Calibrating Predictions Based on a Selected Subset of Features from Bayesian Gaussian Classification Models

Longhai Li[†]

Manuscript, 2 January 2009

Abstract. There is an increasing demand for efficient, accurate and precise methods for predicting a discrete response with a great many of features, such as hyperspectral data generated by remote sensoring system, and gene expression data generated by microarray technology. For computational and other reasons, it is necessary to select a subset of features before fitting a statistical model, by looking at how relevant the features are to predicting the response. However, such feature selection procedure will result in overconfident predictions for future cases. Li, Zhang, and Neal (2008) present a general Bayesian method for avoiding this optimistic bias. The challenge in applying this method is the computation of an adjustment factor – the probability of a feature failing to pass the selection filter under the assumed Bayesian model. Li, Zhang, and Neal (2008) find an efficient method to compute the adjustment factor for naive Bayes classification models for *binary* features. In this paper, I show that efficient computation of the adjustment factor is also possible for Bayesian Gaussian classification models for *continuous* features, when the features are selected by F-statistic. The classification method presented in this paper is tested using a simulated data and a gene expression data that is related to small round blue cell tumors (SRBCT) of childhood.

1 Introduction

With some new technologies, human can now easily measure values of a great many features (also known as covariates, explanatory variables, inputs, etc.) of objects/subjects for the purpose of predicting certain characteristic of them, called response variable. In biology, microarray technology can simultaneously measure the expression levels of thousands of genes of a patient, which can be used to diagnose whether a certain disease is present in this patient (see eg Dudoit et al., 2002; Tibshirani et al., 2002; Khan et al., 2001). Similarly, mass spectrometry technology can produce high-dimensional protein profiles, which can also be used to diagnose diseases (see eg Wu et al., 2003). In engineering, remote sensing technology can measure the radiation responses in many wavelengths for a ground resolution element (pixel), producing a hyperspectral radiation profile of this pixel. For example, the airborne visible/infrared imaging spectrometer (AVIRIS) collects data in 224 spectral bands covering 0.4-2.5μm wavelength region. Engineers are interested in determining the material (eg, corn/wheat/soy) at this pixel from the hyperspectral radiation profile (see eg Tadjudin and Landgrebe, 1998). In document analysis, engineers want to determine the topic of a document by looking at the counts of occurrences of a large number (eg thousands) of words in this document (see eg Forman, 2007). In order to learn the relationship between the features and the response, people collect data of both features and response on a set of objects/subjects, called training cases. In statistics, such activities are called statistical classifi-

[†]Department of Mathematics and Statistics, University of Saskatchewan, Saskatoon, Saskatchewan, S7N5E6, CANADA. Email:longhai@math.usask.ca.

cation (also known as supervised learning), abstracted as building a conditional probability distribution of a discrete response variable given the features, from a set of training cases, for which we know the values of the response variable and features. A common feature of the previous high-dimensional classification problems is that the number of training cases is very small, typically less than a hundred, due to high cost in measuring the response variable, for example diagnosing diseases with conventional methods, and accessing a pixel to determine its material.

More features provide more information for predicting the response. However, there are many difficulties with using a great many of features in statistical inference when the number of training cases is small. Simple statistical methods, such as maximum likelihood estimation, will overfit the data, ie, model the noise in the data rather than the signal. For example, even a linear model will overfit a data set when the number of cases isn't much larger than the number of features. Sophisticated Bayesian methods do not suffer from overfitting problem. However, we will likely need to use Markov chain Monte Carlo methods to sample from the posterior distribution, which is computationally burdensome even for a moderate number of parameters. With more parameters, a Markov chain sampler will take longer for each iteration and require more memory, and may also need more iterations to converge, or get trapped more easily in local modes. Additionally, it is more difficult to come up with a reasonable model, in both data and prior distributions, for a great many features. Therefore, an almost ubiquitously used strategy in building a classification model with a great many features is to select only a subset of features which appear fairly predictive for response in training data (see eg Liu and Motoda, 2007). Often features are selected by looking at some simple score measuring the relevance between the response and the features, such as the correlation (in absolute value), and F-statistic. There are also many other sophisticated methods that look at the goodness of fitting regression models (see eg Tibshirani, 1996).

Unfortunately, feature selection will introduce optimistic bias into statistical inference, ie, the features selected will appear more predictive than they actually are. An extreme example is that even when all the features are irrelevant to the response, the selected features will appear fairly predictive to the response, which is, however, wholly made by chance. The consequence of using only selected features is that the predictive probabilities for future cases are overconfident. For example, for a group of test 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. This problem is also termed as that the predictive probabilities based only on the selected features are lack of calibration (Dawid, 1982), and abbreviated as feature selection bias (Li, Zhang, and Neal, 2008).

Such problems have been discussed by other researchers in different situations from different aspects. Hurvich and Tsai (1990) and Zhang (1992) respectively used Monte Carlo simulation and theoretical analysis to show that in regression models (where response is continuous) the actual coverage rate of confidence regions for regression coefficients based on features selected using certain information criteria is smaller than the nominal probability, and the estimate of residual variance is consistently smaller than the true value. As a remedy to this problem, it was suggested that the feature selection and model fitting should be performed separately on different cases. In the context of high-dimensional data, this is undesirable because high cost in measuring response limits the number of training cases. Recently, this problem is also noticed by researchers who have used gene expression data to perform classification task (Ambroise and McLachlan, 2002; Lecocke and Hess, 2004; Singhi and Liu, 2006; Raudys, Baumgartner, and Somorjai, 2005). It is found that if the cross validation evaluation of classification algorithms are performed on a data set containing only a subset of features selected beforehand based on the whole data set, the error rate could be misleadingly as low as 0. It is therefore suggested that the feature selection should be "internal" to the cross validation procedure, ie, the feature selection should be redone whenever the splitting of data set into training and test set is changed. This method can only evaluate the classification algorithm plus the feature selection method correctly,

but does not provide a way to build better classification method with a selected subset of features. An earlier paper by Dawid and Dickey (1977) did not particularly discuss feature selection in regression and classification problems, but addressed similar issue in more general settings, which showed that, if the reported data of an experiment is subject to manual selection, then the likelihood function of the observed values should be modified to account for the selection.

Li, Zhang, and Neal (2008) propose a Bayesian method to make well-calibrated probabilistic predictions with a selected subset of features. The idea is that our inference for parameters should be conditional on all the information available to us, that is, the values of features selected plus the information that a certain number of features are omitted because they are weakly related to the responses. In terms of the definition of calibration given by Dawid (1982), the predictions on future cases by this bias-corrected method are well-calibrated by averaging over the data sets drawn from the prior. Informally speaking, this solution uses partial information from omitted features to balance the over-estimated degree of relevance between the features and the response. The difficulty in applying this method is that the computation of the adjustment factor, ie, the probability that a feature fails to pass the selection filter, is burdensome, which involves integrating (or summing for discrete values) over the data and prior distributions. In Li, Zhang, and Neal (2008), an efficient method of calculating adjustment factor is found for binary classification models for *binary* features, with features selected by sample correlation. However, the practical application of the proposed method is limited because the models are too restrictive. In this paper, I show that efficient computation is also possible for Gaussian classification models for *continuous* features, which are used widely in practice, when the features are selected by F-statistic. Based on the precursors' work on computing the power function of ANOVA, we can efficiently compute the required adjustment factor, with a time nearly negligible compared to the time for training the models with Markov chain sampling methods.

This paper will be structured as follows. The general method for calibrating predictions based on a selected subset of features is reviewed in Section 2. In Section 3, I discuss in details how to apply the method in Bayesian Gaussian classification models. In Section 4.1, I use a simulated data set to illustrate the method, where I will show empirically that the predictions made by bias-corrected methods are indeed well-calibrated, and the calibration of predictions based on a selected subset of features is necessary in practice when loss incurred by different types of errors are different. In Section 4.2 I use a gene expression data to test the method, where I show that the bias-correction method does improve the predictions. The paper will conclude in Section 5.

2 Calibrating predictions based on a selected subset of features

Suppose we want to predict a response variable, y, based on the information in the numerical features $x_1, \ldots, x_p$, which we sometimes write as a vector, $\boldsymbol{x}_{1:p}$. We assume that we have complete data on n "training" cases, for which the responses are $y^{(1)}, \ldots, y^{(n)}$ (collectively written as $\boldsymbol{y}^{\text{train}}$) and the feature vectors are $\boldsymbol{x}^{(1)}_{1:p}, \ldots, \boldsymbol{x}^{(n)}_{1:p}$. We collectively write the values at all training cases of a feature numbered by t as $\boldsymbol{x}^{\text{train}}_t$, and write values of all p features as $\boldsymbol{x}^{\text{train}}_{1:p}$. (Note that when y, $\boldsymbol{x}$, or x_t are used without a superscript, they will refer to some unspecified case.) We wish to predict the response, y^*, for a "test" case, for which we know only the feature vector, $\boldsymbol{x}^*_{1:p}$. Our predictions will take the form of a distribution for y^*, rather than just a single-valued guess, whose determination depends also on the choice of loss function.

We are interested in problems where the number of features, p, is quite big — perhaps as large as ten or a hundred thousand — and accordingly (for pragmatic reasons) we intend to select a subset of features by some simple criterion measuring the relevance between the features and response, denoted

by $R(\boldsymbol{x}_t^{\text{train}}, \boldsymbol{y}^{\text{train}})$, for $t = 1, \ldots, p$. The examples of such criteria include the sample correlation of the response with features and the F-statistic given by expression (25), which is used here for data modeled by Gaussian distribution, and many others.

Although our interest is only in predicting the response, we assume that we have a model for the joint distribution of the response together with all the features. From such a joint distribution, with probability or density function $P(y, x_1, \ldots, x_p)$, we can obtain the conditional distribution for y given any subset of features, for instance $P(y \,|\, x_1, \ldots, x_k)$, with $k < p$. This is the distribution we need in order to make predictions based on this subset.

We will assume that a subset of features is selected by fixing a threshold, γ, for the relevance measure R of a selected feature with the response. We then omit feature t from the feature subset if $R(\boldsymbol{x}_t^{\text{train}}, \boldsymbol{y}^{\text{train}}) \leq \gamma$, retaining those features with a greater degree of relevance to $\boldsymbol{y}^{\text{train}}$. Another possible procedure is to fix the number of features, k, that we wish to retain, and then choose the k features whose relevance measured by R with the response is greatest, breaking any tie at random. If s is the retained feature with the weakest relevance with the response, we can set γ to $R(\boldsymbol{x}_s^{\text{train}}, \boldsymbol{y}^{\text{train}})$, and we will again know that if t is any omitted feature, $R(\boldsymbol{x}_t^{\text{train}}, \boldsymbol{y}^{\text{train}}) \leq \gamma$. If either the response or the features have continuous distributions, exact equality of relevance measure R will have probability zero, and consequently this situation can be treated as equivalent to one in which we fixed γ rather than k.

Regardless of the exact procedure used to select features, we will denote the number of features retained by k, we will renumber the features so that the subset of retained features is $x_1, \ldots, x_k$, and we will assume we know that $R(\boldsymbol{x}_t^{\text{train}}, \boldsymbol{y}^{\text{train}}) \leq \gamma$ for $t = k{+}1, \ldots, p$.

We can now state the basic principle of our bias-correction method: When forming the posterior distribution for parameters of the model using a selected subset of features, we should condition not only on the values in the training set of the response and of the k features we retained, but also on the fact that the other $p{-}k$ features have relevance measures R with the response less than γ. That is, the posterior distribution should be conditional on the following information:

$$\boldsymbol{y}^{\text{train}}, \quad \boldsymbol{x}_{1:k}^{\text{train}}, \quad \mathcal{S} : R(\boldsymbol{x}_t^{\text{train}}, \boldsymbol{y}^{\text{train}}) \leq \gamma, \text{ for } t = k{+}1, \ldots, p, \tag{1}$$

where $\boldsymbol{x}_{1:k}^{\text{train}} = (x_1^{\text{train}}, \ldots, x_k^{\text{train}})$.

In justifying our claim that this procedure will make well-calibrated predictions, we will assume that our model for the joint distribution of the response and all features, and the prior we chose for it, are appropriate for the problem. Now, imagine that rather than selecting a subset of features ourselves, after seeing all the data, we instead set up an automatic mechanism to do so, providing it with the value of γ to use as a threshold. This mechanism, which has access to all the data, will compute the values of relevance measure R of all the features with the response, select the subset of features by comparing these values with γ, and then erase the values of the omitted features, delivering to us only the identities of the selected features and their values in the training cases. All the information that *we* know is just that of (1) above. If the model describes the actual data generation mechanism, and the actual values of the model parameters are indeed randomly chosen according to our prior, Bayesian inference always produces well-calibrated results, on average with respect to the data and model parameters generated from the Bayesian model. The proof that the Bayesian inference conditional on all the information available to us is well-calibrated can be found in the appendix of Li, Zhang, and Neal (2008).

Our method requires computation of an adjustment factor, $P(\mathcal{S} \,|\, \alpha, y^{\text{train}})$, where α is the set of parameters relevant to this probability, and $\mathcal{S}$ represents the information regarding selection (see expression in (1)). Computing this factor is much easier if the $\boldsymbol{x}_1^{\text{train}}, \ldots, \boldsymbol{x}_p^{\text{train}}$ are conditionally independent given α and $\boldsymbol{y}^{\text{train}}$, since we can then write it as a product of factors pertaining to the various omitted

features. Integrating out the parameters specific to particular features, these factors are also *all the same*, since nothing distinguishes one omitted feature from another. We can then write

$$P(\mathcal{S} \,|\, \alpha,\, \boldsymbol{y}^{\text{train}}) \;=\; \prod_{t=k+1}^{p} P\big(R(\boldsymbol{y}^{\text{train}}, \boldsymbol{x}_t^{\text{train}}) \,\le\, \gamma \,|\, \alpha,\, \boldsymbol{y}^{\text{train}}\big) \tag{2}$$

$$\;=\; \Big[P\big(R(\boldsymbol{y}^{\text{train}}, \boldsymbol{x}_t^{\text{train}}) \,\le\, \gamma \,|\, \alpha,\, \boldsymbol{y}^{\text{train}}\big)\Big]^{p-k}, \tag{3}$$

where in the second expression, t represents *any* of the omitted features. Note that in this expression, $\boldsymbol{y}^{\text{train}}$ is conditional on, and hence considered fixed, whereas $\boldsymbol{x}_t^{\text{train}}$ is random. Since the time needed to compute this adjustment factor does not depend on the number of omitted features, we may hope to save a large amount of computation time by omitting many features.

Computing the single factor we do need is not trivial, however, since it involves integrals over both $\boldsymbol{x}_t^{\text{train}}$ and any parameters specific to particular features. As we will see, however, efficient computation is possible for the Bayesian Gaussian Classification models described here.

3 Application to Bayesian Gaussian classification models

In this section, I apply the general method described in Section 2 to naive Bayes Gaussian models, in which we assume that given the response and parameters all the features are independent and have Gaussian distributions. I will first define the model mathematically, and then discuss how to make predictions for the responses of test cases. Next, I will describe how to compute the adjustment factor for correcting bias from feature selection with F-statistic.

3.1 Bayesian Gaussian Classification models

We are interested in predicting a categorical response variable y, which is called class sometimes, given a set of predictor variables $x_1, \ldots, x_p$. Here we assume that y can take integer value from 1 to G, and all the x_i's are continuous. To learn the relationship between y and the x_i's, we collect a number, n, of observations on y and $x_1, \ldots, x_p$, called training data. 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 value g with a probability of ψ_g. Conditional on $y^{(i)} = g$, the predictor variables $x_1^{(i)}, \ldots, x_p^{(i)}$ are independent, and $x_j^{(i)}$ is distributed with $N(\mu_j^{(g)}, 1/\tau_j^x)$, where the subscript x isn't variable, which distinguishes this τ for predictor variables from those for other parameters below. Cases are also assumed to be independent given parameters. For simplicity of presentation below, we write the vector $\mu_1^{(g)}, \ldots, \mu_p^{(g)}$ collectively as $\boldsymbol{\mu}_{1:p}^{(g)}$, or $\boldsymbol{\mu}^{(g)}$ in the context of no confusion of the indicess of features, and write $\mu_j^{(1)}, \ldots, \mu_j^{(G)}$ as $\boldsymbol{\mu}_j$. Formal mathematical description is given as follows:

$$P(y^{(i)} = g \,|\, \boldsymbol{\psi}) \;=\; \psi_g, \quad \text{for } g = 1, \ldots, G\,, \tag{4}$$

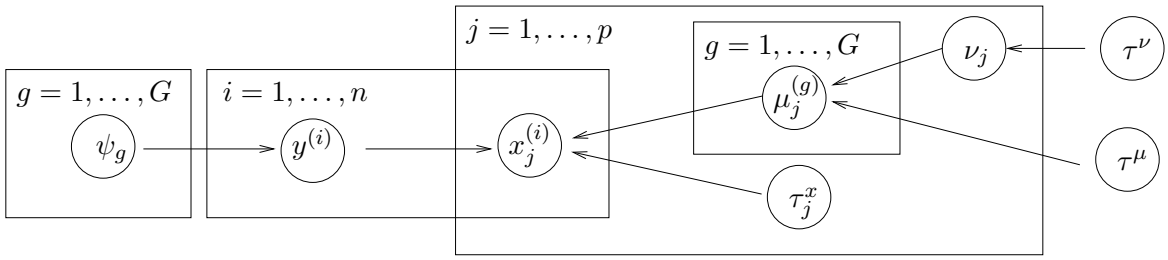
$$x_j^{(i)} \,|\, y^{(i)} = g, \mu_j^{(g)} \;\sim\; N(\mu_j^{(g)}, 1/\tau_j^x), \quad \text{for } j = 1, \ldots, p\,. \tag{5}$$

The mean of the prior for ν_j is fixed at 0. This does not make it lose generality since if the prior mean is believed to some value other than 0, one can subtract it from the data. A natural prior for $\boldsymbol{\psi}$ is a Dirichlet distribution (see eg Gelman et al., 2004) with parameters $c_1, \ldots, c_G$. I assign a hierarchical prior for $\boldsymbol{\mu}_j$ to reflect our prior belief that the mean parameters $\mu_j^{(g)}$ across different classes are close

to a common level ν_j, which is assigned with a Gaussian distribution to reflect our uncertainty on it. I choose conjugate Gamma distributions as priors for all the inverse variance of Gaussian distributions. Explicitely the priors for $\boldsymbol{\psi}$ and $\boldsymbol{\mu}$ are defined as follows:

$$\begin{aligned}
\psi_1, \ldots, \psi_G &\sim \text{Dirichlet}(c_1, \ldots c_G), && (6)\\
\mu_j^{(1)}, \ldots, \mu_j^{(G)} \,|\, \nu_j &\overset{\text{IID}}{\sim} N(\nu_j, 1/\tau^\mu), \quad \text{for } j = 1, \ldots, p, && (7)\\
\nu_1, \ldots, \nu_p &\overset{\text{IID}}{\sim} N(0, 1/\tau^\nu), && (8)\\
\tau_j^x &\sim \text{Gamma}(\alpha^x/2,\, \alpha^x w^x/2), && (9)\\
\tau^\mu &\sim \text{Gamma}(\alpha^\mu/2,\, \alpha^\mu w^\mu/2), && (10)\\
\tau^\nu &\sim \text{Gamma}(\alpha^\nu/2,\, \alpha^\nu w^\nu/2). && (11)
\end{aligned}$$

The above model is diagrammed below using convention of displaying a direct graphical model:



The priors for τ's are parameterized such that α indicates the precision, and w indicates the magnitude of $1/\tau$, variance of relevant Gaussian distribution. When α is no greater than 2, the corresponding distribution for $1/\tau$ has upward heavy tail (infinite mean), which strongly favors small values close to 0, but also allows some extraordinarily large values. Such distributions express appropriately our prior belief for $1/\tau^\mu$ and $1/\tau^\nu$ in high-dimensional data.

The hyperparameter τ^μ plays an important role in controlling the overall degree of relevance between the response and the features. When $1/\tau^\mu$ is larger, more features have large difference amongst $\mu_j^{(1)}, \ldots, \mu_j^{(G)}$, therefore more features are highly predictive to the response. As will be seen in Section 3.3, our adjustment method will have the effect of changing the posterior distribution of τ^μ, with an adjustment factor that is an increasing function of τ^μ.

The assumption that the predictor variables are independent given the response may not be realistic for the original data of a practical problem. When the predictor variables are correlated given the response, and the covariance matrices of the original variables are believed to the same for different classes, we can apply a linear transformation to the original variables such that the resulting variables fit better with the model defined as above. Suppose that given $y^{(i)} = g$, the original variables $z_1^{(i)}, \ldots, z_p^{(i)}$ have a multivariate distribution $N(\tilde{\boldsymbol{\mu}}^{(g)}, \Sigma)$. Using Choleski decomposition, we can factor Σ as LL'. We can then construct new variables $x_1^{(i)}, \ldots, x_p^{(i)}$ with L by letting $(x_1^{(i)}, \ldots, x_p^{(i)})' = L^{-1}\,(z_1^{(i)}, \ldots, z_p^{(i)})'$. Given $y^{(i)} = g$, the transformed variables $x_1^{(i)}, \ldots, x_p^{(i)}$ have a multivariate distribution $N(L^{-1}\tilde{\boldsymbol{\mu}}^{(g)}, I_p)$, where I_p is the p-dimensional identity matrix. We can estimate this unknown Σ using some non-Bayesian methods. In the real data example presented here, I shrink the ususal pooled estimation S (whose definition can be found from Dudoit et al., 2002) towards the diagonal matrix with a reasonable choice of λ:

$$\hat{\Sigma}(\lambda) = \frac{S + \lambda\,\text{diag}(S)}{1 + \lambda}, \tag{12}$$

where $\text{diag}(S)$ is a matrix with the diagonal elements equal to those of S and off-diagonal elements

equal to 0. This estimator keeps the usual unbiased estimates of variances unchanged, but shrinks the correlations towards 0 by a factor of $1/(1+\lambda)$.

3.2 Training models with Markov chain Monte Carlo methods

We want to predict the response y^* of a test case for which we know the values of predictor variables $x_1^*, \ldots, x_p^*$, collectively written as $\boldsymbol{x}_{1:p}^*$. After selecting a subset of features with some relevance measure R, and renumbering the features, we assume that $x_1, \ldots, x_k$ are retained and $x_{k+1}, \ldots, x_p$ are omitted. We want to derive the predictive probability of $y^* = g$ given $\boldsymbol{x}_{1:k}^*$, $\boldsymbol{x}_{1:k}^{\text{train}}$ and $\boldsymbol{y}^{\text{train}}$, for $g = 1, \ldots, G$, where the superscript "train" means collection of values of these variables on training cases, whose explicite definition can be found in Section 2.

If we ignore the feature selection, ie, pretend that these selected data are from the model described in Section 3.1, or when $k = p$, following the Bayes rule, we obtain that:

$$P(y^* = g \,|\, \boldsymbol{x}_{1:k}^*, \boldsymbol{x}_{1:k}^{\text{train}}, \boldsymbol{y}^{\text{train}}) = \frac{P(y^* = g \,|\, \boldsymbol{y}^{\text{train}})\, P(\boldsymbol{x}_{1:k}^* \,|\, \boldsymbol{x}_{1:k}^{\text{train}}, y^* = g, \boldsymbol{y}^{\text{train}})}{\sum_{g=1}^{G} P(y^* = g \,|\, \boldsymbol{y}^{\text{train}})\, P(\boldsymbol{x}_{1:k}^* \,|\, \boldsymbol{x}_{1:k}^{\text{train}}, y^* = g, \boldsymbol{y}^{\text{train}})}. \tag{13}$$

To compute (13), we need to compute the numerator for all $g = 1, \ldots, G$, then divide them by their sum. The first factor in this numerator can be computed by Pòlya urn scheme (Schervish, 1995):

$$P(y^* = g \,|\, \boldsymbol{y}^{\text{train}}) = \frac{n_g + c_g}{n + \sum_{g=1}^{G} c_g}, \tag{14}$$

where $n_g = \sum_{i=1}^{n} I(y^{(i)} = g)$ is the number of training cases with responses equal to g. The second factor can be written as:

$$\begin{aligned} &P(\boldsymbol{x}_{1:k}^* \,|\, \boldsymbol{x}_{1:k}^{\text{train}}, y^* = g, \boldsymbol{y}^{\text{train}}) \\ &= \int\int P(\boldsymbol{x}_{1:k}^* \,|\, \boldsymbol{\mu}_{1:k}, \boldsymbol{\tau}_{1:k}^x, y^* = g)\, P(\boldsymbol{\mu}_{1:k}, \boldsymbol{\tau}_{1:k}^x \,|\, \boldsymbol{x}_{1:k}^{\text{train}}, \boldsymbol{y}^{\text{train}})\, d\boldsymbol{\mu}_{1:k}\, d\boldsymbol{\tau}_{1:k}^x, \end{aligned} \tag{15}$$

where $\boldsymbol{\mu}_{1:k} = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k)$ and $\boldsymbol{\tau}_{1:k}^x = (\tau_1^x, \ldots, \tau_k^x)$. In order to approximate the above integral with Monte Carlo method, we draw samples from the posterior of $\boldsymbol{\mu}_{1:k}$ and $\tau_{1:k}^x$ given selected data, which can be implemented by drawing samples of all relevant parameters, ie, the following joint distribution:

$$P(\boldsymbol{\mu}_{1:k}, \boldsymbol{\nu}_{1:k}, \boldsymbol{\tau}_{1:k}^x, \tau^\mu, \tau^\nu \,|\, \boldsymbol{x}_{1:k}^{\text{train}}, \boldsymbol{y}^{\text{train}}) \tag{16}$$

$$\begin{aligned} \propto\ & \Big[\prod_{j=1}^{k}\prod_{g=1}^{G}\prod_{i\in N_g} \big((\tau_j^x)^{1/2} \exp\big(-\tau_j^x\,(x_j^{(i)} - \mu_j^{(g)})^2/2\big)\big)\Big] \cdot \\ & \Big[\prod_{j=1}^{k}\prod_{g=1}^{G} \big((\tau^\mu)^{1/2} \exp\big(-\tau^\mu\,(\mu_j^{(g)} - \nu_j)^2/2\big)\big)\Big] \cdot \Big[\prod_{j=1}^{k} (\tau^\nu)^{1/2} \exp\big(-\tau^\nu\, v_j^2/2\big)\Big] \cdot \\ & \Big[\prod_{j=1}^{k} (\tau_j^x)^{\alpha^x/2-1} \exp(-\tau_j^x\,\alpha^x\,w^x/2)\Big] \cdot \Big[(\tau^\mu)^{\alpha^\mu/2-1} \exp(-\tau^\mu\,\alpha^\mu\,w^\mu/2)\Big] \cdot \\ & \Big[(\tau^\nu)^{\alpha^\nu/2-1} \exp(-\tau^\nu\,\alpha^\nu\,w^\nu/2)\Big], \end{aligned} \tag{17}$$

where, N_g is the set of indices of training cases in class g. Up to some normalizing constants, the factors in square brackets are respectively the sampling distribution of selected features (assumed from the

model in Section 3.1), the prior for $\boldsymbol{\mu}_{1:k}$, the prior for $(\nu_1, \ldots, \nu_k)$, the prior for $\boldsymbol{\tau}^x_{1:k}$, the prior for τ^μ, and the prior for τ^ν, all conditional on necessary other parameters.

The posterior distribution in (16) can be sampled with Gibbs sampling, a variant of Markov chain Monte Carlo (MCMC) methods (for a comprehensive introduction of MCMC, see eg Neal, 1993; Liu, 2001). That is, we can partition the whole set of parameters into subsets, and sample the conditional distribution of each subset, in some fixed or random order, given others by direct sampling or Metropolis method (Metropolis et al., 1953). For this purpose, from (17) we can derive the following conditional distributions:

$$\mu_j^{(g)} \,|\, \cdots \;\sim\; N\left(\frac{\nu_j\tau^\mu + \bar{x}_j^{(g)} n_g \tau_j^x}{\tau^\mu + n_g\tau_j^x},\;\; \frac{1}{\tau^\mu + n_g\tau_j^x}\right), \tag{18}$$

$$\tau_j^x \,|\, \cdots \;\sim\; \text{Gamma}\left((\alpha^x + n)/2,\;\; \Big(\alpha^x\, w^x + \sum_{g=1}^{G}\sum_{i\in N_g} (x_j^{(i)} - \mu_j^{(g)})^2\Big)/2\right), \tag{19}$$

$$\nu_j \,|\, \cdots \;\sim\; N\left(\frac{\bar{\mu}_j G\tau^\mu}{\tau^\nu + G\tau^\mu},\;\; \frac{1}{\tau^\nu + G\tau^\mu}\right), \tag{20}$$

$$\tau^\mu \,|\, \cdots \;\sim\; \text{Gamma}\left((\alpha^\mu + Gk)/2,\;\; \Big(\alpha^\mu w^\mu + \sum_{j=1}^{k}\sum_{g=1}^{G} (\mu_j^{(g)} - \nu_j)^2\Big)/2\right), \tag{21}$$

$$\tau^\nu \,|\, \cdots \;\sim\; \text{Gamma}\left((\alpha^\nu + k)/2,\;\; \Big(\alpha^\nu w^\nu + \sum_{j=1}^{k} \nu_j^2\Big)/2\right), \tag{22}$$

where "$\cdots$" denotes all the parameters and data except the parameter before "|" in each conditional distribution, $\bar{x}_j^{(g)} = \sum_{i\in N_g} x_j^{(i)}/n$, ie, the average of x_j in class g, and $\bar{\mu}_j = \sum_{g=1}^{G} \mu_j^{(g)}/G$. We can directly sample from the above conditional distributions with standard methods.

The previous procedure is valid only when $k = p$, ie, when no feature selection occurs. When $k < p$, we should condition our inference on not only the values of selected features and response, but also the information $\mathcal{S} : R(\boldsymbol{x}_t^{\text{train}}, \boldsymbol{y}^{\text{train}}) \le \gamma,\ \text{for } t = k{+}1, \ldots, p$. That is, we need to sample from

$$P(\boldsymbol{\mu}_{1:k}, \boldsymbol{\nu}_{1:k}, \boldsymbol{\tau}^x_{1:k}, \tau^\mu, \tau^\nu \,|\, \boldsymbol{x}^{\text{train}}_{1:k}, \mathcal{S}, \boldsymbol{y}^{\text{train}}). \tag{23}$$

Analogous to expanding (16), we can write (23) as:

$$P(\boldsymbol{\mu}_{1:k}, \boldsymbol{\nu}_{1:k}, \boldsymbol{\tau}^x_{1:k}, \tau^\mu, \tau^\nu \,|\, \boldsymbol{x}^{\text{train}}_{1:k}, \boldsymbol{y}^{\text{train}}) \cdot \big(P(R(\boldsymbol{x}_t^{\text{train}}, \boldsymbol{y}^{\text{train}}) \le \gamma \,|\, \boldsymbol{y}^{\text{train}}, \tau^\mu, \tau^\nu)\big)^{p-k}, \tag{24}$$

where the first factor is expanded as in (17).

As consequence of this correction with $\mathcal{S}$, the conditional distributions of τ^μ and τ^ν should be multiplied by what we call adjustment factor: $\big(P(R(\boldsymbol{x}_t^{\text{train}}, \boldsymbol{y}^{\text{train}}) \le \gamma \,|\, \tau^\mu, \tau^\nu)\big)^{p-k}$. As will be seen from Section 3.3, the adjustment factor is unrelated to τ^ν, therefore only the conditional distribution of τ^μ needs adjusting. After this modification, we cannot sample from the conditional distribution of τ^μ directly, for which we could use some Markov chain sampling method, such as Metropolis methods with simple proposal distributions. Computing this adjustment factor isn't trivial, however, we can do this efficiently for the models described in this paper along with F-statistic as selection criterion. In next section, I will discuss it separately.

3.3 Computing adjustment factor

For the Gaussian model described in this paper, an appropriate way of selecting features is by looking at the F-statistic:

$$R_F(\boldsymbol{x}^{\text{train}}, \boldsymbol{y}^{\text{train}}) = \frac{\sum_{g=1}^{G} n_g(\bar{x}^{(g)} - \bar{x})^2/(G-1)}{\sum_{g=1}^{G}\sum_{i\in N_g}(x^{(i)} - \bar{x}^{(g)})^2/(n-G)}, \tag{25}$$

where $\bar{x}^{(g)}$ is the average of the $x^{(i)}$'s with $y^{(i)} = g$, ie, $\sum_{i\in N_g} x^{(i)}/n_g$, and $\bar{x}$ is the overall average $\sum_{i=1}^{n} x^{(i)}/n$. Here I have dropped the index for distinguishing different features, since this is a selection criterion for any feature.

In this section, I discuss how to compute the probability that a feature fails to pass the selection criterion R_F:

$$P(R_F(\boldsymbol{x}^{\text{train}}, \boldsymbol{y}^{\text{train}}) \le \gamma \,|\, \boldsymbol{y}^{\text{train}}, \tau^{\mu}, \tau^{\nu}). \tag{26}$$

The above probability is the same for all omitted features, and we therefore need to compute (26) only once and then raise it to the power of $p-k$.

From the standard results of calculating the power of ANOVA (see Knight, 2000, pages 411-416), we know that given $\mu_1, \ldots, \mu_G, \tau^x$ and $\boldsymbol{y}^{\text{train}}$, F-statistic in (25) has a noncentral F distribution with $G-1$ and $n-G$ degrees of freedom, and noncentrality parameter $\Lambda(\boldsymbol{\mu})\,\tau^x/2$, where:

$$\Lambda(\boldsymbol{\mu}) = \sum_{g=1}^{G} n_g(\mu^{(g)} - \tilde{\mu})^2, \tag{27}$$

where $\boldsymbol{\mu}$ is vector $(\mu^{(1)}, \ldots, \mu^{(G)})$, and $\tilde{\mu} = \sum_{g=1}^{G} n_g\mu^{(g)}/n$. We can therefore obtain that

$$P(R_F(\boldsymbol{x}^{\text{train}}, \boldsymbol{y}^{\text{train}}) \le \gamma \,|\, \boldsymbol{y}^{\text{train}}, \boldsymbol{\mu}, \tau^x) = P(\, F_{(G-1,\, n-G,\, \Lambda(\boldsymbol{\mu})\,\tau^x/2)} \le \gamma\,), \tag{28}$$

where $F_{(G-1,\, n-G,\, \Lambda(\boldsymbol{\mu})\,\tau^x/2)}$ denotes a random variable having a noncentral F distribution with $G-1$ and $n-G$ degrees of freedom and noncentrality parameter $\Lambda(\boldsymbol{\mu})\ \tau^x/2$. A noncentral χ^2 distribution can be expressed as an infinite mixture distribution of central χ^2 distributions with Poisson weights (Knight, 2000). Letting χ^2_{dg} denote a random variable having ordinary χ^2 distribution with degree freedom dg, we can now write the probability in (28) as:

$$P(\, F_{(G-1,\, n-G,\, \Lambda(\boldsymbol{\mu})\,\tau^x/2)} \le \gamma\,) \quad = \quad \sum_{k=0}^{+\infty} \left(f_k \,\text{Pois}\,(k; \Lambda(\boldsymbol{\mu})\tau^x/2)\right), \tag{29}$$

where

$$f_k = P\left(\frac{\chi^2_{G-1+2k}/(G-1)}{\chi^2_{n-G}/(n-G)} \le \gamma\right), \tag{30}$$

which can be computed with the CDF of ordinary F distribution, and $\text{Pois}(x;\lambda)$ is the Poisson probability mass function, ie, $\exp(-\lambda)\,\lambda^k/k!$.

To obtain the adjustment factor in (26), we need to integrate $\boldsymbol{\mu}$ and τ^x out in (29) over their priors (see equations (7),(8) and (9)). It is noticed that the probability in (26) is unrelated to the prior for ν,

therefore unrelated to τ^ν either, since the value of $\Lambda(\boldsymbol{\mu})$ does not change if we add a same constant to all $\mu^{(g)}$. We therefore can simplify the integration using transformation of dummy variables, as follows:

$$\begin{aligned} & P(R_F(\boldsymbol{x}^{\text{train}}, \boldsymbol{y}^{\text{train}}) \le \gamma \,|\, \boldsymbol{y}^{\text{train}}, \tau^\mu, \tau^\nu) \\ &= \int_{\tau^x} \int_{\boldsymbol{\mu}} \int_{\nu} \sum_{k=0}^{+\infty} \left(f_k \,\text{Pois}\,(k; \Lambda(\boldsymbol{\mu})\tau^x/2)\right) \prod_{g=1}^{G} \phi(\mu^{(g)}|\nu, \tau^\mu)\, \phi(\nu|0, \tau^\nu)\, P(\tau^x)\, d\nu\, d\boldsymbol{\mu}\, d\tau^x \quad (31) \\ \mu^{(g)} = m^{(g)} + v \\ = \int_{\tau^x} \int_{\boldsymbol{m}} \sum_{k=0}^{+\infty} \left(f_k \,\text{Pois}\,(k; \Lambda(\boldsymbol{\mu})\tau^x/2)\right) \prod_{g=1}^{G} \phi(m^{(g)}|0, \tau^\mu)\, P(\tau^x)\, d\boldsymbol{m}\, d\tau^x \quad (32) \\ z^{(g)} = m^{(g)} \sqrt{\tau_\mu} \\ = \int_{\tau^x} \int_{\boldsymbol{z}} \sum_{k=0}^{+\infty} \left(f_k \,\text{Pois}\,(k; \Lambda(\boldsymbol{z})\tau^x/(2\tau^\mu))\right) \prod_{g=1}^{G} \phi(z^{(g)}|0, 1)\, P(\tau^x)\, d\boldsymbol{z}\, d\tau^x, \quad (33) \end{aligned}$$

where $\phi(x|\mu, \tau)$ denotes Gaussian PDF with mean μ and variance $1/\tau$, and $P(\tau^\mu)$ denotes the PDF of the prior for τ^μ, which is a Gamma distribution (see (9)). We can now see that the integral in (33), as well as the adjustment factor, are unrelated τ^ν, though I included it at the beginning for formality.

The ratios of integrals in (33) at different τ^μ are needed in simulating a Markov chain for sampling τ^μ. We can approximate the integral over $\boldsymbol{z}$ and τ^x using Monte Carlo method to estimate this expectation:

$$E\left(\sum_{k=0}^{+\infty} \left(f_k \,\text{Pois}\,(k; \Lambda(\boldsymbol{Z})\mathcal{T}^x/(2\tau^\mu))\right)\right), \quad (34)$$

where $\boldsymbol{Z} = (Z^{(1)}, \ldots, Z^{(n)})$ are distributed with independent standard Gaussian distributions, and $\mathcal{T}^x$ is distributed with the prior for τ^x. Since the distributions for $\boldsymbol{Z}$ and $\mathcal{T}^x$ are both free of τ^μ, we can use a common pool of random samples of $\boldsymbol{Z}$ and $\mathcal{T}^x$ in approximating (34) for different τ^μ. There are two benefits in doing this. First, it saves computation time. We need to draw samples of $\boldsymbol{Z}$ and compute $\Lambda(\boldsymbol{Z})$ in (27) only once, regardless of how many iterations of Markov chain sampling are run. More importantly, it improves the accuracy of estimating the ratios of (34) at different values of τ^μ, since the random samples for approximating (34) for different τ^μ are positively correlated. The fact that using positively correlated samples improves the estimate of ratio of $a = E(X)$ to $b = E(Y)$ can be demonstrated with the following expression of the mean square error of estimate $\bar{X}/\bar{Y}$ from a/b, where $\bar{X}$ is the average of samples from X, and $\bar{Y}$ is the average of samples from Y: $E\big((\bar{X}/\bar{Y} - a/b)^2\big) = (1/b^2)\text{Var}(\bar{X}) + (a^2/b^4)\,\text{Var}(\bar{Y}) - 2\,(a/b^3)\,\text{Cov}(\bar{X}, \bar{Y}) + o(1/n^3)$. If $a/b^3 > 0$, when $\text{Cov}(\bar{X}, \bar{Y})$ is positive, the mean square error of $\bar{X}/\bar{Y}$ is smaller than when $\bar{X}$ and $\bar{Y}$ are independent. In our context, X and Y are the random variables in (34) associated with two different τ^μ, and a and b are two probabilities (therefore are positive); since τ^μ is always positive, X and Y are positively correlated, and $a^3/b > 0$.

Finally, we want to mention some minor computational tricks for computing (34). First, the infinite summation over k can be truncated to finite based on the values of f_k, which converge to 0 as k goes to ∞, while the Poisson weights are always less than 1. In addition, f_k needs to be computed only once for one Markov chain, regardless of how many iterations of Markov chain sampling are run. Second, we can switch this summation over k with the expectation in (34), with consequence of that we actually average the Poisson weights for each fixed k over the samples of $\boldsymbol{Z}$ and $\mathcal{T}^x$. This avoids multiplying f_k to Poisson weights at all samples of $\boldsymbol{Z}$ and $\mathcal{T}^x$. Third, the Poisson weights for $k+1$ can be computed from the weights for k by multiplying the mean parameter and $k+1$.

3.4 Testing calibration from single-valued guesses

We can test whether predictive probabilities produced by certain methods are well-calibrated empirically from single-valued guesses for test cases. Let's use $\hat{P}(y^* \,|\, \mathcal{X}^*)$ to denote a predictive probability for response y^* of a test case given $\mathcal{X}^*$, which is all the information on which this predictive probability is conditional. Without correcting for selection bias, $\mathcal{X}^*$ includes $x^*_{1:k}$, $\boldsymbol{x}^{\text{train}}_{1:k}$ and y^{train}, and with correcting for selection bias, it also includes $\mathcal{S}$. To make single-valued guess, we must define a loss function expressing loss incurred by different errors (Schervish, 1995). For discrete y, we use $L(y^* \to y')$ to denote the loss incurred if y^* is guessed as y'. If we trust a predictive probability $\hat{P}(y^* \,|\, \mathcal{X}^*)$, ie, assume that it is equal to the true conditional distribution $P(y^* \,|\, \mathcal{X}^*)$, we should guess y as the value y' minimizing the expected loss:

$$\text{EL}_{\hat{P}}(y'; \mathcal{X}^*) = E_{\hat{P}}(L(y^* \to y') \,|\, \mathcal{X}^*) = \sum_{y^*=1}^{G} \hat{P}(y^* \,|\, \mathcal{X}^*)\, L(y^* \to y') \tag{35}$$

The best guess based on $\hat{P}$ is:

$$Y_{\hat{P}}(\mathcal{X}^*) = \arg\min_{y'} \text{EL}_{\hat{P}}(y'; \mathcal{X}^*), \tag{36}$$

and the corresponding expected loss at this guess is:

$$\text{EL}^B_{\hat{P}}(\mathcal{X}^*) = \text{EL}_{\hat{P}}(Y_{\hat{P}}(\mathcal{X}^*); \mathcal{X}^*) = \min_{y'} \text{EL}_{\hat{P}}(y'; \mathcal{X}^*). \tag{37}$$

From equation (35), we see that if $\hat{P}(y^* \,|\, \mathcal{X}^*)$ is equal to the true conditional distribution $P(y^* \,|\, \mathcal{X}^*)$, by double expectation formula, ie, $E(Y) = E(E(Y \,|\, X))$, the mean of expected loss $\text{EL}^B_{\hat{P}}(\mathcal{X}^*)$ over $\mathcal{X}^*$ is equal to the mean of actual loss, ie,

$$E\big(\, (\text{EL}^B_{\hat{P}}(\mathcal{X}^*)\,\big) = E\big(\, L(y^* \to Y_{\hat{P}}(\mathcal{X}^*)\,\big). \tag{38}$$

We can therefore use equation (38) to test whether the predictive probabilities are equal to the true conditional probabilities, with both sides of equation (38) estimated by averaging over test cases. I will refer to the estimates of the two sides of (38) from test cases as *average of expected loss* and *average of actual loss* respectively. When the predictive probabilities given by $\hat{P}$ are well-calibrated, the average of expected loss, which is known in making predictions, can be used to predict the actual loss in the future.

I now specialize the above discussion for the 0-1 loss: $L(y^* \to y') = I(y^* \neq y')$. Under this loss, $\text{EL}_{\hat{P}}(y'; \mathcal{X}^*) = 1 - \hat{P}(y' \,|\, \mathcal{X}^*)$, the best guess for y^*, $Y^B_{\hat{P}}(\mathcal{X}^*)$, is therefore the mode of $\hat{P}(y^* \,|\, \mathcal{X}^*)$, and the expected loss at this guess, $\text{EL}^B_{\hat{P}}(\mathcal{X}^*)$, is $\min_{y'}(1 - \hat{P}(y' \,|\, \mathcal{X}^*)$. The estimate of the right-hand side of (38) over test cases is just the usual error rate, which I will call *actual error rate*, in contrast, the estimate of the left-hand side of (38) is called *expected error rate*. From equation (38), when $\hat{P}$ is equal to the true conditional distribution, the expected error rate should be empirically close to the actual error rate.

4 Examples

4.1 Experiments with simulated data

Fixing $\tau^\nu = 100, \tau^\mu = 100, \alpha^x = 4, w^x = 1, G = 6$, and $p = 4000$, I generated a data set of 5200 cases from the model described in Section 3.1, with the values of response drawn from uniform distribution

over the set $\{1, \ldots, G\}$. I randomly selected 200 cases as training set, and left the remaining 5000 as test cases to evaluate the predictive performance. With the training cases, I then selected four subsets of features, containing 10, 50, 200, and 1000 features, by comparing their values of F-statistic, giving thresholds 6.20, 4.41, 3.15, and 1.79 respectively for these four subsets. These are the values of γ used by the bias-correction method when computing the adjustment factor of equation (26).

In training the models with MCMC, I set the prior as follows: for $\boldsymbol{\psi}$, $c_1 = 1, \ldots, c_G = 1$, for τ^x, $\alpha^x = 4$ and $w^x = 1$, and for τ^μ and τ^ν, $\alpha^\mu = \alpha^\nu = 1.5$, and $w^\mu = w^\nu = 0.01$. When bias-correction method is applied, for sampling $\log(\tau^\mu)$, I used Metropolis method with Gaussian proposal centering at previous state and with standard deviation 0.5. For each iteration of Gibbs sampling, this Metropolis update was applied 5 times. In computing the adjustment factor with approximation (34), the upper bound of k was set to include f_k larger than e^{-10}, and number of samples of $\boldsymbol{Z}$ and $\mathcal{T}^x$ was set as 1000. These settings are adequate for this problem, different settings may be necessary in other problems.

I ran 6000 iterations of Gibbs sampling with settings as above to draw samples from the posterior distribution of $\boldsymbol{\mu}_{1:k}, \boldsymbol{\nu}_{1:k}, \boldsymbol{\tau}^x_{1:k}, \tau^\mu, \tau^\nu$, with and without correction for selection bias. The first 750 iterations were omitted, and every 15th iteration afterwards was used to make Monte Carlo estimations for test cases. I obtained the predictive probabilities of y equal to each of $g = 1, \ldots, 6$ for 5000 test cases. I examined these predictive probabilities to demonstrate the bias-correction method.

I first directly plotted the predictive probabilities of $y = 1$ for only 500 test cases. Figure 1 plots the predictive probabilities given by the methods with correction for selection bias against without correction. From it, we see clearly that the predictive probabilities without bias-correction tends to be closer to 0 and 1, ie, are more confident than those with bias-correction. This confidence is however incorrect. Let's look at how well calibrated the predictive probabilities of $y = 1$ are in Table 1. We grouped the 5000 test cases into 10 categories according to the first decimals of predictive probabilities, ie, the predictive probabilities of $y = 1$ in category C are between $C/10$ and $(C+1)/10$, for $C = 0, \ldots, 9$. For those test cases in category C, I calculated the average of predictive probabilities, namely "Pred" in Table 1, and the actual fraction of $y = 1$, namely "actual" in Table 1. If the predictive probabilities are well-calibrated, the "Pred" and "Actual" in each category should be close if the number of cases in this category isn't very small (Dawid, 1982). From Table 1, it is clear that without correction for selection bias, the values in "Pred" are not close to those in "Actual". For example the values of "Pred" are incorrectly close to 0 in category $C = 0$, and to 1 in category $C = 9$. After correcting for selection bias, the "Pred" and "Actual" are fairly close, indicating that the predictive probabilities are well-calibrated.

We now examine the effect when the poorly-calibrated predictive probabilities are used in practice to make single-valued guesses, given certain choice of loss function. I compared the average of expected loss agrees with the average of actual loss for each method, and also compared the average of actual loss between uncorrected and corrected methods.

Let's first consider the 0-1 loss $L(y \to y') = I(y \neq y')$, which are discussed specifically in Section 3.4. I compared the actual error rates of the corrected and uncorrected methods, when different subset of features are used, shown by the solid lines on the upper-left plot in Figure 2. Surprising to some readers, the actual error rates are very close for corrected and uncorrected methods, in other words, feature selection does not affect the average loss when the loss incurred by different types of errors are the same. This fact was confirmed also by Singhi and Liu (2006), and by Li, Zhang, and Neal (2008). This is good news for practitioners if the error rate is the only issue in the problem they consider. However, this is often not the case. In particular, people in practice usually need the uncertainty estimate accompanying single-valued guesses. In such situations, an accurate estimate of error rate for future cases, ie, the estimate of the left-hand side of (38), namely expected error rate, is needed too. I plotted the expected error rates in Figure 2 with dashed lines. If the predictive probabilities

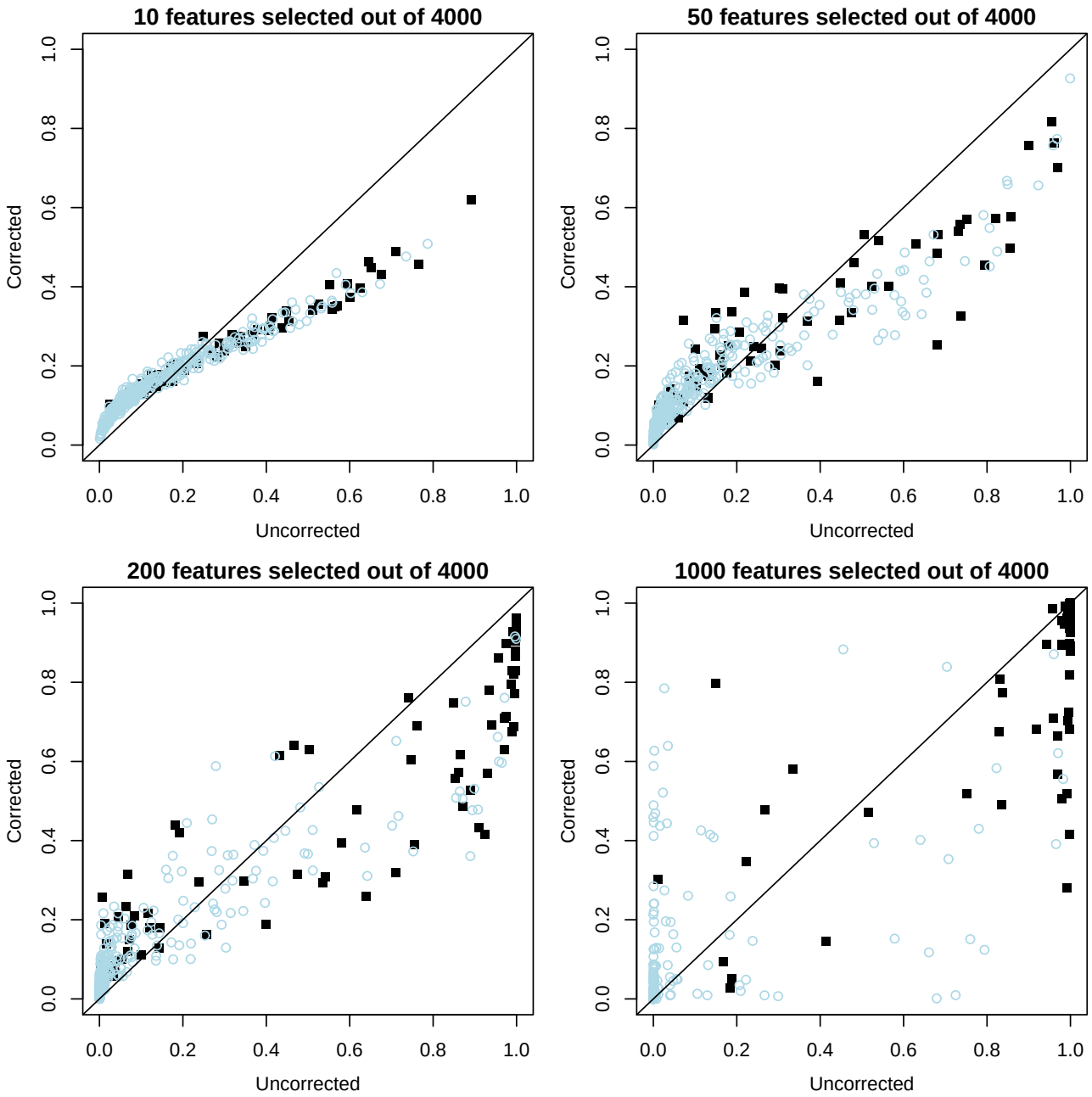


Figure 1: Scatter plots of predictive probabilities of $y = 1$ for 500 test cases simulated from a Bayesian naive Bayes Gaussian classification model, computed by methods with and without correction for selection bias. The solid squares indicate cases with $y = 1$ and the hollow circles indicate cases with $y \neq 1$. Without correction for selection bias, the predictive probabilities tend to be closer to 0 and 1.

	10 features selected out of 4000						50 features selected out of 4000					
	Corrected			Uncorrected			Corrected			Uncorrected		
C	#	Pred	Actual	#	Pred	Actual	#	Pred	Actual	#	Pred	Actual
0	1000	0.072	0.059	1888	0.048	0.090	1968	0.048	0.058	2621	0.023	0.087
1	1827	0.145	0.150	968	0.143	0.175	962	0.144	0.167	436	0.143	0.167
2	839	0.243	0.257	484	0.243	0.240	442	0.243	0.253	237	0.247	0.308
3	237	0.342	0.350	300	0.344	0.310	266	0.347	0.380	158	0.347	0.316
4	80	0.438	0.525	162	0.443	0.290	162	0.443	0.432	123	0.451	0.398
5	15	0.524	0.600	90	0.550	0.411	97	0.544	0.557	107	0.552	0.336
6	2	0.618	1.000	63	0.643	0.476	50	0.646	0.580	83	0.650	0.422
7	0	–	–	36	0.740	0.389	28	0.757	0.750	81	0.747	0.543
8	0	–	–	9	0.846	1.000	19	0.844	0.895	68	0.846	0.544
9	0	–	–	0	–	–	6	0.926	0.833	86	0.954	0.698

	200 features selected out of 4000						1000 features selected out of 4000					
	Corrected			Uncorrected			Corrected			Uncorrected		
C	#	Pred	Actual	#	Pred	Actual	#	Pred	Actual	#	Pred	Actual
0	2490	0.029	0.033	2941	0.010	0.061	3025	0.008	0.011	3126	0.003	0.011
1	501	0.144	0.180	187	0.145	0.257	131	0.143	0.130	54	0.148	0.222
2	263	0.248	0.274	104	0.246	0.288	81	0.246	0.259	41	0.244	0.244
3	197	0.351	0.391	94	0.347	0.287	49	0.349	0.286	32	0.351	0.281
4	143	0.444	0.455	79	0.453	0.342	53	0.447	0.264	38	0.450	0.342
5	120	0.549	0.608	74	0.550	0.405	51	0.553	0.549	20	0.552	0.450
6	91	0.648	0.670	86	0.647	0.547	53	0.652	0.698	22	0.646	0.364
7	79	0.754	0.759	70	0.751	0.414	50	0.748	0.700	27	0.758	0.444
8	66	0.844	0.864	92	0.860	0.598	78	0.851	0.821	53	0.857	0.679
9	50	0.943	0.940	273	0.969	0.777	429	0.979	0.981	587	0.991	0.925

	Complete data		
C	#	Pred	Actual
0	3169	0.003	0.009
1	52	0.147	0.077
2	31	0.241	0.355
3	24	0.348	0.417
4	22	0.443	0.364
5	24	0.546	0.375
6	23	0.644	0.565
7	26	0.751	0.577
8	42	0.850	0.667
9	587	0.990	0.952

Table 1: Comparison of calibration for predictions found with and without correction for selection bias, on data simulated from a Bayesian naive Bayes Gaussian classification model. Results are shown with four subsets of features and with the complete data (for which no correction is necessary). The test cases were divided into 10 categories by the first decimal of the predictive probability of class 1, which is indicated by the 1st column "C". The table shows the number of test cases in each category for each method ("#"), the average predictive probability of class 1 for cases in that category ("Pred"), and the actual fraction of these cases that were in class 1 ("Actual").

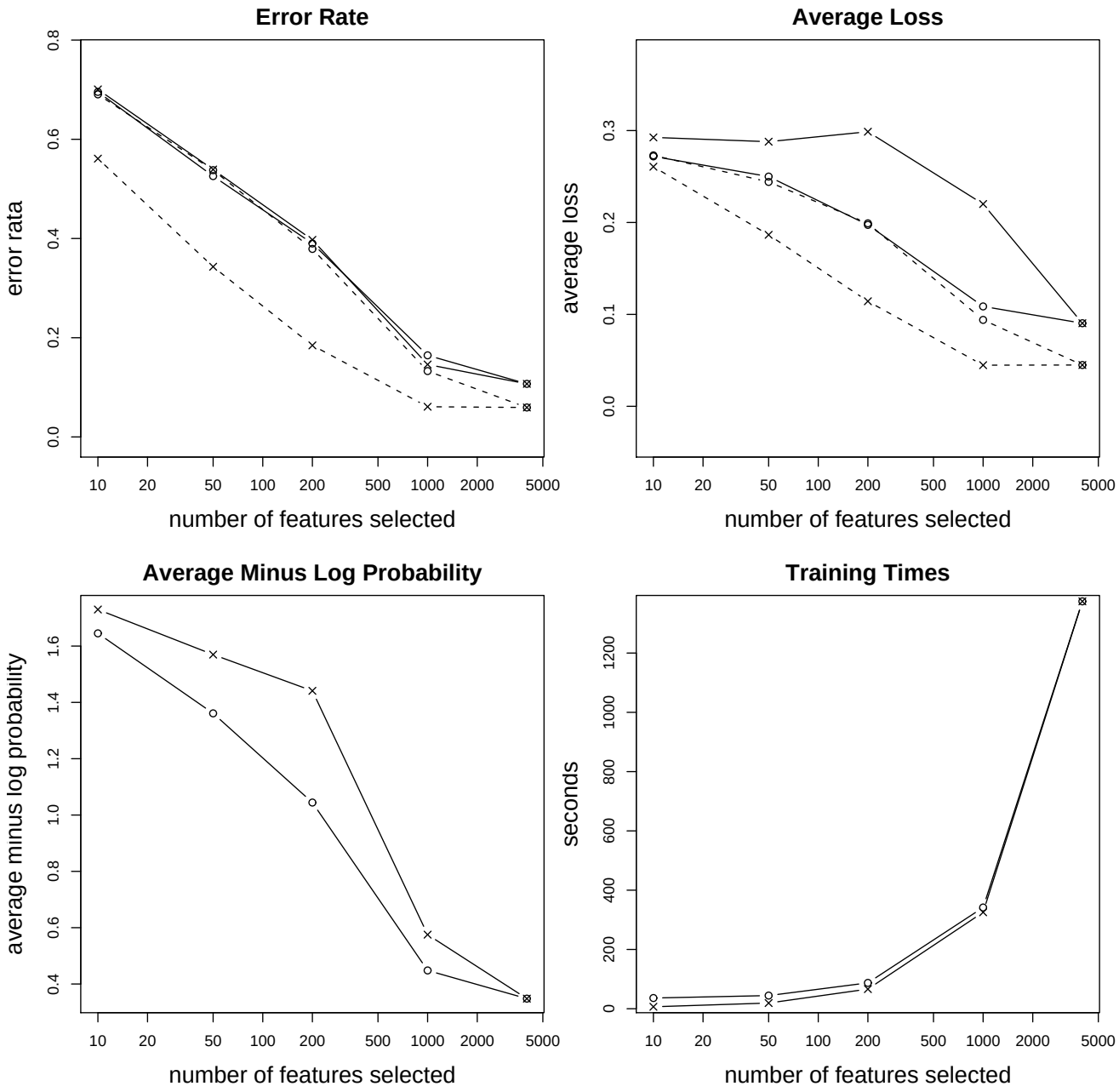


Figure 2: These plots compare the prediction methods with and without correction for selection bias in terms of, error rate (top-left), average of loss based on a loss function whose values are drawn from $\exp(N(0, 2^2))$ (top-right), average of the minus log probabilities of observing the actual symbols (bottom-left), and the times for training the models with Gibbs sampling (bottom-right), on data simulated from a Bayesian naive Bayes Gaussian classification model. The lines with $\times$ show the methods without correction for selection bias, the lines with $\circ$ show the methods with correction. On the top two plots, the dashed lines show the expected error rates (left), or the average of expected loss (right). (Note that the curves of actual error rates and training times overlap for methods with and without correction for selection bias.)

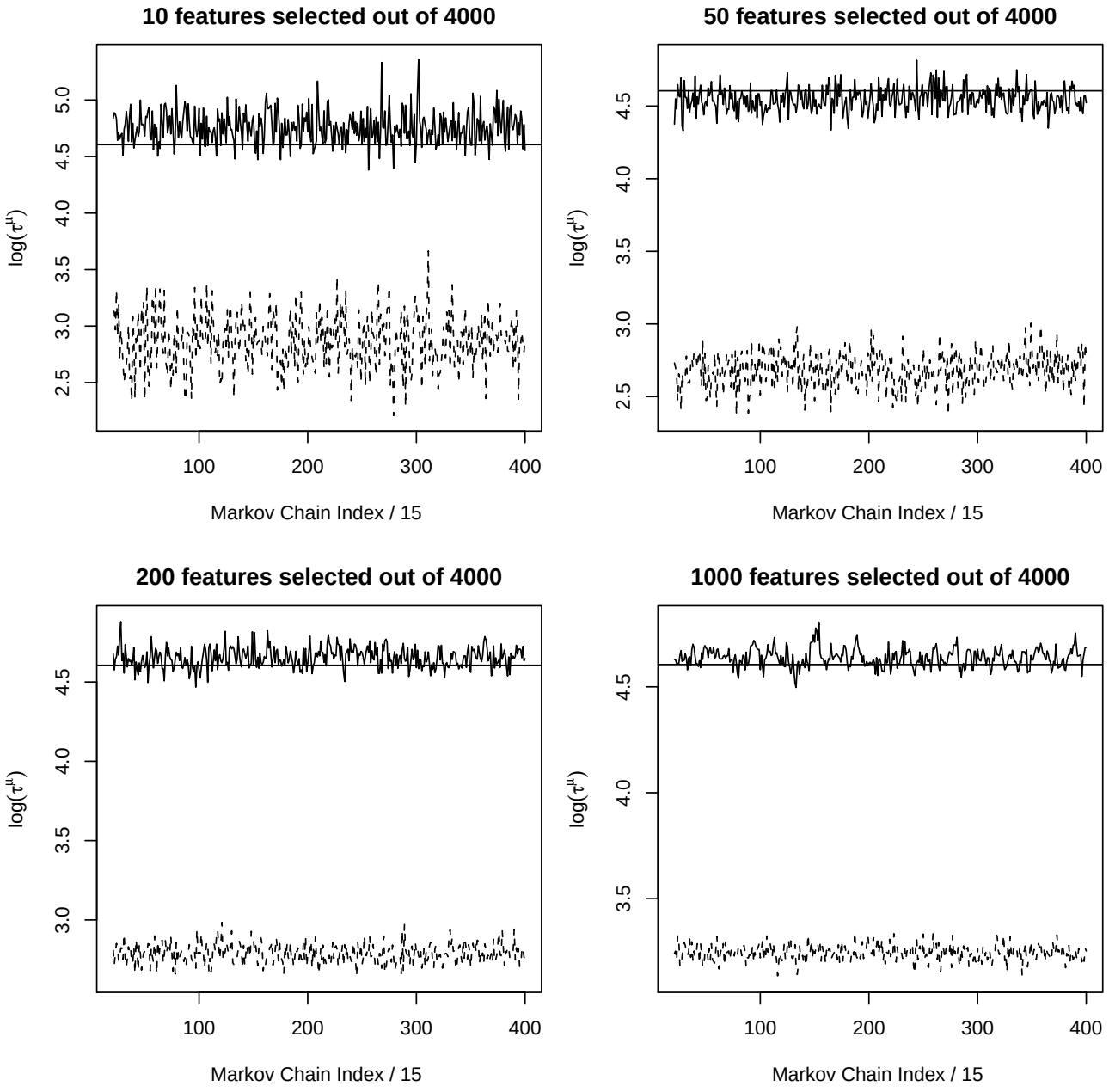


Figure 3: Markov chain traces of $\log(\tau^{\mu})$, on data simulated from a Bayesian naive Bayes Gaussian classification model. The dashed lines show methods without correction for selection bias, and the solid lines show methods with correction. The horizontal straight lines indicate the true values of $\log(\tau^{\mu})$ generating the data, which is log(100). Without correction for selection bias, the Markov chains of $\log(\tau^{\mu})$ move around some values smaller than the true value.

are well-calibrated, the expected error rates are close to the actual error rates. From Figure 2, for the corrected methods, they are fairly close, but they are not for uncorrected methods, with the expected error rates consistently smaller than the actual error rates, indicating the predictive probabilities are overconfident.

In situations where loss incurred by different errors are not the same (eg, missing detecting cancer in a patient incurs more loss than erroneously detecting cancer in a normal patient), poorly-calibrated predictive probabilities could result in more loss on average. I experimented with this by drawing the values of $L(y^* \to y')$ from distribution $\exp(N(0, 2^2))$ when $y^* \neq y'$, and set $L(y^* \to y^*) = 0$. The average of actual loss are shown for corrected and uncorrected methods on the top-right plot in Figure 2, from which we see that the average of actual loss by uncorrected methods are consistently larger than corrected methods. The average of expected loss are also shown with dashed lines on the same plot in Figure 2. For uncorrected methods, the average of expected loss are again consistently smaller than the average of actual loss, whereas for corrected methods, they stay fairly close.

From Figure 2, as well as Table 1, we see that when all of 4000 features were used, some overconfidence in predictive probabilities also occurs, with the averages of expected loss a little smaller than the averages of actual loss, and "pred" more extreme than "actual". In some other experiments (which are not shown here) I have also observed that using all features might even resulte in predictions with higher error rate and average loss. I suspected that this was because of that Markov chains got trapped in some local modes and therefore failed to explore parameter space thoroughly, as the number of parameters is so large. I have experimented with smaller number of total features, such as 1000, and/or with larger number of training cases, such as 1000, and haven't seen such bias or worse predictions. Since such data sets do not look like real data sets, I do not use them as illustrative examples here.

Prediction methods can be evaluated also by the average of minus log probabilities (AMLP) of observing the actual values of responses:

$$\frac{1}{n} \sum_{i=1}^{n} \big[-\log(\hat{P}(y^{(i)} \,|\, \mathcal{X}^{(i)})) \big], \tag{39}$$

where $y^{(i)}$ denotes the true value of response for ith test case, and $\mathcal{X}^{(i)}$ is the information used to predict $y^{(i)}$ by $\hat{P}$. This criterion penalizes heavily those probabilities of actual values close to 0. The plots of AMLPs for uncorrected and corrected methods are shown in Figure 2, indicating that corrected methods perform better than uncorrected methods.

Direct explanation of feature selection bias is that the posterior distribution of τ^μ favors incorrectly smaller values than the true value generating the data set. I plotted the Markov chain traces of $\log(\tau^\mu)$ in Figure 3. With correction for selection bias, the posterior distribution of $\log(\tau^\mu)$ favors values around the true value $\log(100)$, whereas without correction, the posterior distribution concentrates around smaller values than the true value, which was displayed by that the Markov chain traces never touch the true value.

From Figure 2 we can see that the extra time for computing adjustment factor is very little, nearly negligible. Compared to using all 4000 features, training times with a selected subset of features are much less. Though I do not plot prediction times here, one can believe that prediction times will increase rapidly with number of features selected, due to computation of difference between feature value and mean parameter μ. Combining with previous observation that Markov chains for training models with a large number of features more easily get trapped in local modes, which may yield poorly-calibrated, and even less accurate predictions, we can see clearly the benefits of selecting a smaller number of features, provided that the feature selection bias is corrected effectively, as we do here.

4.2 Experiments with SRBCT gene expression data

I also tested the bias-correction method on a gene expression data related to small round blue cell tumors (SRBCT) of childhood, which was first released along with Khan et al. (2001). SRBCTs include neuroblastoma (NB), rhabdomyosarcoma (RMS), non-Hodgkin lymphoma (NHL) and the Ewing family of tumors (EWS), are so named because of their similar appearance on routine histology. However, accurate diagnosis of SRBCTs is essential because the treatment options, responses to therapy and prognoses vary widely depending on the diagnosis. As their name implies, these cancers are difficult to distinguish by light microscopy, and currently no single test can precisely distinguish these cancers. Khan et al. (2001) proposed approaches to diagnosing these four types of tumors from gene expression data. The released data set (available from `http://research.nhgri.nih.gov/microarray/Supplement/`) contains expression data of 2308 genes on 83 patients who have been categorized into one of these tumors with conventional diagnosis.

I divided these 2038 genes randomly into 10 almost equal groups, producing 10 small data sets, each with about 204 features, as well as the same tumor indicators. I applied the corrected and uncorrected methods separately to each of these 10 data sets, allowing some assessment of variability when comparing performance. For each of these 10 data sets, I used 20-fold cross validation to obtain predictive probabilities for the class in the 83 cases. In this cross validation procedure, the cases are divided into 20 almost equal subsets, in turn I left out one of the 20 subsets of cases as test cases, treated the remaining 19 subsets as training cases. For each of such splitting of 83 cases, I estimated pooled variance with training cases, then used it to transform the whole data set in the way given in Section 3.1 with $\lambda = 10$, next selected the 10 features with the largest F-statistic in training cases, and finally ran MCMC to find the predictive probabilities for the left-out test cases, with and without bias correction. Note that here I have made data transformation and feature selection "internal" to cross validation (Lecocke and Hess, 2004), ie, without using any information from test cases in performing data transformation and feature selection. In running each Markov chain, I used the same prior distributions except setting $\alpha^x = 2$ and $w^x = 0.3$, and the same computational methods, as for the demonstration in Section 4.1.

Using the predictive probabilities produced as above, I compared corrected and uncorrected methods. The results are shown by Figure 4. The top two plots show that our bias correction method reduces optimistic bias in the predictions. For each of the 10 data sets, this plot shows the actual error rate against the error rate expected from the predictive probabilities. For all 10 data sets, the expected error rate with the uncorrected method is substantially less than the actual error rate. This optimistic bias is reduced by our bias-correction method, though it is not eliminated entirely. The remaining bias presumably results from the failure in this data set of the models we assume here, especially the assumption that features are independent given class. The bottom-left plot in Figure 4 shows the averages of actual loss based on a random loss functions drawn from $\exp(N(0, 2^2))$, and we can see that corrected methods made much less or no much more loss on average than uncorrected methods. The bottom-right plot shows the average of minus log probabilities of observing true responses, and again we can see that with correction of selection bias, the predictions are much improved.

I have also applied the classification algorithms with and without correction for selection bias on the whole data set with certain numbers of genes selected with F-statistic. Since using all features this data set makes the classification very simple, when more than 50 genes are included, both corrected and uncorrected methods performed almost identically, giving an error rate of 0.024, ie, 2 out of 83 cases were misclassified, and giving similar expected error rates. This error rate is the same as the result reported by Tibshirani et al. (2002), but the paper didn't state it clearly whether they have made feature selection "internal" to their cross validation procedure.

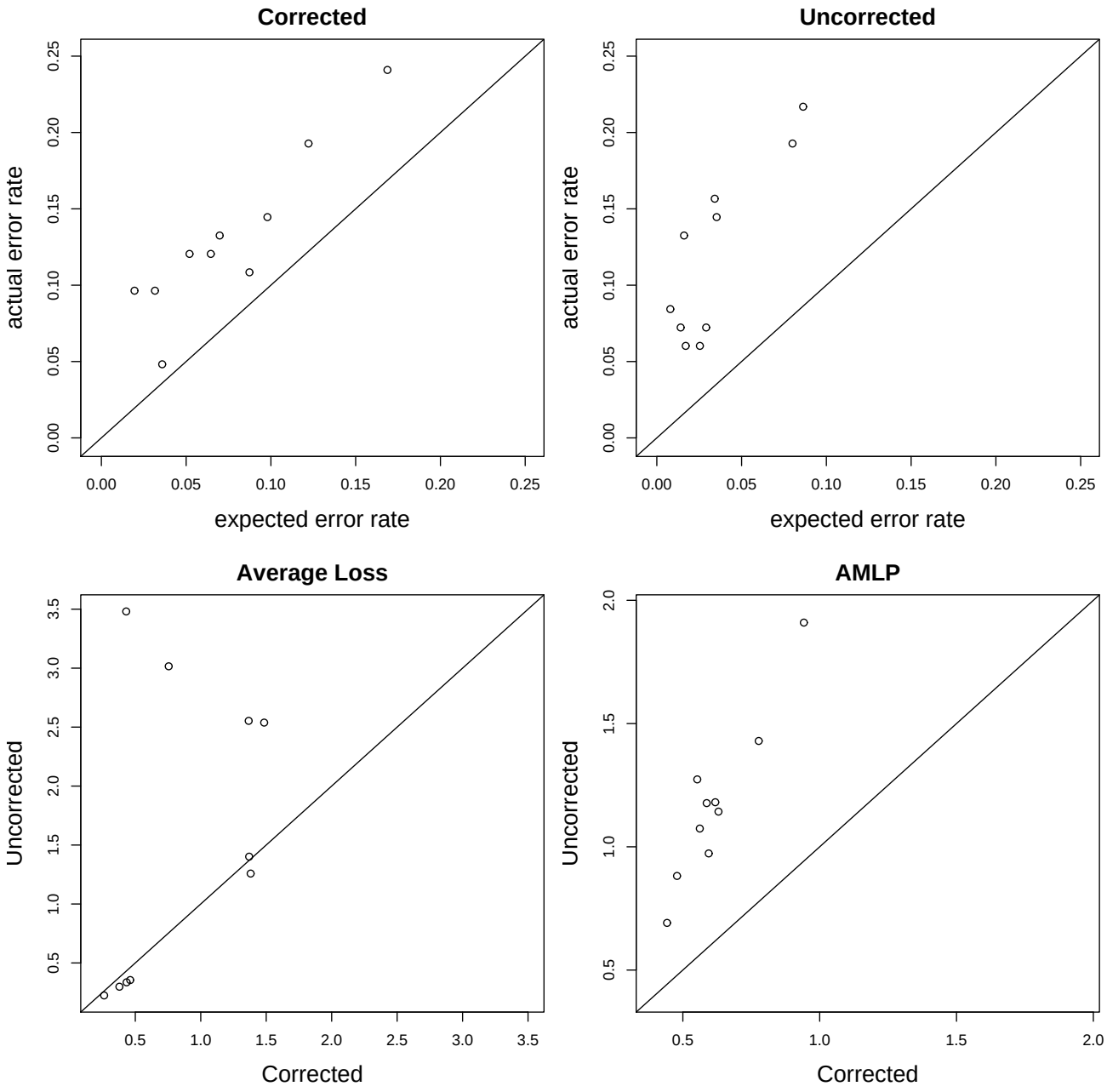


Figure 4: Comparison of corrected and uncorrected methods on small round blue cell tumors (SRBCT) gene expression data. The top two plots compare the expected error rates and actual error rates separately for corrected and uncorrected methods. The bottom two plots compare corrected and uncorrected methods in terms of averages of actual loss based on a random loss function drawn from $\exp(N(0, 2^2))$ and averages of minus log probabilities of observing true responses.

5 Conclusions and future work

In this paper, I show how we can calibrate predictions based on a selected subset of features modeled by Bayesian Gaussian classification models. Specifically I have shown that we can efficiently compute the adjustment factor required to avoid feature selection bias — the probability of a feature is omitted by F-statistic, based on the precursors' work on computing the power function of ANOVA. I have used a simulated data set to show that after correcting for selection bias the predictive probabilities for future cases are well-calibrated. I have also tested the bias-correction method with a gene expression data and found that it does reduce the feature selection bias and yield better predictions.

The method presented here can be applied to a wide variety of practical problems, however it can be further improved. First, a more reasonable prior for $\mu_j^{(g)}$ given ν_j in high-dimensional problems may be some distribution with heavier tail than Gaussian distribution, since we believe in such problems most of the features are useless in predicting the response, while a few of them may be very useful. Student's t distributions with small degree of freedom or Cauchy distributions are attractive. The difficulty of using these distributions is that we cannot directly draw samples of $\mu^{(g)}$ given others as we do here, we therefore may have to use some Markov chain sampling methods, which may hurt the sampling efficiency, such as getting trapped in local modes more easily. However, it is noticed that using these distributions doesn't increase the difficulty of computing the adjustment factor, for which one simply turns to draw samples of $\boldsymbol{Z}$ from t or Cauchy distributions in approximating (34) with Monte Carlo method. Second, our method relies heavily on the assumption of independence amongst features given class, which however may not be realistic for real problems. We have proposed a linear transformation method to remedy this. However, it may be too simple for many real problems, and better methods of extracting independent factors underlying high-dimensional features are needed. Finally, adjusting feature selection bias by modeling features explicitly may be unnecessarily expensive. Another approach is to directly modify the posterior distribution of some hyperparameters controlling the overall relationship between features and response, such as τ^μ here, in light of number of features omitted and threshold used. This modification may need to use some unknown parameters which could be determined by minimizing the difference between the expected error rate and actual error rate.

References

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. (1982), "The well-calibrated Bayesian," *Journal of the American Statistical Association*, 77, 605–610.

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.

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.

Forman, G. (2007), "Feature Selection for Text Classification," in *Computational Methods of Feature Selection*, eds. Liu, H. and Motoda, H., Chapman and Hall/CRC, pp. 255–274.

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

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

Khan, J., Wei, J., Ringnér, M., Saal, L., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C., Peterson, C., et al. (2001), "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," *Nature Medicine*, 7, 673–679.

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

Lecocke, M. L. and Hess, K. (2004), "An Empirical Study of Optimism and Selection Bias in Binary Classification with Microarray Data," UT MD Anderson Cancer Center Department of Biostatistics Working Paper Series.

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.

Liu, H. and Motoda, H. (2007), *Computational Methods of Feature Selection*, Chapman and Hall/CRC.

Liu, J. S. (2001), *Monte Carlo Strategies in Scientific Computing*, Springer-Verlag.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953), "Equation of State Calculations by Fast Computing Machines," *The Journal of Chemical Physics*, 21, 1087–1092.

Neal, R. M. (1993), "Probablistic 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," in *Artificial Intelligence in Medicine*, Springer, pp. 468–472.

Schervish, M. J. (1995), *Theory of Statistics*, Springer Series in Statistics, Springer.

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

Tadjudin, S. and Landgrebe, D. (1998), "Classification of High Dimensional Data with Limited Training samples," Tech. rep., School of Electrical Engineering, Purdue University.

Tibshirani, R. (1996), "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, 58, 267–288.

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.

Wu, B., Abbott, T., Fishman, D., McMurray, W., Mor, G., Stone, K., Ward, D., Williams, K., and Zhao, H. (2003), "Comparison of statistical methods for classification of ovarian cancer using mass spectrometry data," *Bioinformatics*, 19, 1636–1643.

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

Acknowledgement

This research was supported by new faculty start-up grant and NSERC President's Award, both from the University of Saskatchewan.