Bias-corrected Hierarchical Bayesian Classification with a Selected Subset of High-dimensional Features

Longhai Li*

15 July 2011

Abstract: Class prediction based on high-dimensional features has received a great deal of attention in many areas. For example, biologists are interested in using microarray gene expression profiles for diagnosis or prognosis of a certain disease (eg, cancer). For computational and other reasons, it is necessary to select a subset of features before fitting a statistical model, by looking at how strongly the features are related to the response. However, such feature selection procedure will result in overconfident predictive probabilities for future cases, because the signal-noise ratio in the retained features has been exacerbated by the feature selection. In this paper, we develop a hierarchical Bayesian classification method that can correct for this feature selection bias. Our method (called BCBCSF) uses the partial information from the feature selection procedure, in addition to the retained features, to form a correct (unbiased) posterior distribution of certain hyperparameters in the hierarchical Bayesian model that control the signal-noise ratio of the data set. We take MCMC approach to infer the model parameters. MCMC samples are then used to make predictions for future cases. Due to the simplicity of models, the inferred parameters from MCMC are easy to interpret, and the computation is very fast. Our simulation studies and tests with two real microarray data sets related to complex human diseases show that BCBCSF predicts better than two widely used high-dimensional classification methods — PAM and DLDA. An R package called `BCBCSF` for the method described here is available from http://math.usask.ca/~longhai/software/BCBCSF and CRAN.

Short title: Bias-corrected Bayesian Classification with Selected Features
Key words: BCBCSF, optimistic bias, high-dimensional classification, feature selection

*Department of Mathematics and Statistics, University of Saskatchewan, Saskatoon, Saskatchewan, S7N5E6, CANADA. Email: `longhai@math.usask.ca`. Web: http://math.usask.ca/~longhai.

1 Introduction

Nowadays new technologies can easily measure high-dimensional *features* (also known as covariates, explanatory variables) of objects/subjects (called *cases* generally). For example, microarray technologies can simultaneously measure expression levels of thousands of genes of tissues. An interesting use of these high-dimensional measurements is to predict a certain *categorical* characteristic (also known as class label, response variable) of a case. For example, a response may be an indicator of whether a kind of disease is present in a tissue, or different types of a disease. Diagnosis methods by looking at the differences in gene expression profiles from different classes may detect complex diseases at earlier stages than traditional methods, and therefore have received a great deal of attention since about a decade ago. Univariate feature selection by looking at some simple score, such as correlation coefficient or F-statistic, is often applied to high-dimensional data sets before fitting a model for reducing computation time and/or improving prediction accuracy. For example, it was used by most winning entries of 2003 NIPS feature selection competition (Guyon et al., 2006).

Unfortunately, feature selection will introduce optimistic bias into statistical inference, because the signal-noise ratio of the data set is lifted by the feature selection procedure. An extreme example is that all features are irrelevant to a response, but the selected features will still appear fairly predictive to the response, which is, however, wholly made by chance (see Ambroise and McLachlan, 2002, for a demonstration). We will call this problem *feature selection bias*. The effect on predictions caused by feature selection bias is that the predictive probabilities will be *overconfident*. For example, for a group of future cases, the predictive probabilities of their responses being 1 are between 0.9 and 1, but the actual fraction of their responses being 1 is only 0.7. In other words, the predictive probabilities become more extreme than what they actually are.

Correcting for feature selection bias in estimating predictive probabilities for future cases is important in practice. Interestingly, Singhi and Liu (2006) empirically shows that feature

selection doesn't have effect on error rate of a classification method. For example if the response is binary, the predictive probabilities based on a selected subset of features for those cases on the classification boundary are still close to 1/2, therefore the predicted values of response variables for test cases by thresholding at 1/2 are the same after feature selection. However, error rate is useful only when losses incurred from different types of prediction errors are the same. In practice, this is often not the case. For example, classifying a patient with a disease into non-disease may cause much more loss than the opposite error. In such situations, thresholding overconfident predictive probabilities at a value different from 1/2 for binary classification may result in higher loss on average.

To our knowledge, feature selection bias problem in high-dimensional classification problems hasn't drawn much attention in literature. However, some relevant questions have been discussed. Feature selection bias in cross-validation estimate of error rate has been noticed by many researchers working on classification with gene expression data. Ambroise and McLachlan (2002); Raudys et al. (2005) and some others have found that if a cross-validation evaluation of classification algorithms is applied on a data set containing only a subset of features selected beforehand based on the whole data set, the error rate could be misleadingly small. It is therefore suggested that the feature selection should be "internal" to the cross-validation procedure, ie, the feature selection should be redone for each splitting of data set into training and test sets. However, this is only a proper method for evaluating a possibly poorly-calibrated classification procedure, but not a method for giving better predictive probabilities for future cases. Even though it wasn't stated explicitly by the authors, the method called prediction analysis for microarrays (PAM) by Tibshirani et al. (2002) has the effect of correcting for feature selection bias. In PAM, the fewer features are retained, the stronger shrinkage for signals of retained features is imposed. However, we will argue that this method has the problem of reducing discriminative power of retained features. It is worth noting that feature selection bias on confidence region estimation has been long recognized in regression problems. Hurvich and Tsai (1990) and Zhang (1992)

respectively used Monte Carlo simulation and theoretical analysis to show that after feature selection the actual coverage rate of confidence regions for regression coefficients is smaller than the nominal probability. To solve the bias problem, some authors suggested that the feature selection and parameter estimation be performed separately on different cases. In the context of high-dimensional data discussed here, this quick solution is undesirable because the number of training cases is usually very small (very often only *tens*), therefore using fewer cases for fitting models sacrifices the discriminative power, resulting in better calibrated but less accurate predictions for future cases. Recently, Shen et al. (2004) and Wang and Lagakos (2009) propose some methods that use optimal approximation and perturbation/permutation of response values to estimate the mean and variance of the least square estimator of regression coefficients, for finding better calibrated confidence region. However, it seems impossible to apply such methods for finding better calibrated predictive probabilities for classification.

In this paper, we develop a hierarchical Bayesian classification method with the goal of reporting *better calibrated predictive probabilities* for future cases, after correcting for the feature selection bias. A hierarchical Bayesian model is proposed to model high-dimensional features, in which all features are linked with hyperparameters controlling the overall signal-noise ratio. Based on such a Bayesian model, we can naturally incorporate the information that a certain number (probably large) of features were omitted before we obtained the retained features to form a corrected posterior of the signal-noise ratio. An intuitive explanation of the correction method is that the upward bias in signal-noise ratio implied by the retained data will be canceled by the information that many weakly relevant features are present prior to obtaining the retained features. This general correction method was proposed by Li et al. (2008) and used in a naive Bayes model for binary features. However, their model is inappropriate for practical high-dimensional classification primarily due to lack of a sparse prior. Here we apply this general correction method to a Bayesian model assigning heavy-tailed t prior for high-dimensional signals that is appropriate for practi-

cal high-dimensional classification problems, for example with gene expression profiles. We also note that this correction method is indeed a special case of the very general principles proposed by Dawid and Dickey (1977) for statistical inference with selectively reported data.

We will call our classification method by BCBCSF — **b**ias-**c**orrected **B**ayesian **c**lassification with **s**elected **f**eatures. In Section 2, we present the details of BCBCSF. In Sections 3 and 4, we use synthetic and two real microarray data sets related to human cancer to demonstrate BCBCSF by comparing to two other similar and widely used methods — DLDA and PAM. DLDA is a method with totally no correction for selection bias, and PAM is a method that we think overcorrects for the bias. We will show that BCBCSF is a valuable alternative to these two methods with practical values.

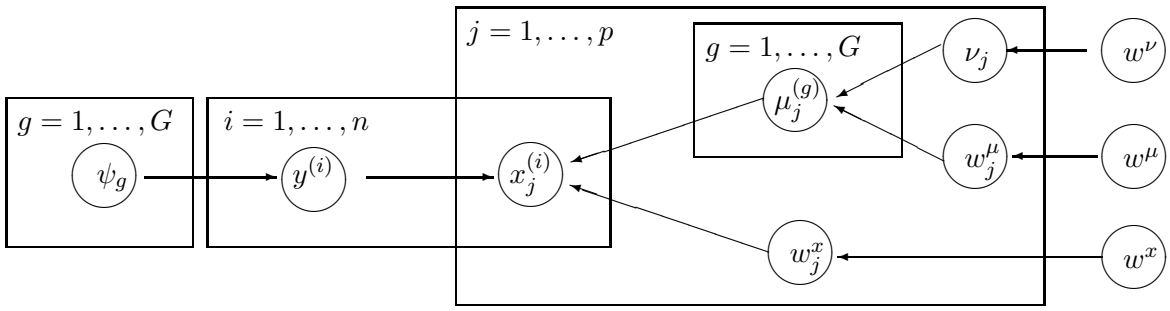
2 The Methodology

2.1 A Hierarchical Bayesian Model for High-dimensional Data

We are interested in predicting a categorical response variable y, which is sometimes called class label, given the information of a set of features $x_1, \ldots, x_p$, for example a gene expression profile of a tissue. Here we assume that y can take integer value from 1 to G, and all the x_i's are continuous. The observation for case i is denoted by $y^{(i)}$ and $x_1^{(i)}, \ldots, x_p^{(i)}$. Given parameter $\psi_1, \ldots, \psi_G$ (collectively written as $\boldsymbol{\psi}$), the response variable $y^{(i)}$ takes a value $g \in \{1, \ldots, G\}$ with a probability of ψ_g. Conditional on $y^{(i)} = g$, the predictor variables $x_1^{(i)}, \ldots, x_p^{(i)}$ are assumed to be independent, and $x_j^{(i)}$ is distributed with $N(\mu_j^{(g)}, w_j^x)$.

The assumptions of independence between features and the same variances across different classes may not be realistic. Considering more realistic models is indeed interesting and important. However, because the number of observations in biomedical data, such as gene expression data, is typically *very small* (often only tens), avoidance of overfitting problems becomes difficult in more flexible models, such as linear models. Therefore, methods based on simple models often outperform sophisticated methods based on more flexible models. A strong evidence was given in the seminal paper in high-dimensional classification by Dudoit,

Figure 1: Graphical representation of the hierarchical Bayesian classification model. A list of notations with brief explanations in this Figure can be found in Appendix A.



Fridlyand, and Speed (2002). They compared their proposed diagonal linear discriminant analysis method (DLDA, which assumes exactly the same model as we outline above), linear discriminant analysis methods (FLDA, LDA) and diagonal quadratic discriminant analysis method (DQDA, which considers *different* variances across classes), decision tree, nearest neighbors (NN), as well as a weighted voting scheme. DLDA is found to perform the best in 3 out of 4 data sets, and makes only 1 misclassification in lymphoma data set, very close to the best, 0, by NN. The assumptions of feature independence and the same variances are also assumed in another highly influential method, called prediction analysis for microarrays (PAM) developed by Tibshirani et al. (2002). Finally we believe that the popularity of DLDA and PAM in biologists, is partially due to easy interpretation of the models and results, and perhaps also fast computation. Therefore in the method reported in this paper we still work under these two assumptions.

Now we will describe how to assign priors to ψ_g, $\mu_j^{(g)}$ and w_j^x using a hierarchical method. Before we give detailed explanation of the priors, we lay out the model for data and priors by the following equations, and display it graphically by Figure 1. For the ease of reference in reading this paper, we also give a list of notations in Appendix A.

$$P(y^{(i)} = g \,|\, \boldsymbol{\psi}) \quad = \quad \psi_g, \quad \text{for } g = 1, \ldots, G\,, \tag{1}$$

$$\psi_1, \ldots, \psi_G \quad \sim \quad \text{Dirichlet}(c_1, \ldots c_G), \tag{2}$$

$$x_j^{(i)} \,|\, y^{(i)} = g, \mu_j^{(g)}, w_j^x \quad \sim \quad N(\mu_j^{(g)}, w_j^x), \quad \text{for } j = 1, \ldots, p\,, \tag{3}$$

$$\mu_j^{(1)}, \ldots, \mu_j^{(G)} \,|\, \nu_j, w_j^\mu \overset{\text{IID}}{\sim} N(\nu_j, w_j^\mu), \quad \text{for } j = 1, \ldots, p, \tag{4}$$

$$\nu_1, \ldots, \nu_p \,|\, w^\nu \overset{\text{IID}}{\sim} N(0, w^\nu), \tag{5}$$

$$w^\nu \sim \text{IG}\left(\frac{\alpha_0^\nu}{2}, \frac{\alpha_0^\nu w_0^\nu}{2}\right), \tag{6}$$

$$w_1^\mu, \ldots, w_p^\mu \,|\, w^\mu \overset{\text{IID}}{\sim} \text{IG}\left(\frac{\alpha_1^\mu}{2}, \frac{\alpha_1^\mu w^\mu}{2}\right), \tag{7}$$

$$w^\mu \sim \text{IG}\left(\frac{\alpha_0^\mu}{2}, \frac{\alpha_0^\mu w_0^\mu}{2}\right), \tag{8}$$

$$w_1^x, \ldots, w_p^x \,|\, w^x \overset{\text{IID}}{\sim} \text{IG}\left(\frac{\alpha_1^x}{2}, \frac{\alpha_1^x w^x}{2}\right), \tag{9}$$

$$w^x \sim \text{IG}\left(\frac{\alpha_0^x}{2}, \frac{\alpha_0^x w_0^x}{2}\right). \tag{10}$$

For simplicity of presentation below, we will use $i{:}j$ to denote the vector of integers from i to j ($i \leq j$), and use $\boldsymbol{A}_{i:j}$ to denote the collection of objects $A_i, \ldots, A_j$, similarly when we use a vector in superscript. In addition, we will use bold-faced letters to denote collections.

We first talk about the lowest level priors for parameters in data distributions. A natural choice of prior for $\boldsymbol{\psi}$ is a Dirichlet distribution with parameters $c_1, \ldots, c_G$. The ratios between $c_1, \ldots, c_G$ indicates our prior preference of which class is more likely, and the sum of $c_{1:G}$ represents how certain we are on this preference. More discussions of this prior can be found from Gelman et al. (2004), pg 83. We can choose them all equal to 1 (or 0.1) to reflect that we have weak prior information of which class is more likely, and let the posterior class probabilities depend mostly on the features. The mean parameters $\boldsymbol{\mu}_j^{(1:G)}$ of the jth feature across G classes are assigned independent normal distributions with mean ν_j and variance w_j^μ (see (4)), which will be assigned higher-level prior. This prior reflects our belief that the mean parameters $\boldsymbol{\mu}_j^{(1:G)}$ vary around a common expression level ν_j — the overall expression level of the jth gene, and will have the effect of shrinking $\boldsymbol{\mu}_j^{(1:G)}$ to ν_j in posterior sampling. The variances $\boldsymbol{w}_{1:p}^x$ for features *within* classes are assigned with the conjugate Inverse-Gamma (IG) prior with shape parameter $\alpha_1^x/2$ and rate parameter $\alpha_1^x\, w^x/2$ (see (9)). This prior is quite standard and convenient for modeling variances, and called scaled-χ^2 distribution;

more explanations of the meanings of its parameters will also be given in next paragraph in discussing $\boldsymbol{w}^{\mu}_{1:p}$. We may set α^{x}_{1} to a fairly large value (for example 10) to reflect our belief that $w^{x}_{1:p}$ are fairly close, but have some variability.

The hyperparameter w^{μ}_{j} represents the signal level of the jth feature for predicting the response. At a higher level, we assign $\boldsymbol{w}^{\mu}_{1:p}$ the conjugate Inverse-Gamma (IG) prior with shape parameter $\alpha^{\mu}_{1}/2$ and rate parameter $\alpha^{\mu}_{1}\, w^{\mu}/2$ (see (7)), in which α^{μ}_{1} and w^{μ} control respectively the sparsity and "mean" (though not exactly the prior mean) of $\boldsymbol{w}^{\mu}_{1:p}$ — the smaller α^{μ}_{1} is, the more sparse the $\boldsymbol{w}^{\mu}_{1:p}$ are; the larger w^{μ} is, the larger values $\boldsymbol{w}^{\mu}_{1:p}$ can be. If we integrate w^{μ}_{j} away, it will lead to a multivariate t distribution for $\boldsymbol{\mu}^{(1:G)}_{j}$ with degree freedom α^{μ}_{1} and scale parameter $\sqrt{w^{\mu}}$. As is well-known, t distribution has heavier tails than normal distribution, and therefore is more suitable to model a group of parameters, most of which are near 0, but a few are extraordinarily large. *This is what we believe for the signal levels of a large number of features.* Explaining with this hierarchy, we assign a different variance w^{μ}_{j} for different features to have the effect of model-based feature selection in fitting the signal levels of a large number of features. The fitted model will have a few w^{μ}_{j} very large while shrinking others to very small. We could make α^{μ}_{1} a hyperparameter, and let it learn from the data. However, this will make the posterior distribution have many local modes, which may result in even worse MCMC fitting results than fixing it to a reasonable value. We therefore leave α^{μ}_{1} to be fixed by an expert for controlling desired sparsity according to his/her belief about the signals. Our experiences indicate that a value of α^{μ}_{1} between 0.5 and 4 works reasonably well for most gene expression data sets related to complex diseases. The prediction results may vary a little bit (such as 1 or 2 more or fewer errors) but generally the results are fairly robust to the choice. In practice, one can also use cross-validation to choose it, since the computation for our method is fast. The insensitivity also explains why we don't choose the family of spike-and-slab priors — mixtures of a continuous distribution and a point mass at 0. The spike-and-slab priors indeed express well our belief for the signals, and is very attractive because it has effect of shrinking many small signals to exact

0. However, the implementation of posterior sampling for real data sets is very difficult. The difficulty arises from the need of calculating many intractable marginalized likelihoods, and more critically that marginalized likelihood is very sensitive to the choice of the width of the continuous distribution, see a relevant discussion in Shafer (1982) and the references therein. Furthermore, the effect of shrinking small signals to exact 0 is achieved in our method by a univariate feature selection that omits features with small signals. The choice of width of continuous part in spike-and-slab priors becomes the choice of the number of retained features, which is more transparent to practitioners. At last the common means $\boldsymbol{\nu}_{1:p}$ are given a normal distribution with mean 0 (assuming that $\boldsymbol{x}_j^{(1:n)}$ have been centralized) and variance w^ν.

The "mean" parameters w^μ and w^x respectively for $\boldsymbol{w}^\mu_{1:p}$ and $\boldsymbol{w}^x_{1:p}$, and variance w^ν of $\boldsymbol{\nu}_{1:p}$ are treated as higher-level hyperparameters, assigned with diffuse IG distribution with fixed parameters α_0 and w_0, which may be different for three groups of parameters (see equations (8), (10), (6)). The values of α_0 and w_0 is to be fixed by an analyst, usually given small positive values (eg, 0.5 and 0.05), defining a very diffuse prior.

When w^μ is larger, more features have large variance amongst $\boldsymbol{\mu}_j^{(1:G)}$, therefore more features are useful in predicting the response. When w^x is larger, the noises in all features are larger. Therefore, w^μ and w^x control respectively the overall signal and noise levels, and w^μ/w^x indicates the overall signal-noise ratio.

2.2 Predictions and Posterior Sampling Given All Features

Suppose we want to predict the response y^* of a test case with feature values denoted by $\boldsymbol{x}^*_{1:p}$. Following Bayes rule, the predictive probability of $y^* = g$ is

$$P(y^* = g \,|\, \boldsymbol{x}^*_{1:p}, \boldsymbol{x}^{(1:n)}_{1:p}, \boldsymbol{y}^{(1:n)}) = \frac{P(y^* = g \,|\, \boldsymbol{y}^{(1:n)})\, P(\boldsymbol{x}^*_{1:p} \,|\, \boldsymbol{x}^{(1:n)}_{1:p}, y^* = g, \boldsymbol{y}^{(1:n)})}{\sum_{g=1}^{G} P(y^* = g \,|\, \boldsymbol{y}^{(1:n)})\, P(\boldsymbol{x}^*_{1:p} \,|\, \boldsymbol{x}^{(1:n)}_{1:p}, y^* = g, \boldsymbol{y}^{(1:n)})}. \quad (11)$$

To compute (11), we need to compute the numerator for all $g = 1, \ldots, G$, then divide them by their sum, which gives the denominator. The first factor in the numerator of (11) can be computed by Pòlya urn scheme: $P(y^* = g \,|\, \boldsymbol{y}^{(1:n)}) = \dfrac{n_g + c_g}{n + \sum_g c_g}$, where n_g is the number of training cases in class g. The second factor can be written as:

$$P(\boldsymbol{x}^*_{1:p} \,|\, \boldsymbol{x}^{(1:n)}_{1:p}, y^*{=}g, \boldsymbol{y}^{(1:n)}){=}\int P(\boldsymbol{x}^*_{1:p} \,|\, \boldsymbol{\mu}^{(g)}_{1:p}, \boldsymbol{w}^x_{1:p}, y^*{=}g)\, P(\boldsymbol{\mu}^{(g)}_{1:p}, \boldsymbol{w}^x_{1:p} \,|\, \boldsymbol{x}^{(1:n)}_{1:p}, \boldsymbol{y}^{(1:n)})\, d\boldsymbol{\mu}^{(g)}_{1:p}\, d\boldsymbol{w}^x_{1:p}, \tag{12}$$

where,

$$P(\boldsymbol{x}^*_{1:p} \,|\, \boldsymbol{\mu}^{(g)}_{1:p}, \boldsymbol{w}^x_{1:p}, y^* = g) = (2\pi)^{p/2} \exp\left(-\frac{1}{2}\sum_{j=1}^{p}\left[\frac{(x^*_j - \mu^{(g)}_j)^2}{w^x_j} + \log(w^x_j)\right]\right). \tag{13}$$

The above classification rule is very similar to DLDA and PAM. What's different for our Bayesian approach is that we average the predictive probabilities over plausible values of parameters in light of data.

We will use Markov chain Monte Carlo (MCMC) (see eg Neal, 1993, and references therein) to approximate the integral in (12) by averaging the quantity in (13) over a pool of samples of $\boldsymbol{\mu}^{(1:G)}_{1:p}$ and $\boldsymbol{w}^x_{1:p}$ drawn by simulating a Markov chain. To draw samples of $\boldsymbol{\mu}^{(1:G)}_{1:p}$ and $\boldsymbol{w}^x_{1:p}$, we can draw samples of ($\boldsymbol{\mu}^{(1:G)}_{1:p}$, $\boldsymbol{\nu}_{1:p}$, $\boldsymbol{w}^\mu_{1:p}$, $\boldsymbol{w}^x_{1:p}$ w_μ, w_ν) from their joint posterior distribution that is proportional to:

$$\prod_{j=1}^{p}\Big(P(\boldsymbol{x}^{(1:n)}_j \,|\, \boldsymbol{\mu}^{(1:G)}_j, w^x_j, \boldsymbol{y}^{(1:n)})\, P(\boldsymbol{\mu}^{(1:G)}_j \,|\, \nu_j, w^\mu_j)\, P(\nu_j \,|\, w^\nu)\, P(w^\mu_j \,|\, w^\mu)\, P(w^x_j \,|\, w^x)\Big) \times$$
$$P(w^\mu)\, P(w^\nu)\, P(w^x), \tag{14}$$

where, $P(\boldsymbol{x}^{(1:n)}_j \,|\, \boldsymbol{\mu}^{(1:G)}_j, w^x_j, \boldsymbol{y}^{(1:n)}) = (2\pi)^{n/2} \exp\left(-\frac{1}{2}\sum_{i=1}^{n}\left[\frac{(x^{(i)}_j - \mu^{(y^{(i)})}_j)^2}{w^x_j} + \log(w^x_j)\right]\right)$, and other probability density functions can be found from model descriptions (3) - (10).

The conditional distributions needed to apply Gibbs sampling to (14) are given below:

$$\mu_j^{(g)} \,|\, \boldsymbol{x}^{(1:n)}, \boldsymbol{y}^{(1:n)}, w_j^\mu, \nu_j, w_j^x \;\sim\; N\left(\frac{\nu_j/w_j^\mu + \ddot{x}_j^{(g)}/w_j^x}{1/w_j^\mu + n_g/w_j^x},\; \frac{1}{1/w_j^\mu + n_g/w_j^x}\right), \tag{15}$$

$$w_j^x \,|\, \boldsymbol{x}^{(1:n)}, \boldsymbol{y}^{(1:n)}, w^x, \boldsymbol{\mu}_j^{(1:G)} \;\sim\; \mathrm{IG}\left(\frac{\alpha_1^x + n}{2},\; \frac{\alpha_1^x\, w^x + \sum_{i=1}^n (x_j^{(i)} - \mu_j^{(y^{(i)})})^2}{2}\right), \tag{16}$$

$$w_j^\mu \,|\, w^\mu, \nu_j, \boldsymbol{\mu}_j^{(1:G)} \;\sim\; \mathrm{IG}\left(\frac{\alpha_1^\mu + G}{2},\; \frac{\alpha_1^\mu\, w^\mu + \sum_{g=1}^G (\mu_j^{(g)} - \nu_j)^2}{2}\right), \tag{17}$$

$$\nu_j \,|\, \boldsymbol{\mu}_j^{(1:G)}, w_j^\mu, w_\nu \;\sim\; N\left(\frac{\ddot{\mu}_j/w_j^\mu}{1/w^\nu + G/w_j^\mu},\; \frac{1}{1/w^\nu + G/w_j^\mu}\right), \tag{18}$$

$$w^\nu \,|\, \boldsymbol{\nu}_{1:p} \;\sim\; \mathrm{IG}\left(\frac{\alpha_0^\nu + p}{2},\; \frac{\alpha_0^\nu\, w_0^\nu + \sum_{j=1}^p \nu_j^2}{2}\right), \tag{19}$$

$$P(w^\mu \,|\, \boldsymbol{w}_{1:p}^\mu) \;\propto\; (w^\mu)^{\frac{p\,\alpha_1^\mu - \alpha_0^\mu}{2} - 1} \exp\left\{-\sum_{j=1}^p \frac{\alpha_1^\mu}{2w_j^\mu}\, w^\mu - \frac{\alpha_0^\mu\, w_0^\mu}{2w^\mu}\right\}, \tag{20}$$

$$P(w^x \,|\, \boldsymbol{w}_{1:p}^x) \;\propto\; (w^x)^{\frac{p\,\alpha_1^x - \alpha_0^x}{2} - 1} \exp\left\{-\sum_{j=1}^p \frac{\alpha_1^x}{2w_j^x}\, w^x - \frac{\alpha_0^x\, w_0^x}{2w^x}\right\}, \tag{21}$$

where $\ddot{x}_j^{(g)} = \sum_{\{i:y^{(i)}=g\}} x_j^{(i)}$, and $\ddot{\mu}_j = \sum_{g=1}^G \mu_j^{(g)}$. The distributions (15) - (19) can be sampled directly with standard methods. The conditional distributions of w^μ and w^x is sampled with Metropolis-Hasting method with Gaussian proposal, applied to the posterior of $(\log(w^\mu), \log(w^x))$.

From the expressions in (15), our Bayesian method shrinks the MLE estimates of $\boldsymbol{\mu}_j^{(1:G)}$ toward the common mean ν_j. We therefore shrink the small signals, as PAM does. What's different in our Bayesian method is that we shrink $\boldsymbol{\mu}_j^{(1:G)}$ differently for different features with w_j^μ, therefore it doesn't punish a lot the real signals.

2.3 Bias-corrected Posterior Given a Selected Subset of Features

When the number of available features — p is very large (for example as large as tens or a hundred of thousands in genomic data), training the model based on all the features with MCMC is slow, and therefore we intend to select a smaller subset of features by some simple

univariate score measuring the usefulness of a feature in predicting the response, denoted by $R(\boldsymbol{x}^{(1:n)}, \boldsymbol{y}^{(1:n)})$. More importantly, feature selection can eliminate the influence of noises in useless features. In high-dimensional problems, the accumulation of noises in a large number of useless features may effectively conceal the signals of a comparatively much smaller subset of useful features.

A widely used feature selection score is F-statistic:

$$R_F(\boldsymbol{x}^{(1:n)}, \boldsymbol{y}^{(1:n)}) = \frac{\sum_{g=1}^{G} n_g(\bar{x}^{(g)} - \bar{x})^2/(G-1)}{\sum_{g=1}^{G} \sum_{i \in N_g} (x^{(i)} - \bar{x}^{(g)})^2/(n-G)}, \tag{22}$$

where N_g is the set of cases in class g, $\bar{x}^{(g)}$ is the average of the $x^{(i)}$'s in N_g, and $\bar{x}$ is the overall average $\sum_{i=1}^{n} x^{(i)}/n$. Because of the simplicity, F-statistic is used very often in practice, see for example Dudoit et al. (2002); Guyon et al. (2006), and many others. The number, k, of retained features may be determined by some automatic way, for example by thresholding the p-values in F-test by a prescribed value. Perhaps more often used are trial-and-error methods, for example by looking at the prediction accuracy with various numbers of k. We will therefore assume that the k is chosen arbitrarily before fitting the model and looking at the prediction results on test cases. For notational convenience, we will assume that the features are renumbered so that the subset of retained features is $x_1, \ldots, x_k$, and features $x_{k+1}, \ldots, x_p$ are to be omitted.

Unfortunately, the overall signal-noise ratio of the data set has been lifted by feature selection. To correct for the bias, instead of omitting those $p-k$ features completely, we will keep partial information — the following set statements about them:

$$\boldsymbol{x}_j^{(1:n)} \in \mathcal{S} = \{\boldsymbol{x}^{(1:n)} \mid R_F(\boldsymbol{x}^{(1:n)}, \boldsymbol{y}^{(1:n)}) \leq \gamma\}, \text{ for } j = k+1, \ldots, p, \tag{23}$$

where γ is the score value of the last retained feature x_k (or a threshold that is actually used in determining k). These set statements for omitted features contain information about the

overall signal-noise ratio — indeed a likelihood based on them favors small signal-noise ratio, and therefore can be used to correct for the lifted overall signal-noise ratio. Formally, we will base our posterior distribution on the following joint distribution:

$$\prod_{j=1}^{k}\Big[P(\boldsymbol{x}_j^{(1:n)}\,|\,\boldsymbol{\mu}_j^{(1:G)}, w_j^x, \boldsymbol{y}^{(1:n)})\,P(\boldsymbol{\mu}_j^{(1:G)}\,|\,\nu_j, w_j^\mu)\,P(\nu_j\,|\,w^\nu)\,P(w_j^\mu\,|\,w^\mu)\,P(w_j^x\,|\,w^x)\Big]\times$$
$$\prod_{j=k+1}^{p}\Big[P(\boldsymbol{x}_j^{(1:n)}\in\mathcal{S}\,|\,\boldsymbol{\mu}_j^{(1:G)}, w_j^x, \boldsymbol{y}^{(1:n)})\,P(\boldsymbol{\mu}_j^{(1:G)}\,|\,\nu_j, w_j^\mu)\,P(\nu_j\,|\,w^\nu)\,P(w_j^\mu\,|\,w^\mu)\,P(w_j^x\,|\,w^x)\Big]\times$$
$$P(w^\mu)\,P(w^\nu)\,P(w^x). \tag{24}$$

It is crucial to note that the set statements (23) are *the same* for all the omitted features. We can therefore integrate away feature-specific parameters and hyperparameters ($\boldsymbol{\mu}_j^{(1:G)}$, ν_j, w_j^μ, and w_j^x) in (24), and then the second line of (24) becomes $p-k$ multiples of

$$C(w^\mu, w^x) = P(\boldsymbol{x}_j^{(1:n)}\in\mathcal{S}\,|\,w^\mu, w^x, \boldsymbol{y}^{(1:n)}). \tag{25}$$

In words, $C(w^\mu, w^x)$ is the probability that a feature will fail the feature selection mechanism with threshold γ when the overall signal variance is w^μ and the overall noise variance is w^x. Here, we omit w^ν in the condition of (25), because the probability is not related to the grand means $\boldsymbol{\nu}_{1:j}$, therefore neither to w^ν, which is shown in Appendix B. As result of the integration, the joint posterior (24) becomes:

$$\prod_{j=1}^{k}\Big[P(\boldsymbol{x}_j^{(1:n)}\,|\,\boldsymbol{\mu}_j^{(1:G)}, w_j^x, \boldsymbol{y}^{(1:n)})\,P(\boldsymbol{\mu}_j^{(1:G)}\,|\,\nu_j, w_j^\mu)\,P(\nu_j\,|\,w^\nu)\,P(w_j^\mu\,|\,w^\mu)\,P(w_j^x\,|\,w^x)\Big]\times$$
$$C(w^\mu, w^x)^{p-k}\times P(w^\mu)\,P(w^\nu)\,P(w^x). \tag{26}$$

We will instead apply MCMC to sample from (26) in replace of (14) for all features. From (26), the conditional distributions of the parameters, $\boldsymbol{\mu}_j^{(1:G)}$, ν_j, w_j^μ and w_j^x for $j = 1,\ldots,k$ are the same as given by equations (15) - (18). The conditional distribution of w^ν

is not affected by $C(w^\mu, w^x)$, still having a form similar to (19), but with only $\boldsymbol{\nu}_{1:k}$ used and p set to k. What's different from considering only selected features is a *bias-corrected* conditional distribution for (w^μ, w^x), which is written as:

$$P(w^\mu, w^x \,|\, \boldsymbol{w}^\mu_{1:k}, \boldsymbol{w}^x_{1:k}, \boldsymbol{x}^{(1:n)}_j \in \mathcal{S}, j \in (k{+}1):p) \propto P(w^\mu \,|\, \boldsymbol{w}^\mu_{1:k})\, P(w^x \,|\, \boldsymbol{w}^x_{1:k})\, C(w^\mu, w^x)^{\,p-k}, \tag{27}$$

where $P(w^\mu \,|\, \boldsymbol{w}^\mu_{1:k})$ is similar to (20), with only $\boldsymbol{w}^\mu_{1:k}$ used and p set to k, and $P(w^x \,|\, \boldsymbol{w}^x_{1:k})$ is similarly modified from (21).

The speed of computing $C(w^\mu, w^x)$ is crucial in applying MCMC sampling for the bias-corrected posterior. An efficient Monte Carlo algorithm for this computation is given in Appendix B.

2.4 Summary of Settings for Prior and MCMC Sampling

For all simulation and real data experiments in Sections 3 and 4, we used the following settings for priors: $\alpha^\mu_1 = 3,\ \alpha^x_1 = 10,\ \alpha^\mu_0 = \alpha^x_0 = \alpha^\nu_0 = 0.5,\ w^\mu_0 = w^x_0 = w^\nu_0 = 0.05,\ c_1 = \ldots = c_G = 1$. Our experiences indicated that these settings are appropriate for many gene expression data sets related to complex diseases. More explanations about how to choose these parameters are given in Section 2.1. The reason we used the same settings for all experiments is to demonstrate that the results are insensitive to the settings, especially α^μ_1.

We used the following computational settings for all experiments in this paper. The number of iterations in sampling $(\log(w^\mu), \log(w^x))$ with Metropolis methods is set to 10. We ran 10000 iterations of Gibbs sampling to draw MCMC samples from the bias-corrected posterior distribution. The first 2000 iterations were omitted as burn-in, and every 10th iteration afterwards was used to make Monte Carlo estimations for test cases. In computing the adjustment factor with approximation (34), the cutoffs for f_ℓ and Poisson weights are set to e^{-10}, the number of random Λ used in Monte Carlo integration is set to 1000.

3 Simulation Studies

3.1 Two Criteria for Evaluating Classification Methods

Before presenting the results, we briefly introduce two criteria used to compare different classification methods. Let's denote the predictive probabilities produced by a classification method by $\hat{p}_g^{(i)}$, where i indicates the identity of a test case, g is the class label. We also denote the true class label by $y^{(i)}$. Suppose we have N test cases. The first criterion is **error rate** is: $\frac{1}{N}\sum_{i=1}^{N} I(\hat{y}^{(i)} \neq y^{(i)})$, where $\hat{y}^{(i)} = \arg\max_g \hat{p}_g^{(i)}$. This criterion is very simple and widely used, but isn't precise enough to find the deviances in predictive probabilities. For example, in binary classification, the true response of a test case is 1, for which method A gives a predictive probability 0.49 that it is from class 1, and method B gives 0.2; using 0.5 as threshold, they are both wrong, but we can see that method A is better. Another method that can better detect the differences in predictive probabilities is the **average of minus log predictive probabilities (AMLP)** at the true value $y^{(i)}$: $\frac{1}{N}\sum_{i=1}^{N} -\log(\hat{p}_{y^{(i)}}^{(i)})$. This criterion punishes heavily the small predictive probabilities at the true class labels.

3.2 Design for Comparisons of Three Classification Methods

Using each data simulation model to be described below, we generated 2100 cases, 100 of which were used to form a training set, and the remaining to form a test set. We trained models on the 100 training cases with various numbers of retained features, then obtained predictive probabilities for the 2000 test cases.

In this paper, we compare three different methods: (1) **BCBCSF**: bias-corrected Bayesian classification with selected features, denoted by symbol **"c"** in the following plots. (2) **MLE (DLDA)** (Dudoit, Fridlyand, and Speed, 2002), which is equivalent to estimating the model parameters $\boldsymbol{\mu}_{1:p}^{(1:G)}$ and $\boldsymbol{w}_{1:p}^{x}$ with MLEs, therefore denoted by **"m"**. (3) **PAM** (Tibshirani et al., 2002), denoted by **"p"**. Roughly, PAM subtracts the absolute values of estimated "centroids" ($|\mu_j^{(g)} - \nu_j|/\sqrt{w_j^x}$, for $g \in 1{:}G, j \in 1{:}p$) by a common threshold, with negative values set to 0. Then it uses the shrunken centroids to reconstruct $\mu_j^{(g)}$ with ν_j

and w_j^x for classification. Only the features with at least one non-zero shrunken centroids will have effect in classification. Therefore PAM is also a feature selection method. A larger threshold for cutting off centroids will retain a smaller subset of features.

The three methods are compared in terms of error rate and AMLP with true response values of the 2000 test cases, when the same number of features were retained for fitting models. Since one cannot specify directly how many features to retain in PAM as described above, we ran first PAM with the R package `pamr` available in CRAN (http://cran.r-project.org/mirrors.html), which returned a set of numbers of retained features. We then chose the same numbers of features with F-statistic (ANOVA) in BCBCSF and DLDA methods, and compared the predictive performances when the same number of features were retained. Note that PAM may have multiple predictions for the same number of retained features, which are resulted from different thresholds used to cut off centroids.

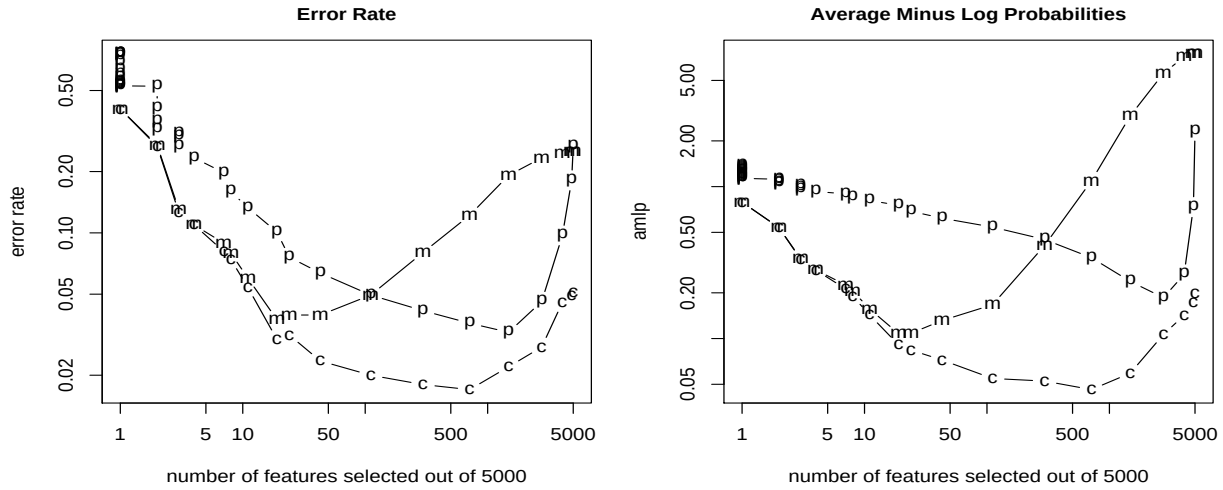
3.3 Experiments on Simulated Data from Our Bayesian Model

Using the Bayesian model described in Section 2.1, with the following fixed top level hyperparameters and degrees of freedom for IG distributions: $\alpha_1^\mu = 3,\ w^\mu = 0.003,\ w^x = 1,\ \alpha_1^x = 10,\ w^\nu = 1$, we generated a data set of $n = 2100$ cases that are evenly distributed in $G = 4$ classes, with $p = 5000$ features. 100 of the cases are randomly selected as training set, and the remaining 2000 are used as test cases. Figure 2 shows the comparison results by plotting the error rates and AMLPs against the numbers of retained features. We can see clearly that BCBCSF predicts the best in terms of both of the criteria.

The first thing we see is that the error rates and AMLPs of all three methods decrease at the beginning when more features are included and then increase. Therefore, feature selection by for example F-statistic is useful to obtain good predictions. It justifies why we should consider feature selection, even though the computation for implementing our Bayesian method with all features is still feasible.

The error rates and AMLPs of BCBCSF and MLE are similar when only a small number

Figure 2: Comparisons of predictive probabilities produced by three classification methods ("p" — PAM, "m" — DLDA, "c" — BCBCSF) in terms of error rate and AMLP. The data was simulated from our Bayesian model. Both x and y axis are in logarithm scale.



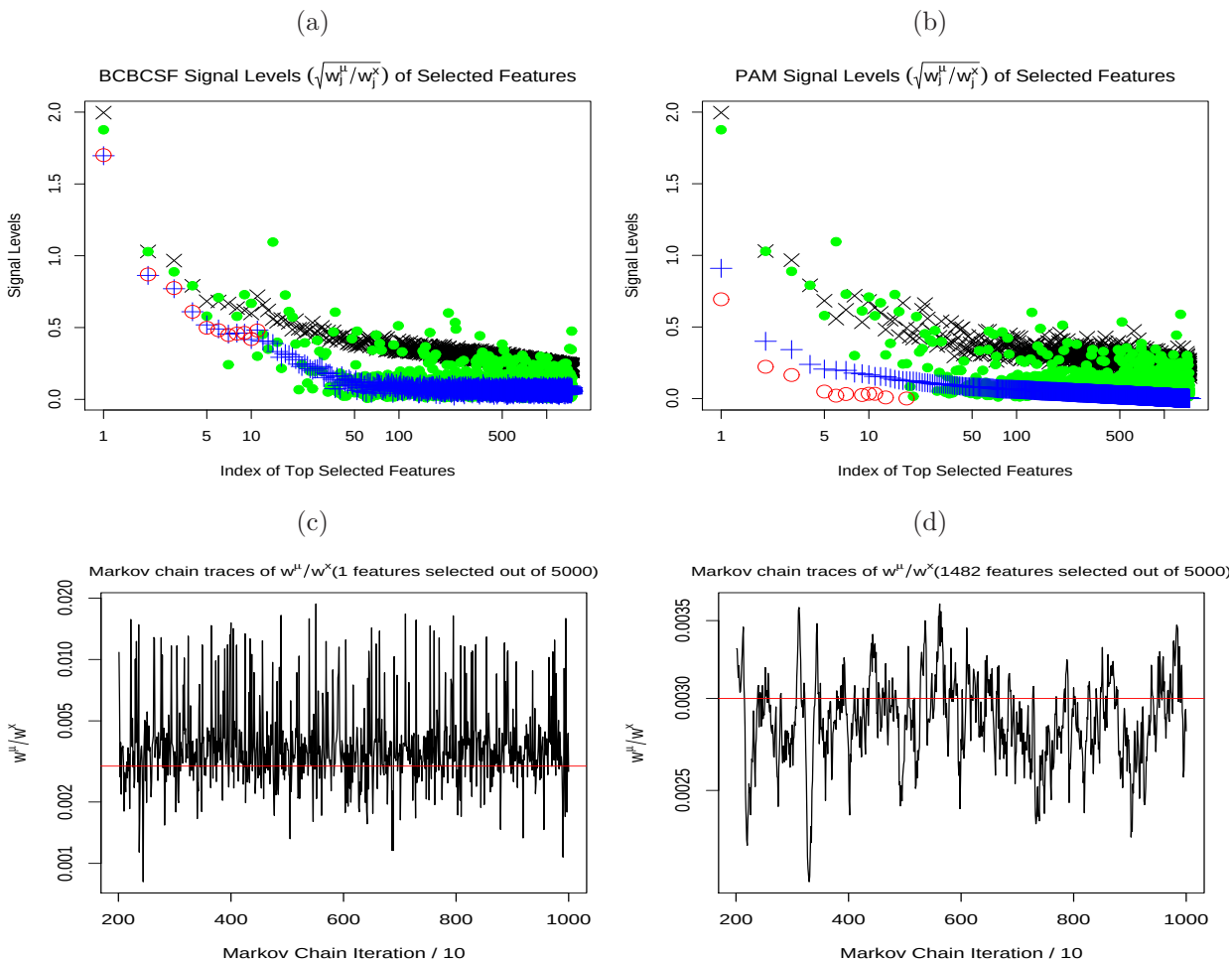
of top features are used. This is because the top features selected by F-statistic are indeed useful, as shown later in Figure 3a. However, BCBCSF is more resistant to the noises in useless features, therefore has better predictions when many features are used. Even when they have similar error rates, the bias-uncorrected MLE produces much more extremely wrong predictive probabilities than BCBCSF for difficult cases. Figure 4b show the individual predictive probabilities at their true class labels for 70 randomly selected test cases, when 8 features are retained. We see that the predictive probabilities of MLE at true labels are mostly smaller than those of BCBCSF, especially for those cases misclassified by both MLE and BCBCSF. The overconfidence of MLE will be more severe when more features are retained. We will discuss the practical disadvantage of MLE resulting from this overconfidence in real data analysis presented in Section 4.

The error rates and AMLPs of PAM decrease more slowly than those of BCBCSF and MLE. This indicates that as a tool for selecting features, PAM is less powerful than F-statistic, since it will need to retain more features to reach its best prediction. The reason is that PAM overcorrects for selection bias — when PAM omits more features, it uses a higher value of threshold, which however also cuts more off the real signals, therefore reduces their

discriminative powers. In contrast, as explained in Section 2.2, BCBCSF shrinks the signals of different features toward 0 with different "thresholds" — $\boldsymbol{w}_j^\mu$, so it doesn't hurt the strong signals while shrinking small signals. In addition, our correction for feature selection bias doesn't impose stronger shrinkage when more features are omitted, instead it adjusts the posterior of hyperparameters w^μ and w^x to center at the "correct" value but with different uncertainty. Therefore, BCBCSF corrects for feature selection bias appropriately without reducing the discriminative powers of real signals. Further illustrations are given below.

The plots of Figure 3 illustrate the difference of bias corrections used by BCBCSF and PAM, and also look into the differences of these three methods in parameter estimation. In subfigures 3a and 3b, we plot the signal levels ($\sqrt{w_j^\mu / w_j^x}$) of top selected features estimated by three methods when different numbers (11,1482) of features are retained, as well as the true signals calculated from the 2000 test cases; for BCBCSF, the signals are taken as the medians of the signals in Markov chain sample, for MLE, PAM and true value, w_j^μ is computed as the sample variance of $\mu_j^{(g)}$ across classes. From subfigure 3a, we see that BCBCSF shrinks the small MLE signals to 0. By comparing to the true signal values of these features with small MLE signals, we can see that most of them are only made by chance, but F-statistic and MLE *overestimate* their signal levels. BCBCSF successfully avoids the feature selection bias by shrinking them to their true values. Meanwhile, the correction for bias doesn't punish significantly the real strong signals, though we still see small amount of shrinkage for a few of them, which may be due to randomness. Most importantly, we see that the estimates of strong signals by BCBCSF are very stable regardless of how many features are retained. This is because that BCBCSF corrects for bias through adjusting the posterior of hyperparameters to center at the true value, rather than cutting off the signals directly, as PAM does. Subfigures 3c and 3d show two Markov chain traces of $\log(w^\mu / w^x)$ when 1 and 1482 (two very different numbers) of features are retained. Both traces move around the true value, but subfigure 3c shows much larger variation due to omission of many features. In contrast, the signal estimates by PAM in subfigure 3b for the top selected features are

Figure 3: Plots in (a) and (b) show the signal levels of top selected features estimated by BCBCSF (red ∘ and blue +), MLE (black ×) and PAM (red ∘ and blue +). For BCBCSF and PAM, red ∘ and blue + represent the estimated signal levels when two different numbers (11 and 1482 here; two other similar numbers used for Figures 5 and 6) of features are retained. *Note that the points in (a) overlap on the left.* The true values are also indicated with green •. Plots in (c) and (d) show Markov chain traces of $\log(w^{\mu}/w^{x})$ in two runs, with the horizontal lines showing the true value. The data was simulated from our Bayesian model.

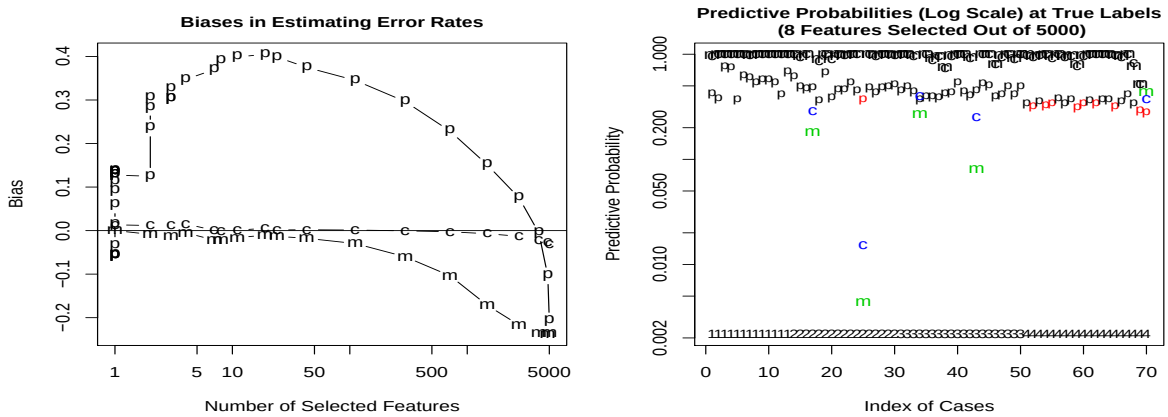


smaller than the true values by large amount. In addition, this shrinkage becomes stronger when fewer features are retained, which is undesired.

We can look at how well-calibrated a set of predictive probabilities by the bias (difference) of *expected error rate* to the true error rate. The expected error rate is the average of the expected probabilities of making prediction errors for all test cases, each of which is equal to "1 - the highest predictive probability in all classes". A negative value for this bias indicates

Figure 4: Plots in (a) show biases of expected error rates to true error rates. Plots in (b) show the predictive probabilities *at the true labels* for different methods. The colorized points represents the cases that are misclassified when the predicted response values of test cases are the class with highest predictive probability, ie, thresholds for predictive probabilities in prediction for all classes are fair. The bottom numbers show the true labels. The data was simulated from our Bayesian model.

(a) (b)



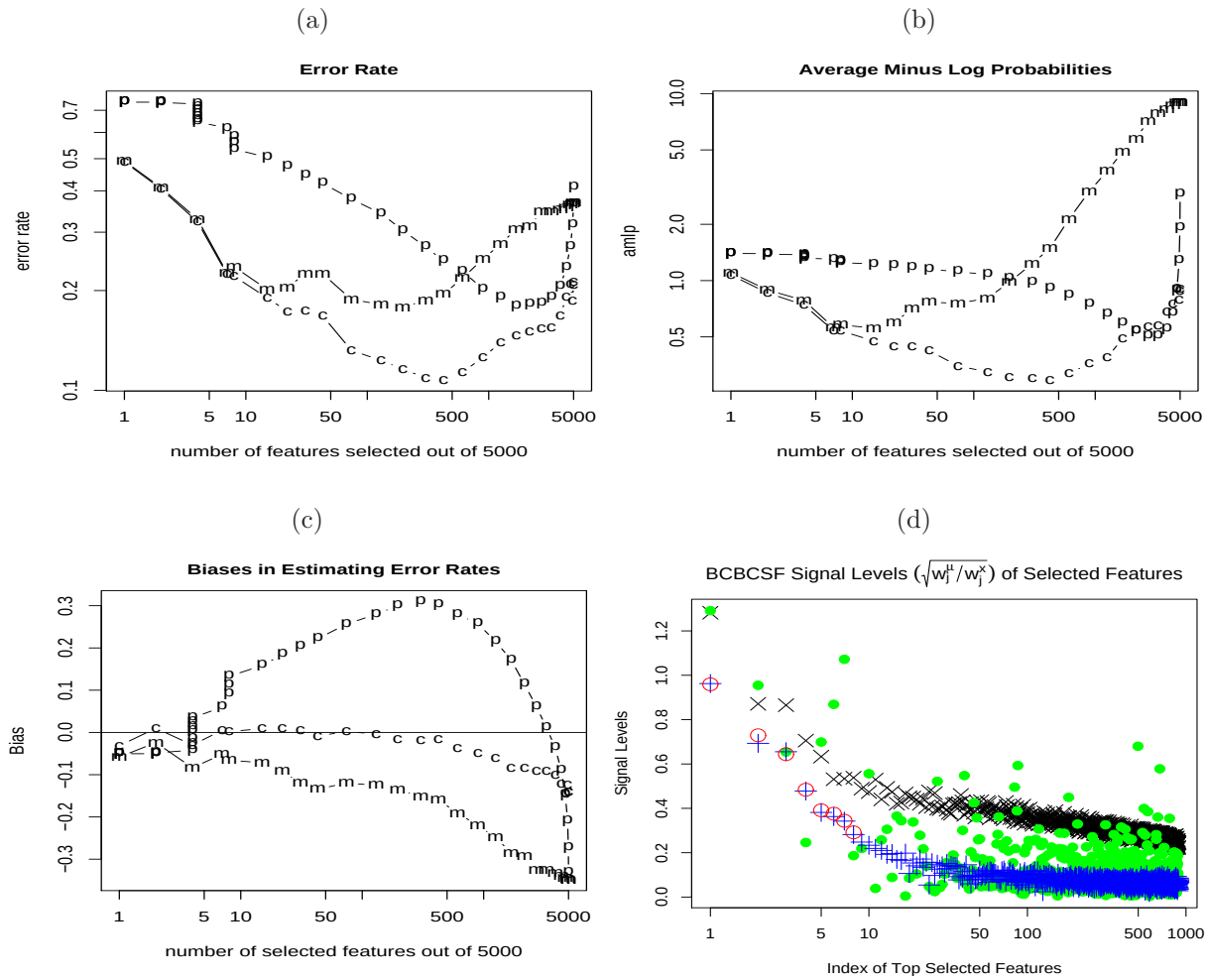
that the predictive probabilities are overconfident, a positive value indicates overconservative, and a value close to 0 is desired, indicating well-calibration. Subfigure 4a shows the biases of three methods. The biases of MLE are mostly negative, and increase in absolute value when more features are retained since more false signals are included in prediction. The biases of PAM are positive at the beginning, indicating that the predictive probabilities produced by PAM are overconservative because it cuts off real signals. The biases of BCBCSF stay close to 0, but become slightly negative at the end, which may be due to computational inaccuracy in MCMC.

3.4 Experiments on Simulated Data with t Noises

In this section, we test BCBCSF when the data contain noises that cannot be modeled by normal distributions. A data set was generated with the same way for generating $\mu_j^{(g)}$ and w_j^x as described in Section 3.3, but the noises for $x_j^{(i)}$ are generated from a scaled t distribution with 4 degrees of freedom (denoted by t_4): $x_j^{(i)} \mid y^{(i)} = g, \mu_j^{(g)}, w_j^x \quad \sim \quad \mu_j^{(g)} + t_4 \times \sqrt{w_j^x}$, for $j = 1, \ldots, p$.

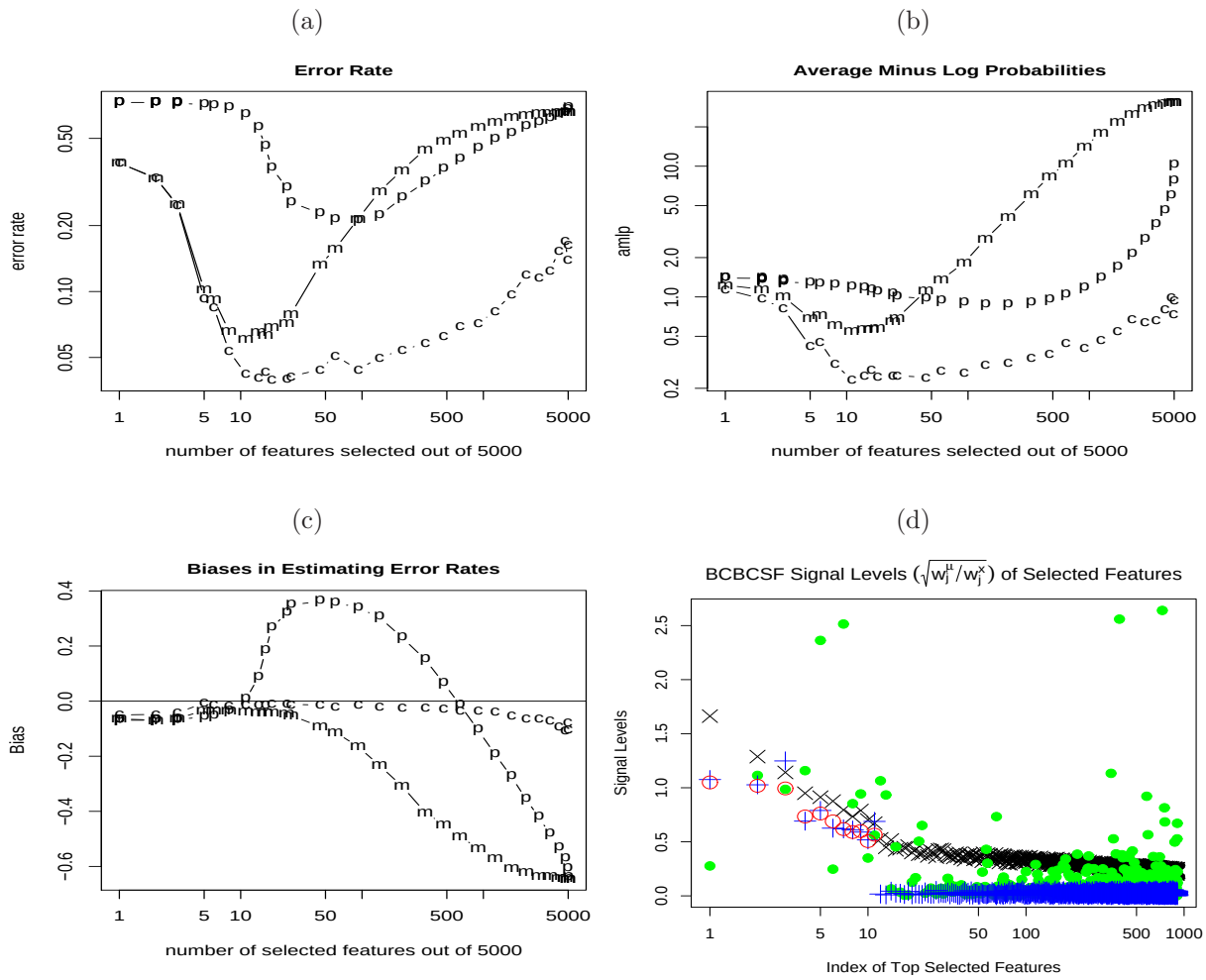
We made similar comparisons of the three classification methods. To save space, we only

Figure 5: These plots show comparisons of BCBCSF, MLE, and PAM on a data set with noises for $x_j^{(i)}$ generated from scaled t distribution with 4 degrees of freedom. The methods for reading these plots can be found from the plots in Section 3.3. Particularly, the true values of signals are computed from the 2000 test cases with MLE method.



show the comparisons of error rates, AMLPs, biases in expected error rates, and BCBCSF signal estimates. The results are shown in Figure 5. These plots confirm all of the discussions about the differences of BCBCSF from MLE and PAM. Briefly, MLE produces overconfident predictive probabilities, which is even clearer for this example from looking at the biases in expected error rates, caused by increased noises in data; PAM produces overconservative predictive probabilities; and BCBCSF corrects for feature selection bias appropriately by reducing the influences of noises in features with little or no signals, producing unbiased and good predictions when appropriate number of features are retained. For this example, we see that BCBCSF are a little overconfident when a large number of features are used

Figure 6: These plots show comparisons of BCBCSF, MLE, and PAM on a data set with noises for $x_j^{(i)}$ generated from scaled t distribution with 4 degrees of freedom, and signals generated with spike-and-slab priors. The methods for reading these plots can be found from the similar plots in Section 3.3. Particularly, the true values of signals are computed from the 2000 test cases with MLE method.



(see subfigure 5c), which confirms the necessity of selecting only a subset of features in high-dimensional problems.

3.5 Experiments on Data Simulated with Spike-and-Slab Prior

In this section we test BCBCSF when the signals of features are generated from a more sparse spike-and-slab priors. We still generate $p = 5000$ features, 0.5% of which have $w_j^\mu = 4$, and the remaining have w_j^μ set to exact 0. Parameters ν_j and w_j^x were generated as in Section 3.3, and the noises for $x_j^{(i)}$ were generated from t_4 distribution as described in Section 3.4.

Similar experiments and comparisons were carried out as in Section 3.3, with results shown in Figure 6. All the differences of BCBCSF from MLE and PAM are confirmed again for this example even though the distributions for generating signals are not the prior of BCBCSF. It shows that BCBCSF is robust to these model mis-specifications in priors and noises. Particularly, we see that t prior can handle the very sparse signals well, shrinking small signals (which are mostly made by chance) to be very close to 0. In subfigure 6d, we see that the signal estimates by BCBCSF are still rather stable regardless of how many features are retained, and there is a big gap between the real signals to the false when around 10 features are retained, correctly reflecting the shape of the spike-and-slab prior.

4 Comparisons on Real Microarray Data Sets

In this section, we apply BCBCSF to two publicly available mircoarray data sets — namely lymphoma, and colon, to demonstrate BCBCSF by comparing to MLE and PAM. The data sets were downloaded from `http://stat.ethz.ch/~dettling/bagboost.html`, and studied by Dettling (2004) and many others. The original lymphoma and colon data sets were published by Alizadeh et al. (2000) and Alon et al. (1999) respectively. Some preprocessing was made by Dettling (2004), from which one can find the information. In summary, lymphoma data set contains expression levels of $p = 4026$ genes from $n = 62$ patients with most prevalent adult lymphoid malignancies: 42 cases of diffuse large B-cell lymphoma, 9 cases of follicular lymphoma and 11 cases of chronic lymphocytic leukemia. Colon data set contains expression levels of 22 normal and 40 tumor colon tissues for 6500 human genes measured using the Affymetrix technology; a selection of 2000 genes with highest minimal intensity across the samples has been made by Alon et al. (1999). We coded the tissue types by integers 1, 2, and 3 (if any) as the order of appearing in the above description for each data set in the following discussions.

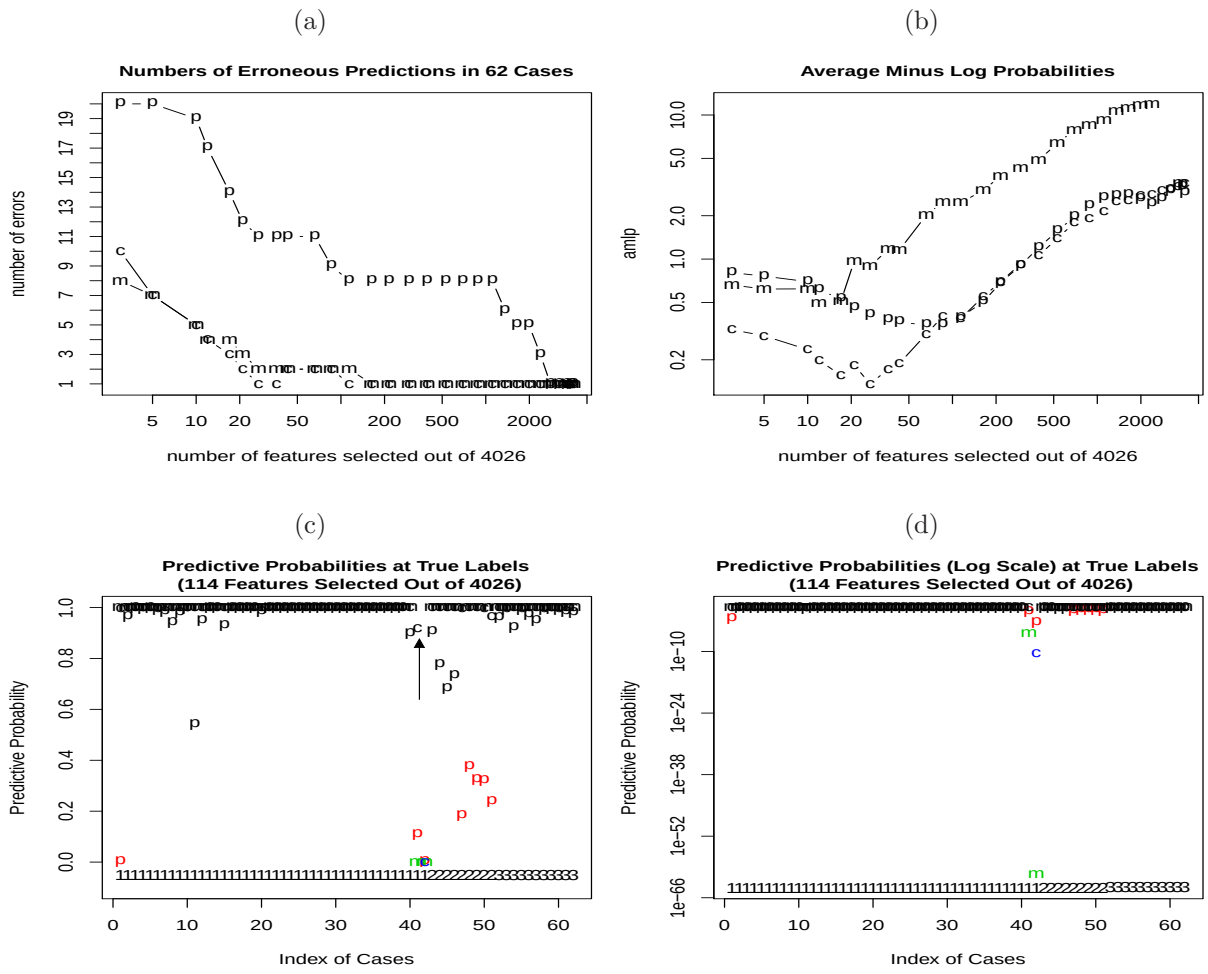
We used the same settings for priors and computation in BCBCSF as in simulation studies, described in Section 2.4. Since the data sets have very small number of cases,

we used cross-validation (CV) to obtain the predictive probabilities for all cases. For fair comparison, we used the same folds (the way to divide cases) as returned by PAM's R function `pamr.cv` with the number of folds set to 10 for all three methods. If some class has fewer than 10 cases, the exact number of folds returned by `pamr.cv` according to its scheme was however smaller than 10. Because PAM cannot specify the number of retained features, as in simulation studies, we first ran PAM's R function `pamr.cv` to obtain a set of number of retained features (as well as folds). We then used the same folds and chose the same numbers of features with F-statistic for BCBCSF and MLE. The comparisons of prediction performance are made when the same number of features are retained. The comparison results are shown in Figures 7 and 8. For direct visualization, we choose one experiment with appropriate number of features are retained at which PAM has nearly the best AMLP to display the predictive probabilities of all cases. Note, however, this doesn't imply that we recommend using PAM to choose the number of retained features for BCBCSF.

As we have seen in simulation studies, the AMLPs decrease first and then increase as the number of retained features increases. Therefore, best predictions for all three methods are attained when an appropriate number of features are selected. In both data sets, the error rates are not corrupted significantly when many features are used, different from what we see from simulation studies. A possible explanation is that the data sets have too small number of cases to contain those cases on the classification boundary for which the predictions are more easily corrupted by small and false signals in training data. Another possibility is that the data sets have been "preprocessed" by experimenters to be nicer before being published.

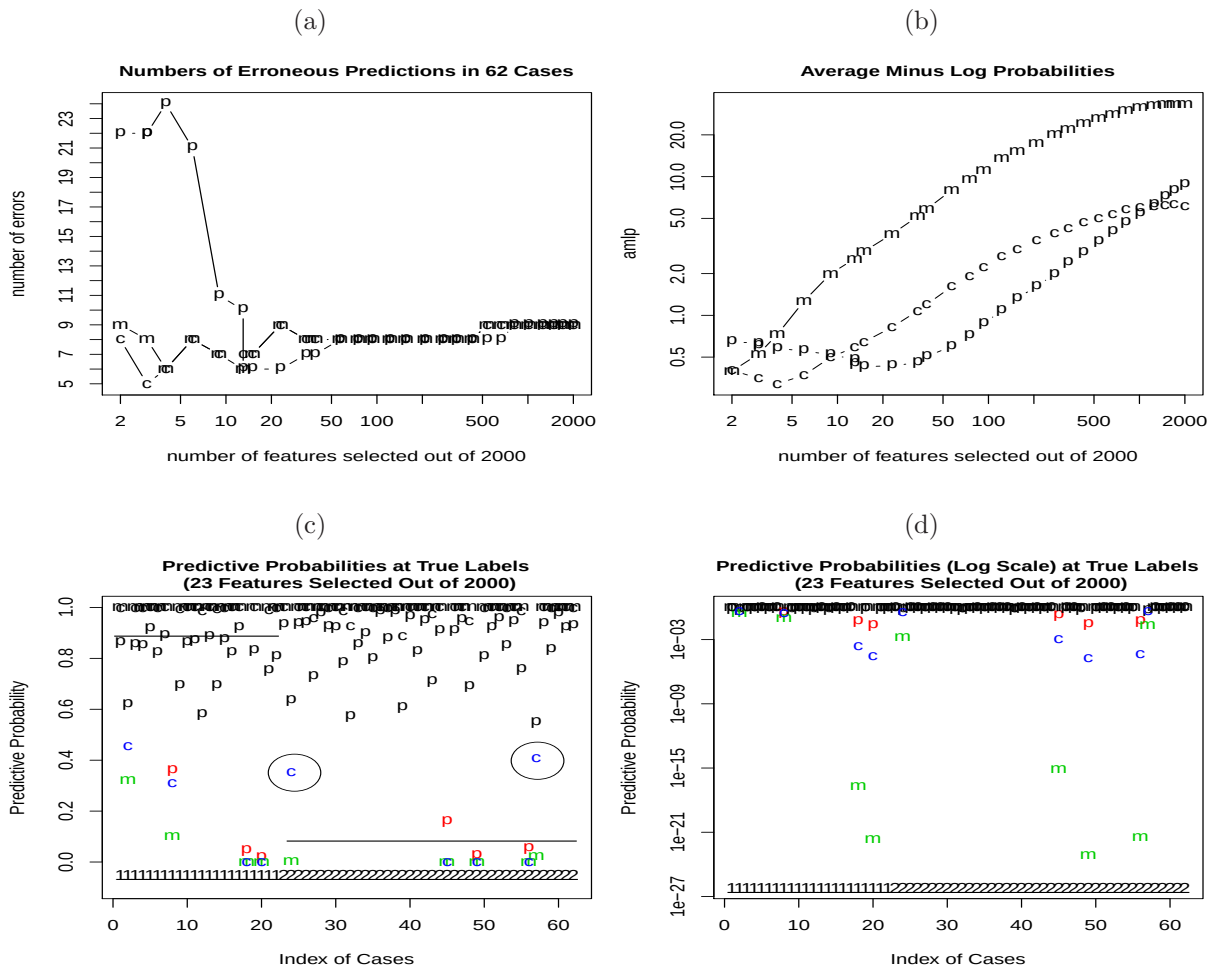
We first look at the comparisons between BCBCSF and MLE. The error rates of BCBCSF and MLE are similar in both data sets. This is not surprising as they use the same features in prediction, especially when we retain only a small number of features, most of which are real signals for separating the classes. Our simulation studies have shown that there are differences in error rates between BCBCSF and MLE when fairly many features are retained. Perhaps the difference can only be detected when they are applied to a large number of

Figure 7: These plots show comparisons of BCBCSF, MLE and PAM on lymphoma data. The methods for reading these plots can be found from the similar plots in Section 3.3.



test cases. The differences in predictive probabilities given by BCBCSF and MLE are clear when we look at the AMLPs in subfigures 7b and 8b because this criterion punishes heavily the small predictive probabilities at true labels. The advantage of BCBCSF over MLE in terms of AMLP comes from that for those difficult cases, the predictive probabilities given by MLE are greatly *overconfident* at the *wrong* labels, therefore are very small at the *true* labels. This is shown by subfigures 7d and 8d in which the log predictive probabilities at true class labels for all cases are displayed with y-axis. From these subfigures, we see clearly that for those difficult cases, the predictive probabilities at true labels given by MLE are smaller than those of BCBCSF by many orders of magnitude. The message to be taken from these sharp differences is that BCBCSF is indeed able to adjust the greatly overconfident predic-

Figure 8: These plots show comparisons of BCBCSF, MLE and PAM on colon data. The methods for reading these plots can be found from the similar plots in Section 3.3. Additionally, the horizontal lines in (c) show classification boundaries if the threshold for predicting tumor is 0.1 (which means that if predictive probability at class tumor of a case is larger than 0.1, then it is classified into tumor); the cases with predictive probabilities at true classes above the lines are correctly classified.



tive probabilities of MLE. However, since these cases are really difficult for both methods, this correction for overconfidence of MLE seems useless in practice. However, BCBCSF's adjustment for overconfident predictive probabilities is practically useful for those less difficult cases, especially when losses from different classification errors are unbalanced. Among so small number of cases, we see three such useful corrections. One is the case indexed by 41 in lymphoma data pointed by an arrow in subfigure 7c. Even when the predictive class label is taken to be the class with highest predictive probability (implied by balanced losses),

BCBCSF correctly predicts the label of this case, but MLE reports a very small predictive probability at its true label. The other two are the cases indexed by 24 and 57 in colon data — the two cases with circles in subfigure 8c. If we use a threshold for predicting tumor (which means that if the predictive probability at tumor is larger than this threshold, then a case is classified into tumor) that is smaller than 0.5, for example 0.1 (shown by the horizontal lines), BCBCSF will classify them into class tumor correctly, while MLE still makes wrong predictions. Note that such small threshold for predicting tumor is reasonable in practice because the loss from classifying a tumor tissue into normal tissue may be more than 9 times of the opposite error. As summary, from the comparison results on these two data sets, we see that BCBCSF can correct for the overconfidence of MLE, and the correction is useful in practice.

Next we look at the comparisons between BCBCSF and PAM. As we have shown in simulation studies, because PAM selects features by cutting off the signal levels, it produces overconservative and therefore less discriminative predictive probabilities. This fact is also observed when it is applied to these two data sets, as shown by subfigures 7c and 8c — most of the predictive probabilities given by PAM at the true labels are smaller than those of BCBCSF. We will discuss two practical disadvantages caused by this overconservation.

First, we see that PAM tends to select more (sometimes much more as in lymphoma data) features to attain its best predictive accuracy. When only a very small number of features are retained and a large cut-off to signal levels is used, PAM predicts very poorly in terms of error rate. Indeed, in such situations, PAM simply predicts the class labels of all cases to the class with the most cases; therefore for each data set, the numbers of erroneous predictions are almost just the total number of cases in minority classes. This loss of accuracy may vanish when a small cut-off to the signal levels is used and more features are retained. However, we see that for lymphoma data set, this only happens when more than 2000 features are retained. The reason is that the signals for separating class 2 in lymphoma data from other two classes are rather weak, so they are cut off even when a fairly small

PAM's threshold is used. The numbers, roughly 15-20, of retained features for PAM to reach good accuracy in colon data set isn't so large (bad), but still much larger than the number (around 4) of retained features needed by BCBCSF. This fact shows that PAM is inferior than BCBCSF (and MLE) when it is used as a tool for selecting a subset of differential features in high-dimensional problems by looking at predictive performance (currently, this type of use of PAM may be more often in practice than as a disease diagnosis tool with gene expression.).

Second, when unfair thresholds for different classes are used (implied by unbalanced losses from different classification errors), the less discriminative power of PAM's predictive probabilities will result in much more erroneous predictions than BCBCSF. We look at the predictive probabilities produced by BCBCSF and PAM for colon data set when 23 features are retained (where PAM almost reaches its smallest AMLP), shown by subfigure 8c. If the threshold for predicting tumor is smaller than 0.1, PAM will result in many erroneous predictions, with most normal cases classified into tumor.

We remark on the inferiority of BCBCSF to PAM in terms of AMLP when many features are retained, as seen in subfigure 8b. As we have shown, feature selection is necessary to obtain good prediction for all three methods. We therefore don't recommend using BCBCSF with a large subset of selected features. Therefore, the best attainable prediction accuracy is what we care in practice. From the comparisons of AMLPs on both data sets, we see that BCBCSF is better than PAM in smallest AMLPs. Indeed, this inferiority occurs also because AMLP cannot punish the overconservative probabilities very well, as we see that the AMLPs of PAM are not so poor even when the error rates are very poor at the beginning when very small subsets of features are retained.

Finally we must point out that the computation of BCBCSF is significantly slower than MLE and PAM, as BCBCSF uses MCMC. The plots in Figure 9 show the computation times needed for the simulation example in Section 3.3, which is very similar to the time needed for

running BCBCSF once for a practical data set. In applying BCBCSF for practical problems, one doesn't need to run it with large numbers of retained features, because we have shown that the best predictive accuracy is often achieved by a very small subset of selected features. From Figure 9, the computation is very fast when the numbers of retained features are small.

5 Concluding Remarks

In this article, we have shown that a proposed high-dimensional classification method, called BCBCSF, is a valuable alternative to two other similar methods — MLE (DLDA) and PAM, with practical values. Our goal in proposing BCBCSF is a classification method that can correct for feature selection bias and therefore report better calibrated predictive probabilities for future cases. Our simulation and real data experiments show that BCBCSF indeed reports predictive probabilities that are better calibrated than MLE and PAM. In Section 4, we have discussed in details the practical advantages of using BCBCSF due to the better calibration. Briefly, as result of bias correction, BCBCSF has better classification accuracy than PAM and MLE, especially when the losses incurred from different classification errors are unbalanced. In addition, compared to PAM, BCBCSF reaches its (better) smallest classification error with a smaller subset of selected features.

Gaussian distributions for $x_j^{(i)}$ given class labels may not be appropriate for some real gene expression data sets in which some extraordinarily large or small expression levels (often called outliers) are recorded. There are two possible extensions of BCBCSF to handle outliers. One is using t-distribution with small degree freedom to model $x_j^{(i)}$ given class label. Without feature selection applied, this is easy to implement in Gibbs sampling by introducing auxiliary variances for each $x_j^{(i)}$. The difficulty lies in a good choice for univariate feature selection score such that the required bias correction factor is easy to compute. When outliers exist, the scores based on ranks, for example Mann-Whitney-Wilcoxon test, are more suitable than F-statistic. However, how to compute the correction factor efficiently remains to be investigated. Another solution for outliers is converting real-valued gene expression

levels into discrete values and select features with sample correlation. The conversion may lose some information in selected features, but the required computation of bias correction factor may be simplified. However, for newer high-throughput sequencing technologies that directly count the numbers of mRNA molecules, a discrete distribution for $x_j^{(i)}$ is natural.

References

Alizadeh, A. A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., Boldrick, J. C., Sabet, H., Tran, T., and Yu, X. (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., and 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 the National Academy of Sciences (USA)*, 96, 6745–6750.

Ambroise, C. and McLachlan, G. J. (2002), "Selection Bias in Gene Extraction on the Basis of Microarray Gene-expression Data," *PNAS*, 99, 6562–6566.

Dawid, A. P. and Dickey, J. M. (1977), "Likelihood and Bayesian Inference from Selectively Reported Data," *Journal of the American Statistical Association*, 72, 845–850.

Dettling, M. (2004), "BagBoosting for tumor classification with gene expression data," *Bioinformatics*, 20, 3583–93.

Dudoit, S., Fridlyand, J., and 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.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004), *Bayesian Data Analysis*, Texts in Statistical Science, Chapman and Hall/CRC.

Guyon, I., Gunn, S., Nikravesh, M., and Zadeh, L. A. (2006), *Feature Extraction: Foundations and Applications*, vol. 207 of *Studies in Fuzziness and Soft Computing*, Springer.

Hurvich, C. M. and Tsai, C.-L. (1990), "The Impact of Model Selection on Inference in Linear Regression," *The American Statistician*, 44, 214–217.

Knight, K. (2000), *Mathematical Statistics*, Texts in Statistical Science, Chapman and Hall/CRC.

Li, L., Zhang, J., and Neal, R. M. (2008), "A Method for Avoiding Bias from Feature Selection with Application to Naive Bayes Classification Models," *Bayesian Analysis*, 3, 171–196.

Neal, R. M. (1993), "Probabilistic Inference using Markov Chain Monte Carlo Methods," Tech. rep., Dept. of Computer Science, University of Toronto.

Raudys, S., Baumgartner, R., and Somorjai, R. (2005), "On Understanding and Assessing Feature Selection Bias," *Artificial Intelligence in Medicine*, 468–472.

Shafer, G. (1982), "Lindleys paradox," *Journal of the American Statistical Association*, 77, 325–334.

Shen, X., Huang, H.-C., and Ye, J. (2004), "Inference After Model Selection," *Journal of the American Statistical Association*, 99, 751–762.

Singhi, S. K. and Liu, H. (2006), "Feature Subset Selection Bias for Classification Learning," *Proceedings of the 23rd International Conference on Machine Learning*, 849–856.

Tibshirani, R., Hastie, T., Narasimhan, B., and Chu, G. (2002), "Diagnosis of multiple cancer types by shrunken centroids of gene expression," *Proceedings of the National Academy of Sciences*, 99, 6567.

Wang, R. and Lagakos, S. W. (2009), "Inference after variable selection using restricted permutation methods," *Canadian Journal of Statistics*, 37, 625–644.

Zhang, P. (1992), "Inference After Variable Selection in Linear Regression Models," *Biometrika*, 79, 741–746.

Acknowledgements

This work was supported by fundings from Natural Sciences and Engineering Research Council of Canada, and Canadian Foundation for Innovation. The author also thanks JASA editors and two anonymous referees for great help in improving the previous drafts.

Appendices

A List of Notations

We also make a list of notations used throughout the paper. More detailed explanations of these notations are given in Section 2.1 and are graphically displayed in Figure 1.

- `data`: $y^{(i)}$ is the label for the ith case, $x_j^{(i)}$ is the value of the jth feature, for $j = 1, \ldots, p$.
- `parameters`: $\mu_j^{(g)}$ is the mean of the jth feature in class g, and w_j^x is the variance of $x_j^{(i)}$ *within* a class, which is the same for all classes. ψ_g is the prior label probability of class g.
- `hyperparameters`: ν_j and w_j^μ are the mean and the variance of $\mu_j^{(g)}$ across classes of feature j. ν_j represent the common expression level of feature j, and w_j^μ represents the signal level of feature j.
- `hyperparameters`: w^ν is the variance of $\nu_1, \ldots, \nu_p$, w^μ is the "mean" of $w_1^\mu, \ldots, w_p^\mu$, representing the overall signal level, and w^x is the "mean" of $w_1^x, \ldots, w_p^x$, representing the overall noise level.

B Approximating Adjustment Factor of BCBCSF

A method for approximating $C(w^\mu, w^x)$ is based on precursors' work on computing the power function of one-way ANOVA. Given $\boldsymbol{\mu}_j^{(1:G)}$, and $\boldsymbol{y}^{(1:n)}$, F-statistic in (22) has a non-central F distribution (Knight, 2000, pg. 411-416), with $G-1$ and $n-G$ degrees of freedom, and a non-centrality parameter $2\,\Lambda(\boldsymbol{\mu}_j^{(1:G)}, w_j^x)$, where

$$\Lambda(\boldsymbol{\mu}_j^{(1:G)}, w_j^x) = \frac{D(\boldsymbol{\mu}^{(1:G)})}{2\,w_j^x}, \quad D(\boldsymbol{\mu}^{(1:G)}) = \sum_{g=1}^{G} n_g(\mu_j^{(g)} - \tilde{\mu}_j)^2, \quad \tilde{\mu}_j = \frac{\sum_{g=1}^{G} n_g \mu_j^{(g)}}{n}. \tag{28}$$

We therefore have

$$P(\boldsymbol{x}_j^{(1:n)} \in \mathcal{S} \,|\, \boldsymbol{y}^{(1:n)}, \boldsymbol{\mu}_j^{(1:G)}, w_j^x) = P\left(F_{(G-1,\, n-G,\, 2\,\Lambda(\boldsymbol{\mu}_j^{(1:G)}, w_j^x))} \leq \gamma \right) \equiv c(\Lambda(\boldsymbol{\mu}_j^{(1:G)}, w_j^x)), \tag{29}$$

where $F_{(G-1,\, n-G,\, 2\,\Lambda(\boldsymbol{\mu}_j^{(1:G)}, w_j^x))}$ denotes a random variable with a non-central F distribution.

The function $c(\Lambda)$ can be computed easily using the fact that a non-central χ^2 distribution can be expressed as an infinite mixture of central χ^2 distributions with Poisson weights (Knight, 2000). With χ^2_ν denoting a random variable having central χ^2 distribution with degree freedom ν, we can now express $c(\Lambda)$ as:

$$c(\Lambda) = \sum_{\ell=0}^{+\infty} f_\ell \, \frac{\exp(-\Lambda)\, \Lambda^\ell}{\ell\,!}, \tag{30}$$

where

$$f_\ell = P\left(\frac{\chi^2_{G-1+2\ell}/(G-1)}{\chi^2_{n-G}/(n-G)} \leq \gamma \right) = P\left(\frac{\chi^2_{G-1+2\ell}/(G-1+2\ell)}{\chi^2_{n-G}/(n-G)} \leq \frac{\gamma\,(G-1)}{G-1+2\ell} \right). \tag{31}$$

Here, f_ℓ can be computed with the CDF of central F distribution. From (31), we can see that f_ℓ decreases to 0 as ℓ tends to $+\infty$. Therefore, $c(\Lambda)$ is a decreasing function of Λ. In addition, since Poisson weights are always between 0 and 1, we can truncate the above infinite summation by setting a threshold for f_ℓ, while controlling a same tolerable error for all Λ.

To obtain the adjustment factor — $C(w^\mu, w^x)$, we need to integrate $c(\Lambda(\boldsymbol{\mu}_j^{(1:G)}, w_j^x))$ with respect to the prior distribution of $\Lambda(\boldsymbol{\mu}_j^{(1:G)}, w_j^x)$, which is induced by the priors for $\boldsymbol{\mu}_j^{(1:G)}$ and w_j^x, conditional on w^μ, w^x, and w^ν. It is useful to note that the prior distribution of $\Lambda(\boldsymbol{\mu}_j^{(1:G)}, w_j^x)$ is unrelated to ν_j, and so neither to w^ν. To show this, we will re-parameterize $\boldsymbol{\mu}_j^{(1:G)}$ and w_j^x as follows:

$$\boldsymbol{\mu}_j^{(1:G)} = \boldsymbol{m}_j^{(1:G)} \sqrt{s_j^\mu} \sqrt{w^\mu} + \nu_j, \qquad w_j^x = s_j^x \, w^x, \tag{32}$$

where, $\boldsymbol{m}_j^{(1:G)} \overset{\text{IID}}{\sim} N(0,1)$, $\quad s_j^\mu \sim \text{IG}(\alpha_1^\mu/2, \alpha_1^\mu/2)$, $\quad s_j^x \sim \text{IG}(\alpha_1^x/2, \alpha_1^x/2)$. Then it can be shown that

$$\Lambda(\boldsymbol{\mu}_j^{(1:G)}, w_j^x) = \frac{1}{2}\, D\left(\boldsymbol{m}_j^{(1:G)}\right) \frac{s_j^\mu}{s_j^x} \frac{w^\mu}{w^x}. \tag{33}$$

From the above expression, we can see readily that the prior distribution of $\Lambda(\boldsymbol{\mu}_j^{(1:G)}, w_j^x)$ is not related to w^ν. Therefore we denote the adjustment factor by a function of only w^μ and w^x — $C(w^\mu, w^x)$. It is explicitly written as:

$$C(w^\mu, w^x) = E(c(\Lambda)) = E_{\boldsymbol{m}_j^{(1:G)}, s_j^\mu, s_j^x}\left(c\left(\frac{1}{2}\, D\left(\boldsymbol{m}_j^{(1:G)}\right) \frac{s_j^\mu}{s_j^x} \frac{w^\mu}{w^x}\right)\right). \tag{34}$$

As shown above, $c(\Lambda)$ is a decreasing function of Λ. Therefore, when w^μ/w^x is larger, $C(w^\mu, w^x)$ is smaller, resulting in fewer features to be omitted using γ as threshold. This explains more precisely that the ratio w^μ/w^x controls the overall strength of the relationship between the features and response. The method presented in this paper corrects for the upward bias in the posterior of w^μ/w^x given the retained features with $C(w^\mu, w^x)$, which is a decreasing function of w^μ/w^x.

The expression of $C(w^\mu, w^x)$ in (34) also indicates that we can use Monte Carlo method to estimate it at different values of w^μ and w^x, with a *common* pool of i.i.d. random samples of $\boldsymbol{m}_j^{(1:G)}$, s_j^μ and s_j^x. There are two advantages of doing this. First, it saves computation time. We need to draw samples of $\boldsymbol{m}_j^{(1:G)}$, s_j^μ, and s_j^x and compute $D\left(\boldsymbol{m}_j^{(1:G)}\right) s_j^\mu/s_j^x$ only once, regardless of how many iterations of Markov chain sampling are to be run. More importantly, it improves the accuracy (measured by mean square error) of estimating the ratios of $C(w^\mu, w^x)$ at different values, which are needed in simulating Markov chain for updating w^μ and w^x, since two random variables with two different sets of values of w^μ and w^x whose expectations are computed with (34) are positively correlated. One can show this explicitly by approximating the mean square error of a ratio of two random variables with Taylor expansion of the ratio at their expected values.

C BCBCSF Computation Times

In this appendix, we show the computation times (seconds) of BCBCSF in Figure 9 for the simulation experiments in Section 3.3. All the computation was done on a Unix machine with Ultra Sparc III processors. Compared with using all the 5000 features, training and prediction with a small subset of selected features is much faster. For example, the computation time with only 713 features is only 20% of that with 5000 features. Therefore, BCBCSF with feature selection and bias correction results in better prediction performance with less computation. The small values of times for training and prediction with BCBCSF also show that the computation with BCBCSF is fast.

Figure 9: Computation times of BCBCSF.