Computational Statistics and Data Analysis 53 (2009) 4198–4209

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

Bayesian binary kernel probit model for microarray based cancer classification and gene selection

Sounak Chakraborty *

Department of Statistics, University of Missouri-Columbia, 209F Middlebush Hall, Columbia, MO 65211-6100, USA

ARTICLE INFO

Article history:
Received 3 February 2009
Received in revised form 6 May 2009
Accepted 8 May 2009
Available online 18 May 2009

ABSTRACT

With the arrival of gene expression microarrays a new challenge has opened up for identification or classification of cancer tissues. Due to the large number of genes providing valuable information simultaneously compared to very few available tissue samples the cancer staging or classification becomes very tricky.

In this paper we introduce a hierarchical Bayesian probit model for two class cancer classification. Instead of assuming a linear structure for the function that relates the gene expressions with the cancer types we only assume that the relationship is explained by an unknown function which belongs to an abstract functional space like the reproducing kernel Hilbert space. Our formulation automatically reduces the dimension of the problem from the large number of covariates or genes to a small sample size. We incorporate a Bayesian gene selection scheme with the automatic dimension reduction to adaptively select important genes and classify cancer types under an unified model. Our model is highly flexible in terms of explaining the relationship between the cancer types and gene expression measurements and picking up the differentially expressed genes. The proposed model is successfully tested on three simulated data sets and three publicly available leukemia cancer, colon cancer, and prostate cancer real life data sets.

1. Introduction

Classification or staging of cancer is a widely studied area for a long time. With exact cancer type identification we can figure out the right treatment. More importantly this reduces and eliminates the risk of damaging healthy tissues from unintentional side effects from the treatment. Before the discovery of gene expression microarrays cancer classification was done according to the site of origin, histopathological examination of samples and the spread of the cancer. Since cancer develops from an alteration in a cell's genetic structure due to mutations to cells with uncontrolled growth patterns, extensive research has shown that we might be able to identify the cancer more accurately by looking at the genetic level (Duggan et al., 1999; Golub et al., 1999; Schena et al., 1995; Alizadeh et al., 2000). Before the arrival of microarrays we used to study the genetic behavior of tissue samples by looking at one gene at a time. With the discovery of DNA microarrays which are "an ordered array of nucleic acids, proteins, small molecules, that enables parallel analysis of complex biochemical samples" (Schena et al., 1995), we are able to monitor the activities of thousands of genes simultaneously. Since most of the genes interact with each other this tool gave us the power to look at the full picture at once.

In a microarray study we obtain the expression level of thousands of genes. Due to prohibitively high cost we can afford a very small number of samples, mostly less than hundred. This launches us into a new study area where the sample size is much smaller than the number of covariates or explanatory variables. Most of the standard statistical classification

* Corresponding author. Tel.: +1 573 882 3916; fax: +1 573 884 5524.
E-mail address: chakrabortys@missouri.edu.

0167-9473/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.csda.2009.05.007

techniques fail or perform poorly if the dimension or the number of covariates are very large. This phenomenon is well known in the statistical literature as the "curse of dimensionality" (Bellman, 1961). It means that the mean integrated sum of squares of error increases nonlinearly as the dimension increases. Classification methods recently developed in the machine learning community like support vector machine (SVM) (Vapnik, 1995) and random forest (RF) (Breiman, 2001) are equipped to handle large gene expression data sets for accurate classification. Although theoretically SVM and RF models can be used with full gene expression microarrays for cancer type identification, during actual implementation it has been observed that the precision largely depends on the set of genes used for fitting the models. With the correct set of differentially expressed genes they can be used very effectively in cancer tumor classification. More recently a new type of SVM based on L_1 penalty, popularly known as L1-SVM (Bradley and Mangasarian, 1998; Wang and Shen, 2007), are introduced which can produce a sparse solution. In addition to that, another type of SVM based on recursive feature elimination known as SVM-RFE (Guyon et al., 2002) has been recently developed that can deal with variable selection and classification simultaneously. But both L1-SVM and SVM-RFE lack a probabilistic interpretation. Apart from the SVM formulations there are few penalty based shrinkage methods for linear and additive regression models (Tibshirani, 1996; Zou and Yuan, 2008; Zou and Hastie, 2005; Belitza and Lang, 2008) that can do simultaneous variable selection and model fitting. Artificial neural networks (Ripley, 1996) are very effectively used in explaining nonlinear relationships between cancer types and the genes. But with huge number of covariates or input nodes the total number of unknown parameters in the model becomes unmanageable. This make the whole optimization highly unstable and computationally impossible without some prior gene subset selection (Chakraborty et al., 2004). The Bayesian neural network by Neal (1996) can handle the full list of genes by the automatic relevance determination or ARD. It shrinks the effect of an insignificant gene near to zero with the help of a normal and inverse gamma prior. Some works by Lee et al. (2003) and Sha et al. (2004) suggested to use Bayesian probit, and multinomial regression models where the genes are selected into the model adaptively with the help of a Bayesian mixture prior. But they have used a linear model to establish the relationship between the genes and the cancer types. Often how the genes finally explain the tumor behavior cannot be tracked down by a simple linear structure. Also under the set up proposed by Lee et al. (2003) the number of genes selected in the model has to be less than the number of samples in the training set. Under a Bayesian model selection scheme where we stochastically search over a large space of microarray covariates this cannot be guaranteed and often leads to a breakdown. In their subsequent work they generalized the approach of Lee et al. (2003). According to their general approach the regression coefficients of the probit model are integrated out and the MCMC chain is designed to visit models of any size (Sha et al., 2004). In a very large dimension there also exists a possibility of multicollinearity, so even with a small set of genes we can end up with a set of covariates which may lead to over fitting or non full rank covariance matrix. In recent literature a Bayesian SVM developed by Mallick et al. (2005) based on the similar RKHS theory has been used for gene based cancer classification. Their method is limited in the sense that it cannot self sufficiently select the important genes. Another approach for non-linear classification as described in Zhou et al. (2003) uses the stochastic search and nonlinear additive model for simultaneous gene selection and classification.

In this paper we suggest a novel technique which does not restrict the classification function to be linear and propose a kernel based Bayesian semi-parametric probit model. Instead of giving a linear model relationship we assume that the function which connects the genes with the cancer class probability belongs to a broad abstract functional space known as the reproducing kernel Hilbert space (RKHS) (Wahba, 1990; Parzen, 1970; Aronszajn, 1950). RKHS is a smooth functional space which encompasses a broad class of linear and nonlinear functions. In Section 2 we will describe how we have used the kernel representation of RKHS (Wahba, 1990) to reduce the model space effectively to a much lower dimension without any loss of information. Along with that we have used a Bayesian variable selection (Sha et al., 2004; Tadesse et al., 2005; Kim et al., 2006) with Bernoulli priors on the covariates to identify the important genes from the model and further improve the classification accuracy. Under this approach the gene selection is done by indexing the covariates of the model not putting a mixture prior on the regression coefficients as originally proposed by George and McCulloch (1993).

In Section 3 we have discussed in details about the prior choices, our implementation scheme, and several other issues like the convergence of MCMC, computing time etc. In this section we also include a brief description of the selected tuning parameters for the other competing models which yield optimal prediction performance. We have applied our model on four different simulated data sets and three gene expression microarray data sets available in the public domain: leukemia cancer (Golub et al., 1999), colon cancer (Alon et al., 1999), and prostate cancer (Liu et al., 2007). The detailed results are reported in Section 4. In our study we also look into the genes that are selected by our model and investigate their biological properties. In all three real life case studies we tabulate the genes that are selected most frequently by our model and discuss their biological relevance. In Section 5 we summarize our findings and discuss about some future possible extensions. Some further analyses are provided in the Appendix A.

2. Bayesian RKHS probit model

Let us assume $(y_1, \boldsymbol{x}_1), (y_2, \boldsymbol{x}_2), \ldots, (y_n, \boldsymbol{x}_n)$ be n samples in the training set. Where $y_i \in 0, 1, i = 1, \ldots, n$ are the class labels for two classes for the i-th sample. In this paper we only consider two class cases. For a tissue sample with p available genes, $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^{\mathrm{T}}$ denote the simultaneous gene expression measurements corresponding to the i-th tumor sample. So according to our notation x_{ij} denotes the gene expression measurement for the j-th gene corresponding to the

i-th tissue sample, $i = 1, \ldots, n$, and $j = 1, \ldots, p$. We model the probability of tumor classes by independent Bernoulli distribution as follows,

$$y_i \overset{ind}{\sim} Bernoulli(p_i), \quad i = 1, \ldots, n \tag{1}$$

where p_i is the probability that $y_i = 1$. Given we have the gene expression measurements of a patient we connect the probability of a tumor of type 1 (p_i) using the probit model as follows

$$P(y_i = 1|\Theta) = \Phi(f(\boldsymbol{x}_i, \Theta)), \quad i = 1, \ldots, n \tag{2}$$

where Θ is the set of all unknown model parameters, $\Phi(.)$ is the standard normal cumulative density function, and $f(.)$ is the unknown function that relates the genes with the class probability. In a recent work on Bayesian classification (Lee et al., 2003) assumed a linear regression structure for $f(\boldsymbol{x}_i)$ as $f(\boldsymbol{x}_i) = \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta}$. This restricts the class of models to a very narrow class. Such strict assumption fails to produce accurate results in cases where the functional relationship is not simple linear. In the simulation and case study section in this paper we will show several such cases where simple Bayesian probit regression model performs poorly.

In this paper we do not impose the $f(\boldsymbol{x}_i) = \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta}$ assumption on $f(.)$. Rather we assume that the unknown function $f(.)$ belongs to an abstract functional space known as reproducing kernel Hilbert space (RKHS) denoted as $\mathcal{H}_K$. A Hilbert space is a infinite dimensional functional space encompassing a large class of functions. For practical purposes the Hilbert functional space is too huge and contains a large number of non-smooth functions. This makes the function estimation in Hilbert space very difficult. We restrict our functional space to be a much smaller class known as the reproducing kernel Hilbert space or RKHS. Hence,

$$f(.) \in \mathcal{H}_K. \tag{3}$$

Thus when the unknown function belongs to the RKHS functional space, using Wahba (1990) representation we can express the unknown function $f(.)$ in Eq. (2) as a linear combination of positive definite kernel functions $K(., \boldsymbol{x}_k) \in \mathcal{H}_K$, $k = 1, \ldots, n$ which spans the RKHS or $\mathcal{H}_K$. So we can write,

$$\begin{aligned} f(\boldsymbol{x}_i; \theta, \beta) &= \sum_{k=1}^{n} \beta_k K(\boldsymbol{x}_i, \boldsymbol{x}_k|\theta) \\ &= \boldsymbol{K}_i(\theta)^{\mathrm{T}}\boldsymbol{\beta}. \end{aligned} \tag{4}$$

In Eq. (4) $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_n)^{\mathrm{T}}$ is the coefficient vector, $\boldsymbol{K}_i(\theta) = (K(\boldsymbol{x}_i, \boldsymbol{x}_1|\theta), K(\boldsymbol{x}_i, \boldsymbol{x}_2|\theta), \ldots, K(\boldsymbol{x}_i, \boldsymbol{x}_n|\theta))^{\mathrm{T}}$ and θ is the kernel parameter. There are several choices of kernel functions. The two most prominent choices are the Gaussian kernel and polynomial kernel. In this paper we will use the Gaussian kernel defined as

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left\{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\theta}\right\} \tag{5}$$

$\|.\|$ is the Euclidean distance. Gaussian kernel is particularly effective for its localized and finite responses across the entire range of the real values of $\boldsymbol{x}$. This is due to the fact that if we take the limit of the Gaussian kernel, as the θ goes down to zero, we get the mathematical delta function, or Delta-Dirac function, $\delta()$. Unlike The kernel parameter θ is left unknown and we will put a prior distribution on θ. We also did an exploratory analysis using the polynomial kernel $(1 + \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}/p)^{\theta}$ but the best prediction performance is mostly achieved with a Gaussian kernel. In Table A.1, we report the results with the polynomial kernel.

Initially, we had p covariates or genes but by using the RKHS theory we can express the unknown function $f(.)$ as a linear combination of n terms as in Eq. (4), where n is the number of samples in the training set. In a typical microarray data set we have large p and small n, so this kernel trick automatically reduces the dimension from p to n. Apart from the default dimension reduction through this trick we are also able to incorporate a large class of linear and non-linear functions since our only assumption about $f(\boldsymbol{x})$ is that it belongs to the RKHS. This provides our model a high degree of flexibility.

The kernel trick as described in Eq. (4) reduces the dimension effectively and enables us to use all available genes in our model. The RKHS theory gives us a finite representation $f(x) = \sum_{i=1}^{n} \beta_i K(x, x_i)$ of an infinite expansion of a function $f(x) = \sum_{i=1}^{\infty} c_i\phi_i(x)$, where $f \in \mathcal{H}_K$ (Wahba, 1990) and $\phi_i()$s are the eigen-functions. Thus by the kernel trick of RKHS we are reducing an infinite dimensional problem to a tractable finite dimensional estimation problem. By our earlier statement “without any loss of information” in introduction we mean that through finite representation we are retaining the same level of accuracy as with the infinite series expansion. This would not happen if we use the infinite series formulation and chose to truncate the series. In practice, out of thousands of genes only a handful are differentially expressed. So the prediction performance can be enhanced by a large margin if we can adaptively select those significant genes in the model. We introduce a $p \times 1$ dimensional binary $\boldsymbol{\gamma}$ vector for adaptive gene selection (Tadesse et al., 2005; Sha et al., 2004). Where the j-th component of $\boldsymbol{\gamma}$ is defined as

$$\begin{aligned} \gamma_j &= 1 \quad \text{if the } j\text{-th gene is included in the model} \\ &= 0 \quad \text{otherwise}, \ j = 1, 2, \ldots, p. \end{aligned} \tag{6}$$

Whenever the j-th gene is not selected or the corresponding $\gamma_j = 0$ then the j-th column of the gene expression matrix will be deleted. If $\boldsymbol{X}$ is the full gene expression matrix with columns as the genes and rows as the samples, we denote $\boldsymbol{X}_{\boldsymbol{\gamma}}$ as the reduced gene expression matrix with columns in corresponding nonzero positions in $\boldsymbol{\gamma}$. We now calculate the kernels $\boldsymbol{K}_{\boldsymbol{\gamma}}$ based on the reduced set of gene expressions. Previously Mallick et al. (2005) used the kernel trick to develop a Bayesian support vector machine model but their treatment lacked the critical component of adaptive gene selection. Eq. (4) now changes to

$$\begin{aligned} f(\boldsymbol{x}_i; \boldsymbol{\gamma}, \theta, \boldsymbol{\beta}) &= \sum_{k=1}^{n} \beta_k K_{\boldsymbol{\gamma}}(\boldsymbol{x}_i, \boldsymbol{x}_k|\theta) \\ &= \boldsymbol{K}_{i\boldsymbol{\gamma}}(\theta)^{\mathrm{T}}\boldsymbol{\beta}. \end{aligned} \tag{7}$$

Adopting the latent variable approach (Albert and Chib, 1993) we introduce n independent latent variables $\boldsymbol{z} = (z_1, z_2, \ldots, z_n)^{\mathrm{T}}$ and connect with $f(\boldsymbol{x}_i; \boldsymbol{\gamma}, \theta, \boldsymbol{\beta})$ from Eq. (7) as

$$\begin{aligned} z_i &= f(\boldsymbol{x}_i; \boldsymbol{\gamma}, \theta, \boldsymbol{\beta}) + \epsilon_i \\ &= \boldsymbol{K}_{i\boldsymbol{\gamma}}(\theta)^{\mathrm{T}}\boldsymbol{\beta} + \epsilon_i, \quad i = 1, \ldots, n. \end{aligned} \tag{8}$$

Where $\epsilon_i \sim N(0, 1)$, and y_i and z_i are connected as

$$\begin{aligned} y_i &= 1 \quad \text{if } z_i > 0 \\ &= 0 \quad \text{if } z_i < 0, j = 1, 2, \ldots, p. \end{aligned} \tag{9}$$

We assign priors to the unknown model parameters $\boldsymbol{\beta}, \theta$, and $\boldsymbol{\gamma}$ as follows

$$\begin{aligned} &\beta_k \overset{iid}{\sim} N(0, a) \\ &\theta \sim \mathit{Uniform}(r_L, r_U) \text{ (For Gaussian kernel)} \\ &\gamma_j \overset{iid}{\sim} \mathit{Bernoulli}(\omega_j) \end{aligned} \tag{10}$$

$k = 1, \ldots, n; j = 1, \ldots, p$. For polynomial kernel we use the Discrete Uniform distribution. Generally we fix the kernel parameter θ to some pre-determined fixed value obtained by cross validation procedure. Here by putting a prior on the kernel parameter θ we are ensuring that instead of one fixed shaped kernel our model makes use of polynomial or Gaussian kernels with flexible shapes. This helps us to produce a much richer class of nonlinear classifier. We can also obtain an estimate of θ by maximizing the marginal posterior distribution. The joint posterior distribution of $\boldsymbol{\beta}, \theta, \boldsymbol{\gamma}$, and $\boldsymbol{z}$ is given by,

$$\begin{aligned} \pi(\boldsymbol{\beta}, \theta, \boldsymbol{\gamma}, \boldsymbol{z}|\boldsymbol{y}, \boldsymbol{X}) \propto{}& \prod_{i=1}^{n} \Phi(z_i)^{y_i}(1 - \Phi(z_i))^{1-y_i} \\ &\times \exp\left(-\frac{\sum_{i=1}^{n}(z_i - \boldsymbol{K}_{i\boldsymbol{\gamma}}(\theta)^{\mathrm{T}}\boldsymbol{\beta})^2}{2}\right) \times \exp\left(-\frac{\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\beta}}{2a}\right) \times \frac{1}{(r_U - r_L)} \times \prod_{j=1}^{p} \omega_j^{\gamma_j}(1 - \omega_j)^{1-\gamma_j}. \end{aligned} \tag{11}$$

For a new sample $(y_{new}, \boldsymbol{x}_{new})$ to predict its class we calculate its the posterior predictive probability distribution given by

$$\begin{aligned} &P(y_{new} = 1|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{x}_{new}) \\ &\quad = \int_{\boldsymbol{\gamma}} \int_{\theta} \int_{\boldsymbol{\beta}} \int_{\boldsymbol{z}} P(y_{new} = 1|\boldsymbol{\beta}, \theta, \boldsymbol{\gamma}, \boldsymbol{z}, \boldsymbol{x}_{new}) \times \pi(\boldsymbol{\beta}, \theta, \boldsymbol{\gamma}, \boldsymbol{z}|\boldsymbol{y}, \boldsymbol{X}) \mathrm{d}\boldsymbol{z}\mathrm{d}\boldsymbol{\beta}\mathrm{d}\boldsymbol{\theta}\mathrm{d}\boldsymbol{\gamma}. \end{aligned} \tag{12}$$

The above equation (12) represents the posterior predictive probability of $y_{new} = 1$ i.e., the cancer corresponding to the new gene expression measurement will be of class 1. If the calculated value of Eq. (12) is greater than 0.5 we assign it to class 1 otherwise we assign it to class 0. To evaluate the multiple integral in Eq. (12) we cannot use the analytical approach since there is no closed form solution of Eq. (12), also standard quadrature method is ineffective and computationally impossible since we need to integrate over a very large dimension. So to get an approximate value for the integral we use the Gibbs sampling (Gelfand and Smith, 1990) and Metropolis–Hastings technique (Metropolis et al., 1953). The detail step by step algorithm is described in Section 3. Through out the rest of the paper we will refer our Bayesian kernel probit model as BKPR.

3. Prior choice and model implementation

The performance of any Bayesian model often remains highly dependent to the choice of prior. To reduce the sensitivity of our model on the prior we use near diffuse but proper priors. The objective behind near diffuse proper prior is, we want to guarantee a proper posterior along with least sensitivity. For prior choice on $\beta \sim N(0, a)$ we fix $a = 10^3$. The large variance makes the prior distribution as flat as possible. The gene selection parameter is $\omega_j, j = 1, \ldots, p$. Since we do not have any information about which genes are more important than other we give equal importance to each gene and fix $\omega_j = \omega$, for $j = 1, \ldots, p$. In reality out of thousands of genes only a handful are active. So to have a sparse model with few genes we assign $\omega = 0.01$. That means on average only 1% of the genes will be included in the model. In the prior literature by Sha et al. (2004), Lee et al. (2003) and Tadesse et al. (2005) it has been also suggested to assign ω a low value. In most of the other approaches it is necessary that we tune ω to keep the number of genes less than the size of the training set, our method is not restricted by any such constraint. For the kernel parameter θ our assigned uniform prior behaves like a noninformative flat prior over the closed support (r_L, r_U). It is advisable not to use the common flat prior over the entire range $(0, \infty)$ since it may lead to a near singular matrix in computation. When the sample size is very small like in most of the microarray experiment cases the prediction performance decays by a large margin if the value of θ is allowed over a very large interval. Our choice of prior on θ can be considered to be locally non-informative, it gives higher prediction accuracy and avoids the computational problem of getting stuck due to a near singular matrix in any step. A detailed scheme of finding an optimal value of the kernel parameter θ is described in Step 5 of our algorithm for model fitting.

As Eq. (12) has no closed form solution we find an approximation using MCMC technique by generating samples from the full conditional distributions of $\boldsymbol{\beta}$, $\boldsymbol{\theta}$, and $\boldsymbol{\gamma}$ as follows:

Step 1. Start with an initial value of $\boldsymbol{z}^0, \boldsymbol{\gamma}^0, \boldsymbol{\beta}^0, \theta^0$

Step 2. Sampling from $\pi(\boldsymbol{\beta}|\theta, \boldsymbol{\gamma}, \boldsymbol{z}, \boldsymbol{y}, \boldsymbol{X})$: The conditional posterior of $\boldsymbol{\beta}$ is given by

$$\boldsymbol{\beta}| \cdots \sim N\left(\mu_*, \sigma_*^2\right)$$

where $\mu_* = (a^{-1}I_n + \boldsymbol{K}_\gamma^{\mathrm{T}}\boldsymbol{K}_\gamma)^{-1}\boldsymbol{K}_\gamma^{\mathrm{T}}\boldsymbol{z}$,

$\sigma_*^2 = (a^{-1}I_n + \boldsymbol{K}_\gamma^{\mathrm{T}}\boldsymbol{K}_\gamma)^{-1}$ and I_n is the $n \times n$ identity matrix

Step 3. Sampling from $\pi(\boldsymbol{z}|\boldsymbol{\beta}, \theta, \boldsymbol{\gamma}, \boldsymbol{y}, \boldsymbol{X})$:

From Eq. (11) we see,

$$z_i| \cdots y_i = 1 \sim N(\boldsymbol{K}_{i\gamma}\boldsymbol{\beta}, 1)\mathbf{1}[z_i > 0]$$
$$z_i| \cdots y_i = 0 \sim N(\boldsymbol{K}_{i\gamma}\boldsymbol{\beta}, 1)\mathbf{1}[z_i < 0] \quad i = 1, \ldots, n.$$

$\mathbf{1}[z \in S]$ is an indicator function. Thus z_is are truncated normal distributions.

Step 4. Sampling from $\pi(\boldsymbol{\gamma}|\boldsymbol{z}, \boldsymbol{\beta}, \theta, \boldsymbol{y}, \boldsymbol{X})$: We sample from $\boldsymbol{\gamma}$ by updating each component individually, using the conditional distribution,

$$\pi(\gamma_k|\boldsymbol{z}, \boldsymbol{\beta}, \theta, \boldsymbol{y}, \boldsymbol{X}, \gamma_{-j}) \propto \exp\left(-\frac{\sum_{i=1}^{n}(z_i - \boldsymbol{K}_{i\gamma}(\theta)^{\mathrm{T}}\boldsymbol{\beta})^2}{2}\right) \times \omega^{\gamma_k}(1-\omega)^{1-\gamma_k}.$$

The above distribution does not belong to any standard distribution. We use Metropolis–Hastings algorithm with Bernoulli(ω) as the proposal distribution to generate γ_k as follows

(i) $\gamma_k^{new} = \text{Bernoulli}(\omega)$.

(ii) Using Metropolis–Hastings algorithm accept the γ_k^{new} with acceptance probability,

$$R_1 = \min\left\{1, \frac{\pi(\gamma_k^{new}|\boldsymbol{z}, \boldsymbol{\beta}, \theta, \boldsymbol{y}, \boldsymbol{X}, \gamma_j, j \neq k)}{\pi(\gamma_k|\boldsymbol{z}, \boldsymbol{\beta}, \theta, \boldsymbol{y}, \boldsymbol{X}, \gamma_j, j \neq k)}\right\}.$$

In $\boldsymbol{\gamma}_{new}$ we have updated only one component $\gamma_k = \gamma_k^{new}$.

Step 5. Sampling from $\pi(\theta|\boldsymbol{\gamma}, \boldsymbol{z}, \boldsymbol{\beta}, \boldsymbol{y}, \boldsymbol{X})$: For fast updating and better mixing of the Markov chain we update the kernel parameter θ by sampling from the marginal posterior of θ obtained by integrating out the $\boldsymbol{\beta}$ parameter. The marginal posterior of $\boldsymbol{\beta}$ is given by

$$\pi(\theta|\boldsymbol{\gamma}, \boldsymbol{z}, \boldsymbol{y}, \boldsymbol{X}) \propto \frac{\exp\left\{\frac{1}{2}\left(\boldsymbol{z}^{\mathrm{T}}\boldsymbol{K}_\gamma\left(\boldsymbol{K}_\gamma^{\mathrm{T}}\boldsymbol{K}_\gamma + a^{-1}I_n\right)^{-1}\boldsymbol{K}_\gamma^{\mathrm{T}}\boldsymbol{z}\right)\right\}}{|\boldsymbol{K}_\gamma^{\mathrm{T}}\boldsymbol{K}_\gamma + a^{-1}I_n|^{1/2}}. \tag{13}$$

We sample from Eq. (13) by following steps:

(i) Maximize the marginal posterior equation (13), to obtain θ_{opt}.

(ii) Fix a proposal distribution around θ_{opt},

$\theta \sim Uniform(\theta_{opt} - L, \theta_{opt} + L)$, and generate a new $\theta = \theta_{new}$.

(iii) Accept θ_{new} with acceptance probability,

$$R_2 = \min\left\{1, \frac{\pi(\theta_{new}|\boldsymbol{\gamma}, \boldsymbol{z}, \boldsymbol{y}, \boldsymbol{X})}{\pi(\theta_{opt}|\boldsymbol{\gamma}, \boldsymbol{z}, \boldsymbol{y}, \boldsymbol{X})}\right\}.$$

The parameter L is the tuning parameter. We tune the range of the proposal distribution of θ to achieve around 30% acceptance rate by controlling L. The parameter θ cannot be extended over the entire real line, because too large a range of θ often leads $(\boldsymbol{K}_{\boldsymbol{\gamma}}^{\mathrm{T}}\boldsymbol{K}_{\boldsymbol{\gamma}} + a^{-1}I_n)$ to be near singular, and thus it becomes impossible for computer to invert. There are several advantages of this sampling scheme for θ. Since there are no good interpretations or insights about the possible value of θ it is very difficult to come up with a reasonable prior choice. Since we have very few data, flat priors or near diffuse priors will not be able to help. The method as mentioned before can be viewed as a type of data driven prior where in each step we find an empirical Bayes type of estimate for θ and perturb around that value to come up with a prior and sample from it.

To make the final class prediction for a new sample y_{new}, we approximate Eq. (11) by

$$P(y_{new} = 1|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{x}_{new}) = \frac{1}{R}\sum_{c=1}^{R} \Phi(z_{new}^{(c)}) \tag{14}$$

where $z_{new}^{(c)} \sim N(\boldsymbol{K}_{new\boldsymbol{\gamma}^{(c)}}(\theta^{(c)})^{\mathrm{T}}\boldsymbol{\beta}^{(c)}, 1)$, and $\theta^{(c)}$, $\boldsymbol{\beta}^{(c)}$, $\boldsymbol{\gamma}^{(c)}$ are parameter values generated at the cth MCMC iteration after the initial burn in.

For adaptive gene selection we calculate the importance of each gene by counting the number of times our model has selected them through out the MCMC chain after the initial burn in. We would expect the genes that are differentially expressed and responsible for discrimination between the classes will be selected by our model more frequently than other unexpressed or dormant genes.

4. Applications

In this section we will illustrate our proposed Bayesian kernel probit model or BKPR for binary classification problems. We demonstrate the effectiveness of our BKPR model with the help of four simulation studies. Subsequently we analyze three benchmark microarray data sets available in the public domain. In all simulations and real case studies we have generated 100,000 samples and used the first 50,000 as burn in. To avoid the problem of multimodality we tried five random starting points for each data set. The convergence of our MCMC is checked by trace plots of the generated samples and calculating the Gelman–Rubin scale reduction factor using the *Coda*() package in *R*. The calculated Gelman–Rubin scale reduction factor is reported at the bottom of the tables summarizing the results. To show that our model is not sensitive to the choice of the prior parameters (b and ω) as long as they belong to the near diffuse class of priors we tried three different prior settings for the simulation studies. Prior choice 1: $a = 10^3$, $\omega = 0.01$; Prior choice 2: $a = 10^2$, $\omega = 0.005$; Prior choice 3: $a = 10^4$, $\omega = 0.02$. The Prior choice 1 is our original choice as described in Section 3. In all real data analysis we work with our Prior choice 1. In all simulations and real case studies we used the Gaussian kernel.

We compared the performance of our BKPR model with several standard binary classification models. The first types of model are support vector machine (SVM), random forest (RF), neural network (NNET), and nearest neighbors (k-NN). These models requires an initial variable selection to have better prediction results. The top variables are selected using "BWSS" (between and within square) criteria (Dudoit et al., 2002) and then used in the respective models. The second group of models are those which can do variable selection and class prediction simultaneously like our BKPR model. Such models are the Bayesian linear probit model (BPR) (Lee et al., 2003), L1-SVM (Bradley and Mangasarian, 1998), and SVM-RFE (Guyon et al., 2002). For each of the standard methods we find out the optimal tuning and other parameter settings by leave one out cross validation technique on the training set. We also report the results of the sane Bayesian kernel probit model when applied without any variable selection. Detailed descriptions are provided in Appendix A.

4.1. Simulation study

We simulated data sets under four different simulation frameworks. For each of the first three individual simulations we generate 100 samples in the training set and 100 samples in the test set. The fourth simulation has 72 samples in total and we randomly spilt it into 36 samples in the training and 36 samples in the test set. All simulations are repeated five times. Finally we calculate the average misclassification error over all these five independent sets for each simulation. In all four simulations we have a greater number of covariates than the number of data points. The covariates are designed so that in all four simulation settings only 1% to 2% of the variables are relevant for classification and rest of the components are redundant.

1. Simulation 1: $p = 500$. Generate $\boldsymbol{x} = (x_1, x_2, \dots, x_p)^{\mathrm{T}}$. Where for Class 1: $x_i \sim N(2, 1), i = 1, \dots, 5$ and $x_i \sim N(0, 0.01)$, $i = 6, \dots, 500$ and for Class 2: $x_i \sim N(-2, 1), i = 1, \dots, 5$ and $x_i \sim N(0, 0.01), i = 6, \dots, 500$.
2. Simulation 2: $p = 1000$. Generate $\boldsymbol{x} = (x_1, x_2, \dots, x_p)^{\mathrm{T}}$. Where for Class 1: $x_i \sim N((-1)^{i+1}\sqrt{2}, 1)$, $i = 1, \dots, 10$ and $x_i \sim N(0, 0.01)$, $i = 11, \dots, 1000$, and for Class 2: $x_i \sim N((-1)^i\sqrt{2}, 1)$, $i = 1, \dots, 10$ and $x_i \sim N(0, 0.01)$, $i = 11, \dots, 1000$.
3. Simulation 3: $p = 500$. Generate $\boldsymbol{x} = (x_1, x_2, \dots, x_p)^{\mathrm{T}}$. Where for Class j, $j = 1, 2$: $x_i \sim N(\sqrt{j}, 1)$, $i = 1, \dots, 10$ and $x_i \sim N(0, 0.01)$, $i = 11, \dots, 500$.

Table 1
Average misclassification percentage for simulated data sets.

Cov	SVM	NNET	RF	k-NN	BPR	L_1-SVM	SVM-RFE	BKPR-1	BKPR-2	BKPR-3
Sim 1										
(5)					(3,18)	(4,31)	(3,27)	(4,9)	(4,8)	(5,12)
5	5.2	6.1	4.3	6.9	7.4	**4.0**	5.2	4.3	4.5	**4.0**
	(1.88)	(1.67)	(1.79)	(1.49)	(1.51)	(1.81)	(1.47)	(1.55)	(1.59)	(1.66)
10	5.2	8.4	4.1	6.6				*1.03	*1.01	*0.96
	(1.69)	(1.77)	(1.91)	(1.44)						
50	8.1	18.9	7.9	12.3						
	(1.99)	(1.79)	(1.90)	(1.64)						
100	12.3	25.2	12.3	16.9						
	(1.90)	(1.99)	(2.01)	(1.94)						
500	19.5	–	18.4	27.2						
	(2.01)	–	(2.11)	(2.14)						
Sim 2										
(10)					(7,26)	(8,41)	(10,52)	(9,16)	(9,17)	(9,20)
10	8.1	13.2	8.4	10.8	10.0	6.2	**5.0**	5.3	**5.0**	5.5
	(1.48)	(1.70)	(1.62)	(1.59)	(1.65)	(1.42)	(1.31)	(1.35)	(1.41)	(1.36)
50	10.4	9.7	4.8	13.7				*0.97	*0.96	*1.06
	(1.52)	(1.67)	(1.42)	(1.63)						
100	17.7	14.8	9.8	26.2						
	(1.74)	(1.81)	(1.83)	(1.97)						
500	26.1	–	19.3	33.7						
	(2.01)	–	(2.23)	(1.93)						
1000	34.2	–	22.8	39.1						
	(2.32)	–	(2.56)	(1.99)						
Sim 3										
(10)					(8,20)	(7,31)	(8,27)	(8,13)	(9,14)	(8,17)
5	8.1	10.4	8.7	8.3	10.6	6.2	5.6	**5.3**	**5.3**	5.6
	(1.56)	(1.42)	(1.61)	(1.59)	(1.85)	(1.62)	(1.50)	(1.61)	(1.70)	(1.58)
10	8.5	13.0	8.5	8.7				*1.05	*0.96	*1.03
	(1.49)	(1.40)	(1.52)	(1.61)						
50	16.2	21.8	9.9	15.1						
	(1.60)	(1.75)	(1.79)	(1.66)						
100	21.5	35.4	12.3	23.6						
	(1.86)	(1.99)	(2.06)	(2.12)						
500	32.6	–	18.4	36.9						
	(2.88)	–	(3.23)	(2.91)						
Sim 4										
(50)					(28,57)	(22,62)	(29,51)	(36,41)	(31,45)	(34,55)
10	5.8	8.8	5.8	8.5	7.3	3.5	3.2	**2.8**	**2.8**	3.1
	(1.98)	(1.87)	(1.89)	(1.72)	(1.92)	(1.85)	(1.96)	(1.95)	(1.91)	(1.98)
50	5.7	13.6	5.1	11.2				*1.04	*1.02	*1.04
	(2.16)	(2.23)	(1.99)	(2.05)						
100	10.1	19.3	9.3	19.5						
	(2.98)	(2.57)	(2.62)	(2.89)						
500	13.7	–	14.4	29.2						
	(2.98)	–	(2.62)	(2.89)						
2000	29.9	–	23.2	38.5						
	(3.15)	–	(3.32)	(3.65)						

The Gelman–Rubin scale reduction factors for MCMC convergence check are reported as * in the BKPR columns. The lowest misclassification errors are highlighted in bold. The numbers in parentheses are the standard deviations.

4. Simulation 4: $p = 2000$. We use the leukemia data set (Golub et al., 1999) and by "BWSS" scheme extract top 50 genes. So, x_i, $i = 1, \dots, 50$ are those microarray measurements from the leukemia data set. Generate $x_i \sim N(0, 0.01)$, $i = 51, \dots, 2000$. They behave as redundant noise variables. The class labels are kept as in the original data set. The goal will be how precisely our model can select the 50 genetic variables from the random noises.

In Table 1, we present the average misclassification error of the competing methods. We report our BKPR model result under three different sets of prior choices. They are denoted as BKPR-1, BKPR-2, and BKPR-3. The number in parentheses in the first column under "Covariates" is the number of true relevant variables. The other numbers are the top variables as selected by the "BWSS" scheme and used. This is relevant for the models which cannot make simultaneous variable selection (e.g. SVM, NNET, RF, & k-NN). The number in parentheses in the L_1-SVM, SVM-RFE, and BKPR columns reported as (,) format is of special interest to us. The first number is the median number of correct variables selected, and the second number is the

Table 2
Cross-validation error for the prostate cancer data.

Genes	SVM	NNET	RF	k-NN	BPR	L_1-SVM	SVM-RFE	**BKPR**
10	3	4	4	4	5 (36)	4 (29)	3 (20)	**2** (23)
25	3	3	6	5				
50	3	3	6	3				
100	4	5	6	6				
6546	10	–		11				

The numbers in parentheses are the median number of genes selected by the models. The lowest LOOCV error is marked in bold. The GR scale reduction factor is 1.05.

median number of total variables selected by these individual methods. Ideally we would like to include only the correct variables in a model. If a model includes too many variables then although it would be possible to capture all the true covariates but too many noise variables will reduce the prediction quality.

The misclassification results as reported in Table 1 indicate that in all the four simulation studies our BKPR model consistently outperforms other standard methods and in terms of the lowest average misclassification error. It is particularly clear that the two step strategy of selecting variables and fitting them for prediction is highly reliable. The BPR, L1-SVM, and SVM-RFE which do a simultaneous variable selection and classification are also quite effective in all four simulation studies. Our BKPR model is able to compete with BPR, L1-SVM and SVM-RFE by either matching their precision or even surpassing it at least twice. Variable selection is the big novelty of our method and from Table 1 we see our BKPR models under all prior parameter settings produce more sparse model than BPR, L1-SVM and SVM-RFE. Along with that on average our BKPR model is able to capture 80%–90% of the true covariates. Although L1-SVM and SVM-RFE can sometimes show a similar success rate on average they select a larger set of variables than our method. Hence they end up selecting too many noise variables and ends up with higher misclassification error than our BKPR. The BPR model has been developed on similar philosophy like ours but a big limitation of BPR that they restrict to only linear type of model, whereas our BKPR is much more flexible in that respect and can encompass a large class of both linear and non-linear models. Similar results under three different prior choices testifies that as long as we remain in the class of near diffuse proper prior our model is not sensitive to the prior choice. A point of caution is, ω should not be kept very large as that may result in including too many irrelevant variables in the model and thus might reduce the prediction accuracy.

4.2. *Real case studies*

4.2.1. *Prostate cancer data*

Prostate cancer is one of the most common type of cancer found in American men. According to National Cancer Institute (http://www.cancer.gov) in 2008 alone 186,320 new cases of prostate cancer were recorded in the USA and out of them 28,660 people will die. A prostate cancer can be localized or it can spread outside the organ. When the prostate cancer is spread outside the organ surgery is the only option. On the basis of the spread physicians need to make a decision whether to go for radical surgery or to go for a combination of medicine and chemotherapy. We use our BKPR to develop a prognostic model based on gene expression microarray to predict seminal vesicle (SV) invasion. We use our model to predict the SV invasion for the prostate cancer data set described in Liu et al. (2007). After initial gene filtering, log base 2 transformation and standardization (Chakraborty et al., 2004) we have 35 samples and 6546 genes (Dhanasekaran et al., 2001). Since there is no separate test set we use leave one out cross validation (LOOCV) to estimate the misclassification error.

In Table 2, we give a comparative performance of our model (BKPR), with the other standard methods for the prostate cancer data set. From the reported result in Table 2, we can clearly see that linear probit model (BPR) cannot be used here as it gives quite high misclassification error. Our BKPR model produces the lowest misclassification rate with only 2 LOOCV error. It also beats the two step procedures like SVM, NNET, RF, and k-NN in terms of prediction accuracy. The L1-SVM and SVM-RFE successfully performs simultaneous gene selection and class prediction but both result in a higher misclassification error than BKPR.

In Table 3, we list the 10 most frequently occurring genes in our model. From the full set of 6546 genes our model on average selected only 23 genes at each MCMC step. The genes marked "*" indicates that it also ranked high under the BWSS scheme. Out of our top 10 genes, 5 are also marked significant under the BWSS scoring rule. CCAAT which is flagged by our model to be important has been established to enhance the α-Methylacyl-CoA racemase (AMACR) expression level in prostate cancer cell lines (Zha and Isaacs, 2005). In a recent study by Kawakami and Nakayama (1997) it is established that the prostate-specific membrane antigen 1 is associated with the malignant prostate cells and a higher level of expression is found in most of the carcinoma cells. Both these two genes were not picked up by the BWSS criteria but effectively located by our model.

4.2.2. *Colon cancer data*

Colon cancer develops in the large intestine and rectal cancer forms in the tissues of rectum. Sometime we refer them together as colorectal cancer. In the year 2008 there are estimated 148,810 new cases of colorectal cancer in USA (http://www.cancer.gov). Out of them 49,960 will die. We apply our model on the colon cancer data (Alon et al., 1999) to

Table 3
Top 10 selected genes by our BKPR model for the prostate cancer data.

Accession no.	Gene names	Frequency (%)
*911	Lymphocyte-specific protein tyrosine kinase	23.1
*7800	Cisplatin resistance associated	23.0
8704	ESTs, Weakly similar to B34087 hypothetical protein [H.sapiens]	18.8
*4176	Hect domain and RLD 2 pseudogene 2	18.8
7602	CCAAT/enhancer binding protein (C/EBP), beta	18.1
4245	Homo sapiens DNA binding peptide mRNA, partial cds	16.5
1793	Folate hydrolase (prostate-specific membrane antigen) 1	15.6
9530	Actin binding LIM protein	15.3
9678	Small nuclear RNA activating complex, polypeptide 5, 19 kD	13.9
*8989	KIAA1501 protein	12.0

Genes that also appeared under BWSS scheme are marked by *.

Table 4
Average misclassification percentage for the colon cancer data.

Genes	SVM	NNET	RF	k-NN	BPR	L_1-SVM	SVM-RFE	**BKPR**
10	22.7 (1.84)	18.2 (2.01)	22.7 (2.45)	36.3 (2.19)	22.7 (Ripley, 1996) (1.99)	18.2 (Tadesse et al., 2005) (2.13)	16.5 (Mallick et al., 2005) (2.03)	**13.6** (Gelfand and Smith, 1990) (2.06)
25	22.7 (2.04)	22.7 (2.16)	18.2 (2.39)	22.7 (2.31)				
50	27.2 (2.11)	27.2 (2.69)	18.2 (2.57)	22.7 (2.73)				
100	27.2 (2.92)	31.4 (3.01)	22.7 (2.85)	27.2 (2.95)				
2000	36.3 (3.02)	– –	40.9 (2.75)	31.8 (3.33)				

The number in square bracket is the average number of genes selected. The number in parentheses is the standard deviation. The lowest test set error is marked in bold. The GR scale reduction factor is 0.99.

Table 5
Top 10 selected genes by our BKPR model for the colon cancer data.

Accession No.	Gene names	Frequency (%)
T47377	S-100P PROTEIN (HUMAN)	13.6
H64489	LEUKOCYTE ANTIGEN CD37 (Homo sapiens)	12.2
*M63391	Human desmin gene, complete cds	11.5
*T71025	Human (HUMAN)	10.3
M76378	Human cysteine-rich protein (CRP) gene, exons 5 and 6	10.1
M26697	Human nucleolar protein (B23) mRNA	9.4
*M22382	MITOCHONDRIAL MATRIX PROTEIN P1 PRECURSOR (HUMAN)	9.3
M23115	Homo sapiens calcium-ATPase (HK2) mRNA	8.5
*R36455	NUCLEOLAR TRANSCRIPTION FACTOR 1 (Homo sapiens)	8.5
H08393	COLLAGEN ALPHA 2(XI) CHAIN (Homo sapiens)	7.8

Genes that also appeared under BWSS scheme are marked by *.

classify the tumor and normal tissues. Here we have a total of 62 samples and 2000 genes. We randomly split it into a training set of 40 samples and a test set of 22 samples five times for checking the predictive performance of our model.

In Table 4, we list the average misclassification percentage in the test set. All standard two step methods perform poorly, on average all of them give at least 20% incorrect classifications. The misclassification rate increases as we include more genes in the models. So it is critically important to select an appropriate set of active genes for classification. The poor performance of BPR might be largely due to the fact that linear structure is unable to explain the functional relationship between the tissue class and the gene expression. Our BKPR model relaxes such linearity assumption, adaptively selects genes, and gives the lowest misclassification error. The BKPR model selects smaller set of genes than either L1-SVM or SVM-RFE and attains better accuracy.

Only 13 genes are selected on average by our model. Studying the top 10 genes selected by BKPR (Table 5), we see that there is a considerable overlap between the genes our model identifies as relevant and also the genes that are scored high under the BWSS scheme. This apparent agreement on some genes assures that through a stochastic search our model is indeed able to pinpoint the biologically significant genes and it is not just producing a random subset. We investigate the role of some newly found genes that are not tracked down by the BWSS scheme but appeared more frequently in our model and hence marked as important. The S-100P protein is documented to have an association with prostate and breast cancer (Gribenko et al., 2002). In a recent study by Huang et al. (2004) it is found that S-100P is highly expressed in colon cancer tissue than a normal tissue sample. The Human cysteine-rich protein also has a strong influence and is differentially expressed in the malignant colon cells (Suzuki et al., 2002).

Table 6
Test set error for the leukemia cancer data.

Genes	SVM	NNET	RF	k-NN	BPR	L_1-SVM	SVM-RFE	**BKPR**
10	2	4	4	4	**0** (32)	**0** (24)	1 (32)	**0** (16)
25	2	3	2	3				
50	1	1	2	1				
100	**0**	1	2	1				
3571	11	–	3	2				

The number in parentheses in the BKPR column is the average number of genes selected by our model. The lowest test set errors are marked in bold. The GR scale reduction factor is 1.03.

Table 7
Top 10 selected genes by our BKPR model for the leukemia cancer data.

Accession no.	Gene names	Frequency (%)
*4847	Zyxin	18.9
*760	CYSTATIN A	16.2
*1882	CST3 Cystatin C (amyloid angiopathy row3 and cerebral hemorrhage)	15.7
5808	Epican, Alt. Splice 11	15.6
*6218	ELA2 Elastatse 2, neutrophil	13.2
*2288	DF D component of complement (adipsin)	10.6
*1120	SNRPN Small nuclear ribonucleoprotein polypeptide N	9.1
*3252	GLUTATHIONE S-TRANSFERASE, MICROSOMAL	8.5
1834	CD33 antigen (differentiation antigen)	7.8
4499	CHRNA7 Cholinergic receptor, nicotinic, alpha polypeptide 7	7.2

Genes that also appeared in Lee are marked by *.

4.2.3. Leukemia cancer data

We use our method on the well documented leukemia data set (Golub et al., 1999). This data set contains in total 72 samples and 7129 genes. After some initial transformation, standardization and filtering (Dudoit et al., 2002) we have 3571 genes to work with. The data set is originally divided into a training set of 38 samples and the a test set of 34 samples. Here we have two types of leukemia cancer acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). Most of the standard methods have done extremely well in accurately detecting the leukemia types in the test set. So here we do not gain much in terms of classification accuracy but, we use this well known data set as a validation of continued reliable performance of our BKPR model. We would also like to see if we can come up with some new sets of genes that might open up some new investigations. From Table 6, it is clear that most of the standard methods like SVM, BPR, L1-SVM and our BKPR are able to give a perfect classification in the test set. But there is a decline in the prediction accuracy if we include more than 100 genes in the model.

With our model we have selected approximately 16 genes. In Table 7, we tabulate some of the top selected genes. Out of the top 10 genes selected by our model 7 are found to be common with the set of gene selected by Lee et al. (2003). We further investigate the role of the genes that appeared frequently in our model but were not selected by the BPR model of Lee et al. (2003). The Epican Alt. Splice 11 is shown to have a down regulatory effect on TGF-α, an important protein involved in cell cycle arrest (Harakeha et al., 2006). The CD33 antigen another gene detected by our model can be targeted for removal of malignant hematopoiesis in some patients with AML (Sievers et al., 1999).

5. Concluding remarks

In this paper we have suggested a nonparametric Bayesian probit model. We call it nonparametric since we do not assume any linear structure for the function that connects the genes with the cancer types. Our only assumption is the unknown function belongs to a broad functional space known as the reproducing kernel Hilbert space. Using this set-up we are able to provide a nonlinear structure on the unknown function thus spanning a much broader class of function. For the choice of kernel parameter θ we kept it random and tuned it adaptively as the model fits the data through MCMC. Instead of selecting the significant genes before fitting the classification model we followed an integrated approach of class prediction and gene selection with the help of the Bayesian mixture prior. In all four simulations and the three real data sets our Bayesian kernel probit model with adaptive gene selection scheme able to produce the lowest misclassification error. Although lowest misclassification cannot be guaranteed for all situations our model has some distinctive advantages over the standard Bayesian probit model, L1-SVM, and SVM-RFE. In general the standard Bayesian probit model set-up we have to tune the gene selection parameter ω so that number of genes selected is not greater than the training set sample size, since with more genes selected than number of samples we will end up with singular covariance matrix for the posterior for β, the regression coefficients. Our model tackles this problem by projecting from the sample space to the kernel space, so theoretically there is no problem if at any step we include more genes than the number of samples we have in the training set. Since out of thousands of genes only few are differentially expressed it is always suggested to keep the number of genes to be included in the model as low as possible. Typically we prefer to include only 1% of the available genes. Our BKPR is

Table A.1
Results with polynomial kernel.

Data sets	BKPR-1	BKPR-2	BKPR-3
Simulation 1	5.1	4.9	4.3
Simulation 2	6.2	5.0	6.2
Simulation 3	5.8	6.5	6.5
Simulation 4	4.2	3.6	4.5
Prostate cancer	4	4	5
Colon cancer	16.6	16.2	18.9
Leukemia cancer	0	0	1

Table A.2
Results with Gaussian kernel and no variable selection.

Data sets	Variables				
	5	10	50	100	ALL
Simulation 1	6.1	6.5	7.3	10.0	16.2
Simulation 2	7.5	6.8	9.7	18.4	33.9
Simulation 3	8.8	10.5	13.3	19.2	34.1
Simulation 4	5.9	4.6	7.8	9.5	25.2
Prostate cancer	4	4	3	4	9
Colon cancer	19.3	18.2	17.4	21.6	32.2
Leukemia cancer	1	0	1	2	8

a non-linear extension to the BPR model. Unlike L1-SVM and SVM-RFE our BKPR models have a probabilistic framework. So instead of hard thresholding we are able to calculate the full posterior distribution of the probability of the class labels. Getting the full distribution is much more informative than a point estimate, as now we can also quantify the amount of uncertainty with each prediction. From Table A.2 in Appendix A we see that simply fitting the Bayesian kernel probit model without the gene selection scheme produces not very impressive results. This further vindicates our claim that integrated variable selection is essential for better prediction performance.

Here we have used a Gaussian kernel with the kernel parameter θ random. A better and more flexible classification can be obtained if we can identify an optimal kernel function. All the top genes selected by our model carry some biological relevance. But, in the random search using Bernoulli prior we also do include some unnecessary genes which to some extent reduces the efficacy of our model. This can be reduced if we can incorporate some prior information about the available genes and their biological role play towards a particular cancer growth. The gene–gene interaction data and genetic networks can be used to construct better informative priors for the gene selection. Several developments in nonlinear models (Zhou et al., 2003) establishes a link between the genes and the odds of the disease. In the kernel trick though we effectively reduce the dimension and achieve better prediction performance but unfortunately in doing that we lose that kind of direct interpretation. However, we can develop statistical tests for the entire nonlinear function $f()$ (Liu et al., 2007) based on calculated Bayes factor and do hypothesis testing. This can be a nice alternative to test the nonlinear gene effect. Instead of the whole microarray if we have genes from a set of known biological pathways, these non-parametric tests will be really important in understanding the overall pathway effects. This type of problem will be handled in future research. With the increased understanding of association between gene expression and a disease, modern medicine is more and more relying on customized drugs. Drugs that target some specific genes of an individual that contribute to the growth of cancer for a particular patient. A better fast gene selection method carries lot of promise in that regard.

Appendix

A.1. Tuning and other parameters of the standard methods:

- We use the *nnet*() function in *R* to fit the neural network models. The number of hidden nodes is selected by LOOCV on the training set. Most of the time working with 50 hidden nodes proved to be sufficient.
- For fitting a random forest we use the *randomForest*() function in *R* with one thousand boosted trees. The out of bag error rate stabilizes with 10000 boosted trees.
- We use the *svm*() function in *R*, with the default value for the RBF kernel parameter which is $\gamma = 1/n$. Several other values of the RBF parameter on a range of (0, 1) are tried but most of the time the best result is obtained using the default setting. The C-constant of the regularization term in the Lagrange formulation of SVM, popularly known as the tuning parameter is chosen by an equispaced grid search on an interval (0, 3) and then picking up the C that corresponds to the best result. For SVM-RFE we used the *varSel.svm.rfe*() function in *R*.
- For Bayesian probit model or BPR we chose $\omega = 0.005$ as suggested in the original paper (Lee et al., 2003).

References

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

Alizadeh, A.A., Eisen, M.B., Davis, R.E., Ma, C., Lossos, I.S., et al., 2000. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature 403, 503–511.

Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra, S., Mack, D., Levine, A.J., 1999. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proceedings of National Academy of Science 96, 6745–6750.

Aronszajn, N., 1950. Theory of reproducing kernels. Transactions of the American Mathematical Society 68, 337–404.

Bellman, R.E., 1961. Adaptive Control Processes. Princeton University Press, Princeton.

Bradley, P., Mangasarian, O., 1998. Feature selection via concave minimization and support vector machines. In: Proceedings of the 15th International Conference on Machine Learning.

Belitza, C., Lang, S., 2008. Simultaneous selection of variables and smoothing parameters in structured additive regression models. Computational Statistics and Data Analysis 53 (1), 61–81.

Breiman, L., 2001. Random forests. Machine Learning 45, 5–32.

Chakraborty, S., Ghosh, M., Maiti, T., Tewari, A., 2004. Bayesian neural networks for bivariate binary data: An application to prostate cancer study. Statistics in Medicine 24, 3645–3662.

Dhanasekaran, S.M., Barrette, T.R., Ghosh, D., Shah, R., Varambally, S., Kurachi, K., Pienta, K.J., Rubin, M.A., Chinnaiyan, A.M., 2001. Delineation of prognostic biomarkers in prostate cancer. Nature 412, 822–826.

Dudoit, S., Fridlyand, J., Speed, T.P., 2002. Comparison of discrimination methods for the classification of tumors using gene expression data. Journal of the American Statistical Association 97, 77–87.

Duggan, D.J., Bittner, M., Chen, Y., Meltzer, P., Trent, J.M., 1999. Expression profiling using cDNA microarrays. Nature Genetics 21, 10–14.

Gelfand, A., Smith, A.F.M., 1990. Sampling-based approaches to calculating marginal densities. Journal of the American Statistical Association 85, 398–409.

George, E., McCulloch, R., 1993. Variable selection via Gibbs sampling. Journal of the American Statistical Association 88, 881–889.

Golub, T.R., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C., Lander, E., 1999. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. Science 286, 531–537.

Gribenko, A.V., Guzmán-Casado, M., Lopez, M.M., Makhatadze, G.I., 2002. Conformational and thermodynamic properties of peptide binding to the human S100P protein. Protein Science 11, 1367–1375.

Guyon, I., Weston, J., Barnhill, S., Vapnik, V., et al., 2002. Gene selection for cancer classification using support vector machines. Machine Learning 46, 389–422.

Harakeha, S., Diab-Assafa, M., Niedzwieckib, A., Khalifea, J., Abu-El-Ardata, K., Rathb, M., 2006. Apoptosis induction by Epican Forte in HTLV-1 positive and negative malignant T-cells. Leukemia Research 30, 869–881.

Huang, E., Fuentes, M., Arumugam, T., Logsdon, C., 2004. The RAGE ligand, S100P, has increased expression in colon cancer. Journal of the American College of Surgeons 199, 18.

Kawakami, M., Nakayama, J., 1997. Enhanced expression of prostate-specific membrane antigen gene in prostate cancer as revealed by in situ hybridization. Cancer Research 57, 2321–2324.

Kim, S., Tadesse, M.G., Vannucci, M., 2006. Variable selection in clustering via Dirichlet process mixture. Biometrika 93, 877–893.

Lee, K.E., Naijun, S., Dougherty, E.R., Vannucci, M., Mallick, B.K., 2003. Gene selection: A Bayesian variable selection approach. Bioinformatics 19, 90–97.

Liu, D., Lin, X., Ghosh, D., 2007. Semiparametric regression of multi-dimensional genetic pathway data: Least squares kernel machines and linear mixed models. Biometrics 63, 1079–1088.

Mallick, B.K., Ghosh, D., Ghosh, M., 2005. Bayesian classification of tumors using gene expression data. Journal of the Royal Statistical Society, B 67, 219–232.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E., 1953. Equations of state calculations by fast computing machines. Journal of Chemical Physics 21, 1087–1092.

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

Parzen, E., 1970. Statistical inferences on time series by RKHS methods. In: Proceedings of the 12th Biennial Seminar, Canadian Mathematical Congress, Montreal, Canada, pp. 1–37.

Ripley, B.D., 1996. Pattern Recognition and Neural Networks. Cambridge University Press, pp. 337–341.

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

Schena, M., Shalon, D., Davis, R., Brown, P., 1995. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. Science 270, 467–470.

Sievers, E.L., Appelbaum, F.R., Spielberger, R.T., Forman, S.J., Flowers, D., Smith, F.O., Berger, M.S., Bernstein, I.D., 1999. Selective ablation of acute myeloid leukemia using antibody-targeted chemotherapy: A phase I study of an anti-CD33 calicheamicin immunoconjugate. Blood 93, 3678–3684.

Suzuki, H., Gabrielson, E., Chen, W., Anbazhagan, R., Engeland, M., Weijenberg, M.P., Herman, J.G., Baylin, S.B., 2002. A genomic screen for genes upregulated by demethylation and histone deacetylase inhibition in human colorectal cancer. Nature Genetics 31, 141–149.

Tadesse, M.G., Sha, N., Vannucci, M., 2005. Bayesian variable selection in clustering high-dimensional data. Journal of the American Statistical Association 100, 602–617.

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

Vapnik, V.N., 1995. The Nature of Statistical Learning Theory. Springer, New York.

Wahba, G., 1990. Spline Models for Observational Data. SIAM, Philadelphia.

Wang, L., Shen, X., 2007. On L1-norm multi-class support vector machines: Methodology and theory. Journal of the American Statistical Association 102, 583–594.

Zha, S., Isaacs, W.B., 2005. A nonclassic CCAAT enhancer element binding protein binding site contributes to alpha-Methylacyl-CoA racemase expression in prostate cancer. Molecular Cancer Research 3, 110–118.

Zhou, X., Wang, X., Dougherty, E.R., 2003. Missing-value estimation using linear and non-linear regression with Bayesian gene selection. Bioinformatics 19, 2302–2307.

Zou, H., Hastie, T., 2005. Regularization and variable selection via the elastic net. Journal of Royal Statistical Society B 67 (2), 301–320.

Zou, H., Yuan, M., 2008. Regularized simultaneous model selection in multiple quantiles regression. Computational Statistics and Data Analysis 52 (12), 5296–5304.