High-dimensional Feature Selection Using Hierarchical Bayesian Logistic Regression with Heavy-tailed Priors

Longhai Li* and Weixin Yao†

22 May 2013

Abstract

The problem of selecting the most useful features from a great many (eg, thousands) of candidates arises in many areas of modern sciences. An interesting problem from genomic research is that, from thousands of genes that are active (expressed) in certain tissue cells, we want to find the genes that can be used to separate tissues of different classes (eg. cancer and normal). In this paper, we report our empirical experiences of using Bayesian logistic regression based on heavy-tailed priors with moderately small degree freedom (such as 1) and very small scale, and using Hamiltonian Monte Carlo to do computation. We discuss the advantages and limitations of this method, and illustrate the difficulties that remain unsolved. The method is applied to a real microarray data set related to prostate cancer. The method identifies only 3 non-redundant genes out of 6033 candidates but achieves better leave-one-out cross-validated prediction accuracy than many other methods.

Key phrases: high-dimensional feature selection, heavy-tailed t prior, MCMC, Hamiltonian Monte Carlo, Gibbs sampling, fully Bayesian.

*Correspondance author, Assistant Professor, Department of Mathematics and Statistics, University of Saskatchewan, Saskatoon, SK, S7N5E6, CANADA. e-mail: longhai@math.usask.ca.

†Assistant Professor, Department of Statistics, Kansas State University, Manhattan, Kansas 66506, U.S.A. e-mail: wxyao@ksu.edu.

1 Introduction

Today, high-throughput biotechnologies (such as microarray and sequence methods) can easily measure expression levels of thousands of genes. An interesting problem in genomic research is to identify the genes whose expression levels are relevant to a certain complex disease, such as cancer or diabetes. Once such genes are found (and verified by biologists), they can be used for prognosis or diagnosis of the disease for future tissues. For this purpose, researchers collect some "training" tissues, for which their true classes (such as cancer and normal) are known. Typically the number of training tissues is very small, such as *tens*, but the number of candidate genes is large, such as *thousands*. Identifying the most relevant genes for a disease from thousands of candidates is statistically challenging — we are looking for a few "needles" (useful features) from a huge "haystack" (irrelevant features).

In practice, it is very common to use univariate methods that measure the strength of relationship between each gene and the class label, such as t or F tests, or model-based inference methods with independence assumption for genes within classes, such as DLDA (Dudoit et al., 2002), and PAM (Tibshirani et al., 2002). However, univariate methods ignore the correlations between genes, which are prevalent in gene expression data due to gene co-regulation, see Ma et al. (2007), Clarke et al. (2008) and Tolosi and Lengauer (2011) for real examples. The consequence is that many redundant differentiated genes are included, meanwhile, useful but weakly differentiated genes may be omitted.

Feature selection methods by directly fitting classification models, such as logistic regression models, can take correlations among genes into account. However, when the number of observations isn't much larger than the number of features, maximizing likelihood of a classification model is certain to overfit the data, with noise rather than signal captured. Penalized likelihood methods based on L_1 penalty (corresponding to Laplace prior for Bayesian methods), may not distinguish the "needles" and "hay" well. In the past few years, penalized likelihood methods that use penalties with heavier tails than L_1, sometimes called hyper-

lasso penalties (Griffin and Brown, 2012), have been shown to be a promising alternative to L_1 penalties in high-dimensional regression probelms, see the articles by Gelman et al. (2008); Carvalho et al. (2010); Polson and Scott (2010); Griffin and Brown (2012), among some others. Meanwhile, fully Bayesian methods for high-dimensional regression and classification problems that used priors with tails similar to Laplace, have been extensively studied, see for example Bae and Mallick (2004); Park and Casella (2008); Yi and Xu (2008); Li et al. (2011). However, fully Bayesian methods that use heavy-tailed priors seemingly have not been investigated much in current literature for the high-dimensional classification and regression. Therefore, we developed a fully Bayesian high-dimensional feature selection method for classification problems that use heavy-tailed priors with small scales and sophisticated Hamiltonian Monte Carlo sampling, called by us hierarchical Bayesian polychotomous logistic regression (HBPLR). In this article, we reported our results of using this method with extensive simulation studies and real data demonstration. The goal of the article is to discuss the advantages, difficulties and specialty of using heavy-tailed priors in fully Bayesian approaches. Particularly, we've found that the advantages of HBPLR include: 1) it can shrink small signals strongly towards 0 (due to small scale), but leave large signals unpunished (due to heavy tails); 2) it can automatically separate a group of many redundant correlated features into different posterior modes, and also makes selection among a group of correlated features automatically; 3) the fitting results are stable for a wide range of small scales for heavy-tailed prior. We hope that our empirical results would benefit other researchers that are interested in this field.

This article will be structured as follows. In Section 2, we first discuss some properties of heavy-tailed priors by comparing to Laplace and Gaussian using geometric illustration. In Section 3, we describe HBPLR in technical details. In Section 4 we use simulated data sets to investigate and test our method. In Section 5, we report the analysis results by applying this method to a real microarray data set with $p = 6033$ related to prostate cancer. The article is concluded in Section 6 with discussions.

2 Simple Demonstration of Heavy-tailed Priors

We first consider the simple logistic regression model for binary class label for explaining why we choose heavy-tailed priors and MCMC for doing computation. Suppose we have collected data of features and responses (class labels) on n training cases. For a case indexed by i, we denote its class label by y_i, which can take integers 1 and 2, and denote p features associated with it by a row vector $\boldsymbol{x}_{i,1:p}$. The logistic regression model for the data is:

$$P(y_i = k+1|\boldsymbol{x}_{i,1:p}, \boldsymbol{\beta}_{0:p}) = \frac{I(k=0) + I(k=1)\exp\left(\beta_0 + \boldsymbol{x}_{i,1:p}\boldsymbol{\beta}_{1:p}\right)}{1+\exp\left(\beta_0 + \boldsymbol{x}_{i,1:p}\boldsymbol{\beta}_{1:p}\right)}, \quad (1)$$

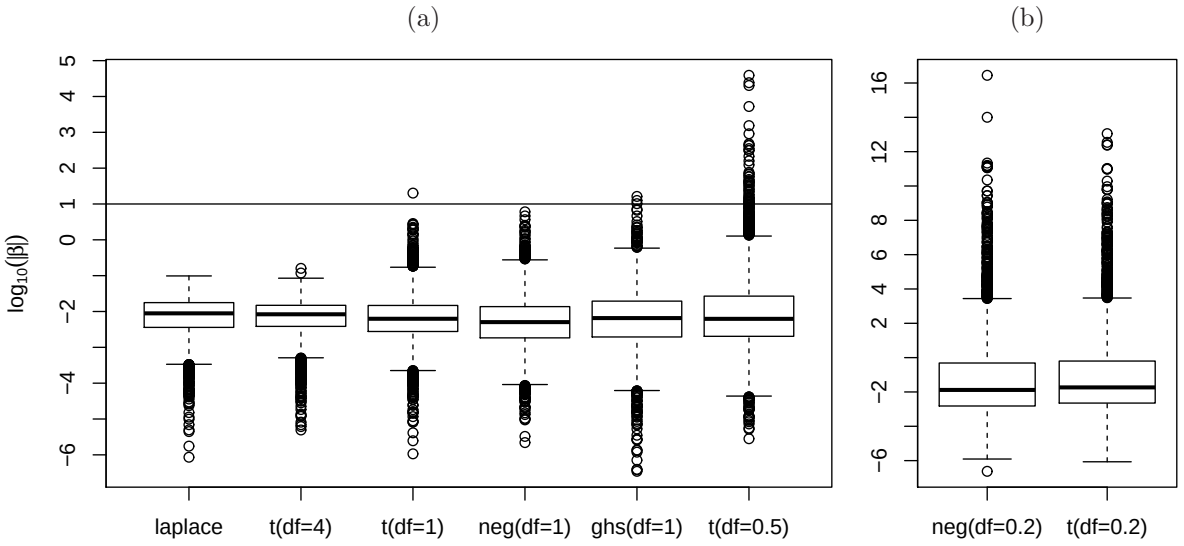
for $k = 0$ and 1, $i = 1, \ldots, n$, where $\boldsymbol{\beta}_{1:p}$ is a column vector of regression coefficients, and $I(\cdot)$ is indicator function, which is equal to 1 if the condition in bracket is true, 0 otherwise. We will assume that all the features are commensurable and we will select features by looking at the values of $\boldsymbol{\beta}_{1:p}$. We will consider how to infer $\boldsymbol{\beta}_{1:p}$ from the training data.

The most important part in HBPLR is the choice of moderately heavy-tailed priors for $\boldsymbol{\beta}_{1:p}$. The simplest choice is independent t distributions with a small degree freedom (shortened by **df** hereafter) such as 1, and a very small scale such as e^{-5}. t distribution is a member of scale-mixture-normal (SMN) family. A t distribution for β_j with df α and scale γ can be expressed with two-level conditional distributions: $\beta_j|\sigma_j^2 \sim N(0, \sigma_j^2), \sigma_j^2 \sim \text{IG}(\alpha/2, \alpha\gamma^2/2)$, where $\text{IG}(a, b)$ stands for Inverse-Gamma distribution, the distribution of the inverse of a Gamma random variable with shape parameter a and *rate* parameter b. In terms of random numbers generator, t is the distribution of the products of two independent random variables and a scale γ: $N(0,1) \times \sqrt{\text{IG}(\alpha/2, \alpha/2)} \times \gamma$. Similarly, Laplace is the distribution of $N(0,1) \times \sqrt{\exp(1)} \times \gamma$, where γ is a scale, and $\exp(1)$ is the standard exponential random variable. Note that a Laplace distribution parametrized by λ with PDF $(\lambda/2)e^{-\lambda|\beta_j|}$ has our scale $\gamma = \sqrt{2}/\lambda$. Recently, some other penalties with SMN interpretation, including Horseshoe by Carvalho et al. (2010), and Normal-Exp-Gamma (NEG) by Griffin and Brown (2012), are shown to be superior than LASSO in high-dimensional regression problems. In the original Horseshoe prior, positive half Cauchy prior is assigned to σ_j. Here we naturally

Table 1: 4 scale-mixture-normal distributions. α is denoted by "df" in Figure 1

Name	**Random Numbers Generator**	**Parameters used in Figure 1**
t	$N(0,1) \times \sqrt{\text{IG}(\alpha/2, \alpha/2)} \times \gamma$	four groups of $(\alpha, \log(\gamma))$ used: $(4, -4.5)/(1, -5)/(0.5, -5.5)/(0.2, -6)$
GHS	$N(0,1) \times \lvert N(0,1)\rvert \times \sqrt{\text{IG}(\alpha/2, \alpha/2)} \times \gamma$	$(\alpha, \log(\gamma)) = (1, -4.5)$
NEG	$N(0,1) \times \sqrt{\exp(1)} \times \sqrt{\text{IG}(\alpha/2, \alpha/2)} \times \gamma$	$(\alpha, \log(\gamma)) : (1, -5)/(0.2, -6)$
Laplace	$N(0,1) \times \sqrt{\exp(1)} \times \gamma$	$\log(\gamma) = -4$

Figure 1: Boxplots of 4000 $\log_{10}(|\beta_j|)$ generated from different SMN priors. The scales can be found from the last column of Table 1.



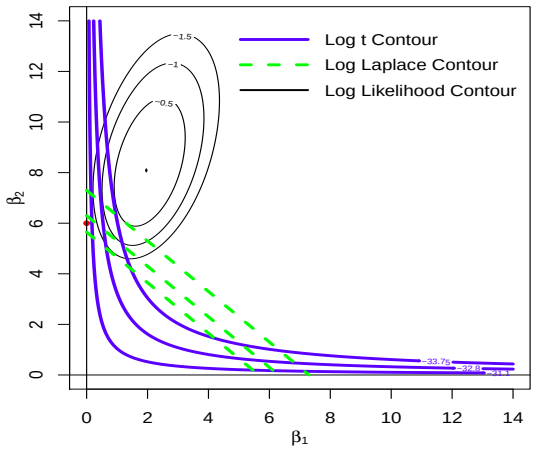
generalize half Cauchy to half t for uniformality of notations for all the priors considered in this article, and call the prior GHS. In Table 1, we describe them with their random numbers generators. The detailed descriptions of GHS and NEG are given in Section 3.1 [primarily by equations (5) and (6)].

We generated 4000 of random β_j from different SMN priors to look at their appropriateness in expressing our belief that a couple of "needles" are present in a huge "haystack". For this, we desire that most of $|\beta_j|$ generated from a prior are very small, but a few are very large for modelling the "needles". To look at their tails more easily, we set the scale for each distribution such that the medians of $\log_{10}(|\beta_j|)$ are around -2, with exact values listed in Table 1. Figure 1 shows the boxplots of $\log_{10}(|\beta_j|)$ for 8 SMN distributions. Note that, with other larger or smaller scale, the boxplots of $\log_{10}(|\beta_j|)$ only move up or down,

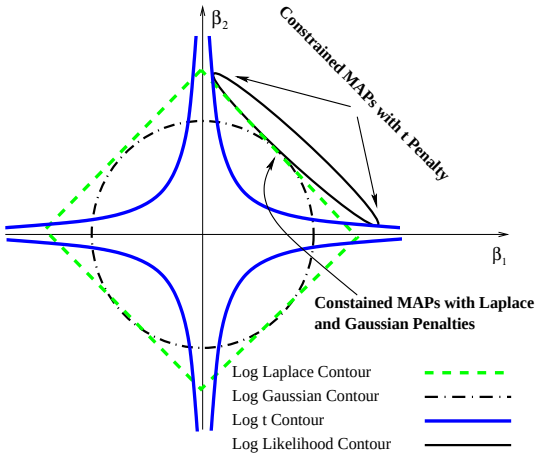
without shape change. From Figure 1a, we see that t, NEG, and GHS with small dfs, such as 1, have moderately heavy tails, allowing a few $|\beta_j|$ much larger than a great many of "hay", expressing well our belief. t with large df (such as 4) and Laplace have very short upward tails, with coefficients very close to each others. On the other hand, t and NEG with overly heavy tails with very small dfs, such as 0.2, are not good either — the upward tails of such priors become overly flat, allowing values from 10^{-6} to 10^{16}, which can't be true.

Figure 2: Geometric illustrations of the properties of t penalty in MAP inference.

(a) Distinguishing Needles from hay



(b) Dividing Redundant Correlated Features



The property of moderately heavy-tailed priors in better separating "needles" from "hay" can be explained also by looking at the "path" of constrained MAPs (maximizer of posterior) — the MAP with the log likelihood constrained to a particular value (ie on a contour). A constrained MAP can be found by shrinking the contour lines of log priors toward the origin until the two lines are tangent. Figure 2a shows three such constrained MAPs for t with df $= 0.5$ and $\gamma = e^{-10}$, and three for Laplace, based on a data set generated with true coefficients $\beta_1 = 0, \beta_2 = 6, \beta_0 = 0$. The (unconstrained) MAP can then be found from the "path" containing these constrained MAPs. Because the contour lines of log t prior indent into the origin, the path of constrained MAPs based on t prior is flatter (with respect to x-axis) than the path based on Laplace; starting from the MLEs, the path based on t prior goes to a point at which β_1 is very close to 0 (but not exact 0), and β_2 is close to its true value (6), whereas, the path based on Laplace goes to the origin. Therefore we see that t

prior can shrink small "hay" without much punishment to large "needles".

From looking at constrained MAPs, we also find that heavy-tailed penalties can *automatically* divide a group of correlated features into different posterior local modes. Figure 2b shows a conceptual illustration not based on a simulated data set. When two features are highly correlated, a contour line of log likelihood is negatively correlated as shown in Figure 2b. With t penalty, the constrained MAPs are at the two ends of the contour of log likelihood, each of which uses only one of them to explain the class label without under-estimating the importance of each of them. Therefore, the coefficients of highly correlated features are divided into different modes, each using only one of them, see Figure 11d for a demonstration with real data. When the predictive abilities of the correlated features are different, these local mode regions have different volumes under the posterior. That is, t prior can also make selection among a group of highly correlated features automatically. The selection within groups is necessary in high-dimensional problems in which often a large group of correlated features exist. By contrast, with Laplace and Gaussian penalties, the constrained MAPs are in the middle of the contour, favoring using all features with coefficients of smaller absolute values to explain the class label. When the group of correlated features is large, they may under-estimate the absolute values of all of them, and miss all of them, see a detailed discussion by Tolosi and Lengauer (2011).

Figure 2b is also helpful for seeing that the primary computational difficulty of using heavy-tailed priors in classification and regression problems is the existing of many local modes in the posterior distribution. An optimization algorithm can easily get trapped in *a* minor local mode arbitrarily depending on the initial values, so the algorithm becomes unstable and some sophisticated methods for choosing the initial values are required, see Griffin and Brown (2012). We therefore choose sophisticated MCMC methods, which can travel across all the modes in theory, and can travel across at least some of them in reality.

3 Methodology

We will now describe HBPLR in technical details. Throughout this article, we will denote matrices with **bold-faced** letters, with row indexes displayed in the first subscript, and column indexes in the second; we denote real-valued vectors with **bold-faced** letters too, but with only a set of indexes in subscript; the indexes of matrices and vectors are denoted by $i\!:\!j$ — integers from i to j, or a single integer for a row or column.

3.1 Hierarchical Bayesian Logistic Regression (HBPLR) Model

Suppose we have collected data of features and responses (class labels) on n training cases. For a case indexed by i, we denote its class label by y_i, which can take integers $1, \ldots, C$, and denote p features associated with it by a row vector $\boldsymbol{x}_{i,1:p}$. The hierarchical Bayesian polychotomous logistic regression model used by us is described as follows:

$$P(y_i = c|\boldsymbol{x}_{i,1:p}, \boldsymbol{\beta}_{0:p,1:C}) = \frac{\exp\left(\beta_{0,c} + \boldsymbol{x}_{i,1:p}\boldsymbol{\beta}_{1:p,c}\right)}{\sum_{c=1}^{C}\exp\left(\beta_{0,c} + \boldsymbol{x}_{i,1:p}\boldsymbol{\beta}_{1:p,c}\right)}, \text{for } c = 1, \ldots, C, \tag{2}$$

$$\boldsymbol{\beta}_{j,1:C}|\sigma_j^2 \sim N\ (0, \sigma_j^2),\ \text{for } j = 0, \ldots, p, \tag{3}$$

$$\boldsymbol{\sigma}_{1:p}^2 \sim \text{IG}\ (\alpha/2, w\alpha/2), \tag{4}$$

where $\boldsymbol{\beta}_{0:p,1:C}$ are regression coefficients, and other variables are called hyperparameters, which are introduced to define the prior for $\boldsymbol{\beta}_{1:p,1:C}$, and for convenience in MCMC sampling.

In this hierarchy, σ_j^2 indicates the importance of jth feature — the feature with larger σ_j^2 is more useful for predicting y, provided that all features are commensurable (which can be enforced by standardization). Note that we fix σ_0^2, not controlled by w, because we believe that the variability of intercepts is quite different from the variability of $\boldsymbol{\beta}_{j,1:C}$ for features. With $\boldsymbol{\sigma}_{1:p}^2$ marginalized with respect to IG prior (4), equations (3) and (4) assign $\boldsymbol{\beta}_{j,1:C}$ $(j > 0)$ with C-dimensional t prior with df α, and $I_C \times \gamma$ as its covariance, whose PDF can be found from Kotz and Nadarajah (2004). For $C = 2$, this bivariate t prior is equivalent to assigning independent t priors for p coefficients $\delta_j = \beta_{j,2} - \beta_{j,1}$, as explained in Section 3.2.

As alternatives of (4), other priors for $\boldsymbol{\sigma}_{1:p}^2$ have been proposed in the recent literature

for regression problems with the goal of shrinking $\boldsymbol{\sigma}^2_{1:p}$ with weak signals more towards 0. Carvalho et al. (2010) propose Horseshoe prior for coefficients by assigning half (positive) Cauchy distribution for $\boldsymbol{\sigma}_{1:p}$. For uninformity of notations, we describe half Cauchy distribution using half t with various degree freedoms for σ_j, inducing a prior for σ_j^2:

$$P_{ghs}(\sigma_j^2) = \frac{\Gamma((\alpha+1)/2)}{\Gamma(\alpha/2)\sqrt{\alpha\pi}\sqrt{w}} \left(\frac{1}{1+\sigma_j^2/(\alpha w)}\right)^{\frac{\alpha+1}{2}} \frac{1}{(\sigma_j^2)^{1/2}}, \quad \text{for } \sigma_j^2 > 0. \tag{5}$$

Griffin and Brown (2012) propose NEG prior for coefficients by assigning exp-gamma prior for $\sigma^2_{1:p}$: $\sigma_j^2|\psi_j \sim \exp(\frac{1}{\psi_j}), \psi_j \sim \text{IG}(\alpha/2, \alpha w/2)$, where ψ_j is the mean parameter of exp distribution. We can marginalize ψ_j, and obtain a closed-form PDF for σ_j^2:

$$P_{neg}(\sigma_j^2) = \frac{\kappa}{\lambda}\left(\frac{1}{1+\sigma_j^2/\lambda}\right)^{\alpha/2+1}, \tag{6}$$

where $\kappa = \alpha/2$ and $\lambda = \alpha w/2$ for notational simplicity. We will call (5) and (6) as **GHS** and **NEG** priors for σ_j^2, though the names are also used for the priors for coefficients. The PDFs of GHS and NEG priors for σ_j^2 don't converge to 0 as σ_j^2 goes to 0 (as IG prior does), therefore possibly regression coefficients are better shrunken towards 0 without punishing large signals. This property is indeed desired. Our numerical studies that follows confirm the improvement over IG prior, though it seems very little for logistic regression models, if small scale $\sqrt{w}$ is used. On the other hand, the sampling methods with adaptive rejection sampling (ARS, Gilks and Wild, 1992) for the posteriors of σ_j^2 given $\boldsymbol{\beta}_{j,1:C}$ based on GHS and NEG priors are substantially slower than the standard sampling method for the IG posterior based on IG prior, which increase the total Markov chain sampling time greatly, for example from 3 hours to 8 or 10 hours in one of our examples.

The value $\sqrt{w}$ (denoted by γ previously) is the scale of prior distribution for regression coefficients. We recommend to fix α and w at some reasonable values, for example $\alpha = 1$, $\log(w) = -10$. Most random numbers generated by t/GHS/NEG distributions with this setting are very small, but contain a few with magnitude 0-2, which can model a wide range of problems, and are recommended in practice. Our examples will show that the

results are insensitive to the choice of very small w. Especially, due to heavy-tails, we don't need to worry that very small values may over-shrink the signals. However, when the $\sqrt{w}$ is too large (such as greater than 0.1), the posterior is close to the likelihood function without any penalty, for which a Markov chain may travel very widely. It seemed to us that an easy method for avoiding choosing w was to treat w as a hyperparameter such that it will be adjusted with actual data information. However, when α is small, very little information about w can be captured from $\sigma^2_{1:p}$ (this is why Cauchy distribution is called weakly informative prior by Gelman et al. (2008)). The consequence is that Markov chain may stick to some region that overfit the data for a very long time with a large value w, before it pins down w. However, when α is large, such as 4, the distribution of $\sigma^2_{1:p}$ is controled much by w (ie, $\sigma^2_{1:p}$ are informative to w), therefore the inference results are sensitive to the choice of w, and we better treat w as a higher-level hyperparameter.

3.2 HBPLR Model with Identifiable and Symmetric Priors

An important issue in *polychotomous* logistic regression models is that the coefficients $\boldsymbol{\beta}_{j,1:C}$ is non-identifiable — if we add a constant to all $\boldsymbol{\beta}_{j,1:C}$, the conditional distribution of y given $\boldsymbol{x}$ in (2) is *exactly* the same. Therefore, the data can identify only the differences of $\boldsymbol{\beta}_{j,1:C}$ to a "baseline" class, say class 1, denoted by $\delta_{j,k} = \beta_{j,k+1} - \beta_{j,1}$, for $k = 1, \ldots, C-1$. To avoid the non-identifiability problem, the coefficient for a baseline class, say $\beta_{j,1}$, is often fixed at 0, and $\boldsymbol{\beta}_{j,2:C}$ is assigned with a prior as in (3). However, such a prior is *asymmetric* for all classes: the prior variance of $\beta_{j,c} - \beta_{j,c'}$ $(c, c' \neq 1)$ double that of $\beta_{j,c} - \beta_{j,1}$. This implication may not be justified for practical problems. In addition, the inferences, especially feature selection results, can vary greatly with the choice of baseline class. The common variance σ_j^2 for all $\boldsymbol{\beta}_{j,1:C}$ makes its posterior identifiable in theory. With fixed $\boldsymbol{\delta}_{j,1:(C-1)}$, for varying $\beta_{j,1}$, the normal prior [equation (3)] for $(\beta_{j,1}, \beta_{j,1} + \delta_{j,1}, \ldots, \beta_{j,1} + \delta_{j,C-1})$ is maximized at $\beta_{j,1} = -(1/C)\sum_{k=1}^{C-1} \delta_{j,k}$, which implies $\sum_{c=1}^{C} \beta_{j,c} = 0$. Therefore, the common variance prior, and so the posterior, identify the centralized $\boldsymbol{\beta}_{j,1:C}$. However, the apparent non-identifiability in likelihood function does harm the efficiency of Markov chain sampling. A

naive Gibbs sampler may stay for a long time where the absolute values of $\boldsymbol{\beta}_{j,1:C}$ are very large, but the differences $\boldsymbol{\delta}_{j,1:(C-1)}$ can be achieved by centralized $\boldsymbol{\beta}_{j,1:C}$ with smaller values.

We can use only the $\boldsymbol{\delta}_{j,1:(C-1)}$ as Markov chain state even when we use symmetric prior, simply by transforming $\boldsymbol{\beta}_{0:p,1:C}$ to a new set of parameters, and marginalizing the non-identifiable parameters. The transformed parameters from $\boldsymbol{\beta}_{0:p,1:C}$ are:

$$\boldsymbol{\delta}_{j,k} = \boldsymbol{\beta}_{j,k+1} - \boldsymbol{\beta}_{0:p,1}, \quad \text{for } k = 1, \ldots, \ K \equiv C-1, j = 0, \ldots, p \tag{7}$$

$$\boldsymbol{\beta}_{j,1} = \boldsymbol{\beta}_{j,1}, \text{for } j = 0, \ldots, p. \tag{8}$$

We will exactly transfer the symmetric prior for $\boldsymbol{\beta}_{j,1:C}$ to $\boldsymbol{\delta}_{j,1:K}$. Conditional on σ_j^2, the transformed parameters $(\beta_{j,1}, \boldsymbol{\delta}_{j,1:K})$ are distributed with a joint multivariate Gaussian distribution. The conditional distribution of y given $\boldsymbol{x}$ with $\boldsymbol{\delta}_{0:p,1:K}$ is:

$$P(y_i = k+1 | \boldsymbol{x}_{i,1:p}, \boldsymbol{\delta}_{0:p,1:K}) = \frac{I(k=0) + I(k>0)\exp\left(\delta_{0k} + \boldsymbol{x}_{i,1:p}\boldsymbol{\delta}_{1:p,k}\right)}{1 + \sum_{k=1}^{K}\exp\left(\delta_{0k} + \boldsymbol{x}_{i,1:p}\boldsymbol{\delta}_{1:p,k}\right)}, \tag{9}$$

for $k = 0, \ldots, K$, and $i = 1, \ldots, n$, where $I(\cdot)$ is an indicator function, equal to 1 if the condition in bracket is true, 0 otherwise. Since $\beta_{j,1}$ is not used in (9), therefore doesn't appear in the likelihood function of new parameters, we can integrate it out from the transferred prior directly conditional on σ_j^2, obtaining the marginal prior for $\boldsymbol{\delta}_{j,1:K}$ given σ_j^2:

$$\boldsymbol{\delta}_{j,1:K} | \sigma_j^2 \sim N_K(\mathbf{0}, (I_K + J_K)\sigma_j^2), \tag{10}$$

where I_K is a $K \times K$ identity matrix and J_K is a $K \times K$ matrix with all elements 1. For $C = 2$, (10) is just a univariate normal for $\delta_{j,1}$ with variance $2\sigma_j^2$. In summary, **HBPLR** model parameterized by $\boldsymbol{\delta}_{j,1:K}$ is now a hierarchy defined by (9), (10) [replacing (2) and (3)], and one of priors (4), (5) and (6) for $\boldsymbol{\sigma}_{1:p}^2$.

From (10), we see that what's different from the asymmetric prior is that the prior for $\delta_{j,1:K}$ are *correlated* for maintaining symmetry. The correlations among $\boldsymbol{\delta}_{j,1:K}$ cause no difficulty for Markov chain sampling, because we can easily evaluate the PDF of (10):

$$P(\boldsymbol{\delta}_{j,1:K} | \sigma_j^2) = (2\pi\sigma_j^2)^{-K/2} \exp\left(-\frac{V(\boldsymbol{\delta}_{j,1:K})}{2\sigma_j^2}\right) \times |I_K + J_K|^{-1/2}, \text{where,} \tag{11}$$

$$V(\boldsymbol{\delta}_{j,1:K}) \;=\; \sum_{k=1}^{K} \delta_{jk}^2 - \left(\sum_{k=1}^{K} \delta_{jk}\right)^2 \Big/ C. \tag{12}$$

We see that $V(\boldsymbol{\delta}_{j,1:K})$ is the sum of variations of $(0, \delta_{j,1}, \ldots, \delta_{j,K})$ from its mean, which is the same as the sum of variations of $\boldsymbol{\beta}_{j,1:C}$. $V(\boldsymbol{\delta}_{j,1:K})$ therefore measures how useful the jth feature is for predicting the class label, if the features are normalized to have standard deviation 1. For selecting features, it is more straightforward to look at **SDB** (standard deviation of $\beta_{j,1:C}$):

$$\mathrm{SDB}(\boldsymbol{\delta}_{j,1:K}) = \sqrt{V(\boldsymbol{\delta}_{j,1:K})/C}. \tag{13}$$

Note that, if $C = 2$ ($K = 1$), $V(\delta_{j,1}) = \delta_{j,1}^2/2$, and $\mathrm{SDB}(\boldsymbol{\delta}_{j,1}) = |\delta_{j,1}/2|$.

3.3 Gibbs Sampling Procedure

We use Gibbs sampling procedure to sample the full posterior distribution of HBPLR model. The full posterior distribution is written as:

$$P(\boldsymbol{\delta}_{0:p,1:K}, \boldsymbol{\sigma}_{1:p}^2 | \boldsymbol{D}) \propto L(\boldsymbol{\delta}_{0:p,1:K}) \times P(\boldsymbol{\delta}_{0:p,1:K} | \boldsymbol{\sigma}_{0:p}^2) \times P\left(\boldsymbol{\sigma}_{1:p}^2 \,|\, \alpha/2, \alpha\, w/2\right), \tag{14}$$

where, $\boldsymbol{D}$ represents the data $y_i, \boldsymbol{x}_{i,1:p}$ for $i = 1, \ldots, p$ and other fixed values in HBPLR models — α, σ_0^2 and L is the likelihood function: $L(\boldsymbol{\delta}_{0:p,1:K}) = \prod_{i=1}^{n} P(y_i | \boldsymbol{x}_{i,1:p}, \boldsymbol{\delta}_{0:p,1:K})$. The last two parts are the PDFs of priors specified by (10), and one of priors (4), (5), and (6). We sample the full posterior in (14) by sampling the conditional distribution of $\boldsymbol{\sigma}_{1:p}^2$ and $\boldsymbol{\delta}_{0:p,1:K}$ given each other alternately for a number of iterations. If IG prior (4) is used, the Gibbs sampling procedure is the alternate of the following two steps:

Step 1: Given $\boldsymbol{\sigma}_{1:p}^2$ fixed, update $\boldsymbol{\delta}_{0:p,1:K}$ jointly with a Hamiltonian Monte Carlo transformation that leaves invariant the following distribution:

$$P(\boldsymbol{\delta}_{0:p,1:K} | \boldsymbol{\sigma}_{0:p}^2, \boldsymbol{D}) \propto L(\boldsymbol{\delta}_{0:p,1:K}) \times P(\boldsymbol{\delta}_{0:p,1:K} | \boldsymbol{\sigma}_{0:p}^2). \tag{15}$$

Step 2: Given value of $\boldsymbol{\delta}_{1:p,1:K}$ from Step 1, update $\boldsymbol{\sigma}_{1:p}^2$ by sampling from

$$\sigma_j^2 | \boldsymbol{\delta}_{1:p,1:K} \;\sim\; \mathsf{IG}\left(\sigma_j^2 \,\middle|\, \frac{\alpha + K}{2}, \frac{\alpha w + V(\boldsymbol{\delta}_{j,1:K})}{2}\right), \tag{16}$$

The sampling for (16) in **Step 2** is straightforward. When Horseshoe and NEG priors are used, the sampling method for **Step 1** can be the same, but we have to use the posterior of σ_j^2 given $\boldsymbol{\delta}_{j,1:K}$ differently in **Step 2**. When we use GHS prior (5) for σ_j^2, the conditional posterior of σ_j^2 given $\boldsymbol{\delta}_{j:1:K}$ [in replace of equation (16)] is:

$$P_{ghs}(\sigma_j^2|\boldsymbol{\delta}_{j,1:K}) \propto \frac{1}{(\sigma_j^2)^{K/2}} \exp\left(-\frac{V(\boldsymbol{\delta}_{j,1:K})}{2\sigma_j^2}\right) \times \left(\frac{1}{1+\sigma_j^2/(\alpha w)}\right)^{\frac{\alpha+1}{2}} \frac{1}{(\sigma_j^2)^{1/2}}. \tag{17}$$

The induced conditional distribution for $\xi_j = \log(\sigma_j^2)$ from the above distribution is log-concave, and can be sampled with ARS (Gilks and Wild, 1992). When we use NEG prior (6) for σ_j^2, the conditional posterior of σ_j^2 given $\boldsymbol{\delta}_{j,1:K}$ [in replace of equation (16)] is

$$P_{neg}(\sigma_j^2|\boldsymbol{\delta}_{j,1:K}) \propto \frac{1}{(\sigma_j^2)^{K/2}} \exp\left(-\frac{V(\boldsymbol{\delta}_{j,1:K})}{2\sigma_j^2}\right) \times \left(\frac{1}{1+\sigma_j^2/\lambda}\right)^{\alpha/2+1}. \tag{18}$$

The induced posterior of the log transformation $\xi_j = \log(\sigma_j^2)$ from the above distribution is log-concave, and can be sampled with ARS.

The key component in the above procedure is the use of Hamiltonian Monte Carlo (HMC) for updating high-dimensional $\boldsymbol{\delta}_{0:p,1:K}$. In Section A.2, we give a concise description of HMC. HMC can greatly suppress the random walk due to correlation (which is common in logistic regression posterior; see our real data examples) with a long leapfrog trajectory (Neal, 2010). The major problem of sampling from posteriors based on heavy-tailed priors is the existing of many local modes due to feature redundancy. Applying HMC in the above Gibbs sampling framework can travel across the modes well. When both σ_j^2 for two correlated features are fairly large, the joint conditional distribution of their coefficients given σ_j^2 in **Step 1** are highly correlated, probably close to their likelihood function as shown in Figure 2b. A fairly long HMC trajectory has much greater chance than ordinary MCMC methods to move from one end of the contour to the other end, by which the Markov chain can travel from one mode to another. For high-dimensional problems with very large p, such as thousands, this step is the bottleneck for the whole Markov chain sampling. This challenge can be relieved greatly by an important trick that we call "restricted Gibbs sampling" — only the coefficients with

σ_j^2 greater than a certain threshold are updated in **Step 1**. The details of this trick are given in Section A. A list of notations for the settings of HBPLR is given in Section B.

3.4 Ranking Features with Markov chain Samples

With posterior samples of $\boldsymbol{\delta}_{1:p,1:K}$, we recommend using *means* over iterations to estimate the coefficients, denoted by $\hat{\delta}_{j,1:K}$. We then compute $\text{SDB}(\hat{\boldsymbol{\delta}}_{j,1:K})$ using formula (13) to obtain an importance index for feature j. The features can then be ranked by $\text{SDB}(\hat{\boldsymbol{\delta}}_{j,1:K})$. As we have discussed in Section 2, the Markov chain sample pool is a mixture of subpools from different modes, each corresponding to a succinct feature subset. The mean over Markov chain is a summary of the importance of the feature, not an estimate of the true coefficient. Obviously, this method omits some useful features that appear with low frequency in Markov chain samples. In the context of high-dimensional problems, there is often a large number of such correlated features, therefore discriminating them according to their predictive ability is desired. The ranking by *means* can omit many of such correlated features with weaker predictive ability, as well as those totally useless, and therefore pin down a very small subset of highly relevant features. *Median* isn't recommended since it may miss useful features that appear in Markov chain iterations with frequencies even *slightly* smaller than 0.5.

After we use *means* to pin down a small number of features, there might be still multiple subpools associated with succinct feature subsets and in our Markov chain samples. The feature subsets can be found manually by plotting Markov chain traces of $V(\boldsymbol{\delta}_{j,1:K})$. A more sophisticated method for interpreting the fitting results is to use a clustering algorithm to divide the whole Markov chain samples into subpools, look at the subpools separately, and deliver a list of succinct feature subsets. Once we can split Markov chain iterations as this, it will then be better to use *median* to obtain an importance index as it can better shrink the coefficients of totally useless features towards 0 and correct for the skewness of the posterior.

4 Demonstrations with Synthetic Data Sets

In this section, we use simulated data sets to demonstrate the three important properties of HBPLR stated in Section 1, and compare methods using t (with various dfs), GHS and NEG priors, as well as the currently most popular method — LASSO.

4.1 Comparisons on a Data Set with 200 Features

We generated a date set of $n = 1100$ cases (of which 100 were used for fitting models, and the other 1000 were used to look at predictive performance), and $p = 200$ features from the following multivariate Gaussian model:

$$P(y_i = c) = \frac{1}{2}, \quad \text{for } c = 1, 2, \qquad \boldsymbol{x}'_{i,1:200} \mid y_i = c \sim N_{100}\left(\boldsymbol{\mu}'_{c,1:200}\,,\, A\,A' + I_{200}\right), \tag{19}$$

where, $\boldsymbol{\mu}_{1,1:200} = (0, \ldots, 0)$, $\boldsymbol{\mu}_{2,1:200} = (2, 0, \ldots, 0)$, $A = (a_{ij})$ with all diagonal elements equal to 1, and $a_{21} = 2$. To make the 200 features commensurable, we standardize them to have means 0 and sds 1. In this model, only the first feature is differential across two classes, and the 2nd is non-differential but correlated with the 1st. Therefore, only the first two features are useful for predicting response y, with **true coefficients** $\boldsymbol{\delta}_{0:2,1} = (0, 2.60, -1.22)$, and all other coefficients are 0. The relationship between y and $\boldsymbol{x}$ is simple, but the signals are placed among the other 198 useless features. Figure 3 shows the scatterplots of the 2nd and 3rd features to the 1st with shapes representing two classes.

Figure 3: Scatterplots of the 2nd and 3rd features against the 1st generated from (19).

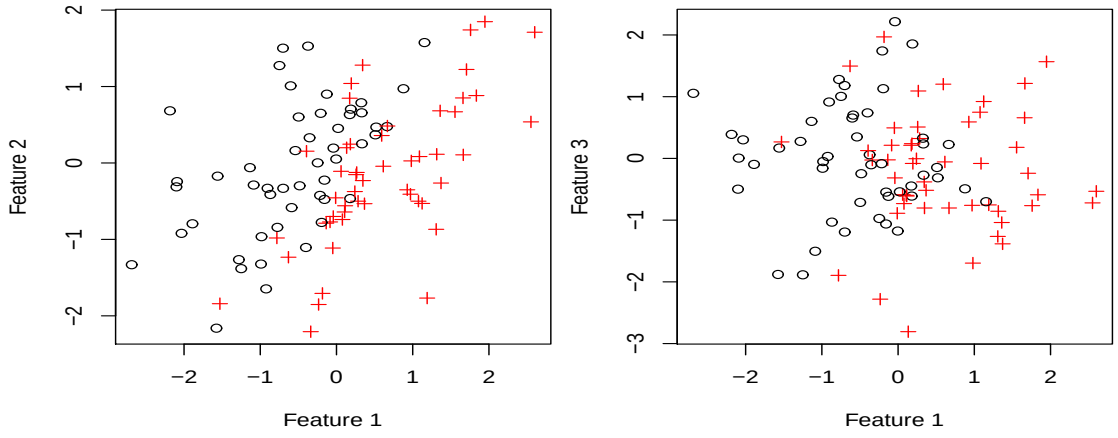
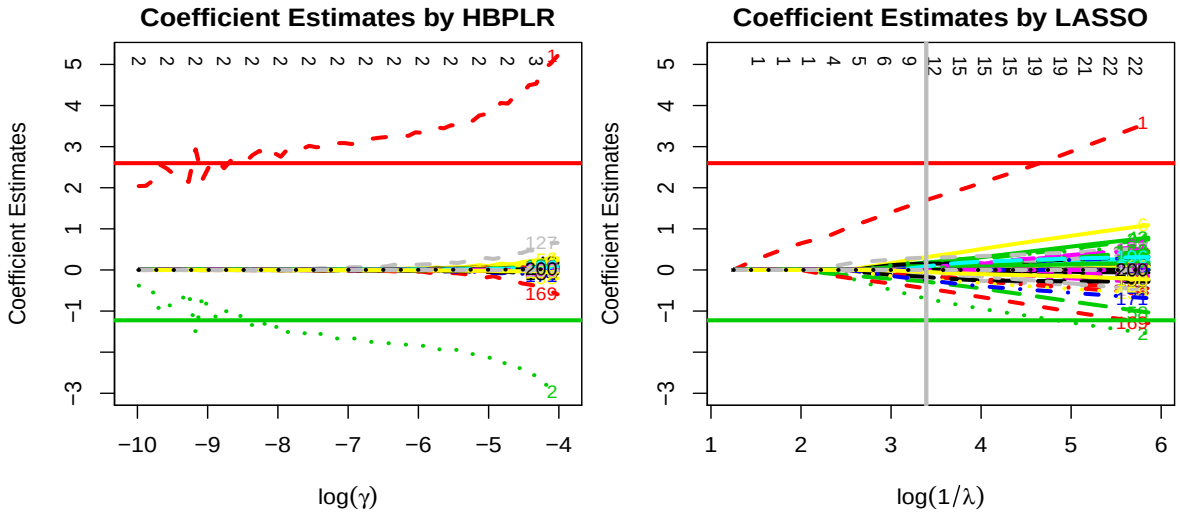


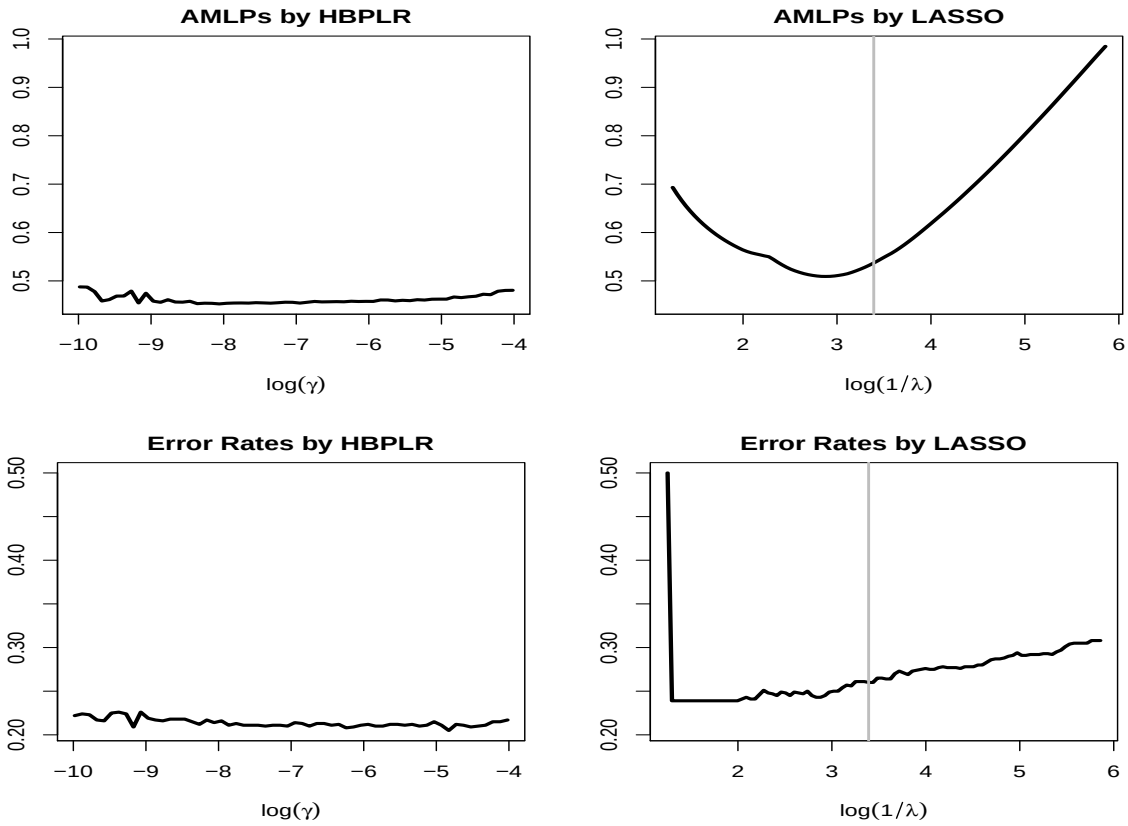
Figure 4: Solution paths of HBPLR (using t prior) and LASSO. The numbers on the top show the number of coefficients with absolute values not smaller than 0.1 times the maximum value. The vertical line in LASSO path shows the cross-validation choice of $\log(1/\lambda)$. The numbers beside paths on the right are feature indice.



We ran HBPLR using ***t* prior** (ie, IG prior for σ_j^2) with prior and MCMC settings $\alpha = 1, n_1 = 50K, \ell_1 = 5, n_2 = 100K, l_2 = 50, \epsilon = 0.3, \zeta = 0$, and 100 different $\log(w)$ spaced evenly from -24 to -8. *The meanings of these setting parameters are listed in Section B.* The Markov chains are unnecessarily long as the dimension is very small, but we used it since the computation is fast, taking about 30 mins for each chain. For each choice of scale, we estimated the coefficients using *means* in Markov chain samples. These results allow us to draw the solution paths (Figure 4) of all the coefficients against $\log(\gamma)$, and compare the path given by LASSO (using R package `glmnet` version 1.7). We can see that HBPLR gives much more distinctive estimates of the two non-zero coefficients from those of the other 198 hay than LASSO. LASSO cannot clearly make the second feature stand out from many other "hay". From comparing these paths, we also see that the estimates by HBPLR are very stable for the choices of γ in a very wide range. There is some upward bias in median estimates when γ is large, because the marginal posterior distributions of two coefficients for two correlated features are skewed to large absolute values (which is explained on page 25 with Figure 11). This bias, however, doesn't affect the predictive performance and feature selection. In addition, we do see that overly large scale $\gamma = \sqrt{w}$ for heavy-tailed t isn't

good in fully Bayesian approach, though it is not so problematic in penalized likelihood approach in which only the mode will be used, see Gelman et al. (2008). However, as explained before, the fitting results of penalized likelihood methods may be sensitive to the intial values of coefficients due to the multimodes of posterior when the data set has many correlated features.

Figure 5: AMLP and error rate paths of HBPLR (using t prior) and LASSO.



We also compare the predictive performance of HBPLR and LASSO in terms of AMLP — *the average minus log predictive probabilities at the true labels*, and error rates. The AMLP and error rate paths are shown in Figure 5. We see that HBPLR predicts better than LASSO. Most importantly, the predictive performance of HBPLR is very stable for the choice of γ. By contrast, LASSO is very sensitive to the choice of γ: if a small scale is used, LASSO omits more useless features, but also over-shrinks the really large coefficients, especially weaker x_2, therefore gives poor prediction; conversely, if a large scale is used, LASSO takes many "hay" into the model, which also downgrade the predictive performance. We see that the cross-

validated choice based on the small training data set didn't pick up the optimal $\log(1/\lambda)$, around 3; and even when this optimal choice is made, LASSO nearly omits x_2. HBPLR doesn't have this difficulty, and can identify the 2 "needles" from the 198 "hay".

4.2 Comparisons on Data Sets with 2000 features

In this section, we will look at how HBPLR works in more realistic scenarios. We generated 50 data sets ($n = 2100$ cases, 100 of which was used for training, the other 2000 was used for looking at predictive performance) as follows. The number of classes C is set to 3, and class labels are equally likely drawn from $1, 2$, and 3. Values of 10 features for each case were generated as follows:

$$\begin{aligned} x_1|y=c &= \mu_{c,1} + z_1 + 0.5\epsilon_1, \\ x_2|y=c &= \mu_{c,2} + 2z_1 + z_2 + 0.5\epsilon_2, \\ x_j|y=c &= \mu_{c,j} + z_3 + 0.5\epsilon_j, \text{ for } j = 3, \ldots, 10, \end{aligned} \qquad \text{where, } (\mu_{c,j})_{3\times 10} = \begin{pmatrix} 0 & 0 & 0 & \ldots & 0 \\ 2 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 2 & \ldots & 2 \end{pmatrix},$$

and, z_j and ϵ_j are independently generated from N(0,1). In addition to these 10 features, we also attached 1990 features simply drawn from N(0,1), which we will call absolute "hay". In this model, x_1 is differential, with different mean in class 2 from classes 1 and 3. x_2 is non-differential, but correlated with x_1, therefore is useful, as shown by Figure 3. $x_3 - x_{10}$ are all differential, with different means in class 3 from classes 1 and 2. However, $x_3 - x_{10}$, which have the same class means and are related to a common factor z_3, are highly correlated and redundant for predicting y; we will call this group of features as "correlated features".

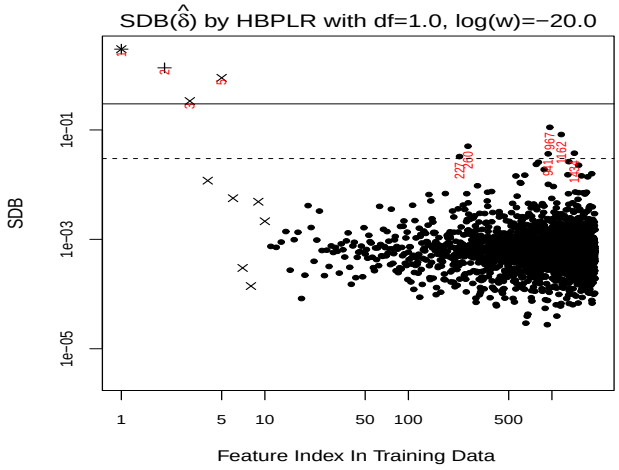
We ran HBPLR using t prior with 4 choices of α: 0.2, 0.5, 1, 4, 10. The setting for $\log(w)$ varies slightly for different α. When $\alpha = 1$, we chose two values of $\log(w)$: -20 and -10. When $\alpha = 4$ and 10, we chose to treat $\log(w)$ as a hyperparameter assigned with a normal prior with variance 100 (The reason is that, for large α, the results for feature selection and prediction are sensitive to the choices of scale and we therefore show the results with $\log(w)$ chosen automatically during MCMC simulation). The values of $\log(w)$ for $\alpha = 0.2/0.5$ are -40/-20 respectively. For setting MCMC, when $\alpha = 1$, we chose

$n_1 = 50K, \ell_1 = 10, n_2 = 500K, \ell_2 = 50, \epsilon = 0.3, \zeta = 0.05$ When α equals to 0.2/0.5/4/10 (other than 1), we set a larger $n_2 = 1M$ for longer chain and the same other settings as $\alpha = 1$. We ran HBPLR using GHS and NEG priors with settings $\alpha = 1$ and $\log(w) = -10$, and the same other settings for running HPBLR using t prior with $\alpha = 1$. We ran LASSO with λ chosen by cross-validated AMLP.

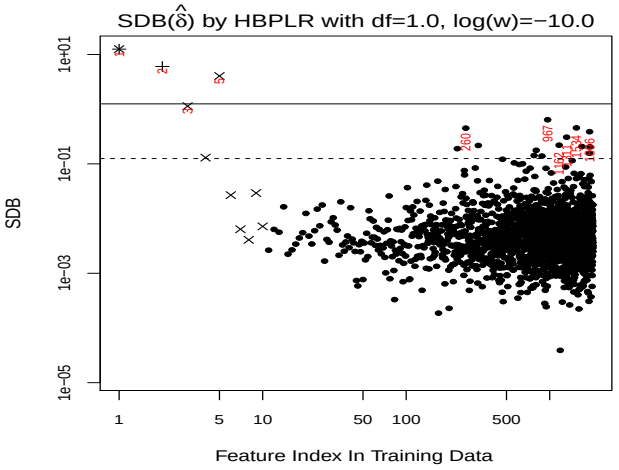
Figures 6 and 7 show the SDBs by different methods of all 2000 features for a data set, which is similar to most other data sets. From Figure 6, we see that HBPLR using t/GHS/NEG priors with $\alpha = 1$ and t prior with $\alpha = 0.5$ does the feature selection very well by looking at relative SDBs with threshold 0.1: 1) they can distinctively separate the absolute "hay" from other useful features, with SDBs mostly only equal to $10^{-2} - 10^{-4}$ times the maximum SDB; 2) they don't miss the 2nd feature which is useful but has weaker relevance; 3) they rank highly one feature (x_5) from the 8 correlated features, recognize another feature x_3 as useful too, and discriminate others; we believe x_5 is the most useful based on the information from the training data because it is consistently identified by all the runs with $\alpha = 1$ and 0.5; 4) the results of HBPLR using t prior with $\alpha = 1$ are stable for the two very different choices of $\log(w)$: -10 and -20. We believe that this applies to GHS and NEG priors too; 5) the results of HBPLR using t prior and GHS and NEG priors with the same α and $\log(w)$ are almost the same. By contrast, 1) LASSO and HBPLR with large α (such as 10) cannot separate the absolute "hay" distinctively from the few "needles", but LASSO is better than HBPLR with large α; 2) they both miss x_2 for this data set (and very often for other data sets), which we think is because they include many "hay" to overfit the data, therefore make x_2 harder to identify; 3) they tend to include many of the correlated features into their unique mode, without clear discrimination for importance; when the group of correlated features is large, they could be entirely missed because the coefficients in such a mode for the correlated features have small absolute values as shown by Figure 2b. HBPLR with very small $\alpha = 0.2$ (very heavy tails) can do feature selection well but their overly flat tails allow the "needles" to go to very large values (such as *thousands*, see Figure 6e and 6f),

Figure 6: SDBs given by HBPLR using t priors and LASSO on a synthetic data set with $p = 2000$ features. The shapes of points $*, +, \times, \bullet$ respectively stand for four groups of features: x_1, x_2, $x_3 - x_{10}$ and $x_{11} - x_{2000}$. The red numbers below points show the indice of top 10 features. Both x and y axises are in logarithm scale. The horizontal lines indicate the values equal to 0.1 and 0.01 times the maximum SDB.

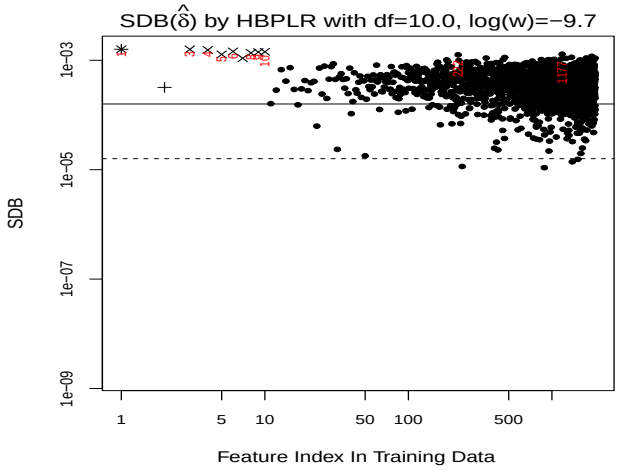
(a)



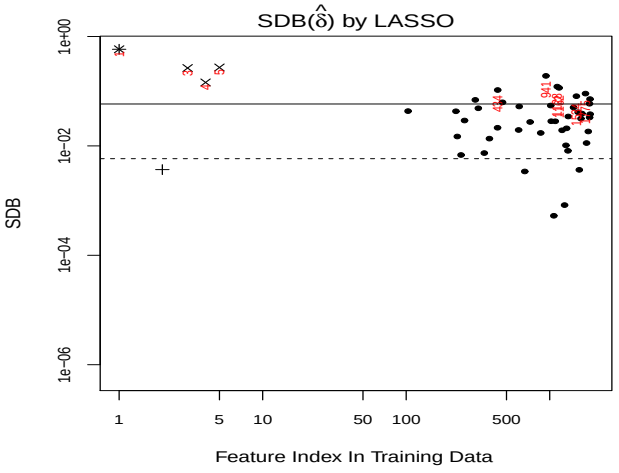
(b)



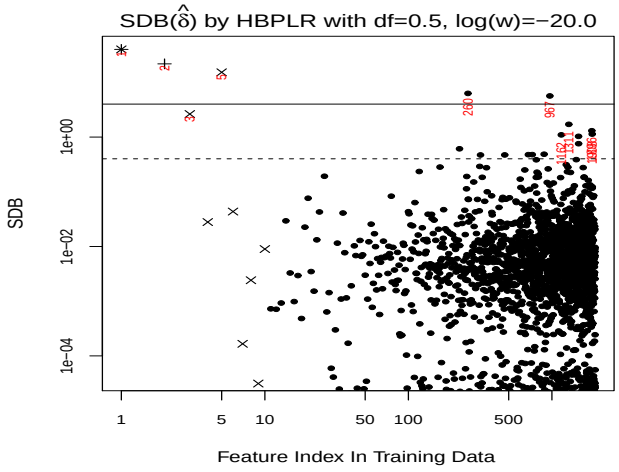
(c)



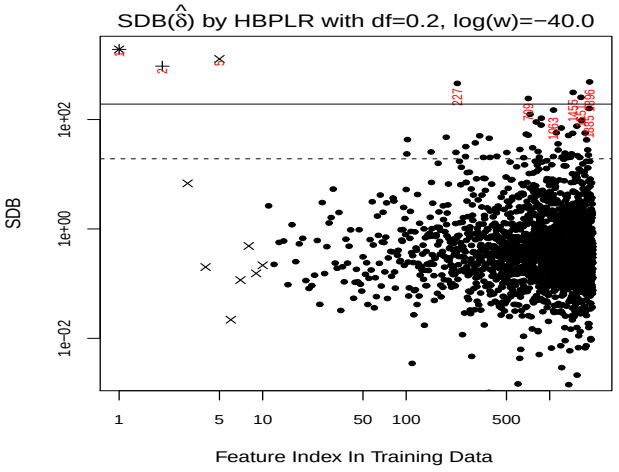
(d)



(e)



(f)



resulting in very poor prediction for a few cases, and infinite AMLP. Therefore priors with overly heavy tails are not good for logistic regression models. This is because the likelihood functions of logistic regression don't punish very large coefficients once the correct subset of features are selected, and very heavy tailed priors don't punish either.

Figure 7: SDBs given by HBPLR using GHS and NEG priors on the same data set used to make plots in Figure 6.

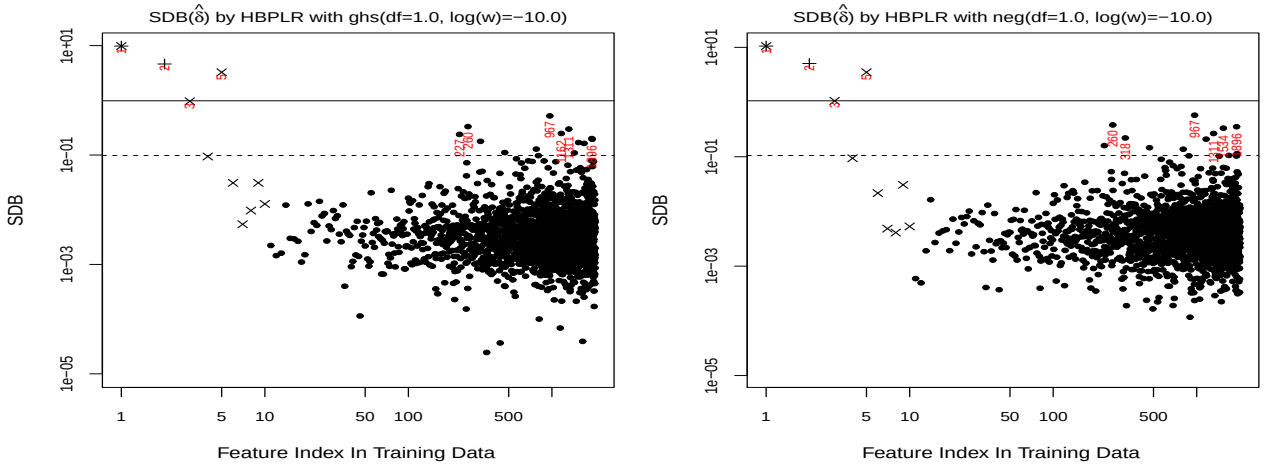
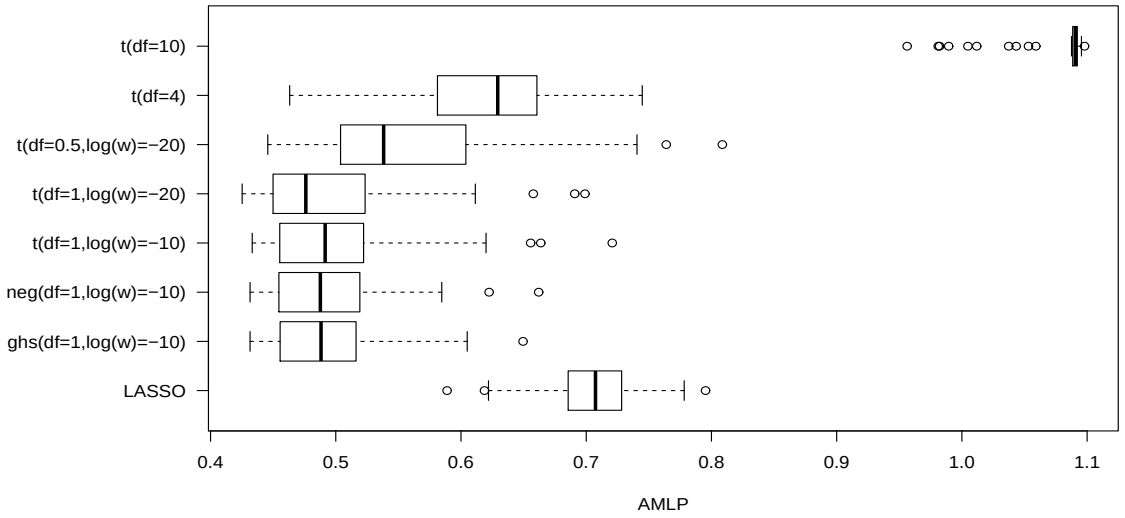


Table 2 shows a summary of numbers of selected features in 4 different groups by retaining features with SDBs not smaller than 0.1 times the maximum SDB (ie, by thresholding relative SDBs with 0.1). Figure 8 shows the boxplots of the 50 AMLPs of the prediction on 2000 test cases given different methods. We see that HBPLR (using t/GHS/NEG priors) with $\alpha = 1$ gives substantially better predictions and feature selections for most of the data sets than other choices of α and LASSO. LASSO and HBPLR with very large and very small α don't predict well. Note that the AMLPs of all runs with $\alpha = 0.2$ are *infinity*, so not shown in Figure 8. HBPLR with moderate $\alpha = 4$ does well in omitting the absolute "hay" from the "needles" with a less distinctive boundary because of the lighter tails. However, we see that it has large chance of missing the weaker feature x_2, therefore gives poorer predictions most of times, which is also due to over-shrinkage of the signals. Therefore, the choice of α is critical for HBPLR to work well for high-dimensional classification, and $\alpha = 1$ is recommended based on our investigations.

Figure 8: Boxplots of AMLPs on 2000 test cases by HBPLR with various priors and LASSO. "df" in the plot is α of priors for HBPLR.



From Table 2 and Figure 8, we see that the feature selection and prediction performance measures using GHS and NEG priors are slightly better on these 50 data sets than using t priors with the same value α and $\log(w)$. This improvement may be attributed to the nonzero values of PDFs of GHS and NEG at 0. But generally they are very similar, with differences without statistical significance. This indicates that t prior with small degree freedom and small scale performs almost as well as these two more complicated priors in logistic regression problems. On the other hand, the additional computation for using GHS and NEG compared to t is not negligible in high-dimensional problems based on our implementation. A Markov chain using t prior with $\alpha = 1$ as described previously took only about 3 hours, whereas a Markov chain using GHS or NEG priors took about 8 hours. The time difference is merely due to the use of the ARS sampling for posteriors [equations (17) and (18)] of σ_j^2 given coefficients rather than standard sampling for IG posterior [equation (16)]. The time difference will be larger as p is larger. However, this time difference is possibly eliminated by using better methods than ARS for sampling (17) and (18).

5 Application to Prostate Microarray Data

We applied HBPLR to a real microarray gene expression data that is related to prostate cancer, which has expression profiles of 6033 genes from 50 normal and 52 cancerous tis-

Table 2: Means of numbers of retained features by thresholding relative SDBs with 0.1 in different groups in 50 data sets. Numbers in () show the sds of the 50 numbers.

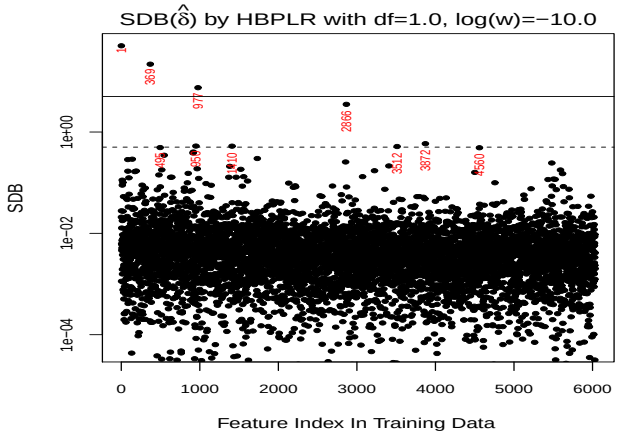
	Groups of Features			
Methods	x_1	x_2	$x_3 - x_{10}$	$x_{11} - x_{2000}$
HBPLR with $t(\alpha$=10)	0.96	0.66	7.42 (1.86)	1354 (580)
HBPLR with $t(\alpha$=4)	1	0.36	1.26 (0.53)	0.00 (0.00)
HBPLR with $t(\alpha$=0.2,$\log(w) = -40)$	1	0.72	1.36 (0.60)	5.74 (3.12)
HBPLR with $t(\alpha$=0.5,$\log(w) = -20)$	1	0.98	1.16 (0.37)	1.14 (0.97)
HBPLR with $t(\alpha$=1,$\log(w) = -20)$	1	0.94	1.14 (0.35)	0.16 (0.37)
HBPLR with $t(\alpha$=1,$\log(w) = -10)$	1	0.96	1.10 (0.30)	0.32 (0.55)
HBPLR with GHS $(\alpha$=1,$\log(w) = -10)$	1	1.00	1.14 (0.35)	0.30 (0.51)
HBPLR with NEG $(\alpha$=1,$\log(w) = -10)$	1	1.00	1.06 (0.24)	0.28 (0.50)
LASSO	1	0.34	2.72 (1.18)	6.92 (4.97)

sues. This data set was originally reported by Singh et al. (2002). We analyzed a data set downloaded from the website http://stat.ethz.ch/~dettling/bagboost.html for Dettling (2004), which contains more descriptions about this data set. For better looking at our results, we re-ordered the features by F-statistic on the whole data set, therefore the indice of the genes discussed below are also the ranks of features according to F-statistic. We ran HBPLR using t/GHS/NEG priors and LASSO with λ chosen with cross-validation, in leave-one-out cross-validation (LOOCV) fashion for comparing predictive performance. Before fitting with HBPLR, we always standardized the features with only training data (LASSO does such standardization too). We ran HBPLR with settings $\alpha = 1, \log(w) = -10, n_1 = 100K, \ell_1 = 10, n_2 = 1M, \ell_2 = 50, \epsilon = 0.3, \zeta = 0.05$ with all 6033 genes. Each Markov chain took about 10 hours if t prior was used, and about 33 hours if GHS/NEG priors were used.

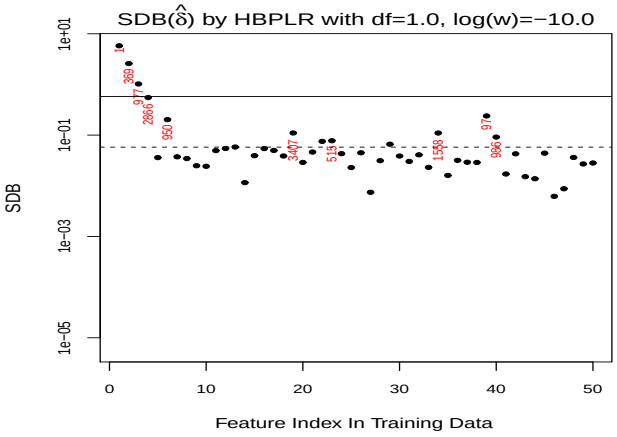
Figure 9 shows the SDBs of HBPLR and LASSO for the fold with the 2nd case left out, and P-values given by F-statistic on the whole data set. Figure 9b shows the results of rerunning HBPLR using t prior on only top 50 genes selected using the run with all 6033 genes. From these plots, we see that HBPLR using t/GHS/NEG priors perform very similarly and stably, and all distinctively select fewer than 10 genes by thresholding relative SDBs with 0.01. Comparing to F-statistic, we see that only the most differentiated gene 1

Figure 9: Gene selection results on Prostate data by different methods. The red numbers under points show the original indice of top 10 ranked genes. The horizontal lines indicate the values equal to 0.1 and 0.01 times the maximum SDB or P-value. (a) shows results using t priors, and (b) shows results from rerunning with top 50 genes selected from (a).

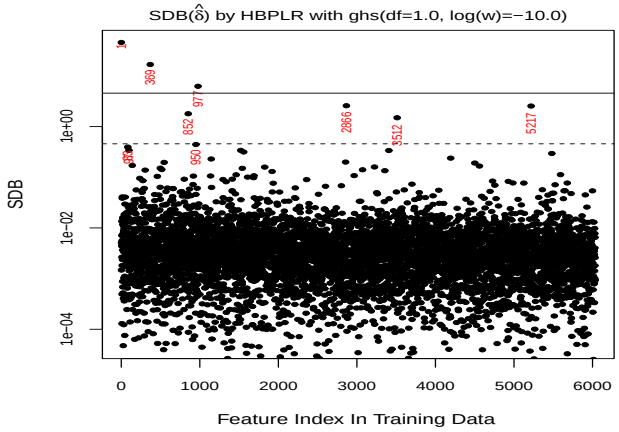
(a)



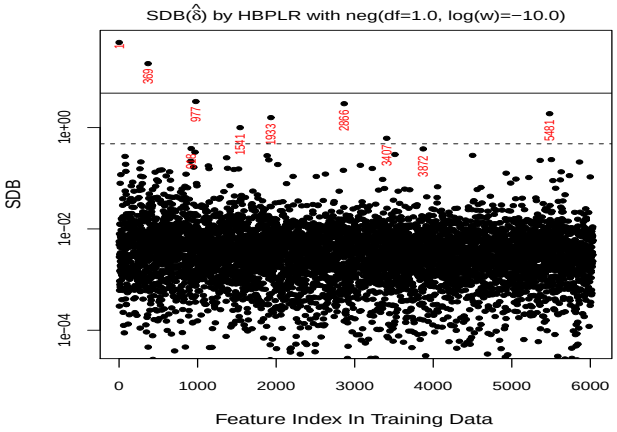
(b)



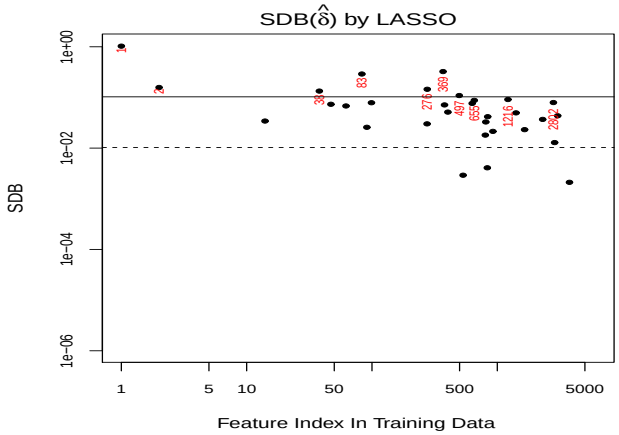
(c)



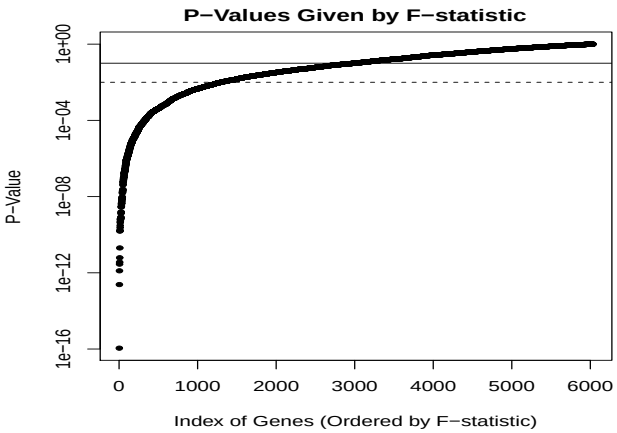
(d)



(e)



(f)



is ranked as top 1 by HBPLR, and thousands of genes with very small P-values are omitted by HPBLR methods; at the same time, other top ranked genes by HBPLR indeed have low ranks (such as 369, 977, 2899) by F-statistic. Comparing to LASSO, we see that 1) many SDBs (other than those 0) by LASSO are narrow around the value equal to 0.1 times the maximum SDB; 2) the maximum SDB value, around 1, by LASSO is much smaller than the maximum SDB value, around 10, by HBPLR, because LASSO over-shrinks the coefficients; 3) LASSO omits gene 977, which is ranked the 3rd by HBPLR, given 0 coefficient by LASSO, and is probably useful after a check with cross-validation, as shown at the end of this section.

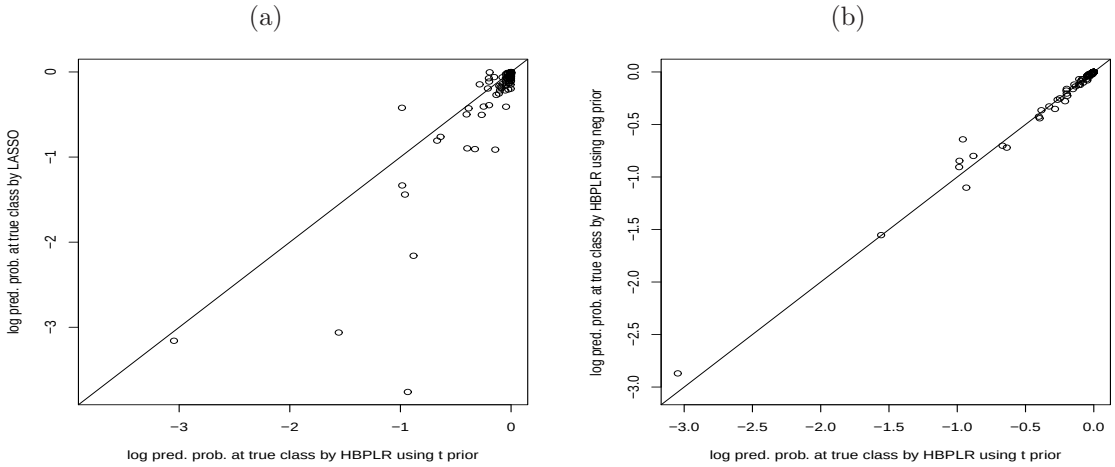
LOOCV predictive performances measured by AMLP and error rate are shown in Table 3, which also includes the results of many other methods reported by Dettling (2004). Figure 10 shows the scatterplots of log predictive probabilities at true class labels by HBPLR using t and NEG priors and LASSO. HBPLR is substantially better than many other methods, especially LASSO. The difference in AMLPs between HBPLR (using t prior) and LASSO is statistically significant, with p-value 0.00046 by paired one-sided t test. The performances of HBPLR using t/GHS/NEG priors are very similar without statistical significant difference, as shown by Figure 10b. Finally, we point out that the predictive performances of HBPLR can be improved further if we rerun them with fewer top genes selected with the runs with all genes due to improved MCMC sampling.

Table 3: Comparisons of LOOCV predictive performances of HBPLR and others. HBt, HBghs and HBneg are HBPLR using t, GHS and NEG priors respectively.

Methods	HBt	HBghs	HBneg	LASSO	Bagboost	PAM	DLDA	SVM	RanFor	kNN
AMLP	.156	.158	.152	.274	-	-	-	-	-	-
ER (%)	6.86	7.84	7.84	10.8	7.53	16.5	14.2	7.88	9.00	10.59

We have looked at the expression values of the top genes (such as 1, 369, 977, and 2866) and the MCMC samples of their coefficients. Figure 11 shows some results with the Markov chain by running HBPLR using t priors on the data set with the 2nd case left out and only top 50 genes that were selected from a previous run using all genes (the SDB values of these two runs are shown in Figure 9a and 9b respectively). First, from these plots and others

Figure 10: Comparisons of log predictive probabilities at true class labels.

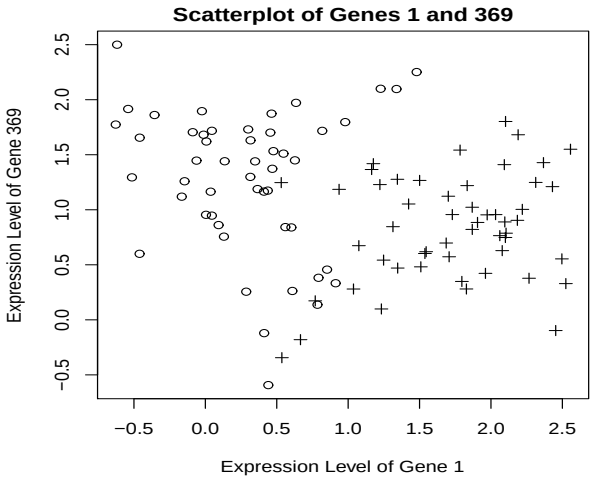


not shown, we see that genes 369, 977, and 2866 are weakly differentiated across two classes, but they are useful because they are correlated with the most differentiated gene 1. The joint posterior of coefficients for genes 1 and 369 is highly correlated, and skewed to large absolute values, as shown by Figure 11c. The reason is that in logistic regression, only the ratio between coefficients can be fairly determined by the classification boundary, but the "slope" of hyperplane cannot, therefore both log likelihood function and heavy-tailed priors allow large proportional absolute values. These observations indicate that it is necessary to use HMC to overcome the random walk of ordinary MCMC methods. Second, Figure 11b shows that genes 369 and 2899 are also correlated but both are weakly differentiated, therefore they are redundant in explaining the response. Markov chain simulation (results shown by Figure 11d) separates genes 369 and 2866 into different modes, and uses only 1 in each (or none). However, the absolute coefficient values of each gene are clearly large in each mode, indicating that both are useful but redundant. Generally, we see that the posterior of HBPLR using heavy-tailed priors is very multimodal.

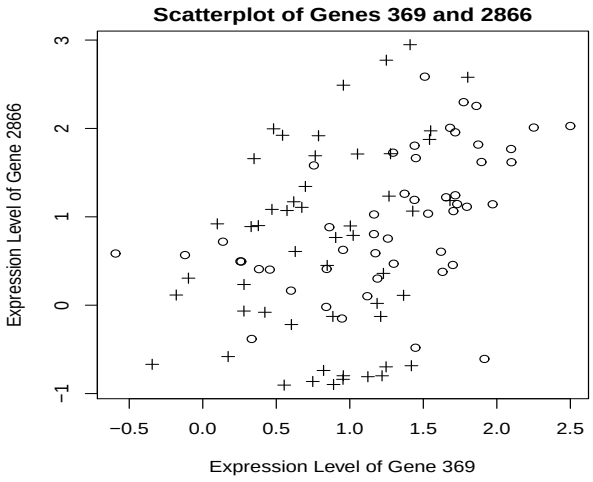
Finally, we compare the predictive ability of the top 3 genes found by HBPLR with other small gene subset, by looking at LOOCV prediction results given by running HBPLR (using t priors with $\alpha = 1$) on the data set containing only genes of a fixed subset. The results are shown in Table 4. The subset of genes 1, 369 and 977 is substantially better than other

Figure 11: MCMC coefficient samples, expression scatterplots, and predictive performance of some top genes selected by HBPLR. The two red numbers or + in (e) and (f) label the two cases misclassified in LOOCV.

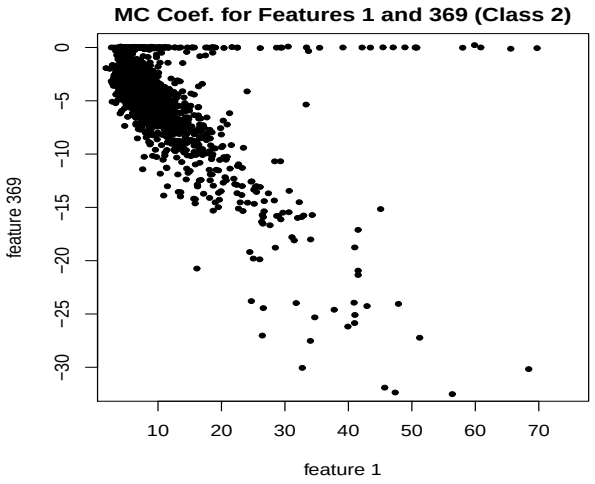
(a)



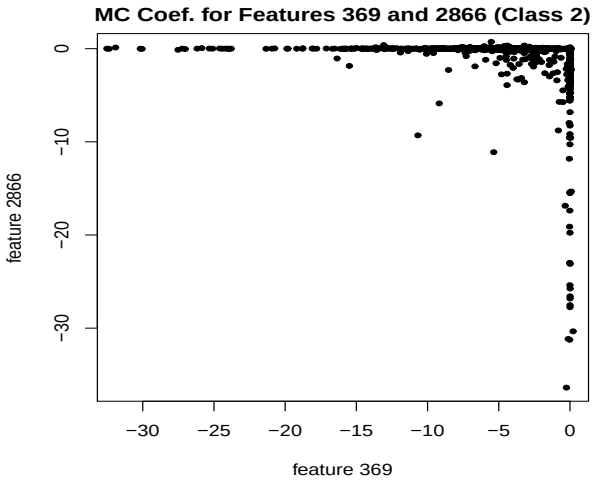
(b)



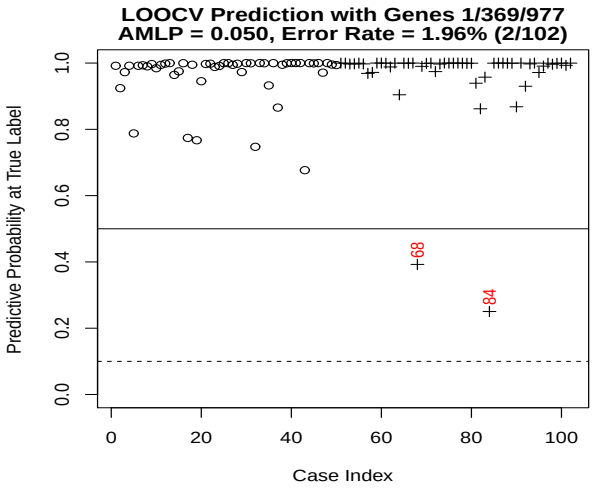
(c)



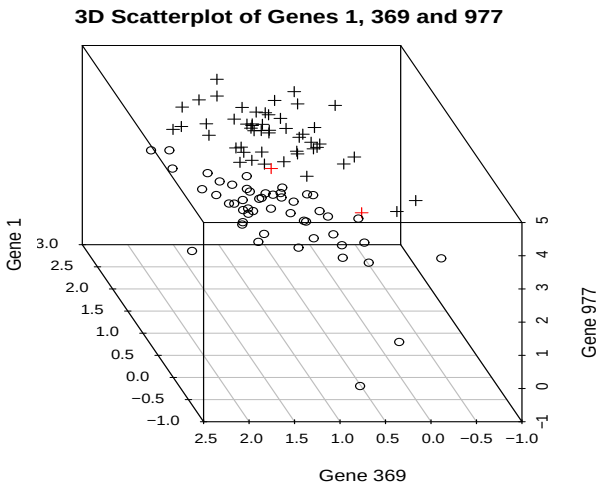
(d)



(e)



(f)



subsets in separating the two classes. The 3D scatterplots (Figure 11f) of the top 3 genes (1, 369, and 977) suggests that this gene subset can separate the two classes very well. The LOOCV predictive probabilities at the true class labels given by HBPLR using only this feature subset are very good, shown by Figure 11e. We think that the subset of genes 1, 369, and 977 is worthy of further biological investigations. We also see that gene 977 is useful, providing additional information than genes 1 and 369 for separating the two classes. Note that, as stated previously, this gene is missed by LASSO.

Table 4: LOOCV predictive performances of various gene subsets.

Gene subset	1, 369, 977	1, 369	1, 2, 3	1, 369, 83
Selected by	HBPLR	HBPLR and LASSO	F-Statistic	LASSO
AMLP	.050	.232	.240	.163
ER (%)	1.96	8.82	9.80	7.84

6 Conclusions and Discussions

In this article, we introduce a fully Bayesian high-dimensional feature selection method that is based on continuous heavy-tailed priors with small scale and Gibbs sampling with HMC, called HBPLR, for classification problems. We have investigated HBPLR with many different choices of priors intensively with simulated data sets and a real microarray data set by comparing to LASSO and other methods. Our main conclusion is that this simple method with little tuning in implementation works very well for problems of very high dimension within reasonable amount of time, with the three noteworthy properties as summarized in Section 1. Our empirical studies also show that some special settings are necessary for HBPLR to work well in high-dimensional problems: 1) choosing Cauchy as prior seems to be optimal in logistic regression, since it has moderately heavy tails; 2) it is neither necessary nor good to treat the scale of Cauchy distribution as a hyperparameter, therefore one can fix it to a reasonable value such as e^{-5}. Meanwhile, we have pointed out some difficulties of this method: 1) the marginal posterior of a coefficient that is highly correlated with others is biased to large absolute values when heavy-tailed prior is used; 2) there are multiple subpools of Markov chain samples that correspond to different feature subsets. These two difficulties

call for more sophisticated methods for interpreting the Markov chain samples. A solution may be to divide Markov chain samples into subpools using clustering methods, and then use median or sample mode to summarize the importance of coefficients in each subpool.

The main drawback of HBPLR is that they are still much slower than many others, such as penalized likelihood methods. Therefore, there is still much room for improving the computational efficiency. The difficulties lie in the high-dimensionality and the existing of multiple posterior modes. A possible solution may be to rotate the original feature space with PCA method, and then apply Gibbs sampling to the posterior distribution of the new coefficients for the transformed feature space with lower dimension. An MCMC simulation for the transformed coefficients may travel across multiple modes with fewer iterations. The crucial step of this solution is to devise a method for sampling the original coefficients conditional on the values of transformed coefficients, a technique similar to what described by Li and Neal (2008). Other methods that have the potential of travelling across the multiple modes more effectively, such as tempered transition described by Neal (1996), may be investigated too.

References

Bae, K. and Mallick, B. K. (2004), "Gene selection using a two-level hierarchical Bayesian model." *Bioinformatics*, 20, 3423–30.

Carvalho, C. M., Polson, N. G., and Scott, J. G. (2010), "The horseshoe estimator for sparse signals," *Biometrika*, 97, 465.

Clarke, R., Ressom, H. W., Wang, A., Xuan, J., Liu, M. C., Gehan, E. A., and Wang, Y. (2008), "The properties of high-dimensional data spaces: implications for exploring gene and protein expression data," *Nature Reviews Cancer*, 8, 37–49.

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., Jakulin, A., Pittau, M. G., and Su, Y.-S. (2008), "A weakly informative default prior distribution for logistic and other regression models," *The Annals of Applied Statistics*, 2, 1360.

Gilks, W. R. and Wild, P. (1992), "Adaptive rejection sampling for Gibbs sampling," *Applied Statistics*, 41, 337–348.

Griffin, J. E. and Brown, P. J. (2012), "BAYESIAN HYPER-LASSOS WITH NON-CONVEX PENALIZATION," *Australian & New Zealand Journal of Statistics*, forthcoming.

Kotz, S. and Nadarajah, S. (2004), *Multivariate t distributions and their applications*, Cambridge Univ Pr.

Li, J., Das, K., Fu, G., Li, R., and Wu, R. (2011), "The Bayesian lasso for genome-wide association studies," *Bioinformatics*, 27, 516–523.

Li, L. (2012), "Bias-Corrected Hierarchical Bayesian Classification With a Selected Subset of High-Dimensional Features," *Journal of the American Statistical Association*, 107, 120–134.

Li, L. and Neal, R. M. (2008), "Compressing Parameters in Bayesian High-order Models with Application to Logistic Sequence Models," *Bayesian Analysis*, 3, 793–822.

Ma, S., Song, X., and Huang, J. (2007), "Supervised group Lasso with applications to microarray data analysis," *BMC bioinformatics*, 8, article 60.

Neal, R. M. (1996), "Sampling from multimodal distributions using tempered transitions," *Statistics and Computing*, 6, 353–366.

— (2010), "MCMC using Hamiltonian dynamics," in *Handbook of Markov Chain Monte Carlo* (eds S. Brooks, A. Gelman, G. Jones, XL Meng). Chapman and Hall/CRC Press.

Park, T. and Casella, G. (2008), "The bayesian lasso," *Journal of the American Statistical Association*, 103, 681–686.

Polson, N. G. and Scott, J. G. (2010), "Shrink globally, act locally: Sparse Bayesian regularization and prediction," *Bayesian Statistics*, 9, 76.

Singh, D., Febbo, P. G., Ross, K., Jackson, D. G., Manola, J., Ladd, C., Tamayo, P., Renshaw, A. A., DAmico, A. V., Richie, J. P., et al. (2002), "Gene expression correlates of clinical prostate cancer behavior," *Cancer cell*, 1, 203–209.

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.

Tolosi, L. and Lengauer, T. (2011), "Classication with correlated features: unreliability of feature ranking and solution," *Bioinformatics*, 27, 1986–1994.

Yi, N. and Xu, S. (2008), "Bayesian LASSO for quantitative trait loci mapping," *Genetics*, 179, 1045–55.

Appendices

A Computational Method for HBPLR

This section is a continued discussion from Section 3.3 about our computational method.

A.1 Initial Values for Gibbs Sampling

The initial values for $\delta_{0:p,1:K}$ are the coefficients of Bayes discriminant rule based on Gaussian distributions, whose mean vectors are estimated by the medians of Markov chain samples produced by the method described in Li (2012), and whose covariance matrix is estimated by equally weighted average of the sample covariance and the identity matrix.

A.2 Updating $\delta_{0:p,1:K}$ with Hamiltonian Monte Carlo

Suppose we want to sample from a d-dimensional distribution with PDF proportional to $\exp(-U(\boldsymbol{q}))$, or construct a transformation leaving it invariant. For our problem, $U(\boldsymbol{q})$ is minus log of posterior distribution of $\boldsymbol{q} = \boldsymbol{\delta}_{0:p,1:K}$ (ie, minus log of (15)).

We will augment $\boldsymbol{q}$ with a set of auxiliary variables $\boldsymbol{p}$ independently distributed with $N(0,1)$, also independent of $\boldsymbol{q}$. For this purpose we will randomly draw a $\boldsymbol{p}$ independently from $N(0,1)$. In physics, $\boldsymbol{p}$ is interpreted as momentums of particles. Next we will transform $(\boldsymbol{q},\boldsymbol{p})$ in a way that leaves invariant $\exp(H(\boldsymbol{q},\boldsymbol{p}))$ — the joint distribution of $(\boldsymbol{q},\boldsymbol{p})$, where $H(\boldsymbol{q},\boldsymbol{p})$ is often called **Hamiltonian**, which is given by:

$$H(\boldsymbol{q},\boldsymbol{p}) = U(\boldsymbol{q}) + K(\boldsymbol{p}) = U(\boldsymbol{q}) + \frac{1}{2}\sum_{i=1}^{d} p_i^2.$$

At the end of this transformation, we will discard $\boldsymbol{q}$, obtaining a new $\boldsymbol{p}$ that is still distributed with $\exp(-U(\boldsymbol{p}))$.

The method for transforming $(\boldsymbol{q},\boldsymbol{p})$ is inspired by the Hamiltonian dynamics, in which

$(\boldsymbol{q}, \boldsymbol{p})$ moves along a continuous time τ according to the following differential equations:

$$\begin{aligned} \frac{dq_i(\tau)}{d\tau} &= \frac{\partial H}{\partial p_i} = \frac{\partial K}{\partial p_i} = p_i \\ \frac{dp_i(\tau)}{d\tau} &= -\frac{\partial H}{\partial q_i} = -\frac{\partial U}{\partial q_i} \end{aligned}$$

It can be shown that Hamiltonian dynamic keeps H unchanged and preserves volume (see details from Neal (2010)). These are the crucial properties of Hamiltonian dynamics that make it a good proposal distribution for Metropolis sampling.

In computer implementation, Hamiltonian dynamics must be approximated by discretized time, using small stepsize ϵ. Leapfrog transformation is one of such methods, which is shown to be better than several other alternatives. *One* leapfrog transformation with stepsize ϵ_i is described as follows:

One Leapfrog Transformation

$$\begin{aligned} p_i &\leftarrow p_i - (\epsilon_i/2)\, \frac{\partial U}{\partial q_i}(q_i), \\ q_i &\leftarrow q_i + \epsilon_i\, p_i, \\ p_i &\leftarrow p_i - (\epsilon_i/2)\, \frac{\partial U}{\partial q_i}(q_i). \end{aligned}$$

Note that, above we apply leapfrog transformations independently to each pair (q_i, p_i), with different stepsizes. Applying a series of leapfrog transformations, we *deterministically* transform (q_i, p_i) to a new state, denoted by (q_i^*, p_i^*), for $i = 1, \ldots, d$. This transformation has the following properties:

- Value of H is nearly unchanged, if ϵ_i is small enough. This is because each leapfrog transformation is a good approximation to Hamiltonian dynamics.
- Reversibility: following the same series of leapfrog transformations, $(q_i^*, -p_i^*)$ will be transformed back to $(q_i, -p_i)$. We therefore add a negation ahead of these leapfrog transformations to form an exactly "reversible" transformation between $(q_i, -p_i)$ and (q_i^*, p_i^*).
- Volume preservation: the Jacobian of this transformation is 1.

A series of leapfrog transformations cannot leave H exactly unchanged. But we will use it only as a proposal distribution in Metropolis sampling. That is, at the end of the leapfrog transformations, $(\boldsymbol{q}^*, \boldsymbol{p}^*)$ will be accepted or rejected randomly according to Metropolis acceptance probability. As a summary, the algorithm of Hamiltonian Monte Carlo is presented completely below:

Hamiltonian Monte Carlo (HMC) with Leapfrog Transformations

Starting from current state $\boldsymbol{q}$, update it with the following steps:

Step 1: Draw elements of $-\boldsymbol{p}$ independently from $N(0,1)$

Step 2: Transform $(\boldsymbol{q}, -\boldsymbol{p})$ with the following two steps:

(a) Negate $-\boldsymbol{p}$ to $\boldsymbol{p}$.

(b) Apply leapfrog transformation ℓ times to transform $(\boldsymbol{q}, \boldsymbol{p})$ to a new state $(\boldsymbol{q}^*, \boldsymbol{p}^*)$. A trajectory connecting the states along these ℓ transformations are called *leapfrog trajectory* with length ℓ.

Step 3: Decide whether or not to accept $(\boldsymbol{q}^*, \boldsymbol{p}^*)$ with a probability given by:

$$\min\left(1, \quad \exp\left(-\Big[H(\boldsymbol{q}^*, \boldsymbol{p}^*) - H(\boldsymbol{q}, -\boldsymbol{p})\Big]\right)\right). \tag{20}$$

If the result is a rejection, set $(\boldsymbol{q}^*, \boldsymbol{p}^*) = (\boldsymbol{q}, -\boldsymbol{p})$.

At last, retaining $\boldsymbol{q}^*$, with $\boldsymbol{p}^*$ discarded.

To implement HMC, we need to choose appropriate stepsizes ϵ_i and ℓ — length of leapfrog trajectory. The stepsizes determine how well the leapfrog transformation can approximate Hamiltonian dynamics. If ϵ_i is too large, leapfrog transformation may diverge, resulting in very high rejection rate, and very poor performance; otherwise, it may move too slowly, even though with very low rejection rate. An ad-hoc choice is a value close to the reciprocal of square root of the 2nd-order partial derivative of U with respect to q_i, which automatically accounts for the width of posterior distribution of q_i. We therefore adjust the reciprocals by an adjustment factor ϵ (which usually should be between 0.1 and 0.5, called *HMC stepsize adjustment*. The exact value of the adjustment factor can be chosen empirically such that the HMC rejection rate is less than but close to 0.2. There is often a critical point beyond which Hamiltonian diverges. A value slightly smaller than this critical point often works the best. According to our experiences, a value close to 0.25 often works well. A good thing for this choice is that it is independent of the choice of ℓ — the length of leapfrog trajectory, because the value of Hamiltonian, actually the whole leapfrog trajectory, changes nearly cyclically as long as it doesn't diverge.

After we determine the stepsize adjustment ϵ, we will determine the length of leapfrog trajectory ℓ. The fact is that the appropriate values of ℓ are different in two phases. In the *initial phase*, a small value ℓ_1 should be used such that the Gibbs sampling can quickly dissipates the value of U, as well as more frequently update the hyperparameter w. The exact choice of ℓ for initial phase can be made empirically by looking at how fast the Gibbs

sampling converges with different values of ℓ. For our problem, $\ell_1 = 10$ or 5 seemingly works fairly well. Another reason for the initial phase is that a very long trajectory starting from the initial value has very high chance of being rejected. After running initial phase for a while, we need to choose a larger value, denoted by ℓ_2 to suppress random walk. This phase is called *sampling phase*. The advantage of using HMC instead of other samplers is that HMC can keep moving in the direction determined by the gradients of U without random walk. We therefore should choose fairly large ℓ (at least larger than 1, when $\ell = 1$, HMC is Langevin Metropolis-Hasting method) such that the leapfrog transformation can reach a distant point from the starting one. However, if ℓ is excessively large, the leapfrog transformation will reverse the direction and move back to the region near the starting point. The choice of ℓ for this phase can be made empirically by looking at the curve of distance of $\boldsymbol{q}$ from the origin along a very long trajectory, which changes cyclically. We will choose the largest ℓ such that leapfrog transformation can move in a direction, ie, the distance of $\boldsymbol{q}$ changes monotonically. From our experiences, $\ell_2 = 50$ or 100 works well for many problems.

To apply HMC to sample (15), we need to compute the 1st-order partial derivatives of $-\log(P(\boldsymbol{\delta}_{0:p,1:K}|\boldsymbol{\sigma}^2_{0:p}))$ with respect to δ_{jk} for $j = 0, \ldots, p, k = 1, \ldots, K$, which is equal to the sum of the following two partial derivatives of L and minus log prior:

$$-\frac{\partial \log(L(\boldsymbol{\delta}_{0:p,1:K}))}{\partial \delta_{jk}} = \sum_{i=1}^{n} x_{ij}(P(y_i = k+1|x_{ij}, \boldsymbol{\delta}_{0:p,1:K})) - I(y_i = k+1)), \quad (21)$$

$$-\frac{\partial \log(P(\boldsymbol{\delta}_{0:p,1:K}|\boldsymbol{\sigma}^2_{0:p}))}{\partial \delta_{jk}} = \left(\delta_{jk} - \sum_{k=1}^{K} \delta_{jk}/C\right) \Big/ \sigma_j^2. \quad (22)$$

An ad-hoc stepsizes ϵ_{jk} is a value close to the reciprocal of the 2nd-order derivatives of U. We also use an estimate of the 2nd-order derivatives of U, which should be independent of current values of $\boldsymbol{\delta}_{0:p,1:K}$ but could be dependent on $\boldsymbol{\sigma}^2_{0:p}$. They are the sum of the following two values:

$$-\frac{\partial^2 \log(L(\boldsymbol{\delta}_{0:p,1:K}))}{\partial^2 \delta_{jk}} \approx \sum_{i=1}^{n} x_{ij}^2/4,$$

$$-\frac{\partial^2 \log(P(\boldsymbol{\delta}_{0:p,1:K}|\boldsymbol{\sigma}^2_{0:p}))}{\partial^2 \delta_{jk}} = \frac{C-1}{C}\frac{1}{\sigma_j^2}.$$

A.3 Restricted Gibbs Sampling

When p is large, the dominating computing in applying HMC is for obtaining values of the linear functions $\delta_{0,k} + \boldsymbol{x}_{i,1:p}\boldsymbol{\delta}_{1:p,k}$ for $i = 1, \ldots, n$ and $k = 1, \ldots, K$, with which we can compute the log likelihood and its partial derivatives with respect to $\boldsymbol{\delta}_{0:p,1:K}$ very easily.

A belief in high-dimensional classification is that most features are irrelevant and there-

fore most coefficients concentrate very close to 0 in a local mode of posterior. It is therefore useless to update them very often. A useful computational trick for reducing computation time is that, for each iteration of Gibbs sampling, we update only those features with σ_j greater than a small threshold ζ without much loss of efficiency. However, even a fairly small ζ can cut off many coefficients from being updated. The consequence is that the computation time for each iteration of Gibbs sampling is reduced substantially, since we can reuse from the last iteration the sum of a large number of $x_{i,j}\delta_{j,c}$ related to those small coefficients, which are to be fixed. We call this trick **restricted Gibbs sampling**. We want to point out that this method can be justified with Markov chain theory, therefore our computation using this trick is still an exact Markov chain simulation. The essential effect of this trick is updating those important features more often than a large number of coefficients for irrelevant features. However, note that when ζ is chosen to be very large, only a few (say ones) coefficients are updated using HMC, and we therefore lose the ability of HMC in suppressing random walk. The consequence is that Markov chain may be harder to travel across the modes, as travelling from one mode to the other needs to update a very small coefficient to a large value. Further research is needed to find the optimal choice of ζ. The implementation used in our examples chose $\zeta = 0.05$ when α is set to 1, for which about 10% of coefficients are updated in each iteration.

B Notations of Prior and MCMC Settings for HBPLR

First, one needs to choose the prior type from `t`, `ghs`, and `neg`. For each choice of prior type, one needs to set these parameters for the prior and MCMC computation:

- α, $\log(w)$: degree freedom (df) and log square scale of `t`/`ghs`/`neg` prior.
- n_1, ℓ_1: number of Gibbs sampling iterations and length of trajectory in initial phase.
- n_2, ℓ_2: number of Gibbs sampling iterations and length of trajectory in sampling phase.
- ζ: the coefficients with σ_j smaller than ζ are fixed in current HMC updating.
- ϵ: stepsize adjustment multiplied to the 2nd order partial derivatives of log posterior.

In addition to these settings, the prior variances σ_0^2 for the intercepts are set to 2000.

Acknowledgements

The research of Longhai Li is supported by Natural Sciences and Engineering Research Council of Canada (NSERC), and Canadian Foundation of Innovations (CFI). Longhai Li acknowledges inspiring conversations with Radford Neal.