Stat Comput manuscript No.
(will be inserted by the editor)

Bayesian Kernel Projections for Classification of High Dimensional Data

Katarina Domijan · Simon P. Wilson

Received: date / Accepted: date

Abstract A Bayesian multi-category kernel classification method is proposed. The algorithm performs the classification of the projections of the data to the principal axes of the feature space. The advantage of this approach is that the regression coefficients are identifiable and sparse, leading to large computational savings and improved classification performance. The degree of sparsity is regulated in a novel framework based on Bayesian decision theory. The Gibbs sampler is implemented to find the posterior distributions of the parameters, thus probability distributions of prediction can be obtained for new data points, which gives a more complete picture of classification. The algorithm is aimed at high dimensional data sets where the dimension of measurements exceeds the number of observations. The applications considered in this paper are microarray, image processing and near-infrared spectroscopy data.

Keywords Bayesian inference · multinomial logistic regression · reproducing kernel Hilbert spaces · kernel principal components analysis · Bayesian decision theory

K. Domijan

Mathematics Department, NUI Maynooth, Ireland

Tel.: +353-1-7083374

Fax: +353-1-7083913

E-mail: Katarina.Domijan@maths.nuim.ie

S. P. Wilson

Statistics Department, Trinity College Dublin, Ireland

1 Introduction

Supervised learning for classification can be formalized as the problem of inferring a function $f(\mathbf{x})$ from a set of n training samples $\mathbf{x}_i \in \mathbb{R}^J$ and their corresponding class labels $\mathbf{y}_i$. The model developed in this paper is aimed at multi-category classification problems. Of particular interest is classification of high dimensional

data, where each sample is defined by hundreds or thousands of measurements, usually concurrently obtained. Such data arise in many application domains, for example, the genomic and proteomic technologies, and their rapid emergence in the last decade has generated much interest in the statistical community, as analysis of such data requires novel statistical techniques. The applications considered in this paper are microarray, image processing and near-infrared (NIR) spectroscopy data where the dimension of the variables J exceeds ten to twenty - fold the number of samples n.

In this paper we consider classifiers based on the reproducing kernel Hilbert spaces (RKHS) theory. RKHS methods allow for nonlinear generalization of linear classifiers by implicitly mapping the classification problem into a high dimensional feature space where the data is thought to be linearly separable. Due to the reproducing property of the RKHS, the classification is actually carried out in the subspace of the feature space which is of dimension $n << J$. Therefore kernel methods are especially useful for high dimensional data, such as the data sets considered here.

A drawback of the RKHS models is that they can become over-parameterized if n regression coefficients need to be estimated when n samples are available. Therefore, these parameters are not identifiable in the statistical sense, i.e. different combinations of non - identifiable parameters lead to the same likelihood. This, in Bayesian framework, results in a posterior distribution that is multimodal, even if sparse priors are placed on regression parameters. Furthermore, choosing a globally optimal subset of regression coefficients out of 2^n subsets is tricky as many combinations of the parameters yield the same result. Bayesian framework allows for arbitrarily complex models to be specified, however inferences based on overparameterized models are not always legitimate as MCMC samplers mix poorly and maximum a posteriori (MAP) estimates are suboptimal.

In this paper, a RKHS classifier is constructed that performs the classification of the projections of the data to the principal axes of the feature space. Thus, the sparsity is achieved by removing the principal axes with zero-eigenvalues. The degree of sparsity can be further regulated in a Bayesian decision theoretic framework, where the optimal model maximizes the expected utility function with respect to all the unknowns, including the model parameters and future data. The decision space is discrete and of dimension $\leq n$, therefore it is possible to do an exhaustive search and stochastic search algorithms are not required. A sparse model of uncorrelated principal axes requires estimating a small number of identifiable regression coefficients, which simplifies the convergence and optimization issues. This

approach to sparsity is computationally efficient and we argue that it is simpler than estimating MAPs from multimodal n-dimensional posterior. In addition, we show that computational savings and improved classification performance can be achieved if the underlying structure of the feature space can be adequately summarized by a small subset of the principal axes.

Kernel methods were first introduced into statistical learning by [Aizerman et al., 1964] and later reintroduced by [Boser et al., 1992] who constructed the Support Vector Machine, a generalization of the optimal hyperplane algorithm for binary classification. Bayesian treatments of this popular deterministic statistical learning method were motivated by the need to overcome the problem of quantifying uncertainty of SVM predictions, as Bayesian framework allows for probabilistic outputs to be obtained from the predictive distribution.

Many Bayesian treatments of deterministic kernel methods have been developed, but only a subset of most relevant approaches are discussed here. [Sollich, 2002, Seeger, 2000, Opper and Winther, 2000, Herbrich et al., 1999, Kwok, 1999] use Gaussian process priors to SVM classification models. For other basis function models that have been fitted in Bayesian framework via Gaussian processes see [Neal, 1996, 1998, Williams and Barber, 1998, Rasmussen, 1996].

The Relevance Vector Machine (RVM) [Tipping, 2000] is an alternative Bayesian formulation of SVM, developed for both classification and regression with the aim of obtaining a sparse solution. The sparseness is induced in the model through the prior structure; see [Tipping, 2001] for an in-depth discussion on the sparsity in RVM. Following the work of [Wahba, 1990], [Tipping, 2000] recast the SVM as regularization problem where the aim is to minimize a loss function L subject to a penalty term over a set of regression coefficients β:

$$\min_{\beta} \left[L(\mathbf{y}, \mathbf{K}\beta) + \tau \beta^T \mathbf{K} \beta \right]. \quad (1)$$

The model function, i.e. the separating hyperplane, is a linear combination of the reproducing kernels and is in the dual form:

$$f(x) = \sum_{i=1}^{n} \beta_i K(x, x_i|\theta). \quad (2)$$

[Tipping, 2000] use a binary logistic likelihood to model loss and assume a relatively standard prior structure for regression coefficients. [Figueiredo, 2003] proposed a similar model to the RVM, but uses a probit likelihood for binary classification and places double exponential priors on regression coefficients, which are known to promote sparseness [Figueiredo, 2002, Bishop and Tipping, 2000]. Note that the RVM model can be viewed as an implicit formulation of the Gaussian process, where the prior is a Gaussian process over then model functions f expressed in the primal form, i.e. as a (possibly

infinite) linear combination of the feature space bases:

$$f(x) = \sum_{p=1}^{P} c_p \phi_p(x) \tag{3}$$

where c_p are some coefficients, $P \leq \infty$ is the dimension of the feature space and $\phi_p(x)$ are the basis functions of the feature space. For a more detailed discussion see [Rasmussen, 1996].

The approach of [Figueiredo, 2003] obtains MAP estimates for the model parameters via expectation maximization algorithm. The RVM [Tipping, 2000] employs the empirical Bayes approach. [Mallick et al., 2005] adopt the same model construction and prior structure as the RVM, however, rather than estimating the hyperparameters, they assign distributions to them and employ an MCMC sampling algorithm. The practical advantage of the full probabilistic approach is that probability distributions of prediction can be obtained for new observations, which gives a more complete picture of classification. By assigning priors to the hyperparameters, the binary classifier of [Mallick et al., 2005] accounts for the uncertainty due to their estimation. In addition to the binary logistic likelihood, [Mallick et al., 2005] also consider a stochastic version of the SVM likelihood. [Zhang and Jordan, 2006] extend this model to multi-category problems by employing the stochastic version of the multi-category support vector machine of Lee et al. [2004]. Chakraborty et al. [2007] also follow the model construction and choice of prior architecture of [Mallick et al., 2005], however, employ multinomial logistic likelihood. Krishnapuram et al. [2005] extend the approach of [Figueiredo, 2003] by also employing multinomial logistic likelihood. The paper is organized as follows; Sect. 2 describes a Bayesian multi-category kernel classifier (BMKC) where the likelihood is modeled through the multinomial logistic regression model and the relatively standard hierarchical prior structure for Bayesian generalized linear models is assumed. This is a natural multi-category extension of the model of [Mallick et al., 2005] and very similar to algorithms presented in Krishnapuram et al. [2005], Zhang and Jordan [2006], Chakraborty et al. [2007]. This model is developed for illustrative purposes and is used as as a reference for further discussion and comparison to Bayesian Kernel Projection Classifier (BKPC), which is presented in Sect. 3. Section 4 outlines the variable selection approach for this algorithm which is based on Bayesian decision theory. The reduction of model complexity and the implementation advantages of this algorithm are discussed. Sect. 5 gives a brief description of the data sets used. The classification results are presented in Sect. 6 and the concluding remarks are given in Sect. 7.

2 Bayesian Multi-category Kernel Classifier (BMKC)

2.1 Multinomial Logistic Regression Model

The training data are n samples $(\mathbf{x}_1, \mathbf{y}_1)$,...,$(\mathbf{x}_n, \mathbf{y}_n)$ where the predictors $\mathbf{x}_i = (x_{i1}, ..., x_{iJ})$ are real valued J- dimensional vectors of feature values and $\mathbf{y}_i = (y_{i1}, ..., y_{iK})$ are K- dimensional categorical response variables with $y_{ik} = 1$ if $\mathbf{x}_i$ belongs to a class k and 0 otherwise. A standard approach to this classification problem is the multinomial logistic regression model given by:

$$\mathbb{P}(\mathbf{y}|\mathbf{z}) = \prod_{i=1}^{n} \prod_{k=1}^{K} \mathbb{P}(y_{ik} = 1|z_{ik})^{y_{ik}}, \tag{4}$$

where $\mathbb{P}(y_{ik} = 1|z_{ik})$ is defined as:

$$\mathbb{P}(y_k = 1|\mathbf{x}) = \frac{\exp(z_k)}{\sum_{l=1}^{K} \exp(z_l)}, \tag{5}$$

and z_{ik} are linear combinations of the kernel functions:

$$z_{ik}(\mathbf{x}_i, \beta_k, \theta) = \sum_{l=1}^{n} \beta_{kl} K(\mathbf{x}_i, \mathbf{x}_l|\theta) + \epsilon_{ik} = \mathbf{K}_i \beta_k + \epsilon_{ik}, \tag{6}$$

for $i = 1, ..., n$ where β_k are regression parameters $\beta_k = [\beta_{1k}, \beta_{2k}, ..., \beta_{nk}]$ corresponding to class k, for $k = 1, ..., K-1$. $\mathbf{K}_i$ is the i^{th} row of matrix $\mathbf{K}$ and ϵ_{ik} are i.i.d. $N(0, \sigma^2)$. In this application only Gaussian kernels are considered:

$$K(\mathbf{x}_i, \mathbf{x}_l|\theta) = \exp(-\theta||x_i - x_l||^2). \tag{7}$$

2.2 Prior Specification

In a Bayesian inference approach, priors are assigned to the model parameters. The prior model is specified as:

$$z_{ik} \sim N(\mathbf{K}_i \beta_k, \sigma^2),$$

$$\beta_k \sim MVN(0, \sigma^2 \mathbf{T}_k^{-1}),$$

$$\sigma^2 \sim IG(\gamma_1, \gamma_2),$$

$$\tau_{ik} \sim G(\gamma_3, \gamma_4).$$

$\mathbf{T}_k$ is a matrix with diagonal entries $\tau_{1k}, ..., \tau_{nk}$. G denotes a gamma prior, IG an inverse gamma and MVN is a multivariate normal of dimension n.

Note that this is a relatively standard hierarchical prior structure for generalized linear models and is used by [Mallick et al., 2005] for binary classification as well as [Chakraborty et al., 2007] for the multinomial extension. In order to improve the mixing and convergence of the MCMC algorithm, the latent variables are given a normal prior with means $\mathbf{K}_i \beta_k$ and standard deviation σ^2. This allows for direct block updating of regression coefficients from the joint conditional density [Holmes and Held, 2005, Denison et al., 2002].

2.3 Inference

A Metropolis-within-Gibbs algorithm was used for sampling from the posterior. The output from the MCMC is a set of samples ($\beta^{(m)}$, $\mathbf{z}^{(m)}$, $\sigma^{2\,(m)}$, $\tau^{(m)}$), for $m =$

$1, ..., M$ iterations, obtained from the joint posterior distribution after a period of 'burn-in' iterations. The joint posterior distribution is given by:

$$\mathbb{P}(\beta, \mathbf{z}, \tau, \sigma^2|\mathbf{y}) \propto \mathbb{P}(\mathbf{y}|\mathbf{z}, \beta, \tau, \sigma^2)\mathbb{P}(\mathbf{z}|\beta, \sigma^2) \times \mathbb{P}(\beta|\tau, \sigma^2)\mathbb{P}(\tau)\mathbb{P}(\sigma^2). \quad (8)$$

The full conditional distributions that were sampled from for each parameter in the model are given in Appendix A.

The MCMC algorithm is implemented so that it iterates through block updates of the parameters starting with $\mathbf{z}$. Each $\mathbf{z}_i = [z_{i1}...z_{i(K-1)}]$ is proposed to be updated conditionally on the rest of the parameters including the matrix $\mathbf{z}$ without the ith element. The proposal density for $\mathbf{z}_i$ is a random walk and is sampled using a Metropolis step within the Gibbs algorithm. Subsequently, parameters β, σ^2 and τ are block updated directly from their conditionals via Gibbs steps.

2.4 Practical Aspects of Implementation

The MCMC algorithm was implemented in the C programming language. The most time consuming aspect of the algorithm is the block updating of the regression parameters β from their conditional distribution $\mathbb{P}(\beta|\mathbf{z}, \tau, \sigma^2) = \prod_{k=1}^{K-1} MVN(\mathbf{m}_k, \sigma^2\mathbf{V}_k)$, where $\mathbf{m}_k = \mathbf{V}_k\mathbf{K}^T\mathbf{z}_k, \mathbf{V}_k = (\mathbf{K}^T\mathbf{K} + \mathbf{T}_k)^{-1}$. Note that at each Gibbs iteration, the update of β_k for $k = 1, ..., K-1$ involves inverting matrices of dimension $n \times n$. The fact that the matrices are symmetric can be exploited to make the computation easier by using Cholesky decomposition, which runs in time proportional to n^3, see [Thisted, 1988] or [Press et al., 1986]. The Cholesky decomposition of matrix $\mathbf{V}_k = \mathbf{L}\mathbf{L}^T$ is used to compute the determinant of $\mathbf{V}_k$, which is the square of the product of the diagonal elements of $\mathbf{L}$ and to generate vector valued samples from $MVN_{(n)}(\mathbf{m}_k, \sigma^2\mathbf{V}_k)$. If ε is a vector of components that are i.i.d. $N(0, 1)$ then $\beta_k^{(m)} = \mathbf{m}_k + \sigma\mathbf{L}\varepsilon$.

2.5 Prediction

The BMKC allows for posterior distributions to be obtained through simulation, as opposed to just maximum a posteriori (MAP) estimates, which gives a more complete picture of classification. Thus, for each new observation $\mathbf{x}^*$, the probability

$$\mathbb{P}(k|\mathbf{x}^*, \mathbf{x}, \mathbf{y}, \beta^{(m)}) = \frac{\exp(\mathbf{K}_*\beta_k^{(m)})}{1 + \sum_{q=1}^{K-1}\exp(\mathbf{K}_*\beta_q^{(m)})} \quad (9)$$

is calculated for each class $k = 1, ..., K-1$ for sets of samples $\beta^{(m)}$ from $m = 1, ..., M$ samples of the parameters from the joint posterior. Note that

$$\mathbf{K}^* = [K(\mathbf{x}^*, \mathbf{x}_1|\theta), K(\mathbf{x}^*, \mathbf{x}_2|\theta), ..., K(\mathbf{x}^*, \mathbf{x}_n|\theta)]$$

. Consider Ripley's synthetic data Ripley [1996] where each class is set to be a mixture of two Gaussians with

the optimal error rate of 0.08. There are 200 training and 1,000 testing samples. Figure 1 displays histograms of realizations from the posterior distributions $\mathbb{P}(y = 1|\mathbf{x}^*, \mathbf{x}, \mathbf{y}, \beta^{(m)})$ of predictions for four test observations from Ripley's synthetic data set. Note that this information can be particularly useful for examining borderline observations.

The MAP estimate can be obtained from the usual Monte Carlo Integration approximations:

$$\mathbb{P}(k|\mathbf{x}^*, \mathbf{x}, \mathbf{y}) \approx \frac{1}{M} \sum_{m=1}^{M} \frac{\exp(\mathbf{K}^* \beta_k^{(m)})}{1 + \sum_{q=1}^{K-1} \exp(\mathbf{K}^* \beta_q^{(m)})}, \quad (10)$$

$\forall k = 1, ..., K-1$. The result of a classification of Ripley's two-dimensional data set can be graphically displayed. The multinomial regression model obtains a classification probability surface across the domain of the training data. However the BKMC results in a set of realizations of the classification probability surfaces from the posterior density. From these realizations, it is possible to estimate the MAP classification probability surface and information about the certainty of this estimate is available. Whereas it is difficult to plot a set of overlaid surfaces $\mathbb{P}(y = 1|\mathbf{x}^*, \mathbf{x}, \mathbf{y}, \beta^{(m)})$, for some samples $m \in \{1, ..., M\}$, Figure 2 shows the classification boundary, i.e. $\mathbb{P}(y = 1|\mathbf{x}^*, \mathbf{x}, \mathbf{y}, \beta^{(m)}) = 0.5$ obtained for 25 samples of β from the posterior and the mean boundary curve.

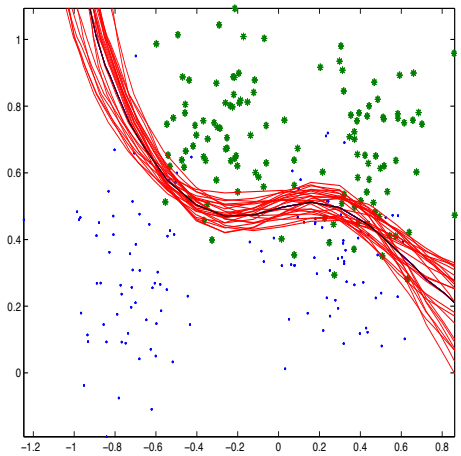

Fig. 2 Twenty-five classification boundaries from the posterior distribution, including the posterior mean boundary for the two-dimensional synthetic data set.

The classification results of the BMKC are good, obtained was the error rate of 0.098 which is comparable to results reported by Tipping [2000], and Figueiredo [2002] who obtain 0.093 and 0.095 respectively.

2.6 Mixing and Convergence Issues

The model is over-parameterized in the sense that all n reproducing kernel bases $K(x_i, \cdot)$, $i = 1, ..., n$, i.e. support vectors, are utilized, whereas only a subset might be required for a good classification model. The parameters β of the model are not identifiable; different combinations of nonidentifiable parameters lead to the same likelihood, making it impossible to decide among the potential parameter values based on the data. Large correlations among the parameters and the multi-modality in the posterior probability distribu-

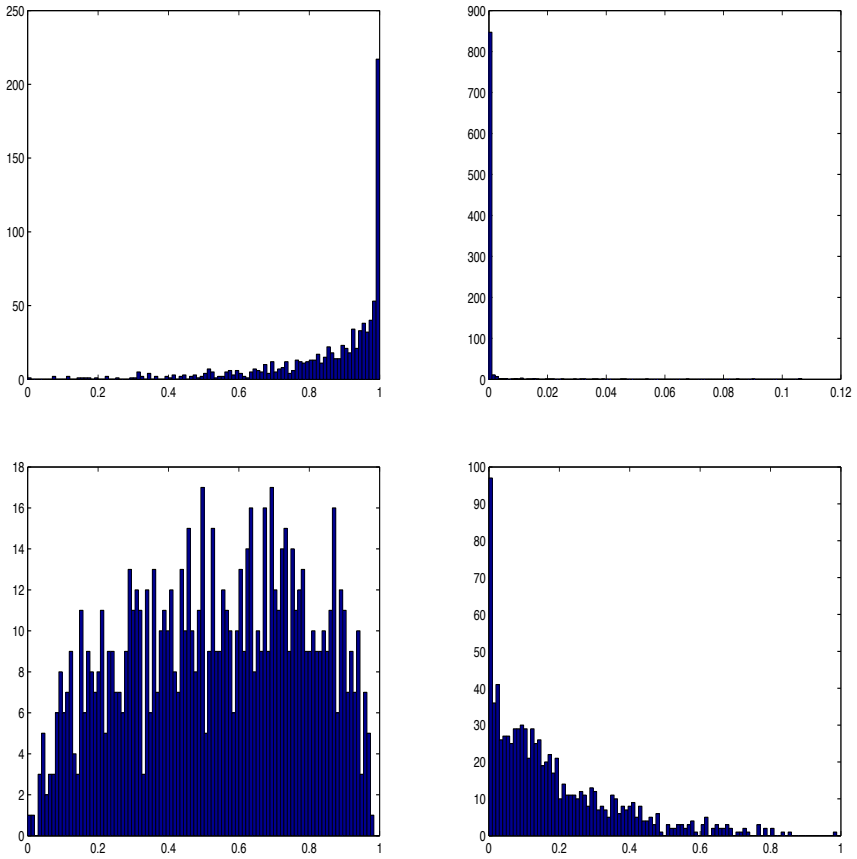

Fig. 1 Histograms of realizations from the posterior distribution of predictions $\mathbb{P}(y = 1|\mathbf{x}^*, \mathbf{x}, \mathbf{y}, \beta^{(m)})$ calculated at some $m \in \{1, ..., M\}$ for four observations from Ripley's test data set. The range of values $\mathbb{P}(y = 1|\mathbf{x}^*, \mathbf{x}, \mathbf{y}, \beta^{(m)})$ can take is between 0 and 1. The first MAP estimate for the first observations will place it in class 1, the second observation will be placed in class 2 etc.

tion result in slow convergence and poor mixing of the MCMC algorithm. Multiple runs of the MCMC with different starting values for the parameters show that the algorithm tends to get stuck in the local optima of the multimodal joint posterior and fails to explore the full support of the distribution. The different starting values of the parameters had little effect on the misclassification rates. This indicates that convergence to a good classification algorithm has been reached. However, the predictive distributions obtained through simulation discussed in Section 2.5 are no longer legitimate.

3 Bayesian Kernel Projection Classifier (BKPC)

In this section, the Bayesian Kernel Projection Classifier (BKPC) is proposed. This is a modification to BMKC, but instead of working with the data mapped to some feature space via $\Phi(\mathbf{x})$, the classification is per-

formed in the space spanned by the principal axes of the feature space. This approach works well if the underlying structure of the feature space can be adequately summarized by a small subset of the principal axes. The mapping of the data and the eigen-decomposition of the covariance matrix $Cov(\Phi(\mathbf{x}))$ is carried out implicitly via the kernel matrix. This is also the mechanism behind the Kernel Principal Components Analysis (KPCA) of [Schölkopf et al., 1998] and the data projections to the principal axes are the kernel principal components (KPCs).

KPCA maps the data $\mathbf{x}_i \in \mathbb{R}^J$ into a high dimensional feature space and then projects the mapped data $\Phi(\mathbf{x})$ to a subspace of the feature space. In the KPCA literature, the vector $\mathbf{x}_i$ is often referred to as the *pre-image* of $\Phi(\mathbf{x}_i)$. Note that, typically, the KPCA subspace will not have a pre-image in the input space. Techniques have been proposed for finding approximate pre-images of data projected on a subset of the eigenvectors, see for example [Schölkopf et al., 1999, Bakir et al., 2004].

[Schölkopf et al., 1998] and [Schölkopf and Smola, 2002] note that the first few eigenvectors of the KPCA can be used for separating clusters in two dimensional data, see, for example, the simulated data in Figure 3. They suggest extracting nonlinear principal components and then training a support vector machine, thus

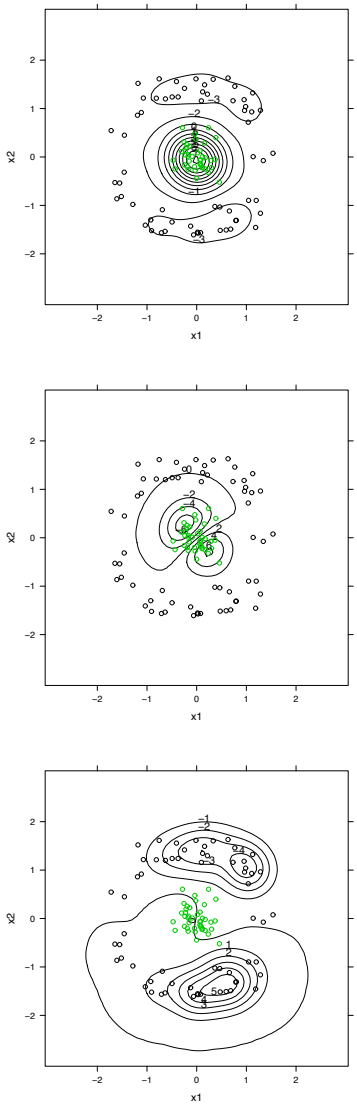

Fig. 3 A simulated dataset with the lines of constant principal component value for the first three eigenvectors (given from left to right). A Gaussian kernel with bandwidth $\theta = 5$ was used.

constructing a multi-layer SVM. The multi-layer formulation evades pre-image reconstruction, but the evident disadvantage of this algorithm is loss of interpretability as the data are mapped to a feature space twice.

The Bayesian Kernel Projection Classifier is a somewhat different approach to using KPCA to aid classi-

fication. It follows the model construction of BMKC, however, the kernel matrix $\mathbf{K}$ is replaced with the matrix of kernel principal components:

$$\underline{\mathbf{K}} = (n\Lambda)^{-1/2}\tilde{\mathbf{K}}\mathbf{U}, \tag{11}$$

where $\tilde{\mathbf{K}}$ is a kernel matrix of the 'centered' mapping, given by:

$$\tilde{\mathbf{K}} = \mathbf{K} - \mathbf{A}\mathbf{K} - \mathbf{K}\mathbf{A} + \mathbf{A}\mathbf{K}\mathbf{A} \tag{12}$$

where $\mathbf{A}$ is a $n \times n$ matrix with all entries equal to $1/n$ [Schölkopf et al., 1998] and $\mathbf{U}$ and $n\Lambda$ are matrices of eigenvector and eigenvalues obtained from:

$$\tilde{\mathbf{K}} = \mathbf{U}n\Lambda\mathbf{U}^T. \tag{13}$$

Thus, the latent variables z_{ik} become:

$$z_{ik} \sim N(\underline{\mathbf{K}}_i\beta_k, \sigma^2) + \epsilon_{ik}, \quad i = 1, ..., n, \tag{14}$$

where $\underline{\mathbf{K}}_i$ is the i^{th} row of matrix $\underline{\mathbf{K}}$.

3.1 Sparsity and Identifiability from the Projection Step

Consider the two dimensional, two class 'circle data' from Figure 3. By employing the Gaussian kernel, BMKC from Sect. 2 fits a logistic regression model to $\Phi(\mathbf{x})$, the data mapped to an infinitely dimensional feature space. However, by application of the kernel trick, the algorithm is actually working in the small subspace of the full feature space. This subspace is spanned by the reproducing kernels $K(\mathbf{x}_i, \cdot)$ and its dimension is $\leq n$. This is a direct extension of the conventional Bayesian logistic regression by using reproducing kernels $K(\mathbf{x}_i, \cdot)$ as the new space of input features. The first nine reproducing kernels of the 'circle data' are plotted in Figure 4(a). The graph shows that the reproducing kernels are highly correlated and only a subset is needed for a good classification model. Such correlation is to be expected given the nature of the new input features: the Gaussian kernel $K(\mathbf{x}_i, \cdot)$ maps each point to a Gaussian centered at $\mathbf{x}_i$, which captures the similarity of $\mathbf{x}_i$ to all other points. Thus, two reproducing kernels $K(\mathbf{x}_i, \cdot)$ and $K(\mathbf{x}_l, \cdot)$ will be correlated if $\mathbf{x}_i$ and $\mathbf{x}_l$ are neighbouring points. This leads to non-identifiability problems discussed in section 2.6. The BKPC, however, fits the logistic model to the projections of the data to the principal axes of the feature space $\Phi(\mathbf{x})$. Thus the space of new input features is spanned by the kernel principal components, which are by definition uncorrelated. Figure 4(b) shows the first three bases of the KPCA subspace for the 'circle data'.

Furthermore, for highly correlated mapped data, the diagonalization of the kernel matrix will yield many eigenvalues $n\lambda_l$ equal to zero. The corresponding principal axes can be removed from the analysis as the variance of the principal component is zero. This ef-

fectively means setting regression parameters $\beta_{lk} = 0$ for $k = 1, ..., K-1$.

Thus the parameters included in the sparse model are σ^2, z_{ik},τ_{kI} and β_{kI}, $\forall k = 1, ..., K-1$ where $I = \{l = 1, ..., n'\}$ and n' is the number of principal components with non-zero eigenvalues.

Note that the proposed model does not require pre-image calculations as the classification is performed in the same feature space as the PCA. This is the main (but subtle) difference between the BKPC and the multilayer formulations of 'first run KPCA then and SVM' suggested by [Schölkopf et al., 1998] and [Schölkopf and Smola, 2002].

3.2 Inference for Sparse Model

Consider a sparse model where some regression parameters β_l are set equal to 0. Let $I = \{l = 1, ..., n' | \beta_l \neq 0\}$ and $\bar{I} = \{l = n', ..., n | \beta_l = 0\}$. The conditional distribution for $\beta_I | \beta_{\bar{I}} = 0$ is given by:

$$\mathbb{P}(\beta_I | \beta_{\bar{I}} = 0, \mathbf{z}, \tau, \sigma^2) = \prod_{k=1}^{K-1} MVN_{(n')}(\tilde{\mathbf{m}}_k, \sigma^2 \tilde{\mathbf{V}}_k), \tag{15}$$

where $\tilde{\mathbf{m}}_k = \mathbf{m}_{kI} - \mathbf{V}_{k2}\mathbf{V}_{k4}^{-1}\mathbf{m}_{k\bar{I}}$ is of dimension $n' \times 1$, $\tilde{\mathbf{V}}_k = (\mathbf{V}_{k1} - \mathbf{V}_{k2}\mathbf{V}_{k4}^{-1}\mathbf{V}_{k3})$, is of dimension $n' \times n$. Note that $\mathbf{m}_{kI}$ and $\mathbf{m}_{k\bar{I}}$ are block components of $\mathbf{m}_k = \begin{pmatrix} \mathbf{m}_{kI} \\ \mathbf{m}_{k\bar{I}} \end{pmatrix}$ with sizes $\begin{pmatrix} n' \times 1 \\ (n-n') \times 1 \end{pmatrix}$ and $\mathbf{V}_{k1}$, $\mathbf{V}_{k2}$, $\mathbf{V}_{k3}$ and $\mathbf{V}_{k4}$ are block components of $\mathbf{V}_k = \begin{pmatrix} \mathbf{V}_{k1} & \mathbf{V}_{k2} \\ \mathbf{V}_{k3} & \mathbf{V}_{k4} \end{pmatrix}$ with sizes

$$\begin{pmatrix} n' \times n' & n' \times (n-n') \\ n' \times (n-n') & (n-n') \times (n-n') \end{pmatrix},$$

where $\mathbf{m}_k = \mathbf{V}_k \underline{\mathbf{K}}^T \mathbf{z}_k$ and $\mathbf{V}_k = (\underline{\mathbf{K}}^T \underline{\mathbf{K}} + \mathbf{T}_k)^{-1}$

The conditional distributions of the other model parameters are given in Appendix B.

3.3 Implementation Issues in Sparse Classifiers

Prior to the MCMC run, the implementation of the BKPC algorithm involves spectral decomposition of $\tilde{\mathbf{K}}$, the kernel matrix of the 'centered' mapping, in order to obtain $\underline{\mathbf{K}}$. Let:

$$\underline{\mathbf{K}} = \begin{pmatrix} \mathbf{K}_1 & \mathbf{K}_2 \\ \mathbf{K}_3 & \mathbf{K}_4 \end{pmatrix} \text{ and } \mathbf{T}_k = \begin{pmatrix} \mathbf{T}_{kI} & 0 \\ 0 & \mathbf{T}_{k\bar{I}} \end{pmatrix}$$

both with sizes $\begin{pmatrix} n' \times n' & n' \times (n-n') \\ n' \times (n-n') & (n-n') \times (n-n') \end{pmatrix}$.

It can be shown using Shur complement that:

$$\begin{aligned} \tilde{\mathbf{V}}_k^{-1} &= (\mathbf{V}_{k1} - \mathbf{V}_{k2}\mathbf{V}_{k4}^{-1}\mathbf{V}_{k3})^{-1} \\ &= \mathbf{K}_1^T\mathbf{K}_1 + \mathbf{K}_3^T\mathbf{K}_3 + \mathbf{T}_{kI}, \end{aligned} \tag{16}$$

and

$$\begin{aligned} \tilde{\mathbf{m}}_k &= \mathbf{m}_{kI} - \mathbf{V}_{k2}\mathbf{V}_{k4}^{-1}\mathbf{m}_{k\bar{I}} \\ &= (\mathbf{K}_1^T\mathbf{K}_1 + \mathbf{K}_3^T\mathbf{K}_3 + \mathbf{T}_{kI})^{-1} \\ &\times \mathbf{K}_1^T\mathbf{z}_{kI} + \mathbf{K}_3^T\mathbf{z}_{k\bar{I}}, \end{aligned} \tag{17}$$

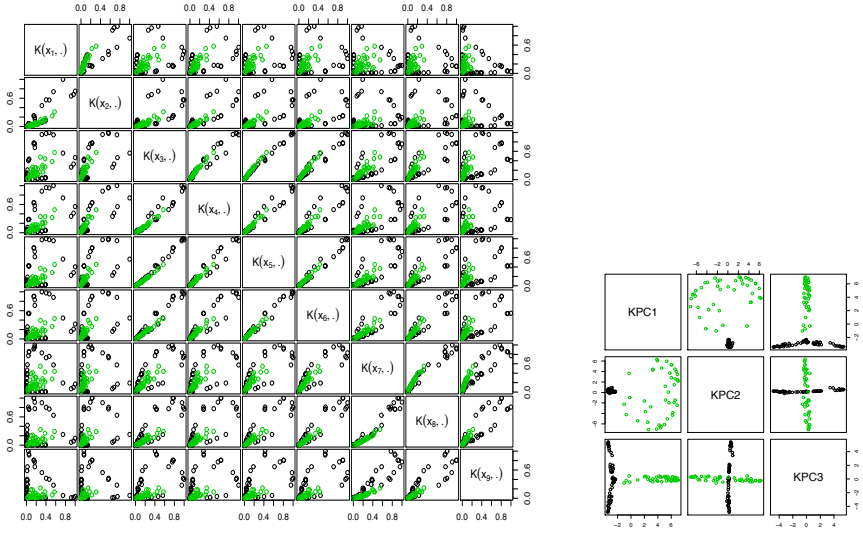

Fig. 4 The graph in (a) shows the 'circle' data mapped to the feature space spanned by the reproducing kernels $K(\mathbf{x}_i, \cdot)$. Only the first nine reproducing kernels are plotted. The graph in (b) shows the 'circle' data mapped to the KPCA subspace. Only the first three KPCs are plotted. Note that the first eigenvector separates the two classes of observations.

where $\mathbf{z}_k = \begin{pmatrix} \mathbf{z}_{kI} \\ \mathbf{z}_{k\bar{I}} \end{pmatrix}$ with sizes $\begin{pmatrix} n' \times 1 \\ (n - n') \times 1 \end{pmatrix}$.

Therefore, instead of first calculating $\mathbf{m}_k = \mathbf{V}_k \underline{\mathbf{K}}^T \mathbf{z}_k$ and $\mathbf{V}_k = (\underline{\mathbf{K}}^T \underline{\mathbf{K}} + \mathbf{T}_k)^{-1}$, and subsequently decomposing them to block components in order to get $\tilde{\mathbf{m}}_k$ and $\tilde{\mathbf{V}}_k$ at each iteration of the MCMC, see equation (15) , the result in (16) and (17) enables us to work directly with $\underline{\mathbf{K}} = \begin{pmatrix} \mathbf{K}_1 \\ \mathbf{K}_3 \end{pmatrix}$, i.e. matrices $\underline{\mathbf{K}}$ whose columns corresponding to $\bar{I} = \{l = n', ..., n | \beta_l = 0\}$ are deleted. It follows that Cholesky decomposition and other computationally demanding operations of the proposed algorithm BKPC are only applied to matrices of dimension $n' \times n'$ at each parameter update, hence large computational gains can be achieved for sparse models.

3.4 Prediction

For test points $\mathbf{x}_i^* \in \mathbb{R}^J$, where $i = 1, ..., n^*$, the $n^* \times n$ inner product kernel matrix is given by:

$$\mathbf{K}^*_{il} = K(\mathbf{x}_i^*, \mathbf{x}_l | \theta), \forall i = 1, ..., n^*, \forall l = 1, ..., n. \tag{18}$$

Similar to (12), inner product matrix of the test observations centered in the feature space can be expressed in terms of $\mathbf{K}^*$:

$$\tilde{\mathbf{K}}^* = \mathbf{K}^* - \mathbf{A}^* \mathbf{K} - \mathbf{K}^* \mathbf{A} + \mathbf{A}^* \mathbf{K} \mathbf{A}, \tag{19}$$

where $\mathbf{A}^*$ is a $n^* \times n$ matrix with all entries equal to $1/n$. The new observation is projected on the principal axes of the mapping $\Phi(\mathbf{x}^*)$ by:

$$\underline{\mathbf{K}}^*_l = (n\lambda_l)^{-1/2} \tilde{\mathbf{K}}^* \mathbf{u}_l, \tag{20}$$

where $l = 1, ..., n'$ and $\underline{\mathbf{K}}_l^*$ denotes the l^{th} column of the $n \times n'$ matrix $\underline{\mathbf{K}}^*$. The observation $\mathbf{x}^*$ is classified in class $k^* = \arg\max_k \mathbb{P}(k|\mathbf{x}^*, \mathbf{x}, \mathbf{y})$ by employing the Monte Carlo integration approximations:

$$\mathbb{P}(k|\mathbf{x}^*, \mathbf{x}, \mathbf{y}) \approx \frac{1}{M} \sum_{m=1}^{M} \frac{\exp(\underline{\mathbf{K}}^* \beta_k^{(m)})}{1 + \sum_{q=1}^{K-1} \exp(\underline{\mathbf{K}}^* \beta_q^{(m)})} \quad (21)$$

$\forall k = 1, ..., K-1$, and

$$\mathbb{P}(K|\mathbf{x}^*, \mathbf{x}, \mathbf{y}) = 1 - \sum_{k=1}^{K-1} \mathbb{P}(k|\mathbf{x}^*, \mathbf{x}, \mathbf{y}). \quad (22)$$

4 The Choice of the Number of Projections

It is possible to work with an even sparser model, if the projections with small corresponding eigenvalues are removed from the analysis. We approach the problem of selecting the optimal number of projections $\hat{n'}$ using Bayesian decision theory, via maximization of expected utility $\mathbb{E}[u(n', y^*)]$, where y_i^* denotes future observations. The utility is formulated so that it trades off predictive accuracy against the complexity of the model. Since data on future observations is not available, we use a utility form that is approximated by crossvalidatory fit (e.g. Gelfand et al. [1992], Bernardo and Smith [1994], Key et al. [1996], Marriott et al. [2001]) where the dataset is randomly split into a training subset used for creation of predictions and a testing set which serves as a proxy for future observations.

For observations $\mathbf{x}_i^* \in \mathbb{R}^J$, y_i^*, $i = 1, ..., n^*$ in the test set of size n^* and some constant c, utility is defined as:

$$u(n', y^*) = \frac{n^* - \sum_{i=1}^{n^*} I_i(n', y_i^*)}{n^*} - \frac{cn'}{n}, \quad (23)$$

where

$$I_i(n', y_i^*) = \begin{cases} 1 \text{ if } y_i^* = k^* \text{ and } k^* = \arg\max_k \mathbb{P}(k|\mathbf{x}^*, \mathbf{x}, \mathbf{y}); \\ 0 \text{ otherwise,} \end{cases} \quad (24)$$

and $\mathbb{P}(k|\mathbf{x}^*, \mathbf{x}, \mathbf{y})$ is defined in equations (21) and (22). The first term in the utility expression (23) measures the predictive capability of the model based on the misclassification rate of the test set while the second term penalizes the inclusion of projections in the model.

The expectation is taken over all possible cross - validation splits of the data. Since the number of such splits is far too large to evaluate the expectation directly, we use Monte Carlo methods to approximate it, averaging over N random splits of the data into training and testing sets.

Figure 5 provides an illustration of this approach for the data sets described in Section 5. For $N = 10$, $c = 1$ the algorithm evaluates the expected utility as a function of the number of kernel projections retained. The optimal number of components $\hat{n'}$ is the one that maximizes the expected utility. At $N = 10$ the Monte

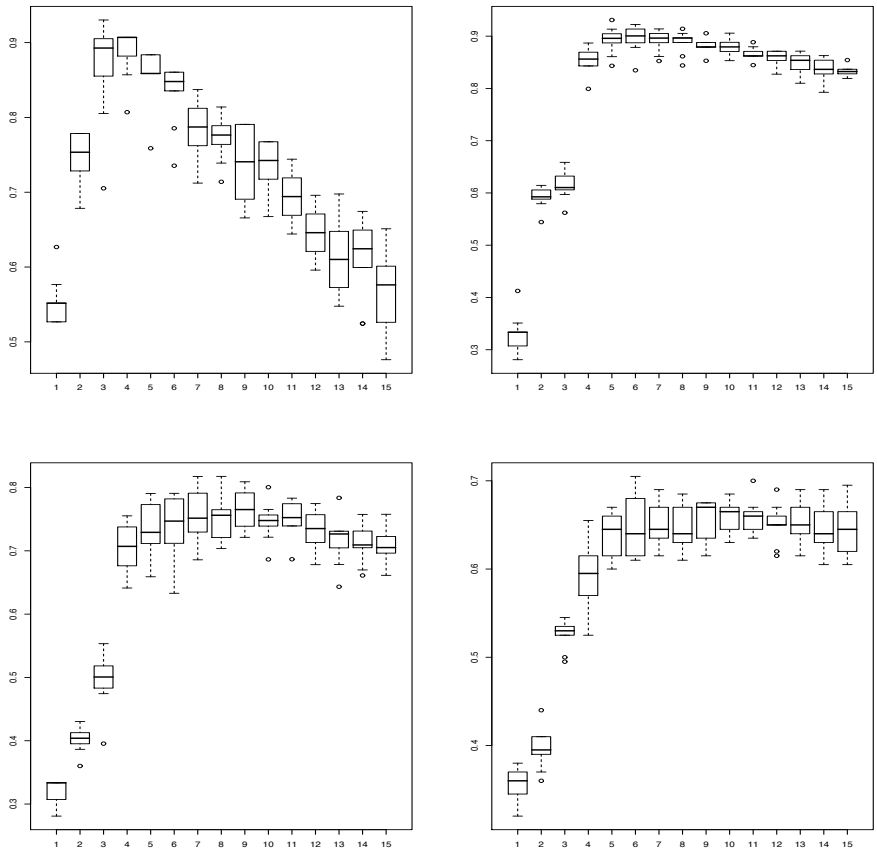

Fig. 5 Estimated expected utility as a function of $n' = 1, 2, ..., 15$ for the microarray, NIR spectroscopy with four and five groups and the image data sets respectively. For all data sets $N = 10$, $c = 1$.

Carlo standard error for each expected utility estimate is at most 0.03 in all the datasets.

Choosing the number of projections can be viewed as a model choice problem. In Bayesian literature on variable selection most approaches focus on a probabilistic fit, see for example George and McCulloch [1993] and George and McCulloch [1997] who put a two component mixture priors on the regression parameters. Therefore a latent binary indicator variable with a Bernoulli prior is used to determine whether a variable is included in the model or not. A drawback of this approach is that the parameter space for the latent variable is discrete and of dimension 2^n, thus an MCMC algorithm is unlikely to to explore the full support of the posterior distribution. Bayesian decision theoretic approach to variable selection was first suggested by Lindley [1968] for univariate multiple regression. Note that it this framework variables are omitted not because their coefficients are believed to be zero, but because they are too costly relative to their predictive benefit [Brown et al., 1999].

The approach taken in this paper is similar to that of Fouskakis and Draper [2002] who apply it to regression coefficient selection in a logistic regression model. One important difference between the problem of choosing of number of projections and the more general setting of variable selection is that in the latter the number of possible models is 2^n, which, as n increases, requires stochastic search methods. Note that the discrete nature of the search space makes these algorithms very sensitive to local optima and its high dimensionality further exacerbates this problem. On the other hand, the number of possible models in the BKPC is $n'' < n$ where n'' is the number of components with non-zero eigenvalues, thus it is possible to evaluate the expected utility for all candidate models. The BKPC algorithm thus proceeds as follows: for each random split, the algorithm carries out a spectral decomposition of the kernel matrix of the centered mapping of the training data. The algorithm then searches exhaustively though the space of n'' models starting with $n' = 1$ where only the projection with the largest corresponding eigenvalue is included and subsequently adding components with decreasing eigenvalue order. For $n' = 1, ..., n''$, the expected utility is obtained by averaging over the utility defined in (23) evaluated for N random splits.

5 Application: High Dimensional Data

5.1 Microarray Data

[Khan et al., 2001] describe gene expression profile data consisting of eighty-three mRNA microarray slides. Each microarray slide corresponds to an individual suffering from one of four tumour types (EWS, BLC, NB and RMS). The total of 2308 genes profiles are reported for each slide. This corresponds to a four category classification problem with a large number of features ($J = 2308$) and small number of observations ($n = 83$). The aim of the analysis is to classify the slides into one of four tumour types on the basis of the gene profiles.

5.2 NIR Spectroscopy Data

The data come from a food authenticity study [Dean et al., 2006]: analysis of spectra of raw homogenized meat samples recorded over the visible and near infrared wavelength range ($400 - 2498$ at intervals of 2 nm, so recorded are 1050 reflectance values) in order to classify samples into five individual species (chicken, turkey, pork, beef and lamb). A four class problem where chicken and turkey are grouped together into a 'poultry' class is also considered for the purposes of classification. Altogether, there are 1050 features and 231 samples in the study A plot of the data is given in Fig-

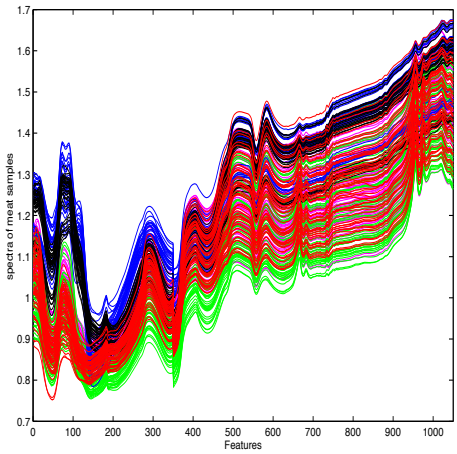

Fig. 6 Individual observations in the NIR spectroscopy data are plotted and coloured by groups: blue and black correspond to the red meat, green is pork and magenta and red correspond to poultry. The visible range of the spectra corresponds to the range [0,150] in this graph.

ure 6. Each meat sample is plotted across the feature space and coloured according to its classification group. The plot shows the most apparent differences between the groups in the visible range of the spectra, which corresponds to the [0,150] section of the feature space. Note that these wavelengths differentiate the colour of the samples so the segregation is between the red and white meat groups.

5.3 Animal Categorization Data

Object recognition is a widely studied problem which has been tackled by a variety of different models. The long term aim of such research is to achieve human levels of recognition accuracy across a large number of object classes in images varying in location, scale, orientation, illumination and subject to occlusions. Animals in natural scenes constitute a challenging problem due to large intra-class variability in terms of shape, texture, size, pose, location in the scene, number of animals etc.

The data set is made up of images that are a subset of the Corel database, which contains 59,795 images of a wide variety of scenes, 8,114 of which are of animals. Four classes of animals were considered: tiger, elephant, goat and lion. 100 images from each class were randomly selected.

The success of the classification depends on the quality of the features summarizing the images. For this task local features which form the 'bag of keypoints' histogram with order of 3,013 features were considered. This set of features was obtained by first detecting the areas of high interest in each image and then extracting the colour, texture and structure information from each area. This information is combined into a histogram of frequencies of the occurrence of certain structures in the image. The data was scaled to have equal standard error across the features.

6 Results

The BKPC was used to fit the data sets described in Sect. 5. For all of the data sets, ten even random splits into training and testing data were used

and the cost parameter c was set to one. In each case, the MCMC algorithm was run for 100,000 iterations, of which the first 9,000 were discarded as 'burn-in'. The misclassification rates of BKPC at $\hat{n'}$ the optimal number of included components are given in Table 1. The results of the proposed method were compared with BMKC and two standard multi-category RHKS classifiers: the Gaussian processes for classification [Williams and Barber, 1998] implemented in library(kernlab) [Karatzoglou et al., 2004] and multi-category SVM with one-against-one technique that fits all the binary sub-classifications and finds the correct class by a voting mechanism implemented in library(e1071) [Dimitriadou et al., 2005], R package version 2.6.1 [R Development Core Team, 2008].

In the proposed method the empirical estimate $\hat{\theta} = 10/max(\mathbf{K})$ is used for the Gaussian kernel bandwidth parameter. The same estimate for θ is used for for the mSVM and the GPs.

The BKPC resulted in improved classification results for all high dimensional data sets. The optimal models that maximized the expected utility were significantly sparser than the full model even though a utility with a very small penalty on the number of included components was used. The BMKC algorithm presented in Sect. 2 performed slightly worse than the BKPC, but its classification results are still comparable to the other kernel classifiers. The main drawback of this model is that it suffers from over - parameterization, as all of the reproducing kernel basis functions are utilized by the model. As a result, the algorithm exhibits poor mixing, and the predictive distributions obtained through simulation are unreliable. In comparison, the BKPC algorithm works with input variables that are by definition uncorrelated.

Another practical disadvantage of BKMC is the relative slow convergence rate caused by the block updating of regression parameters which requires computations involving matrices of dimension $n \times n$, where n is the number of training samples, at each iteration of the MCMC algorithm. The computational speed gain of the BKPC depends on the data set, however, it is considerable for sparse models since the most computationally demanding operations run in time proportional to n'^3. For illustration purposes, consider the NIR spectroscopy data. The number of regression coefficients in this model is $n \times (K-1) = 117 \times 3 = 351$ in the four class model. The 100,000 iterations of MCMC took 110 minutes to run. On the other hand, the first seven feature space projections account for 99% of the variation in the data. The graph in Figure 7(a) plots the proportion of the total variation explained for $n' = 1, ..., n$. The graph in Figure 7(b) shows the computation time required for running 100,000 iterations of MCMC with $n' = 1, ..., 40$.

Table 1 Average misclassification error in the test set obtained from ten random splits of the data sets. Standard deviations are given in brackets. The results are given for runs of the BMKC algorithm proposed in Sect. 2 and the BKPC algorithm described in Sect. 3.The results are given for runs of the proposed algorithm, BMKC algorithm described in Sect. 2, a multi-category SVM (mSVM) with one-against-one technique and the Gaussian processes (GPs) for classification.

Data set	J	n	$\hat{n'}$	BKPC	BMKC	mSVM	GPs	Better?
Images	3013	200	10	0.28 (0.02)	0.37 (0.06)	0.27 (0.05)	0.37 (0.06)	$\surd$
Microarray	2308	43	4	0.02 (0.03)	0.06 (0.04)	0.14 (0.05)	0.17 (0.08)	$\surd$
NIR (4 groups)	1050	117	7	0.05 (0.02)	0.1 (0.03)	0.11 (0.03)	0.11 (0.02)	$\surd$
NIR (5 groups)	1050	117	9	0.16 (0.05)	0.24 (0.04)	0.22 (0.04)	0.23 (0.04)	$\surd$

The total computation time for the BKPC, where the algorithm exhaustively searches through the space of $n'' = 15$ candidate models is 7.57 minutes for each random split of the data. For the optimal model, the number of regression coefficients is $7 \times 3 = 21$, as opposed to 351 in the full model. In addition, the MCMC algorithm for this sparse model of uncorrelated variables achieves better mixing.

Another advantage of this approach is improved data visualization; since BMKC performs the classification in the feature space spanned by reproducing kernels, the number of bases n is usually too large for a matrix plot. However, is possible to visualize a small number of principal component bases of the feature space that the BKPC works in. Figure 8 shows the KPCs, i.e. the ordered columns of matrix $\underline{\mathbf{K}}$, with the largest eigenvalues for the NIR spectroscopy data. The image of the matrix $\underline{\mathbf{K}}$ can be seen in Figure 9.

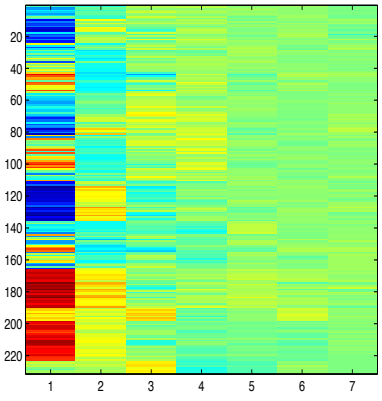

Fig. 9 The image of the matrix of projections $\underline{\mathbf{K}}$ is plotted. Only the first seven KPCs were included in the analysis. The sections of the matrix correspond to: $1-55$ chicken, $55-110$ turkey, $110-165$ pork, $166-197$ beef and $198-231$ lamb.

Multiple chains for different initial values of parameters were run and the classification algorithm was shown to yield similar misclassification error rates. To examine the impact of Monte Carlo error on correct classification rates, ten chains with different initial values for the regression coefficients were run for the single split of the five data sets into a training and testing

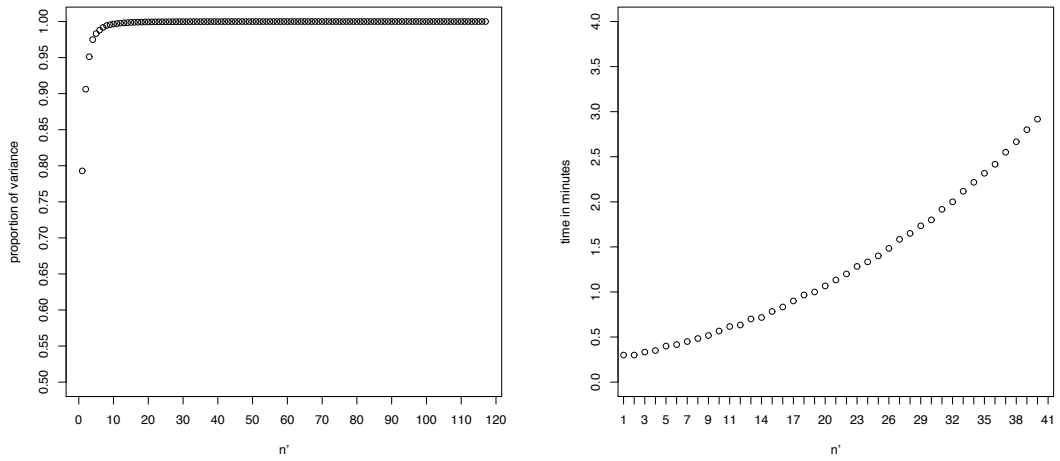

Fig. 7 The graph in (a) plots the proportion of variance explained by $n' = 1, ..., 117$ components. The graph in (b) plots the time taken in minutes, for 100,000 iterations of the MCMC for a model where $n' = 1, ..., 40$.

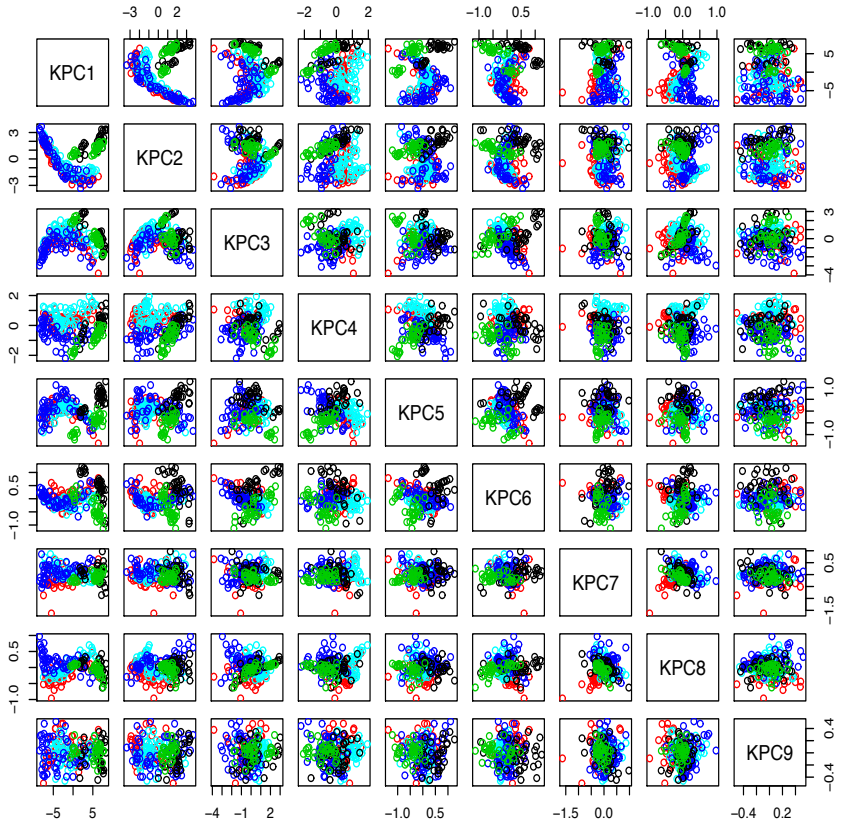

Fig. 8 The first nine KPCs for the NIR spectroscopy data. The colours correspond to the meat type (red=chicken, cyan=turkey, blue=pork, black=beef and green=lamb). Only the first seven KPCs were used for the classification.

Table 2 Average misclassification error for different starting values of β obtained from running the chain ten times on the same random split of the data set. Standard deviations are given in brackets.

Data set	Misclassific. rate
Images	0.28 (0.0)
Microarray	0.038 (0.052)
NIR (4 groups)	0.04 (0.016)
NIR (5 groups)	0.17 (0.02)

data. The regression coefficients were initially either set equal to 1, or were randomly drawn from normal and uniform distributions. For each data set, the number of included projections was set to $\hat{n'}$ given in Table 1. Average misclassification rate for the ten runs is given in Table 2. The results are comparable to those obtained by multiple runs of the chain with the same initial values, but with different random splits seen in Table 1. This shows relative insensitivity of the algorithm to the starting values of these parameters and indicates that 'convergence' to a good classification algorithm has been reached.

7 Discussion

RKHS classifiers, of which BKMC is an example, suffer from over-parameterization if n regression coefficients need to be estimated when n samples are available. Different combinations of nonidentifiable regression coefficients lead to the same likelihood, which results in a multimodal posterior distribution, even if sparse priors are placed on regression parameters. As a result, MCMC samplers mix poorly and maximum a posteriori (MAP) estimates are suboptimal. Regression coefficient selection for these models is tricky as the number of possible models is often near-infinite and many of the 2^n possible combinations of the parameters yield the same result. In practice it is only possible to explore a small subspace of the huge and discrete model space.

The proposed algorithm BKPC is a kernel classifier that performs the classification of the projections of the data to the principal axes of the feature space. The degree of sparsity is regulated through a novel framework based on Bayesian decision theory. Since the number of the possible models is relatively small, it is possible to exhaustively search through the entire model space. We argue that this is a more efficient approach to sparsity for RKHS classifiers. For the high dimensional data sets considered, sparser sets of uncorrelated principal axes were able to adequately summarize the underlying structure of the feature space and improved classification rates were obtained. The sparse optimal models of uncorrelated principal axes required estimating a small number of identifiable regression coefficients and therefore achieved better mixing and faster convergence.

Future work on this topic could involve exploring other prior structures, for example, in the current construction, both the mean and variance of the latent variables depend on σ^2. Whereas, this is a standard assumption for a normal-gamma model which is widely used for tractability in the posterior model, it would be worth exploring the relaxation of this dependence. Furthermore, the inverse-gamma (γ_1, γ_2) distribution is the most common prior distribution used for variance parameters, but it is well recognized that the inverse-gamma priors can be problematic [Lambert, 2006, Gelman, 2006]. Instead of the standard [Spiegelhalter et al., 1996a,b] uninformative prior $\sigma^2 \sim IG(\gamma_1 = 0.001, \gamma_2 = 0.001)$ on the variance parameter, it is possible to use a truncated prior, or as [Gelman, 2006] suggests a proper uniform

A Conditional Distributions for Parameters of BMKC

The conditional distributions for the parameters are given by:

$$\mathbb{P}(\tau|\beta) = \prod_{i=1}^{n}\prod_{k=1}^{K-1} G(\gamma_3 + \frac{1}{2}, \gamma_4 + \frac{\beta_{ik}^2}{2\sigma^2}), \tag{25}$$

$$\mathbb{P}(\beta|\mathbf{z}, \tau, \sigma^2) = \prod_{k=1}^{K-1} MVN_{(n)}(\mathbf{m}_k, \sigma^2\mathbf{V}_k), \tag{26}$$

$$\mathbb{P}(\sigma^2|\beta, \mathbf{z}, \tau) = IG(\gamma_1 + n(K-1), \tilde{\gamma}_2), \tag{27}$$

where $\mathbf{m}_k = \mathbf{V}_k\mathbf{K}^T\mathbf{z}_k, \mathbf{V}_k = (\mathbf{K}^T\mathbf{K} + \mathbf{T}_k)^{-1}$ and $\tilde{\gamma_2} = \gamma_2 + \frac{1}{2}\sum_{k=1}^{K-1}(\mathbf{z}_k^T\mathbf{z}_k - \mathbf{m}_k^T\mathbf{V}_k^{-1}\mathbf{m}_k)$,

$$\mathbb{P}(\mathbf{z}_i|\mathbf{z}_{-i}, \mathbf{y}, \beta, \tau, \sigma^2) \propto \exp\left[\sum_{k=1}^{K-1} y_{ik}z_{ik} - \right.$$
$$\left. - \log\sum_{k=1}^{K}\exp(z_{ik}) - \sum_{k=1}^{K-1}\frac{1}{2\sigma^2}(z_{ik} - \mathbf{K}_i\beta_k)^2\right]. \tag{28}$$

B Conditional Distributions for Parameters of BKPC

The conditional distributions for the parameters are given by:

$$\mathbb{P}(\beta_I|\beta_{\bar{I}} = 0, \mathbf{z}, \tau, \sigma^2) = \prod_{k=1}^{K-1} MVN_{(n')}(\tilde{\mathbf{m}}_k^{(m)}, \sigma^{2(m)}\tilde{\mathbf{V}}_k^{(m)}), \tag{29}$$

$$\mathbb{P}(\mathbf{z}_i|\mathbf{z}_{-i}, \mathbf{y}, \beta, \tau, \sigma^2) \propto \exp\left[\sum_{k=1}^{K-1} y_{ik}z_{ik} - \right.$$
$$\left. - \log\sum_{k=1}^{K}\exp(z_{ik}) - \sum_{k=1}^{K-1}\frac{1}{2\sigma^2}(z_{ik} - \underline{\mathbf{K}}_i\beta_{kI})^2\right], \tag{30}$$

$$\mathbb{P}(\sigma^2|\beta, \mathbf{z}, \tau) = IG(\gamma_1 + n'(K-1), \gamma_2 + $$
$$+ \frac{1}{2}\sum_{k=1}^{K-1}(\mathbf{z}_k^T\mathbf{z}_k - \tilde{\mathbf{m}}_k^T\tilde{\mathbf{V}}_k^{-1}\tilde{\mathbf{m}}_k)), \tag{31}$$

$$\mathbb{P}(\tau_I|\beta, \tau_{\bar{I}} = 0) = \prod_{l=1}^{n'}\prod_{k=1}^{K-1}\sum_{G}(\gamma_3 + \frac{1}{2}, \gamma_4 + \frac{(\beta_{kl})^2}{2\sigma^2}), \tag{32}$$

where $I = \{l = 1, ..., n'\}$, $\tilde{\mathbf{V}}_k^{(m)} = (\underline{\mathbf{K}}^T\underline{\mathbf{K}} + \mathbf{T}_{kI}^{(m-1)})^{-1}$ and $\tilde{\mathbf{m}}_k^{(m)} = \tilde{\mathbf{V}}_k^{(m)}\underline{\mathbf{K}}^T\mathbf{z}_k^{(m)}$.

Acknowledgements We are indebted to A. Teynor from Albert-Ludwigs-Universität Freiburg for providing us with the animal categorization data.

References

M. Aizerman, E. Braverman, and L. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.

G.H. Bakir, J. Weston, and B. Schölkopf. Learning to find pre-images. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16, pages 449–456, Cambridge, MA, 2004. MIT Press.

J. M. Bernardo and A. F. M. Smith. *Bayesian theory.* Wiley, Chichester, 1994.

C. M. Bishop and M. E. Tipping. Variational relevance vector machines. In *UAI '00: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 46–53, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152, Pittsburgh, PA, 1992. ACM Press.

P. J. Brown, T. Fearn, and M. Vannucci. The choice of variables in multivariate regression: a non-conjugate bayesian decision theory approach. *Biometrika*, 86(3):635–648, 1999.

S. Chakraborty, B. K. Mallick, D. Ghosh, M. Ghosh, and E. Dougherty. Gene expression-based glioma classification using hierarchical Bayesian vector machines. *Sankhya*, 69:514–547, 2007.

N. Dean, T.B. Murphy, and G. Downey. Using unlabelled data to update classification rules with applications in food authenticity studies. *J. Roy. Statist. Soc. C*, 55(1):1–14, 2006.

D.G.T. Denison, C.C. Holmes, B.K. Mallick, and A.F.M. Smith. *Bayesian Methods for Nonlinear Classification and Regression.* John Wiley and Sons, Chichester, 2002.

E. Dimitriadou, K. Hornik, F. Leisch, D. Meyer, and A. Weingessel. e1071: Miscellaneous functions of the department of statistics (e1071), TU-Wien, version 1.5-11., 2005. URL `http://CRAN.R-project.org/`.

M. Figueiredo. Adaptive sparseness using jeffreys prior. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 697–704, Cambridge, MA, 2002. MIT Press.

M. Figueiredo. Adaptive sparseness for supervised learning. *IEEE Trans. Pattern Anal. Machine Intell.*, PAMI-25(9): 1150–1159, 2003.

D. Fouskakis and D. Draper. Stochastic optimization: a review. *International Statistical Review*, 70:315–349, 2002.

A. E. Gelfand, D. K. Dey, and H. Chang. Model determination using predictive distributions with implementations via sampling – based methods. In J. M. Bernardo, J.O. Berger, A. P. Dawid, and A.F.M. Smith, editors, *Bayesian Statistics 4*, pages 147–167. Oxford Univ. Press, 1992.

A. Gelman. Prior distributions for variance parameters in hierarchical models. *Bayesian Analysis*, 1(3):515–533, 2006.

E. George and R.E. McCulloch. Variable selection via gibbs sampling. *J. Amer. Statist. Assoc*, 88:881–889, 1993.

E. George and R.E. McCulloch. Approaches for bayesian variable selection. *Statistica Sinica*, 7:339–373, 1997.

R. Herbrich, T. Graepel, and C. Campbell. Bayesian learning in reproducing kernel hilbert spaces – the usefulness of the bayes point. Technical Report TR-99-11, Technical University Berlin, 1999.

C. Holmes and L. Held. Bayesian auxiliary variable models for binary and multinomial regression. *Bayesian Analysis*, 1:145–168, 2005.

A. Karatzoglou, A. Smola, K. Hornik, and A. Zeileis. Kernlab an S4 package for kernel methods in R. *Journal of Statistical Software*, 11, 2004.

J.T. Key, L. R. Pericci, and A.F.M. Smith. Bayesian model choice: what and why? In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, editors, *Bayesian Statistics 6*,

pages 343–370. Oxford University Press, 1996.

J. Khan, J. S. Wei, M. Ringnr, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, and P. S. Meltzer. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, 7(6):673–679, June 2001.

B. Krishnapuram, L. Carin, M. Figueiredo, and A. Hartemink. Sparse multinomaial logistic regression: Fast algorithms and generalization bounds. *IEEE Transactions on Pattern Analysis and Machine Intelligence - PAMI*, 27(6):957–968, 2005.

J.T.Y. Kwok. Moderating the outputs of support vector machine classifiers. *IEEE Transactions on Neural Networks*, 5:1018 – 1031, 1999.

P. C. Lambert. Comment on article by browne and draper. *Bayesian Analysis*, 1(3):543–546, 2006.

Y. Lee, Y. Lin, and G. Wahba. Multicategory Support Vector Machines: Theory, and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, 99:67 – 81, 2004.

D. V. Lindley. The choice of variables in multiple regression (with discussion). *J. Roy. Statist. Soc. B*, 30:31–66, 1968.

B. K. Mallick, D. Ghosh, and M. Ghosh. Bayesian classification of tumors using gene expression data. *J. Royal Statistical Soc. B*, 67:219–234, 2005.

J.M. Marriott, N. M. Spencer, and A. N. Pettitt. A bayesian approach to selecting covariates for prediction. *Scand. J. Stat.*, 28:87–97, 2001.

R. M. Neal. *Bayesian Learning for Neural Networks*. Springer Verlag, New York, 1996.

R. M. Neal. Regression and classification using gaussian process priors (with discussion). In J. M Bernardo et al., editors, *Bayesian Statistics 6*, pages 475–501. Oxford University Press, 1998.

M. Opper and O. Winther. Gaussian process classification and svm: Mean field results and leave one out estimator. In A.J.Smola, P. Bartlett, B. Schölkoph, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 43–65, Cambridge, MA, 2000.

W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical recipes: the art of scientific computing*. Cambridge University Press, New York, 1986.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. URL `http://www.R-project.org`. ISBN 3-900051-07-0.

C. E. Rasmussen. *Evaluation of Gaussian Processes and Other Methods for Non-linear Regression*. Phd, Dept. of Computer Science, University of Toronto, 1996.

B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, 1996.

B. Schölkopf and A. Smola. *Learning with Kernels- Support Vector Machines, Reproducing Kernel Hilbert Spaces , Regularization, Optimization and Beyond*. Cambridge, MA: MIT Press, 2002.

B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

B. Schölkopf, S. Mika, C. J. C. Burges, et al. Input space vs feature space in kernel-based methods. *IEEE Trans. on Neural Networks*, 10(5):1000–1017, 1999.

M. Seeger. Bayesian model selection for support vector machines, gaussian processes and other kernel classifiers. In T.K. Leen S.A. Solla and K. R. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12, pages 603–609, MIT Press, 2000.

P. Sollich. Bayesian methods for support vector machines: Evidence and predictive class probabilities. *Machine Learning*, 46 (1-3):21–52, 2002.

D. J. Spiegelhalter, A. Thomas, N.G. Best, and W. R. Gilks. *BUGS Examples, Volume 1, Version 0.5.* Cambridge: MRC Biostatistics Unit., 1996a.

D. J. Spiegelhalter, A. Thomas, N.G. Best, and W. R. Gilks. *BUGS Examples, Volume 2, Version 0.5.* Cambridge: MRC Biostatistics Unit., 1996b.

R. A. Thisted. *Elements of Statistical Computing.* Chapman and Hall, New York, 1988.

M. E. Tipping. The relevance vector machine. In S. A. Solla, T. K. Leen, and K. R. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12, pages 652–658. MIT Press, 2000.

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

G. Wahba. *Spline models for observational data.* SIAM [Society for Industrial and Applied Mathematics], 1990.

Christopher K. I. Williams and David Barber. Bayesian classification with gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1342–1351, 1998.

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