high-dimensional data.

Keywords: Dirichlet process priors; Kernel models; Reproducing kernel Hilbert space; Semi-supervised learning; Nonparametric Bayesian analysis.

1 Introduction

Kernel models for regression and classification have been used extensively in machine learning [1, 2, 3, 4]. The appeal of these models are their predictive accuracy, flexibility, and simple extension to high-dimensional data analysis. In the case of certain kernels they are universal approximators [5] and can thus approximate any square integrable function – the basis for non-parametric modeling.

The basic formulation of kernel methods is a penalized loss functional [5]

$$\hat{f} = \arg \min_{f \in \mathcal{H}} [L(f, \text{data}) + \lambda \|f\|_k^2], \quad (1)$$

where L is a loss function, \mathcal{H} is often an infinite-dimensional reproducing kernel Hilbert space (RKHS) induced by a kernel function $k(\cdot, \cdot)$, $\|f\|_k$ is the norm of a function in this space, and λ is a tuning parameter chosen to balance the trade-off between fitting errors and the smoothness of the function. The data are input-output pairs $\{(x_i, y_i)\}_{i=1}^n$ with $x_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$ (for regression) or $y_i \in \{0, 1\}$ (for binary classification). Due to the representer theorem [6] the solution of the penalized loss functional will be a kernel representation

$$\hat{f}(x) = \sum_{i=1}^n w_i k(x, x_i), \quad (2)$$

where w_i are coefficients.

A Bayesian formulation for kernel methods would provide a natural framework to further the richness and interpretability of kernel models – a program driving much research in data mining and

machine learning. Crucially it would provide a formal model of uncertainty and allow for the inference of a full posterior rather than a point estimator.

Bayesian versions of kernel models, in particular Bayesian SVMs, have been proposed before [7, 8, 9]. Here the Bayesian analysis is applied on the finite representation from equation (2). This direct adoption of (2) is not a proper model from a Bayesian perspective since the model changes with the sample size without a coherent argument.

In this paper we develop a fully Bayesian framework and theory for kernel regression and classification addressing the above issues and specifying priors over the entire RKHS. Our model also allows for variable or feature selection by placing hyperparameters on the kernel. The summary of this analysis is that our model performs as well as or better than well studied and popular methods such as SVMs and provides a formal measure of uncertainty.

2 The Bayesian kernel model

In this Section we propose a prior specification on the RKHS via priors on signed measures rather than directly on the regression function space. Consider the space of functions defined as a convolution of the kernel with a signed (Borel) measure

$$\mathcal{G} = \left\{ f \mid f(x) = \int k(x, u) d\gamma(u), \gamma \in \Gamma \right\}, \quad (3)$$

with $\Gamma(\cdot)$ as a subset of the space of signed Borel measures. Placing a prior on Γ implies a prior on \mathcal{G} . An equivalence between \mathcal{G} and \mathcal{H} exists for an appropriate choice of the prior distribution on Γ [10]. The prior specified in our paper is such a choice.

The model in (3) can be rewritten as

$$f(x) = \int k(x, u) d\gamma(u) = \int k(x, u) w(u) dF(u), \quad (4)$$

where the random signed measure $\gamma(u)$ is modeled by a random probability distribution $F(u)$ and random coefficients $w(u)$ and $F(u)$ and $\gamma(u)$ share the same support. We assume that $F = F_X$, the marginal distribution of X . This is a reasonable assumption as long as F_X and γ share the same support. A Dirichlet Process (DP) prior, $\text{DP}(\alpha, F_0)$ with mass parameter α and base measure F_0 , is used to model uncertainty about the distribution function F . A result of the conjugacy of the DP prior and marginalization over the posterior distribution is that given data $X_n = (x_1, \dots, x_n)$

$$\mathbb{E}[f \mid X_n] = (\alpha + n)^{-1} \left[\alpha \int k(x, u) w(u) dF_0(u) + \sum_{i=1}^n w(x_i) k(x, x_i) \right],$$

taking the limit $\alpha \rightarrow 0$, an uninformative prior, the above takes the same form as (2) derived via the representer theorem with $w_i = \frac{w(x_i)}{n}$.

This results the following regression model when the response y is continuous

$$y_i = f(x_i) + \varepsilon_i = w_0 + \sum_{j=1}^n w_j k(x_i, x_j) + \varepsilon_i \quad \text{for } i = 1, \dots, n, \quad (5)$$

where $\varepsilon_i \sim \text{No}(0, \sigma^2)$. In case of binary classification when $y_i = 0/1$, we have $z_i = w_0 + \sum_{j=1}^n w_j k(x_i, x_j) + \varepsilon_i$, $i = 1, \dots, n$, where z_i is a latent variable that is related to y_i by a probit link function, $\mathbf{P}(y_i = 1) = \Phi(z_i)$, with $\Phi(\cdot)$ being the standard normal distribution function and $\varepsilon_i \sim \text{No}(0, 1)$ (i.e., $\sigma^2 \equiv 1$).

To allow for variable or features selection a scale parameter is added for each coordinate in the kernel [11]

$$k_\nu(x, u) = k(\sqrt{\nu} \otimes x, \sqrt{\nu} \otimes u)$$

where $a \otimes b$ is the element-wise product of two vectors and $\nu = (\nu_1, \dots, \nu_p)$ is a p -dimensional vector with $\nu_k \in [0, \infty]$. For the linear and Gaussian kernels the parameterization is

$$k_\nu(x, u) = \sum_{k=1}^p \nu_k x_k u_k,$$

$$k_\nu(x, u) = \exp\left(-\sum_{k=1}^p \nu_k (x_k - u_k)^2\right).$$

We next specify priors on parameters. Specifying priors over the w_i can be done by defining appropriate sample size dependent priors and addressing key questions of inducing regression shrinkage appropriately coupled to the structure of the kernel design space by using ridge regression or g-prior modeling [12]. [13] defined and exemplified the use of a flexible and practically very effective class of generalised g-priors that allow for different degrees of shrinkage estimation of regression parameters in different principal component directions on the induced design space for any regression model, and we adopt that strategy here. This is particularly relevant when dealing with many covariates, as it provides an ability to “shrink away” the effects of many irrelevant component dimensions while highlighting those of predictive value. This class of priors explicitly models the distribution $p(w|K)$, so that the sample size dependence is directly induced and the class of priors adapts as the sample size changes.

Specifically, denote by K the kernel matrix with entry $k(x_i, x_j)$. The hierarchical prior involves an eigen-decomposition of the kernel matrix, $K = F\Delta F'$. A result of the decomposition is a reparameterization of the regression into the orthogonal factors, F with $Kw = F\beta$ and $w = F\Delta^{-1}\beta$. Other prior specifications are:

$$\begin{aligned} \pi(w_0, \sigma^2) &\sim 1/\sigma^2, \quad \beta_j \sim \text{No}(0, \tau_j), \quad j = 1, \dots, m \\ \nu_j &\sim (1 - \gamma)\delta_0 + \gamma \text{Gamma}(a_\nu, a_\nu s), \quad j = 1, \dots, p \\ \tau_j &\sim \text{InvGamma}\left(\frac{a_\tau}{2}, \frac{b_\tau}{2}\right), \quad s \sim \text{Exp}(a_s), \quad \gamma \sim \text{Beta}(a_\gamma, b_\gamma) \end{aligned}$$

$a_\nu, a_s, a_\gamma, b_\gamma, a_\tau, b_\tau$ are all hyperparameters that are prespecified. (Note that in the binary response case $\sigma^2 \equiv 1$). Due to numerical stability not all n factors are included in the decomposition of the kernel matrix, that is, we include only the first m ($m < n$) columns of F .

The posterior is simulated based on a Markov chain Monte Carlo (MCMC) analysis. We list for the binary response case the sampling scheme with feature selection below. The scheme for the continuous response case can simply be obtained with minor modifications.

1. Update w_0 : w_0 is drawn from the normal posterior with mean $n^{-1}\iota'(Z - F\beta)$ and variance σ^2/n . where ι is a vector with elements all 1.
2. Update (ν, β, Z) jointly, in the following two steps.
 - 2.1. Update (ν, β) :

2.1.1. Propose ν^* : Let p_g, p_l, p_u denote the probabilities for a *global move*, *local move*, or *update move* respectively.

- For the *global move*, draw ν^* from the prior.
- For the *local move*, set $\nu^* = \nu$ then randomly pick a dimension k . If $\nu_k \neq 0$, set $\nu_k^* = 0$; otherwise draw $\nu_k^* \sim \text{Ga}(a_\nu, a_\nu s)$, the continuous part of the prior.
- For the *update move*, set $\nu^* = \nu$ and then, for all dimensions k where $\nu_k \neq 0$, draw $\nu_k^* \sim \text{Ga}(a_\nu, a_\nu s)$.

Our proposals use $p_g = .25, p_l = .5, p_u = .25$.

2.1.2. Propose β^* : Compute the proposed kernel matrix K^* with entries $k_{\nu^*}(x_i, x_j)$ and its spectral factors F^* and Δ^* . Set $\hat{Z} \equiv (\hat{z}_1, \dots, \hat{z}_n)' = w_0 + F^*\beta$ and simulate Z^* via, for each $i = 1, \dots, n$,

$$z_i^* \sim \begin{cases} N(\hat{z}_i, 1)^+, & \text{if } z_i = 1, \\ N(\hat{z}_i, 1)^-, & \text{if } z_i = 0. \end{cases}$$

Then, propose $\beta^* \sim N(b^*, V)$ where $V = \text{diag}(V_1, \dots, V_m)$ with $V_i = \tau_i/(1 + \tau_i)$, and $b^* = VF^{*'}(Z^* - w_0)$.

2.1.2. Acceptance ratio to compare and test (ν^*, β^*) against the current values (ν, β) : The Metropolis-Hastings acceptance ratio is

$$r = \frac{p(Y | \nu^*, \beta^*, w_0) \pi(\nu^*, \beta^* | s, \gamma) q(\nu, \beta | T, w_0, s, \gamma)}{p(Y | \nu, \beta, w_0) \pi(\nu, \beta | s, \gamma) q(\nu^*, \beta^* | T, w_0, s, \gamma)}$$

where the terms $p(Y | \dots)$ are binary likelihood evaluations

$$p(Y | \nu, \beta, w_0) = \prod_{i=1}^n \Phi(\mu_i)^{y_i} (1 - \Phi(\mu_i))^{1-y_i}, \quad \mu_i = w_0 + \sum_{j=1}^n w_j k(x_i, x_j)$$

and $\pi(\cdot), q(\cdot)$ denote the prior distribution function and proposal distribution function, respectively. With probability $\min\{r, 1\}$ accept the proposed values and hence set $\nu = \nu^*$ and $\beta = \beta^*$; otherwise, retain the current values.

Denote the accepted or retained values by $\{\nu, \beta, K, F, \Delta\}$, and set $w = F\Delta^{-1}\beta$.

2.2. Update Z : $\hat{Z} = w_0 + F\nu\beta$ and resample Z via, for each $i = 1, \dots, n$,

$$z_i \sim \begin{cases} N^+(\hat{z}_i, 1), & \text{if } z_i = 1, \\ N^-(\hat{z}_i, 1), & \text{if } z_i = 0. \end{cases}$$

where N^+ and N^- denote the positive and negative parts of a truncated normal.

3. Update hyper-parameters: (s, γ, T)

3.1. Update s : $s \sim \text{Ga}(a_\nu + 1, a_s + a_\nu \sum \nu_k)$.

3.2. Update γ : $\gamma \sim \text{Be}(a_\gamma + p_1, b_\gamma + p - p_1)$ where p_1 is the number of nonzero elements in ν .

3.3. Update T : For $j = 1, \dots, m$, $\tau_j^{-1} \sim \text{Ga}((a_\tau + 1)/2, (b_\tau + \beta_j^2)/2)$.

Given a new input x^* we provide posterior probabilities of its label $y^* = 1$ as $\Phi(z^{*(d)}) = \Phi(w_0^{(d)} + \sum_{j=1}^n k_{\nu^{(d)}}(x^*, x_j) w_j^{(d)})$, where $\{z^{*(d)}, w_0^{(d)}, \nu^{(d)}, w_j^{(d)}\}$ are samples $d = 1, \dots, D$ from the Markov chain.

3 Examples

3.1 Synthetic Data Sets

A simulated example considers binary classification with variable selection, using two synthetic data sets to illustrate different aspects of the model. For the MCMC in this subsection, we used 5000 iterations including an initial 2500 iterations for burn-in.

The first data set is in \mathbb{R}^{30} but only the first two dimensions influence the classification. The x data for two classes are sampled from Gaussian mixture models with

$$\begin{aligned} (x|y=0) &\sim 0.5N(\mu_{01}, \Sigma) + 0.5N(\mu_{02}, \Sigma) \\ (x|y=1) &\sim 0.5N(\mu_{11}, \Sigma) + 0.5N(\mu_{12}, \Sigma) \end{aligned}$$

where $\Sigma = \text{diag}(.38, .38, 1, \dots, 1)$ and $\mu_{01} = (1, 1, 0, \dots, 0)$, $\mu_{02} = (-1, -1, 0, \dots, 0)$, $\mu_{11} = (-1, 1, 0, \dots, 0)$ and $\mu_{12} = (1, -1, 0, \dots, 0)$. We drew 30 samples from each class as training data, and a further 100 samples from each class as test data. The data on the first two x dimensions are plotted in Figure 1.

To provide an initial, baseline comparison, we fitted a binary model, with a Gaussian kernel on the first two dimensions and $\nu = (1.5, 1.5)$, to the test data alone; no feature selection was used here. The value of ν was chosen to be the one that produces the smallest test error. In terms of posterior means of the resulting classification probabilities, the resulting test error was .5% (only one sample being misclassified). The predictive probability of $(y_* = 1 | x_*)$ with respect to the first two dimensions of x_* is displayed in Figure 2(a).

We compared the kernel model analysis with and without variable selection to this baseline kernel model. For the kernel model without variable selection, we set the bandwidth parameter to be

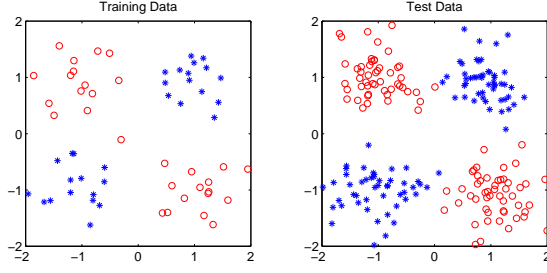


Figure 1: Synthetic data set 1. Scatter plot of the training data (60 observations) and the test data (200 observations) on the first two dimensions, with cases $y_i = 0$ in blue stars and $y_i = 1$ in red circles.

constant in all dimensions, $\nu = (\nu, \dots, \nu)$. For a variety of choices of ν the test error never fell below 33.5% and the training error was 0. This poor performance is illustrated in the prediction plot in Figure 2(b), where we plot $(y_* = 1 | x_*)$ again with respect to the first two dimensions of x_* . We then applied the kernel model with variable selection to this data with hyper-parameters

$$a_\tau = b_\tau = 2, a_\gamma = b_\gamma = 5, a_\rho = 1, a_s = 1, m = 5. \quad (6)$$

The test error of .5% was comparable to the “optimal” model results as in Figure 2(a); the prediction plot in Figure 2(c) shows the efficacy of the variable selection component of the analysis in honing in on the truly predictive variables and adapting the non-linear predictive model appropriately.

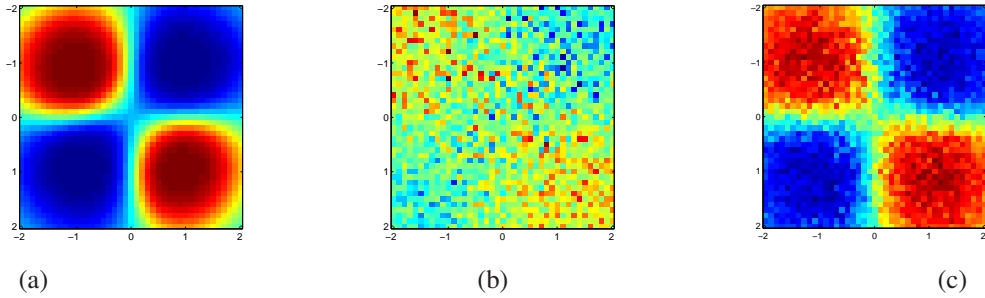


Figure 2: Synthetic data set 1. The color images represent the posterior predictive probability $\Pr(y_* = 1 | x_*, \text{data})$ when the first two dimensions of x_* varies, coded such that the predictive probability of $y_* = 1$ increases from near 0 (blue) to near 1 (red). (a) Only the first two dimensions of the data are used in the classification model and the hyper-parameters are optimized with respect to the test error. (b) All 30 dimensions are used in a kernel classification model without variable selection. (c) All 30 dimensions are used in a kernel classification model with variable selection.

The second synthetic data set is analysed to explore variable selection further as well as to provides a sense of scale for each of the x variables. The data set is in \mathbb{R}^{20} but only the first two dimensions are relevant. The two classes are sampled from Gaussian mixture models with

$$\begin{aligned} (x|y = 0) &\sim \frac{1}{3}N(\mu_{01}, \Sigma) + \frac{1}{3}N(\mu_{02}, \Sigma) + \frac{1}{3}N(\mu_{03}, \Sigma) \\ (x|y = 1) &\sim \frac{1}{3}N(\mu_{11}, \Sigma) + \frac{1}{3}N(\mu_{12}, \Sigma) + \frac{1}{3}N(\mu_{13}, \Sigma) \end{aligned}$$

where $\Sigma = \text{diag}(.38, .38, 1, \dots, 1)$ and $\mu_{01} = (-1, -1, 0, \dots, 0)$, $\mu_{02} = (0, 1, 0, \dots, 0)$, $\mu_{03} = (1, -1, 0, \dots, 0)$, $\mu_{11} = (-1, 1, 0, \dots, 0)$ and $\mu_{12} = (0, -1, 0, \dots, 0)$, $\mu_{13} = (1, 1, 0, \dots, 0)$. The

training data consist of 45 points from each class. The first two dimensions are plotted in Figure 3(a). This plot as well as the generative distributions suggest that the first two dimensions should scale differently and this should be reflected in posterior draws of the corresponding bandwidth parameters ν_1, ν_2 . Specifically, we should expect $\nu_1/\nu_2 \approx 2/3$. We applied the kernel model with variable selection to this data with the same hyper-parameter values. Figure 3(b) displays the predictive probability as a function of the first two relevant variables. Figure 3(c) displays the 90% credible interval for ν_k for $k = 1 : 20$. Examining the posterior distribution of the elements of ν we found that $P(\nu_1 \neq 0 \mid \text{data}) = P(\nu_2 \neq 0 \mid \text{data}) = 1$ and $P(\nu_k = 0 \mid \text{data}) \geq 86\%$ for the irrelevant dimensions $k = 3, \dots, 20$ where P stands for the empirical posterior probability estimated from the MCMC outputs. Meanwhile, the posterior mean and median are 3.01, 2.84 for ν_1 and 1.78, 1.42 for ν_2 . This illustrates how the analysis is capable of inferring appropriate scales of variables in addition to their relative inclusion probabilities.

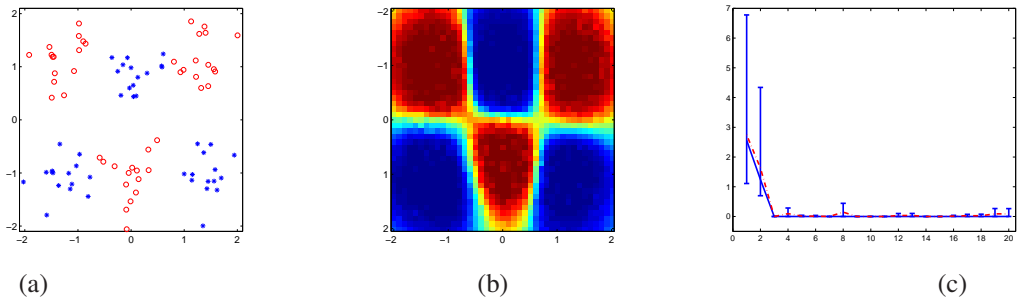


Figure 3: Synthetic data set 2. (a) Scatter plot of test data on the first two dimensions with cases $y_i = 0$ in blue stars and $y_i = 1$ in red circles. (b) Plot of the posterior predictive $\Pr(y_* = 1 | x_*, \text{data})$ as the first two dimensions of x_* varies. (c) Credible interval plot for ν_k for $k = 1, \dots, 20$. The blue solid line indicates the posterior median and the red dashed line indicates the posterior mean for each dimension.

3.2 Real Data: The MNIST Data Set

A standard data set used in the machine learning community is the MNIST data set¹. This data set contains 60,000 images of handwritten digits $\{0, 1, 2, \dots, 9\}$, where each image consists of $p = 28 \times 28 = 784$ gray-scale pixel intensities.

We considered all pairwise comparisons among the 10 different digits resulting in 45 binary classification problems. For each classification problem we randomly selected 50 training samples from each class as training data and 50 samples from each class as test data. This was repeated 5 times and the average test error was computed.

Since the 784 pixels in the image are strongly correlated we pre-processed the data by projecting the training and test data onto the first 50 principle components computed on the training data. We then applied the kernel model analysis twice – with and without variable selection. We used a linear kernel and the same hyper-parameter values as above with the exception that we restricted to m kernel principal components and the reported analysis summaries are based on choosing m to optimize 5-fold cross-validation classification errors within the training data set in each analysis. Note that for linear kernel model without variable selection, using $\nu = (\nu, \nu, \dots, \nu)$ is equivalent to setting $\nu = 1$. We ran the MCMC for 5000 iterations after an initial 5000 iterations for burn-in for each experiment. The results for the 45 comparisons are reported in Figure 4. The performance of the kernel model with variable selection is substantially superior to that without selection for all 45 classification problems.

We further explored variable selection by focusing on the task of classifying “3” vs “5”, one of the most challenging comparisons. We ordered the variables by their approximate posterior model inclusion probabilities averaged over the 5 repeat experiments. Due to the image processing underlying the raw data, each variable is not precisely a single pixel from the original image; rather, it is a locally-weighted linear combination of all 784 pixels. We visualize each variable by plotting

¹Available at <http://yann.lecun.com/exdb/mnist/>

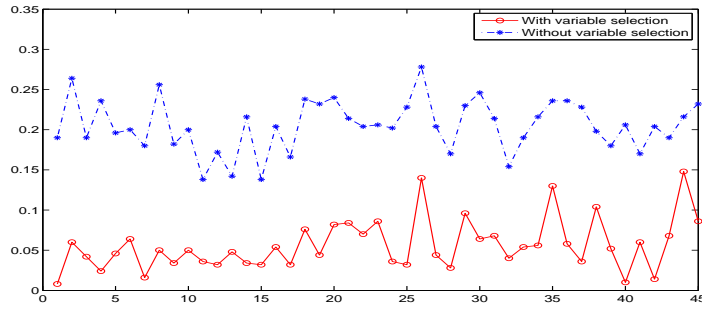


Figure 4: The MNIST data. Plot of the 45 classification errors for the kernel model with variable selection (solid line with circles) and without variable selection (dashed line with stars).

the corresponding 784 weights on the 28×28 grid. In Figure 5 we plot a few apparently relevant variables corresponding to ν_k with high posterior probabilities (upper panel), together with a few apparently irrelevant variables corresponding to ν_k with low posterior probabilities (lower panel). Visually, it is clear that the relevant variables capture geometric differences between “3” and “5”, while the irrelevant variables do not. Since the training and test data sets vary in the 5 experiments, we then randomly select a new data set with 100 samples from each class. Projections of this new data set onto sets of two relevant variables are displayed in Figure 6; similar projections onto sets of two irrelevant variables are in Figure 7. It is clear that the two classes show some separation in the relevant variables but not in the irrelevant variables.

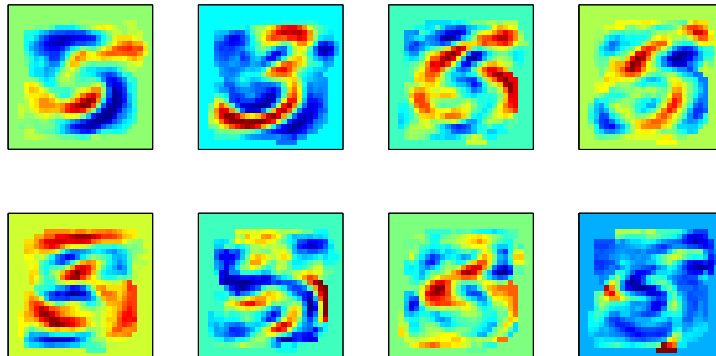


Figure 5: The MNIST data. Upper panel: plot of relevant variables (the 1st, 2nd, 4th and 5th variables). Lower panel: plot of irrelevant variables (the 3rd, 6th, 10th and 11th variables).

4 Discussion

With the growth of interest in statistical classification and prediction methods in the machine learning communities, and an escalation of interest in applications among practitioners, there is a consequent need for refined theoretical understanding of the underlying statistical models as well as improved methodology and algorithms. We address each of these issues here. The theoretical foundation of our Bayesian kernel models is based on the equivalence between a class of functions induced by a nonparametric prior specification and a reproducing kernel Hilbert space. This Bayesian framework of the model allows for coherent inference, assessment of uncertainty, and access to the posterior distributions via Markov chain Monte Carlo sampling. Practical issues such as choice of hyper-parameters and variable selection are automatically incorporated into the Bayesian modeling and inference.

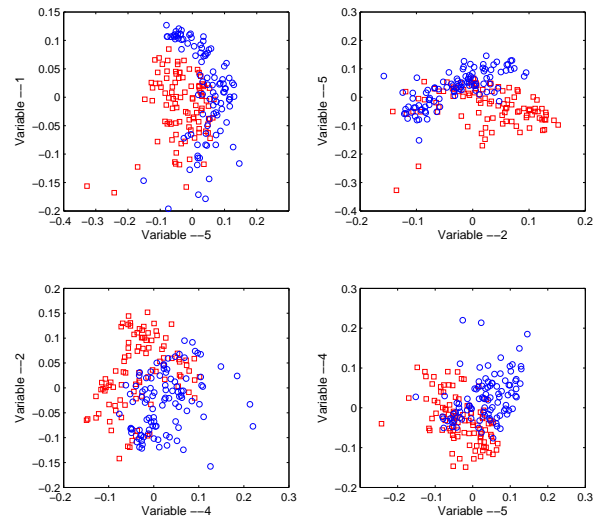


Figure 6: The MNIST data. Plot of projections onto sets of two relevant variables, where circle represents “3” and square represents “5”. The two classes show some separation in the relevant variables.

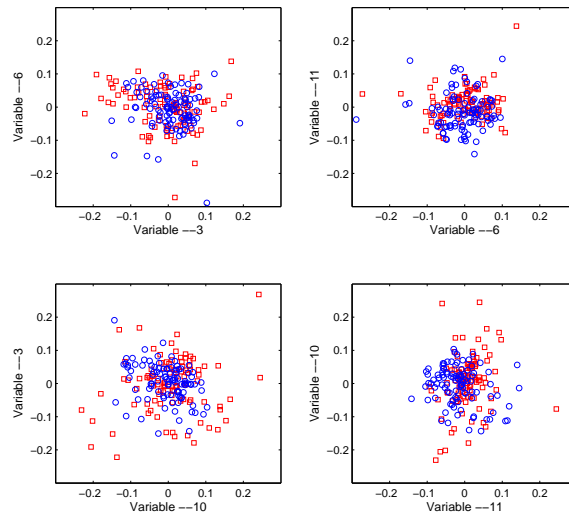


Figure 7: The MNIST data. Plot of projections onto sets of two irrelevant variables, where circle represents “3” and square represents “5”. The two classes are mixed in the irrelevant variables.

References

- [1] T. Poggio and F. Girosi. Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, 247:978–982, 1990.
- [2] V.N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.
- [3] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support vector machines, regularization, optimization, and beyond*. The MIT Press, Cambridge, 2001.
- [4] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge Univ. Press, Cambridge, 2004.
- [5] Theodoros Evgeniou, Massimiliano Pontil, and Tomaso Poggio. Regularization Networks and Support Vector Machines. *Advances in Computational Mathematics*, 13:1–50, 2000.
- [6] G.S. Kimeldorf and G. Wahba. A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *Ann. Math. Statist.*, 41(2):495–502, 1971.
- [7] Michael E. Tipping. Sparse Bayesian learning and the relevance vector machine. *J. Mach. Learn. Res.*, 1:211–244, 2001.
- [8] S. Chakraborty, M. Ghosh, and B.K. Mallick. Bayesian non-linear regression for large p small n problems. Preprint is available at <http://www.stat.ufl.edu/schakrab/svmregression.pdf>, 2005.
- [9] P. Sollich. Bayesian methods for support vector machines: Evidence and predictive class probabilities. *Machine Learning*, 46:21–52, 2002.
- [10] N. Pillai, Q. Wu, F. Liang, S. Mukherjee, and R. Wolpert. Characterizing the function space for Bayesian kernel models. ISDS Discussion Paper Series 2006-18, Duke University, Institute of Statistics and Decision Sciences, 2006. <http://ftp.stat.duke.edu/WorkingPapers/06-18.html>.
- [11] O. Chapelle, V.N. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1-3):131–159, 2002.
- [12] A. Zellner. Bayesian estimation and prediction using asymmetric loss functions. *J. Amer. Statist. Assoc.*, 81:446–451, 1986.
- [13] M. West. 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, editors, *Bayesian Statistics 7*, pages 723–732. Oxford University Press, Oxford, 2003.