Journal of Data Science 22 (1), 152–172
January 2024
DOI: 10.6339/23-JDS1102
Computing in Data Science

Efficient Bayesian High-Dimensional Classification via Random Projection with Application to Gene Expression Data

Abhisek Chakraborty[1,*]

[1]*Department of Statistics, Texas A&M University, 3143 TAMU, College Station, TX 77843, USA*

Abstract

Inspired by the impressive successes of compress sensing-based machine learning algorithms, data augmentation-based efficient Gibbs samplers for Bayesian high-dimensional classification models are developed by compressing the design matrix to a much lower dimension. Ardent care is exercised in the choice of the projection mechanism, and an adaptive voting rule is employed to reduce sensitivity to the random projection matrix. Focusing on the high-dimensional Probit regression model, we note that the naive implementation of the data augmentation-based Gibbs sampler is not robust to the presence of co-linearity in the design matrix – a setup ubiquitous in $n < p$ problems. We demonstrate that a simple fix based on joint updates of parameters in the latent space circumnavigates this issue. With a computationally efficient MCMC scheme in place, we introduce an ensemble classifier by creating R ($\sim$ 25–50) projected copies of the design matrix, and subsequently running R classification models with the R projected design matrix in parallel. We combine the output from the R replications via an adaptive voting scheme. Our scheme is inherently parallelizable and capable of taking advantage of modern computing environments often equipped with multiple cores. The empirical success of our methodology is illustrated in elaborate simulations and gene expression data applications. We also extend our methodology to a high-dimensional logistic regression model and carry out numerical studies to showcase its efficacy.

Keywords *collapsed Gibbs sampler; data augmentation; dimensionality reduction; ensemble learning; parallel processing*

1 Introduction

With the advent of modern technologies, it is now commonplace in many disciplines, including but not limited to bioinformatics, ecology, remote sensing etc., to collect data containing massive numbers of predictors, ranging from thousands to millions or more. In such settings, it is commonly of interest to consider classification models (Albert and Chib, 1993; Loaiza-Maya and Nibbering, 2022; Cao et al., 2022) such as

$$y_i \sim \text{Bernoulli}(1, w_i) \quad \text{where} \quad w_i = \Phi(x_i^{\mathrm{T}}\beta), \tag{1.1}$$

where $\Phi(\cdot)$ is the cumulative distribution function of $\mathrm{N}(0, 1)$, X is an $n \times p$ matrix of predictors, $p \gg n$, y is an $n \times 1$ binary response vector. As traditional techniques such as maximum likelihood cannot be used, a rich variety of alternatives have been proposed mainly in the context of linear regression models, ranging from frequentist penalized optimization methods (Tibshirani,

* Email: cabhisek@stat.tamu.edu or zovialpapai@gmail.com.

Received January 10, 2023; Accepted April 26, 2023

1996; Zou and Hastie, 2005; Zou, 2006; Zhang, 2010; Xie and Huang, 2009) to Bayesian variable selection or shrinkage priors. Examples include the classical "two-group" discrete mixture priors with a point mass at zero (Mitchell and Beauchamp, 1988; George and McCulloch, 1993; Shin et al., 2015), and continuous shrinkage priors expressed as global-local variance mixtures of Gaussian distribution (Polson and Scott, 2011; Park and Casella, 2008; Hans, 2009; Brown and Griffin, 2010; Carvalho et al., 2009, 2010; Bhadra et al., 2017; Piironen and Vehtari, 2017; Armagan et al., 2013; Bhattacharya et al., 2015). The Bayesian approaches are particularly attractive since they provide probabilistic characterization of uncertainty in the high-dimensional regression coefficients and in the resulting predictions, while penalization methods tend to focus on point estimation. It is well known that computing the posterior under Bayesian variable selection priors is an intractable problem, so that one can at best hope for a rough approximation using Markov chain Monte Carlo (MCMC) sampling unless p is small. However, recent developments in this regard have largely improved the computational aspects of the "two-group" (Biswas et al., 2022) and continuous shrinkage priors (Bhattacharya et al., 2016), but scalability still remains to be an quite open area of enquiry. Alternatively, a commonly used approach is to approximate the posterior with a computationally tractable distribution. This gives rise to variational Bayes approximations (Girolami and Rogers, 2006; Titsias and Lawrence, 2010; Faes et al., 2011; Mukherjee and Sen, 2021).

Guhaniyogi and Dunson (2015) proposed a new approach for high-dimensional regression problems based on random projections of the scaled predictor vector prior to analysis which solved several problems simultaneously. In particular, for linear regression models, it completely avoids the computational bottleneck due to the enormous p. Their approach is inspired by the data squashing literature (DuMouchel, 2002; Madigan, 2004; Owen, 2003; Lee et al., 2010) and dramatic success of compressed sensing to facilitate storage and analysis, while retaining the ability to reconstruct the compressed signals with high accuracy under sparsity conditions (Donoho, 2006; Candes et al., 2006).

While such data compression based approaches have largely proved to be extremely successful in various contexts (Guhaniyogi and Dunson, 2015; Cannings and Samworth, 2017), MCMC computation in the context of Bayesian classifications approaches is still illusive. In this article we primarily focus on Bayesian high-dimensional probit regression. We propose a compressed sensing framework equipped with a data augmentation based Gibbs sampler that calculates a conjugate Gaussian-inverse gamma posterior for the regression coefficients corresponding to the compressed predictors in parallel for different random projections in the latent space. Finally, we aggregate the outcomes corresponding to the different compression via a simple but adaptive voting rule. We propose a principled approach to tune the cut-off parameter α of the binary classifier that dramatically improves classification accuracy across extensive simulation and real data examples. Importantly, our proposed methodology is inherently paralizable and continues to enjoy computational tractability even in ultra high-dimensional set up. We also note that, a naive implementation of the data augmentation Gibbs sampler results in poor mixing in the MCMC chains. It turns out that a simple fix based on joint update of the parameters in the latent space alleviates the problem to a large extent, and improves computational efficacy. We end our discussion with an extension to the high-dimensional logistic regression case, where we employ a Polya-Gamma data augmentation with the compressed covariate matrix.

In summary, our primary contributions in this article are three-fold. First, we present a scalable and inherently parallelizable compressed sensing framework equipped with a data augmentation based Gibbs sampler for high-dimensional probit regression. Secondly, for the classification task, we adapt an alternative Gibbs sampling scheme with joint update of the parameters

in the latent space that showcases improved mixing. Thirdly, we present an adaptive voting rule involving a data-driven choice a cut-off parameter for our ensemble classifier, that improves the accuracy of our classifiers without significant increase in compute time.

Rest of the paper is organised as follows. Section 2 introduces a data compression strategy, a data augmentation based Gibbs sampler equipped with adaptive cut-off to carry out Bayesian high-dimensional probit regression. Section 3 presents elaborate empirical studies to demonstrate the efficacy of our methodology. Section 4 includes Micro-array gene expression data analyses to showcase practical utility and scalability of our prescription. Section 5 features an extension to our methodology to Bayesian high-dimensional logit regression, as well as an supporting empirical study. In section 6, we conclude.

2 Algorithm

2.1 Notations

For subjects $i = 1, ..., n$, let $y_i \in \mathbb{Y}$ denote a response and $x_i = (x_{i1}, ..., x_{ip}) \in \mathbb{X} \in \mathbb{R}^p$ denote predictors. We consider compressed regression models having the form

$$y_i \sim \text{Bernoulli}(1, w_i), \quad w_i = \Phi([\Psi x_i]^{\mathrm{T}}\beta), \quad \beta \sim \pi(\beta), \tag{2.1}$$

where Φ is the cumulative distribution function of $\mathrm{N}(0, 1)$, Ψ is an $m \times p$ projection matrix with $m < \min(n, p)$, and $\beta = (\beta_1, ..., \beta_m)^{\mathrm{T}}$ are coefficients on the compressed predictors which a priori are from some distribution $\pi(\cdot)$. To ensure that our inferential procedure is robust to the specific choice of random projection Ψ, we consider R different random compression matrices. The sparsity of a projection matrix Ψ is controlled by a parameter s, refer to (2.2) for details. Now we are in a position to systematically unfold the pieces of our proposal.

2.2 Compression Mechanism

The choice of the projection scheme in (2.1) is a vital component of our methodology, and there is potential merit in attempting to utilize data-driven dimension reduction techniques, i.e, estimate Ψ based on the data. To that end, we can turn to the enormous body of literature on linear dimension reduction techniques, including principal component analysis (Hotelling, 1933; Jolliffe and Cadima, 2016), non-negative matrix factorization (Sra and Dhillon, 2005), singular value decomposition (Banerjee and Roy, 2014), sufficient dimension reduction (Adragni and Cook, 2014), semantic mapping (Corrêa and Ludermir, 2007), multi-dimensional scaling (Cox and Cox, 2001), to name a few. Besides, various non-linear dimensionality reduction techniques are available in our arsenal, like kernel-PCA (Mika et al., 1998), locally linear embedding (Roweis and Saul, 2000), stochastic neighbourhood embedding (Hinton and Roweis, 2002), t-SNE (van der Maaten and Hinton, 2008), etc. Such techniques are routinely incorporated in predictive models to ensure scalability, but developing data driven projection schemes still convey a huge computational price that is often practically infeasible in ensuing applications. Further, some of the linear dimension reduction techniques, e.g singular value decomposition, principal components analysis, classical multidimensional scaling, etc. suffer from unfavorable local properties (van der Maaten and Hinton, 2008). We make this precise in the next paragraph, focusing on singular value decomposition.

Given the design matrix $X = (x_1, x_2, \ldots, x_n)^{\mathrm{T}}$, in order to embed the n points in R^p into R^m via the singular value decomposition, we project them onto the m-dimensional space spanned

by the singular vectors corresponding to the m largest singular values of X. This produces an optimal rank m approximation of X under several popular matrix norms. But this optimality implies no guarantees regarding local properties of the resulting embedding. That is, we can easily devise examples where the new distance between a pair of points is arbitrarily smaller than their original distance. In increasing number of modern machine learning applications where dimensionality reduction is desirable, the absence of such local guarantees can make it hard to exploit embeddings algorithmically.

With these limitations in mind, following Guhaniyogi and Dunson (2015), Cannings and Samworth (2017), we do not attempt to estimate Ψ based on the data. Instead, we take refuge to random projection schemes, owing to their favourable local properties, and computational simplicity of the resulting algorithms. The seminal paper by Johnson and Lindenstraus (1984) proved the *existence* of lower dimensional projection mechanisms that satisfy favorable local properties, under certain sufficient conditions. To add on to the impressive theoretical developments in Johnson and Lindenstraus (1984), Achlioptas (2003) provided a concrete *construction* of such projection mechanisms as follows: (i) Let $\mathbb{U} \subset \mathbb{R}^p$ be an arbitrary set of n points collected in a $n \times p$ design matrix X. Also, given $\varepsilon, \nu > 0$, define $m_0 \propto \log n$, where the proportionality constant depends on ν and ε. (ii) For any integer $m \geqslant m_0$; define $\Psi = ((\psi_{ij}))$ to be a $p \times m$ random matrix such that ψ_{ij} are independent random variables from the following probability distribution:

$$\psi_{ij} = \sqrt{s} \begin{cases} -1, & \text{with probability } \frac{1}{2s}, \\ 0, & \text{with probability } 1 - \frac{1}{s}, \\ +1, & \text{with probability } \frac{1}{2s}, \end{cases} \tag{2.2}$$

where, for example, $s = 1$ or 3. (iii) Let $E = (1/\sqrt{m})X\Psi$, and suppose $f : \mathbb{R}^p \to \mathbb{R}^m$ projects the i-th row of X to the i-the row of E. Then, for any $u, v \in \mathbb{U}$, $(1-\varepsilon)||u-v||^2 \leqslant ||f(u) - f(v)||^2 \leqslant (1+\varepsilon)||u-v||^2$, with probability at least $1 - n^{-\nu}$. Random compression matrices with similar local properties can also be obtained via populating a random matrix with elements drawn from the standard normal distribution (Dasgupta, 2013; Li et al., 2006b). But the sparse random projections scheme described in equation (2.2), only processes $1/s$-th of the data, and only involves simulation from uniform density. Hence it is typically preferred in many applications. Further, Li et al. (2006a,b) demonstrated that we can even use very sparse random projection with $s \gg 3$, e.g., $s = \sqrt{p}$, or $s = p/\log p$ to significantly speed up the computation. However, for improved robustness, they recommended choosing s less aggressively, e.g, $s = \sqrt{p}$. In practice, we observed that fixed $s = 5$ or 10 works reasonably well in varied empirical settings.

Further, to ensure that our methodology is robust to the specific choice of random projection Ψ, we consider R different random compression matrices and utilize an ensemble classifier that combines outputs obtained corresponding to each of the random projection. In this context, readers may be aware that the usage of Bayesian ensemble methods is ubiquitous in classification and prediction settings; including Bayesian model averaging (Hoeting et al., 1999), Bayesian additive classification and regression tree (Chipman et al., 1998, 2006), Bayesian version of bagging (Clyde and Lee, 2001), to name a few. Bayesian Model Averaging (Hoeting et al., 1999) describes an umbrella of techniques where we not only quantify model parameter uncertainty, but also the associated model uncertainty. Clyde and Lee (2001) introduced a Bayesian version of bagging based on the Bayesian bootstrap that often results in more efficient estimators. Bayesian CART (Chipman et al., 1998, 2006) is constructed via an ensemble of binary decision trees built by dividing the predictor space repeatedly into partitions based on splitting rules and

it has enjoyed immense empirical success in this context. Other related ideas include Bayesian classifier combination (Kim and Ghahramani, 2012), Bayesian boosting (Lorbert et al., 2012), cascading classifiers (Li et al., 2010), bucket of models, stacking etc.

In this article, we adapt an *adaptive voting scheme* in 2.5 to combine outputs obtained corresponding to each of the random projections. In our empirical studies, we typically observe that the performance of the classifier improves with the increase in R, but usually plateaus after a while. A choice of $R = 25$ to 50 provided favorable results across the different numerical studies we considered. Next, we describe MCMC algorithms to learn the parameters in (2.1).

2.3 A Naive Blocked Gibbs Sampler

The probit regression model in (2.1) enjoys an equivalent representation via latent variables (Tanner and Wong, 1987; Albert and Chib, 1993),

$$y_i = \delta(z_i > 0), \quad z_i \sim \mathrm{N}([\Psi x_i]^{\mathrm{T}}\beta, 1), \quad \beta \sim \pi(\beta), \tag{2.3}$$

where $\delta(\cdot)$ is the Dirac delta measure, y_i is simply the deterministic conditional on the sign of the stochastic latent variable z_i. In what follows, we denote $y = (y_1, \ldots, y_n)^{\mathrm{T}}$ and $z = (z_1, \ldots, z_n)^{\mathrm{T}}$. Under the conditional independence of $\{z_i \mid \beta\}_{i=1}^n$, the marginal likelihood $\mathrm{L}(\beta \mid y)$ in model (2.3) is exactly same as in (2.1). The advantage of working with representation in (2.3) is that, for judicious choice of $\pi(\beta)$, we easily device efficient blocked Gibbs sampler (Albert and Chib, 1993). In particular, we assume a normal prior on β, i.e, $\pi(\beta) \equiv \mathrm{N}(\mu, \Sigma)$, where μ is set to vector of p zeros, and Σ is diagonal matrix of order p. Then, the full conditional distribution of β is remains to be normal:

$$\beta \mid z \sim \mathrm{N}((\Sigma^{-1} + \Psi X X^{\mathrm{T}} \Psi^{\mathrm{T}})^{-1}(\Sigma^{-1}\mu + \Psi^{\mathrm{T}} X^{\mathrm{T}} z),\ (\Sigma^{-1} + \Psi X X^{\mathrm{T}} \Psi^{\mathrm{T}})^{-1}). \tag{2.4}$$

The full conditional for each element z_i is then truncated normal,

$$z_i \mid \beta, y \sim \delta(y_i = 1)\mathrm{TN}_{(0,\infty)}(\Psi^{\mathrm{T}} x_i,\ 1) + \delta(y_i = 0)\mathrm{TN}_{(-\infty,0)}(\Psi^{\mathrm{T}} x_i,\ 1), \tag{2.5}$$

where $\mathrm{TN}_{a,b}(\cdot, \cdot)$ is the pdf of truncated normal distribution restricted to (a, b). The full conditional distributions in equations (2.4)–(2.5) are extremely straight forward to sample from, and we refer to this algorithm as *Algorithm 1: AC* from here on.

The above latent variable based augmentation method offers a convenient framework to device simple Markov chain Monte Carlo (MCMC) algorithm by iteratively sampling from the full conditional densities. However, a potential problem lurks in that there is strong posterior correlation between regression coefficients β and the latent variables z, clearly indicated in the above model. In the standard Albert-Chib iterative updates, this correlation is likely to cause slow mixing of the MCMC chain.

2.4 An Improved Blocked Gibbs Sampler

To combat the issue of auto-correlation in the MCMC chain, based on Held and Holmes (2006), we suggest a simple approach that reduces auto-correlation and dramatically improves the mixing in the Markov chain. Here, we put the factorisation, $\pi(\beta \mid z, y) = \pi(z \mid y)\ \pi(\beta \mid z)$ to use, and update β and z jointly. Note that, in the above display, the distribution $\pi(\beta \mid z)$ is unchanged from earlier (2.4), but now we update z from its marginal distribution obtained via integrating over β. In particular, we assume that the prior for $\pi(\beta)$ is a mean zero normal density $\mathrm{N}(0, \Sigma)$,

Algorithm 1 Ensemble AC/ AC+.

Input : **(a) Data:** binary response vector $y^{n\times 1}$, design matrix $X^{n\times p}$; a query point $x \in \mathbb{R}^p$.
(c) Hyper-parameters: compression dimension m, number of projections R, sparsity parameter s.

Step 1: Generate R projection matrices $\Psi^{(k)}, k = 1, \ldots, S$, each of order $m \times p$ in parallel, via the scheme in (2.2).

Step 2: Run R classification models in parallel by setting $\Psi = \Psi^{(k)}$, $k = 1, \ldots, R$ in (2.1) respectively, and iterating over steps in (2.4)-(2.5) to calculate corresponding $\hat{\beta}^{(k)}$.

Step 3: For the x, compute $z^{(k)} \sim \mathrm{N}([\Psi^{(k)}x]^{\mathrm{T}}\hat{\beta}^{(k)}, 1)$ and $y^{(k)}(x) = \delta(z^{(k)} > 0)$, $k = 1, \ldots, R$.

Output: Compute the ensemble estimate at x as $y_{\mathrm{vote}}(x) = \delta\big[\sum_{k=1}^{R} y^{(k)}(x) > R \times \alpha\big]$, where $\alpha = 0.5$ for AC, and $\alpha = \hat{\alpha}_{\mathrm{oracle}}$ from (2.10) for AC+.

where Σ is a diagonal matrix of order p. Then we have $\pi(z \mid y) \sim \mathrm{N}(0,\ I_n + \Psi X V X^{\mathrm{T}}\Psi^{\mathrm{T}})$ truncated to an appropriate region. Direct sampling from the multivariate truncated normal is known to be difficult, however, it is straightforward to Gibbs sample the distribution,

$$z_i \mid z_{-i} \ \propto\ \delta(y_i = 1)\mathrm{TN}_{(0,\infty)}(\eta_i, v_i) + \delta(y_i = 0)\mathrm{TN}_{(-\infty,0)}(\eta_i, v_i) \tag{2.6}$$

where $z_{-i} = (z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n)^{\mathrm{T}}$, and

$$\eta_i = x_i B - u_i(z_i - x_i B), \quad v_i = 1 + u_i, \quad u_i = h_i/(1 + h_i) \tag{2.7}$$

with $B = (\Sigma^{-1} + \Psi X X^{\mathrm{T}}\Psi^{\mathrm{T}})^{-1}$, $V_z = (\Sigma^{-1} + \Psi X X^{\mathrm{T}}\Psi^{\mathrm{T}})^{-1}(\Sigma^{-1}\mu + \Psi^{\mathrm{T}}X^{\mathrm{T}}z)$, and $h_i = ((\Psi X V_z X \Psi^{\mathrm{T}}))_{ii}$. Following an update to each z_i we recalculate B via

$$B = B^{\mathrm{old}} - F_i(z_i - z_i^{\mathrm{old}}) \tag{2.8}$$

where B^{old} and z_i^{old} denote the values of B and z_i prior to the update of z_i, and F_i denotes the i-th column of $F = V_z X^{\mathrm{T}}\Psi^{\mathrm{T}}$. The full conditional distributions in equations (2.4) and (2.6) describes our *Algorithm 2: HH*. It is important to note that, we only need to calculate F, u_i and v_i need only be performed once before we run the MCMC iterations. Consequently, the algorithm carries little increase in computational burden over the naive Gibbs sampling approach in section 2.3. The simple modification of the sampler based on use of joint updates dramatically improves mixing and sampling efficiency in the Markov chain across all the numerical studies that we have performed.

With a computationally efficient MCMC scheme in place, we next introduce an ensemble classifier via first creating R ($\sim$ 25–50) projected copies of the design matrix, and then running R classification models with the R projected design matrices in parallel. Finally, we combine the output from the R replications via an adaptive voting scheme equipped with a data driven approach, reminiscent of leave one out cross validation, to choose cut-off parameter introduced next.

Algorithm 2 Ensemble HH/ HH+.

Input : **(a) Data:** binary response vector $y^{n\times 1}$, design matrix $X^{n\times p}$; a query point $x \in \mathbb{R}^p$.
(c) Hyper-parameters: compression dimension m, number of projections R, sparsity parameter s.

Step 1: Generate R projection matrices $\Psi^{(k)}, k = 1, \ldots, R$, each of order $m \times p$ in parallel, via the scheme in (2.2).

Step 2: Run S classification models in parallel by setting $\Psi = \Psi^{(k)}$, $k = 1, \ldots, S$ in (2.1) respectively, and iterating over steps in (2.4) & (2.6) to calculate corresponding $\hat{\beta}^{(k)}$.

Step 3: For the x, compute $z^{(k)} \sim \mathrm{N}([\Psi^{(k)}x]^{\mathrm{T}}\hat{\beta}^{(k)}, 1)$ and $y^{(k)}(x) = \delta(z^{(k)} > 0)$, $k = 1, \ldots, S$.

Output: Compute the ensemble estimate at x as $y_{\mathrm{vote}}(x) = \delta\big[\sum_{k=1}^{S} y^{(k)}(x) > S \times \alpha\big]$, where $\alpha = 0.5$ for HH, and $\alpha = \hat{\alpha}_{\mathrm{oracle}}$ from (2.10) for HH+.

2.5 Adaptive Voting Scheme

We calculate an ensemble of predictions corresponding to the R projections, and combine the results via a simple voting scheme following Cannings and Samworth (2017). Suppose $\{y_k(x)\}_{k=1}^{R} \in \{0, 1\}^R$ are the predictions at x corresponding to the R projections, then the combined classifier takes the form:

$$y_{\mathrm{vote}}(x) = \delta\bigg[\sum_{k=1}^{R} y^{(k)}(x) > R \times \alpha\bigg], \tag{2.9}$$

where $\alpha \in (0, 1)$ is a hyper-parameter, and $\delta(\cdot)$ denotes the Dirac delta measure. We emphasise that additional flexibility is afforded by not pre-specifying the voting threshold α to be 0.5.

In order to develop a data-driven approach to determine α, we introduce some notations. Suppose that the pair (X, Y) takes values in $\mathbb{R}^p \times \{0, 1\}$ with joint distribution characterised by

$$(\pi_0,\ \pi_1) = (\pi[Y = 0],\ \pi[Y = 1]); \quad \Psi^{\mathrm{T}}X \mid Y = r \sim \pi_{(r)}$$

where $\pi_{(r)}$ has the cumulative distribution function $\mathrm{G}_{n,r}(\cdot)$, $r = 0, 1$. Note that, the oracle choice of the cut-off parameter α minimises the miss-classification error rate, i.e,

$$\alpha_{\mathrm{oracle}} = \underset{\alpha\in[0,1]}{\arg\min}\Big[\pi_0\ \mathrm{G}_{n,1}(\alpha) + \pi_1\ (1 - \mathrm{G}_{n,0}(\alpha))\Big].$$

Obviously, we cannot calculate α_{oracle} in practice, since $(\pi_r, \mathrm{G}_{n,r})$, $r = 0, 1$ are unknown. So, we estimate it via replacing the unknown quantities by their sample counter parts, i.e,

$$\hat{\alpha}_{\mathrm{oracle}} = \underset{\alpha\in[0,1]}{\arg\min}\Big[\hat{\pi}_0\ \hat{\mathrm{G}}_{n,1}(\alpha) + \hat{\pi}_1\ (1 - \hat{\mathrm{G}}_{n,0}(\alpha))\Big], \tag{2.10}$$

where

$$\hat{\pi}_r = \frac{1}{n}\sum_{i=1}^{n}\delta[y_i = r], \quad \text{and} \quad \hat{\mathrm{G}}_{n,r}(t) = \frac{\frac{1}{n_r}\sum_{i:y_i=r}\delta[y_{\mathrm{vote}}(x_i) < t]}{\frac{1}{n}\sum_{i=1}^{n}\delta[y_i = r]},\ r = 0, 1,$$

where $\delta(\cdot)$ is the Dirac delta measure. Since empirical distribution functions are piece-wise constant, the objective function in (2.10) does not have a unique minimum, so we choose $\hat{\alpha}_{\text{oracle}}$ to be the average of the smallest and largest minimisers. From here on, we refer to the versions of AC and HH equipped with the adaptive voting scheme in (2.9)–(2.10) as AC+ and HH+, respectively. AC+ and HH+ enjoy superior performance across numerous empirical studies, compared to AC and HH that uses the default choice of $\alpha = 0.5$.

3 Simulation Study

In this section we compare the predictive performance of various versions of the proposed high dimensional probit regression methodology equipped with the data augmentation based Gibbs sampler (2.3). We also consider the alternative implementation of data augmentation based Gibbs sampler that potentially enjoys improved computational efficacy (2.4). We take the proposed principled approach to tune the cut-off parameter α of the binary classifier (2.5). Before presenting the empirical results, we propose default set ups for our algorithms.

The algorithms described in Sections 2.3 and 2.4 involve three tuning-parameters: (1) the dimension of the compressed linear subspace m, (2) the sparsity parameter s of the compression matrix, and (3) the number of random projections R. We propose default choices of these hyper-parameters for ease of use of the practitioners:

1. Based on the recommendation in Guhaniyogi and Dunson (2015), we also propose to use the dimension of the linear subspace to compress to, $m = 40$. This choice works reasonably well in practice while preserving the computational convenience.
2. We set the sparsity parameter s of the compression matrix Ψ to 10, for both sparse and dense examples in Sub-sections 3.1 and 3.2, respectively. It is important to note that the sparsity in the projection matrix increases as s increases, refer to equation (2.2) for details. Consequently, there is potential merit in using smaller s for dense cases in 3.2. We prefer a default choice of $s = 10$ since it works reasonably well under both the set ups, and provides concrete guideline to the users. Moreover, Li et al. (2006a,b) demonstrated that we can use very sparse random projection with $s \gg 3$, e.g., $s = \sqrt{p}$, or $s = p/\log p$ to significantly speed up the computation. In particular, when the data are approximately normal, $\log p$ of the data usually suffice, i.e. $s = p/\log p$, because of the exponential tail bounds in normal-like distributions. A less aggressive choice of $s = 10$ provides a good balance between computational convenience and robustness of the procedure in a wide range of empirical studies.
3. We typically observe that the performance of our classifiers improve with the increase in R, but usually plateaus after a while, refer to Figures 1, 3 for details. A choice of $R = 25$ to 50 provide favorable results across the different numerical studies we considered. For the sake of objectivity, we set the number of random projections $R = 50$.

In the next two subsections, we carry our repeated simulations and benchmark our proposed classifiers: AC/AC+ and HH/HH+. The performance metric we mainly focus on is the median of the missclassification error rates obtained from the different repetitions of a simulation. For a query point $x \in \mathbb{R}^p$, a missclassification is observed if $y_{vote}(x)$ calculated via (2.9) is different from the true class level. We also report the between repetition standard deviation of missclassification error to demonstrate the stability of the numerical results.

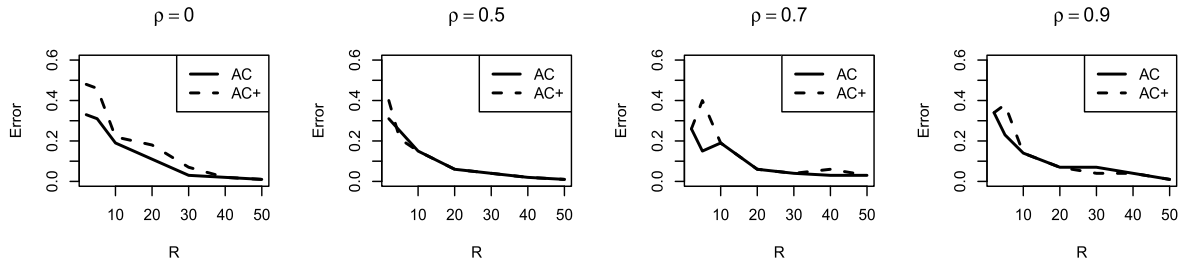

Figure 1: **Choice of R in sparse cases.** Miss-classification error rates with varying number of weak classifiers R, for $(n, p, \zeta) = (10^2, 10^4, 10)$ and $\rho \in \{0.0, 0.5, 0.7, 0.9\}$ for algorithms AC and AC+. We typically observe that the performance of our classifiers improve with the increase in R, but usually plateaus after a while. This observation holds for all other (n, p, ζ) combinations presented in Tables 1 and 3 both for AC/AC+ and HH/HH+, and hence the additional plots are not presented to avoid repetitiveness.

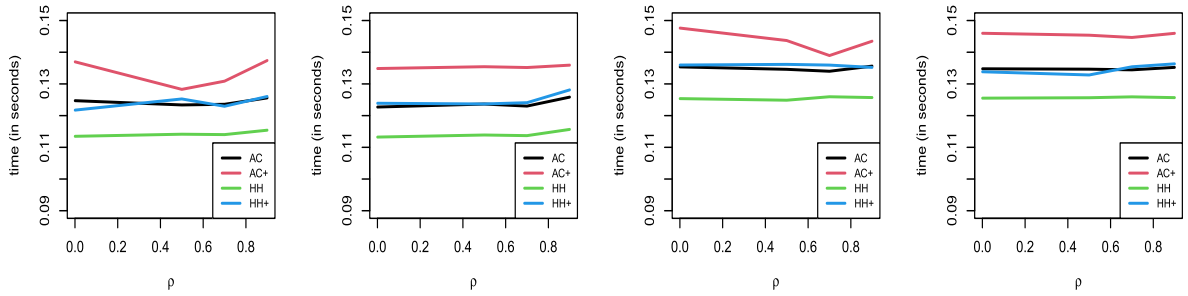

Figure 2: **Time comparison in sparse cases.** Time comparison (in seconds) for drawing 10^4 MCMC samples via algorithms AC, AC+, HH, HH+ for $(n, p, \zeta) = (10^2, 10^3, 5)$ and $\rho \in \{0.0, 0.5, 0.7, 0.9\}$. We typically observe that AC+ (HH+) incurs $5 - 10\%$ extra computational expense in terms of time compared to AC (HH), while enjoying significantly improved miss-classification error rates . This observation holds for all other (n, p, ζ) combinations presented in Tables 1 and 3, and hence the additional plots are not presented to avoid repetitiveness.

3.1 Sparse Cases

We generate observations from the high-dimensional Probit Regression model. We consider the following scenarios, and in each of the scenarios we simulate 50 data sets. We keep the sample size fixed at $n = 10^2$ but vary the number of covariates $p = 10^3,\ 10^4$ and number of non-zero regressions coefficients $\zeta = 5,\ 10$, in order to assess how sparsity impacts performance. We set first ζ regression coefficient at 1, and remaining $p - \zeta$ at 0. Further, we generate the design matrix X such that $\text{corr}(x_i, x_j) = \rho^{|i-j|}$ and vary $\rho = 0.0,\ 0.5,\ 0.7,\ 0.9$, in order to assess the sampling efficiency under correlated design. To complete the model-prior specification, we set the prior variance Σ to be the identity matrix.

For MCMC based model implementations, we discard the first 5000 samples as a burn-in and draw inference based on the next 5000 samples. In particular, we report median miss classification error rates and corresponding 95% confidence intervals obtained from the repeated simulations. We also report *effective sample size* (ESS) as an empirical measure of sampling efficiency under the two MCMC schemes, in order to investigate the mixing behavior of our

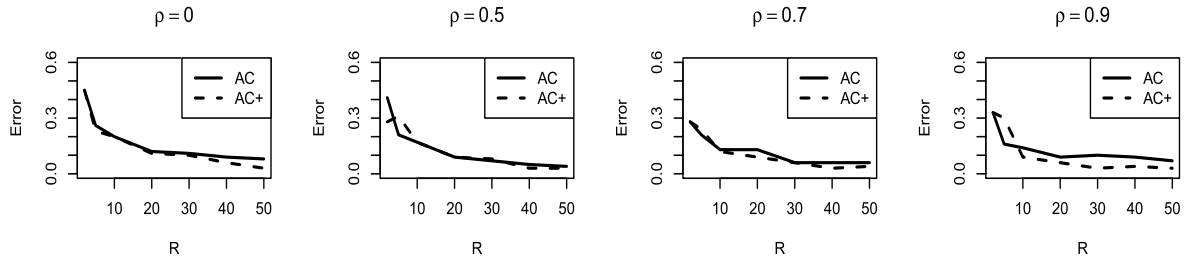

Figure 3: **Choice of R in dense cases.** Miss-classification error rates with varying number of weak classifiers R, for $(n, p, \zeta) = (10^2, 10^4, 10)$ and $\rho \in \{0.0, 0.5, 0.7, 0.9\}$ for algorithms AC and AC+. We typically observe that the performance of our classifiers improve with the increase in R, but usually plateaus after a while. This observation holds for all other (n, p, ζ) combinations presented in Tables 1 and 3 both for AC/AC+ and HH/HH+, and hence the additional plots are not presented to avoid repetitiveness.

samplers. The effective sample size is a measure of the amount of the auto-correlation in a Markov chain, and essentially amounts to the number of independent samples in the MCMC path. From an algorithmic robustness perspective, it is desirable that the effective sample sizes remain stable across varying sparsity and co-linearity in the design matrix, and this is the aspect we wish to investigate here.

We present the miss-classification error rates of the classifiers averaged over the repetitions and the corresponding standard errors under various simulation scenarios for $(n, p) = (10^2, 10^3)$ in Table 1. While all the versions of our methodology enjoyed similar accuracy, the classifiers equipped with the data-driven choice of the cut-off parameter α compared to the default choice $\alpha = 0.5$, seem to slightly improve the performance. Next, we present the corresponding effective sample sizes of the classifiers averaged over the repetitions and the corresponding standard errors under various simulation scenarios in Table 2. The alternative implementation of the data augmentation Gibbs sampler seems to be more robust to the presence of co-linearity in the design matrix, compared to the vanilla implementation. In particular, the effective sample size of AC/AC+ drops by 15% as the ρ changes from 0 to 0.9, whereas the drop is only about 1% for HH/HH+. Moreover, the alternative implementation of the data augmentation Gibbs sampler enjoys significantly higher effective sample size. Although, the gain is about 10% for the independent design, we observe a more pronounced improvement of 25% at $\rho = 0.9$. This indicates

Table 1: **(Sparse cases)** median miss-classification error proportions and between repetition standard error in the subscript for $(n, p) = (10^2, 10^3)$ with varying sparsity ζ.

	Independent		$\rho = 0.5$		$\rho = 0.7$		$\rho = 0.9$	
	$\zeta = 5$	$\zeta = 10$	$\zeta = 5$	$\zeta = 10$	$\zeta = 5$	$\zeta = 10$	$\zeta = 5$	$\zeta = 10$
AC	$0.05_{0.02}$	$0.04_{0.02}$	$0.04_{0.02}$	$0.03_{0.02}$	$0.05_{0.02}$	$0.04_{0.02}$	$0.09_{0.02}$	$0.08_{0.02}$
AC+	$0.03_{0.02}$	$0.02_{0.02}$	$0.03_{0.02}$	$0.03_{0.02}$	$0.05_{0.02}$	$0.04_{0.02}$	$0.09_{0.02}$	$0.07_{0.02}$
HH	$0.05_{0.02}$	$0.04_{0.02}$	$0.04_{0.02}$	$0.03_{0.02}$	$0.05_{0.02}$	$0.05_{0.02}$	$0.09_{0.02}$	$0.08_{0.03}$
HH+	$0.03_{0.02}$	$0.04_{0.02}$	$0.03_{0.01}$	$0.03_{0.02}$	$0.04_{0.02}$	$0.04_{0.02}$	$0.09_{0.02}$	$0.07_{0.02}$

Table 2: **(Sparse cases)** median effective sample size and between repetition standard error in the subscript for $(n, p) = (10^2, 10^3)$ with varying sparsity ζ.

	Independent		$\rho = 0.5$		$\rho = 0.7$		$\rho = 0.9$	
	$\zeta = 5$	$\zeta = 10$	$\zeta = 5$	$\zeta = 10$	$\zeta = 5$	$\zeta = 10$	$\zeta = 5$	$\zeta = 10$
AC/AC+	4718_5	4718_5	4623_5	4624_5	4475_8	4477_6	4002_{12}	3997_{11}
HH/HH+	5008_4	5007_5	5005_5	5005_5	4999_5	5001_5	4959_5	4960_4

Table 3: **(Sparse cases)** median miss-classification error proportions and between repetition standard error in the subscript for $(n, p) = (10^2, 10^4)$ with varying sparsity ζ.

	Independent		$\rho = 0.5$		$\rho = 0.7$		$\rho = 0.9$	
	$\zeta = 5$	$\zeta = 10$	$\zeta = 5$	$\zeta = 10$	$\zeta = 5$	$\zeta = 10$	$\zeta = 5$	$\zeta = 10$
AC	$0.03_{0.02}$	$0.02_{0.02}$	$0.02_{0.02}$	$0.02_{0.01}$	$0.02_{0.01}$	$0.04_{0.02}$	$0.02_{0.02}$	$0.03_{0.02}$
AC+	$0.01_{0.01}$	$0.01_{0.01}$	$0.01_{0.01}$	$0.01_{0.01}$	$0.01_{0.01}$	$0.04_{0.02}$	$0.02_{0.02}$	$0.02_{0.02}$
HH	$0.03_{0.03}$	$0.03_{0.02}$	$0.02_{0.01}$	$0.01_{0.01}$	$0.02_{0.02}$	$0.02_{0.01}$	$0.03_{0.02}$	$0.03_{0.01}$
HH+	$0.01_{0.02}$	$0.01_{0.02}$	$0.01_{0.01}$	$0.01_{0.01}$	$0.01_{0.01}$	$0.02_{0.01}$	$0.03_{0.01}$	$0.03_{0.01}$

Table 4: **(Sparse cases)** median effective sample size and between repetition standard error in the subscript for $(n, p) = (10^2, 10^4)$ with varying sparsity ζ.

	Independent		$\rho = 0.5$		$\rho = 0.7$		$\rho = 0.9$	
	$\zeta = 5$	$\zeta = 10$	$\zeta = 5$	$\zeta = 10$	$\zeta = 5$	$\zeta = 10$	$\zeta = 5$	$\zeta = 10$
AC/AC+	4989_4	4987_5	4978_5	4980_5	4965_5	4965_5	4971_5	4874_5
HH/HH+	5010_4	5012_4	5012_5	5010_5	5011_5	5011_5	5013_5	5010_4

that HH/HH+ will be particularly preferred when the design matrix is highly correlated.

For the case $(n, p) = (10^2, 10^4)$, we present the miss-classification error rates of the classifiers averaged over the repetitions and the corresponding standard errors under various simulation scenarios in Table 3, and the corresponding effective sample sizes of the classifiers averaged over the repetitions and the corresponding standard errors under various simulation scenarios in Table 4. Notably, the effective sample size for AC/AC+ plummeted less compared the case earlier, i.e it drops by 10% as the ρ changes from 0 to 0.9, whereas the is practically no drop for HH/HH+.

3.2 Dense Cases

We stick to the same data generation scenarios, except now we set all the regression coefficients to 1. These dense cases correspond to a one dimensional subspace with no sparsity, and are motivated by the practical applications where each of the covariates has small effect on the outcome. We continue to use the same MCMC configurations as before, and focus the same performance metric.

We present the miss-classification error rates of the classifiers and effective sample size

Table 5: **(Dense cases)** median miss-classification error proportions / effective sample size (ESS) and between repetition standard error in the subscript for $(n, p, \zeta) = (10^2, 10^3, 10^3)$.

	Independent		$\rho = 0.5$		$\rho = 0.7$		$\rho = 0.9$	
	Error	ESS	Error	ESS	Error	ESS	Error	ESS
AC	$0.03_{0.01}$	4718_5	$0.03_{0.02}$	4623_8	$0.05_{0.02}$	4474_6	$0.06_{0.02}$	4474_6
AC+	$0.02_{0.01}$		$0.03_{0.02}$		$0.04_{0.02}$		$0.04_{0.01}$	
HH	$0.03_{0.02}$	5007_4	$0.03_{0.02}$	5006_5	$0.04_{0.02}$	5000_5	$0.06_{0.02}$	4952_5
HH+	$0.02_{0.01}$		$0.03_{0.01}$		$0.03_{0.03}$		$0.05_{0.02}$	

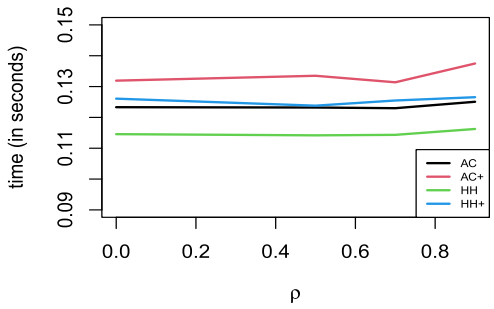

Figure 4: **Time comparison in dense cases.** Time comparison (in seconds) for drawing 10^4 MCMC samples via algorithms AC, AC+, HH, HH+ for $(n, p, \zeta) = (10^2, 10^3, 5)$ and $\rho \in \{0.0, 0.5, 0.7, 0.9\}$. We typically observe that AC+ (HH+) incurs $5 - 10\%$ extra computational expense in terms of time compared to AC (HH), while enjoying significantly improved miss-classification error rates .

averaged over the repetitions and the corresponding standard errors under various simulation scenarios in Table 1. The classifiers equipped with the data-driven choice of the cut-off parameter α compared to the default choice $\alpha = 0.5$, still slightly improve the performance and remains to be more robust to the presence of co-linearity in the design matrix. In particular, the effective sample size of AC/AC+ drops by 5% as the ρ changes from 0 to 0.9, whereas the drop is only about 1% for HH/HH+. Moreover, the alternative implementation of the data augmentation Gibbs sampler enjoys significantly higher effective sample size. Although, the gain is only about 1% for the independent design, we observe a more pronounced improvement of more than 10% at $\rho = 0.9$ that still demonstrates that HH/HH+ may render more efficient when the design matrix is highly correlated.

4 Micro-array Gene Expression Cancer Data Analysis

4.1 Leukemia Data

Leukemia data from high-density Affymetrix oligonucleotide arrays were previously analyzed in Golub et al. (1999), and is freely available on the website data.mendeley.com. There are $p = 7129$ genes and $n = 72$ samples coming from two classes: 47 in class ALL (acute lymphocytic leukemia) and 25 in class AML (acute mylogenous leukemia). Before classification, we standardize each

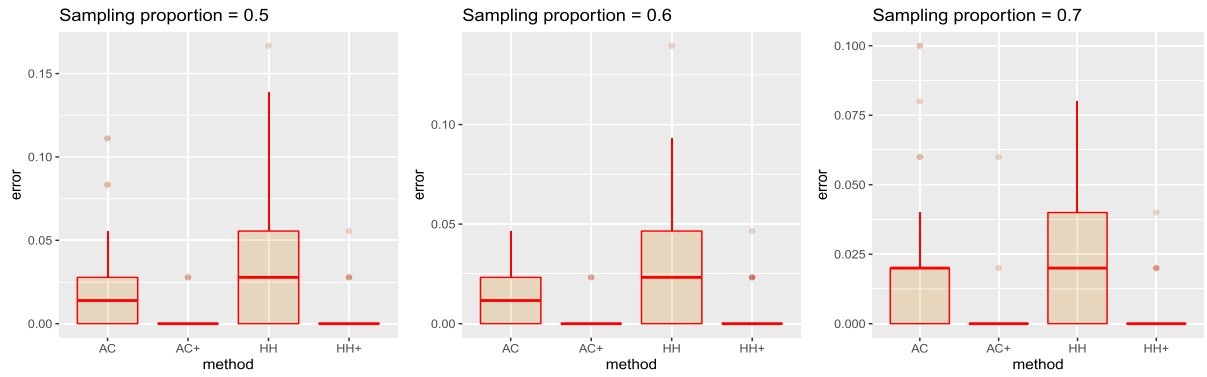

Figure 5: **Leukemia data:** box plots of miss-classification error rates of AC, AC+, HH, HH+ over 50 random splits of 72 samples, where $100 \times \gamma\%$ of the samples are set as training samples. The three plots from left to right correspond to $\gamma = 0.5,\ 0.6,\ 0.7$, respectively.

sample to zero mean and unit variance.

To complete the specification of the compression mechanism, we set the dimension m of the linear subspace to compress to at 40, the sparsity parameter of the compression matrix Ψ at 5, and the number of random projections $R = 25$. For MCMC based model implementations, we discard the first 5000 samples as a burn-in and draw inference based on the next 5000 samples.

To evaluate the performance of the classifiers, we randomly split the 72 samples into training and test sets. Specifically, we set approximately $100 \times \gamma\%$ of the observations as training samples, and the rest as test samples. The various version of our algorithm are used to the training data, and their performances are evaluated by the test samples. The above procedure is repeated 50 times for $\gamma = 0.4,\ 0.5,\ 0.6$, respectively, and the distributions of miss-classification errors in Figure 5. The classifier AC+(HH+) equipped with the data-driven choice of the cut-off parameter α, significantly improve the performance compared to the classifier AC (HH) across all the set ups. Further, as γ increases, i.e, we use more and more training samples, the miss-classification error rates decrease. Moreover, the alternative implementation of the data augmentation Gibbs sampler (HH, HH+) enjoys 3–6 times higher effective sample size compared to AC, AC+.

4.2 Lung Cancer Data

We evaluate our method by classifying between malignant pleural mesothelioma (MPM) and adenocarcinoma (ADCA) of the lung, freely available on data.mendeley.com. Lung cancer data were analyzed by Gordon et al. (2002). There are 181 tissue samples (31 MPM and 150 ADCA). Each sample is described by 1626 genes.

As in the Leukemia data set, we first standardize the data to zero mean and unit variance, and then apply various classification methods to the standardized data set. We follow the same procedure as that in Leukemia example to randomly split the 181 samples into training and test sets, and utilize the same compression and MCMC specifications. Various classification methods are applied to the training data, and the test errors are calculated using the test data. The procedure is repeated 50 times with $\gamma = 0.4,\ 0.5,\ 0.6$, respectively, and the distributions of miss-classification errors in Figure 6. The classifier AC+(HH+) equipped with the data-driven choice of the cut-off parameter α, quite dramatically improve the performance compared to the classifier AC (HH) across all the set ups. Moreover, the alternative implementation of the

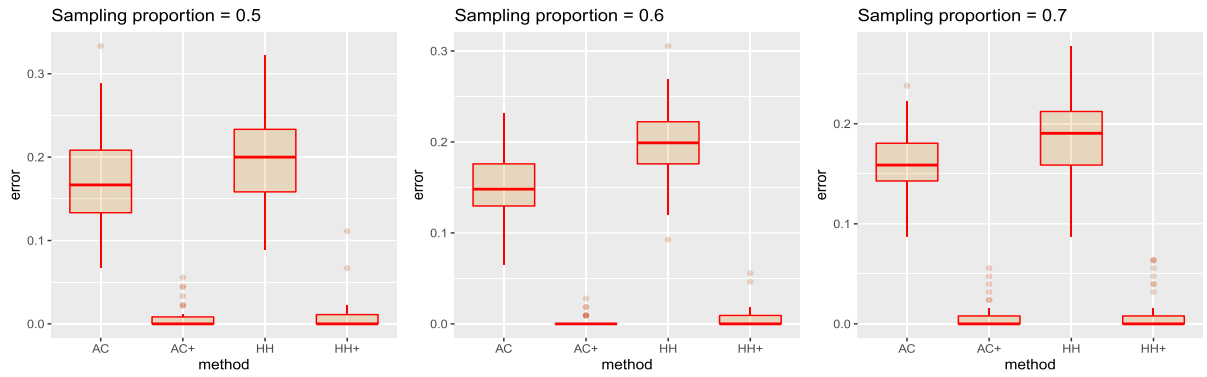

Figure 6: **Lung cancer data:** box plots of miss-classification error rates of AC, AC+, HH, HH+ over 50 random splits of 181 samples, where $100 \times \gamma\%$ of the samples are set as training samples. The three plots from left to right correspond to $\gamma = 0.5,\ 0.6,\ 0.7$, respectively.

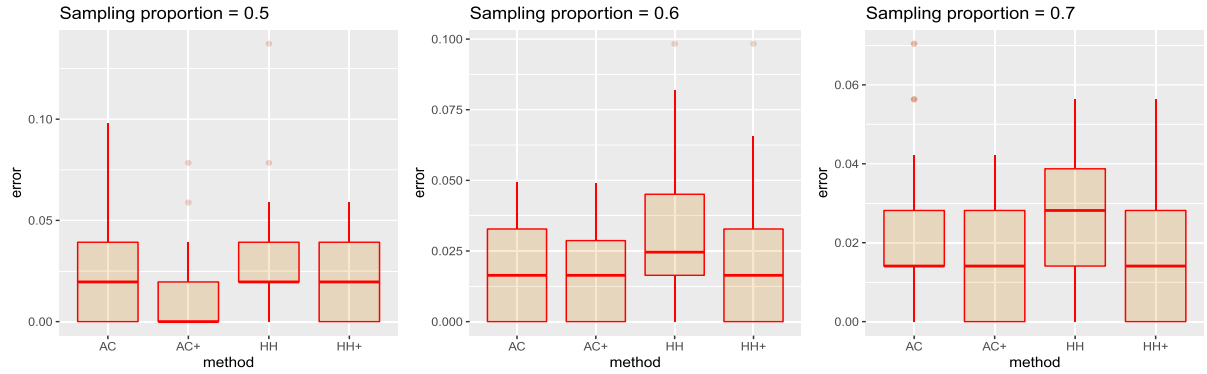

Figure 7: **Prostate cancer data:** box plots of miss-classification error rates of AC, AC+, HH, HH+ over 50 random splits of 102 samples, where $100 \times \gamma\%$ of the samples are set as training samples. The three plots from left to right correspond to $\gamma = 0.5,\ 0.6,\ 0.7$, respectively.

data augmentation Gibbs sampler (HH, HH+) enjoys 1.5–2 times higher effective sample size compared to AC, AC+.

4.3 Prostate Cancer Data

The last example uses the prostate cancer data studied in Singh et al. (2002), also freely available on data.mendeley.com. The training data set contains 102 patient samples, 52 of which (labeled as "tumor") are prostate tumor samples and 50 of which (labeled as "Normal") are prostate samples. There are around 329 genes.

As in the Leukemia data set, we first standardize the data to zero mean and unit variance, and then apply various classification methods to the standardized data set. We follow the same procedure as that in Leukemia example to randomly split the 102 samples into training and test sets, and utilize the same compression and MCMC specifications. Various classification methods are applied to the training data, and the test errors are calculated using the test data. The procedure is repeated 50 times with $\gamma = 0.4,\ 0.5,\ 0.6$, respectively, and the distributions of miss-classification errors in Figure 7. The classifier AC+(HH+) equipped with the data-driven

choice of the cut-off parameter α, tend to improve the performance compared to the classifier AC (HH) across all the set ups. Moreover, the alternative implementation of the data augmentation Gibbs sampler (HH, HH+) enjoys $2.5-3$ times higher effective sample size compared to AC, AC+.

5 Extension: Sparse Binary Logistic Regression

Maintaining parity with the section (2.3), we very briefly consider compressed logistic regression models having the form

$$y_i \sim \text{Bernoulli}(1,\ \text{Logit}([\Psi x_i]^{\mathrm{T}}\beta)\ ),\quad \beta \sim \pi(\beta), \tag{5.1}$$

where $\text{Logit} : t \to \log(t/1-t)$, $0 < t < 1$; Ψ is an $m \times p$ projection matrix with $m < \min(n, p)$; and $\beta = (\beta_1, ..., \beta_m)^{\mathrm{T}}$ are coefficients on the compressed predictors which a priori are from some distribution $\pi(\cdot)$. The data augmentation mechanism that we adapt here, known as the Polya-Gamma data augmentation scheme (Polson et al., 2013) enjoyed the most empirical success in practice. To introduce the methodology, we first present the following definitions of Polya-Gamma family of distributions. In particular, the Polya-Gamma distribution PG(b, 0), b > 0 (Polson et al., 2013) is defined as the distribution with the characteristic function: $t \to (\cosh\sqrt{t/2})^{-b}$, $b > 0$. The general form PG(b, c), b > 0 of the Polya-Gamma distribution (Polson et al., 2013) has the probability density function:

$$\pi(z \mid b, c) = \frac{\exp(-\frac{c^2}{2}z)\ \pi(z \mid b, 0)}{\mathbb{E}_z\big\{\exp(-\frac{c^2}{2}z)\big\}}.$$

This family of distributions has been carefully constructed to yield a simple Gibbs sampler for the Bayesian logistic-regression model. We assume $\pi(\beta) \equiv \text{N}(\mu, \Sigma)$, and to sample from the posterior distribution using the Polya-Gamma method, simply iterate two steps:

$$z_i \mid \beta \sim \text{PG}(1,\ [\Psi x_i]^{\mathrm{T}}\beta),\quad \beta \mid y, z \sim \text{N}(V_z(X^{\mathrm{T}}\Psi^{\mathrm{T}}\kappa + \Sigma^{-1}\mu),\ V_z) \tag{5.2}$$

where $V_z = (X^T\Psi^{\mathrm{T}}\Omega\Psi X + \Sigma^{-1})^{-1}$, $\kappa = (y_1 - 1/2, \ldots, y_n - 1/2)^{\mathrm{T}}$, and Ω is a diagonal matrix with $\Omega_{ii} = z_i$. Noteworthy, the two basic differences in the sampler described in (5.2) from the AC sampler in (2.4)–(2.5) for probit regression are that the full conditional distribution of $[\beta \mid \cdot]$ is a scale mixture of Gaussian distributions rather than a location mixture; and the full conditional distribution of $[z_i \mid \cdot]$ are the Polya-Gamma latent distribution instead of the truncated normals. From here on, we refer to the data augmentation based Gibbs sampler in (5.2) by *Algorithm 3: PG*. As an alternative, we also consider PG equipped with the adaptive choice of cut-off α in (2.5), referred to as PG+.

We consider simulated examples to demonstrate the efficacy of the proposed extension. We generate observations from the high-dimensional Probit Regression model, with specifications described in sections (3.1)–(3.2). Note that, we do not generate data from a high-dimensional logit regression model in order to assess our methodology under mild model misspecification. To conduct inference, we consider the Gibbs sampler augmented with the Polya-gamma scheme in the latent space. We continue to utilize the compression and MCMC specifications in the simulations presented in sections (3.1)–(3.2). We present the miss-classification error rates of the classifiers averaged over the repetitions and the corresponding standard errors under various simulation scenarios in Table 6. While all the versions of our methodology enjoyed similar

Table 6: **(Sparse and dense cases)** median miss-classification error proportions / effective sample size (ESS) and between repetition standard error in the subscript for $(n, p) = (10^2, 10^3)$.

ζ	Method	Independent		$\rho = 0.5$		$\rho = 0.7$		$\rho = 0.9$	
		Error	ESS	Error	ESS	Error	ESS	Error	ESS
5	PG	$0.02_{0.01}$	1294_{131}	$0.01_{0.01}$	1138_{123}	$0.01_{0.01}$	972_{156}	$0.03_{0.02}$	549_{166}
	PG+	$0.00_{0.01}$		$0.01_{0.01}$		$0.01_{0.01}$		$0.02_{0.01}$	
10	PG	$0.02_{0.02}$	1263_{102}	$0.02_{0.01}$	1086_{66}	$0.01_{0.01}$	873_{143}	$0.02_{0.02}$	404_{195}
	PG+	$0.01_{0.01}$		$0.01_{0.01}$		$0.01_{0.01}$		$0.02_{0.01}$	
1000	PG	$0.01_{0.01}$	1262_{110}	$0.01_{0.01}$	1048_{127}	$0.02_{0.01}$	769_{122}	$0.01_{0.01}$	132_{60}
	PG+	$0.00_{0.01}$		$0.01_{0.01}$		$0.01_{0.01}$		$0.01_{0.01}$	

accuracy, the classifier equipped with the data-driven choice of the cut-off parameter α compared to the default choice $\alpha = 0.5$, seems to slightly improve the performance. We also present the effective sample sizes of the classifiers averaged over the repetitions and the corresponding standard errors under various simulation scenarios. The effective sample size of PG/PG+ drops drastically as the ρ changes from 0 to 0.9. In order to maintain the focus of the document on high-dimensional probit regression, and given non-robust simulation results with respect to co-linearity in this set up, we leave this as an avenue for future enquiry aimed at designing MCMC schemes robust to presence of co-linearity in the design matrix.

6 Conclusions

In this article, we presented efficient data augmentation based Gibbs samplers for Bayesian high-dimensional Probit and logit models. Focusing on high-dimensional Probit regression model, we demonstrate that the naive implementation of the data augmentation based Gibbs sampler is not robust to the presence of co-linearity in the design matrix– a set up ubiquitous in $n < p$ problems, and considered a simple fix based on joint updates of parameters in the latent space that seems to circumnavigate this issue. With a computationally efficient MCMC scheme in place, we introduced an ensemble classifier via first creating R (25 to 50) projected copies of the design matrix, and then running R classification models with the R projected design matrix in parallel. Finally, we combine the output from the R replications via an adaptive voting scheme reminiscent of leave one out cross validation. Notably, each of the projected design matrix is $n \times m$, compared to the actual design matrix which is $n \times p$. Since $m \ll p$, each of the projected design matrix induces significantly less storage burden. Moreover, perhaps more importantly, since our scheme is inherently parallelable, it's extremely computationally convenient and is capable of taking advantages of modern multiple cores computing environments.

In principle, ensembles of data augmentation based Gibbs samplers like ours can be developed for high-dimensional multinomial Probit or logit models with ordinal as well as nominal categories. Ensuring the stability of the resulting samplers is an interesting alley for future enquiry. For the sake of brevity of presentation in this article, we do not explore such extensions here. Moreover, a criticism of the compressed regression frameworks is it's inability to carry out variable selection, and this too provides a scope for future exploration.

Acknowledgments and Fundings

We thank the Department of Statistics, Texas A&M University, for letting us access the departmental servers to carry out our elaborate numerical studies. There was not external or internal funding for the work.

Supplementary Material

Software implementation of the methodologies developed in the article is available for use at zovialpapai/Bayesian-classification-with-random-projection. Here, we present a short description about the directories in the repository, as follows: **(1) functions:** The directory contains utility functions in two R scripts, that are utilised in the repeated simulations and real data analysis conducted in the paper. (a) "BCC_Functions.R" contains functions for compression matrix generation; Probit regression via Albert & Chib and Holmes & Held data augmentation schemes; Logit regression via Polya-Gamma data augmentation scheme; hyper-parameter tuning; and associated helper functions. (b) Probit_HH_cpp.R contains Probit regression via Holmes & Held data augmentation scheme, written in Rcpp. **(2) repeated simulations**: The directory contains three R scripts, named BCC_sims.R, Weakleaners.R, and time_comparison.R. (a) BCC_sims.R can be utilised to carry out the simulations presented in Section 3 on High-dimensional Probit regression, and Section 5 on High-dimensional Logit regression, along with hyper-parameter tuning. (b) Weakleaners.R can be utilized to study the effect of number of replications of compression matrix (or number of weak classifiers) on the accuracy of classifiers AC, AC+, HH, HH+. The results are presented in Section 3. (c) time_comparison.R can be utilised to study comparative computional time of our classifiers. The results are presented in Section 3. **(3) data**: Micro-array gene expression cancer data sets utilized in the article is freely available on the website data.mendeley.com. Copies of the data sets are available in the data directory in the our repository. **(4) real data analysis**: The directory contains the a R script named BCC_data.R that can be utilised to carry out the analysis of micro-array gene expression cancer data sets (Leukemia, Lung Cancer, Prostate cancer), presented in Section 4 of the paper.

References

Achlioptas D (2003). Database-friendly random projections: Johnson-lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4): 671–687. Special Issue on PODS 2001. https://doi.org/10.1016/S0022-0000(03)00025-4

Adragni KP, Cook RD (2014). Sufficient dimension reduction and prediction in regression. *Philosophical Transactions of Royal Society A*, 367: 1–21.

Albert JH, Chib S (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422): 669–679. https://doi.org/10.1080/01621459.1993.10476321

Armagan A, Dunson D, Lee J (2013). Generalized double pareto shrinkage. *Statistica Sinica*, 23(1): 119–143.

Banerjee S, Roy A (2014). *Linear Algebra and Matrix Analysis for Statistics*. Chapman and Hall/CRC.

Bhadra A, Datta J, Polson NG, Willard B (2017). The horseshoe+ estimator of ultra-sparse signals. *Bayesian Analysis*, 12(4): 1105–1131. https://doi.org/10.1214/16-BA1028

Bhattacharya A, Chakraborty A, Mallick BK (2016). Fast sampling with Gaussian scale mixture priors in high-dimensional regression. *Biometrika*, 103(4): 985–991. https://doi.org/10.1093/biomet/asw042

Bhattacharya A, Pati D, Pillai NS, Dunson DB (2015). Dirichlet–laplace priors for optimal shrinkage. *Journal of the American Statistical Association*, 110(512): 1479–1490 PMID: 27019543. https://doi.org/10.1080/01621459.2014.960967

Biswas N, Mackey L, Meng XL (2022). Scalable spike-and-slab. In: *Proceedings of the 39th International Conference on Machine Learning* (K Chaudhuri, S Jegelka, L Song, C Szepesvari, G Niu, S Sabato, eds.), volume 162 of *Proceedings of Machine Learning Research*, 2021–2040. PMLR.

Brown PJ, Griffin JE (2010). Inference with normal-gamma prior distributions in regression problems. *Bayesian Analysis*, 5(1): 171–188. https://doi.org/10.1214/10-BA507

Candes EJ, Romberg JK, Tao T (2006). Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(8): 1207–1223. https://doi.org/10.1002/cpa.20124

Cannings TI, Samworth RJ (2017). Random-projection ensemble classification. *Journal of the Royal Statistical Society Series B*, 79(4): 959–1035. https://doi.org/10.1111/rssb.12228

Cao J, Durante D, Genton MG (2022). Scalable computation of predictive probabilities in probit models with Gaussian process priors. *Journal of Computational and Graphical Statistics*, 31(3): 709–720. https://doi.org/10.1080/10618600.2022.2036614

Carvalho CM, Polson NG, Scott JG (2009). Handling sparsity via the horseshoe. In: *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics* (D van Dyk, M Welling, eds.), volume 5 of *Proceedings of Machine Learning Research*, 73–80. PMLR, Hilton, Clearwater Beach Resort, Clearwater Beach, Florida USA.

Carvalho CM, Polson NG, Scott JG (2010). The horseshoe estimator for sparse signals. *Biometrika*, 97(2): 465–480. https://doi.org/10.1093/biomet/asq017

Chipman H, George E, Mcculloch R (2006). Bayesian ensemble learning. In: *Advances in Neural Information Processing Systems* (B Schölkopf, J Platt, T Hoffman, eds.), volume 19, 1–8. MIT Press.

Chipman HA, George EI, McCulloch RE (1998). Bayesian cart model search. *Journal of the American Statistical Association*, 93(443): 935–948. https://doi.org/10.1080/01621459.1998.10473750

Clyde M, Lee H (2001). Bagging and the bayesian bootstrap. In: *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics* (TS Richardson, TS Jaakkola, eds.), volume R3 of *Proceedings of Machine Learning Research*, 57–62. PMLR. Reissued by PMLR on 31 March 2021.

Corrêa RF, Ludermir TB (2007). Dimensionality reduction of very large document collections by semantic mapping. In: Proceedings of the 6th International Workshop on Self-Organizing Maps. volume 6. 1–6.

Cox T, Cox M (2001). *Multidimensional Scaling*. Chapman and Hall/CRC.

Dasgupta S (2013). Experiments with random projection. arXiv preprint: https://arxiv.org/abs/1301.3849.

Donoho D (2006). Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):

1289–1306. https://doi.org/10.1109/TIT.2006.871582

DuMouchel W (2002). Data Squashing: Constructing Summary Data Sets. 579–591. Springer US, Boston, MA.

Faes C, Ormerod JT, Wand MP (2011). Variational bayesian inference for parametric and non-parametric regression with missing data. *Journal of the American Statistical Association*, 106(495): 959–971. https://doi.org/10.1198/jasa.2011.tm10301

George EI, McCulloch RE (1993). Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, 88(423): 881–889. https://doi.org/10.1080/01621459.1993.10476353

Girolami M, Rogers S (2006). Variational Bayesian multinomial probit regression with gaussian process priors. *Neural Computation*, 18(8): 1790–1817. https://doi.org/10.1162/neco.2006.18.8.1790

Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, et al. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286: 531–537. https://doi.org/10.1126/science.286.5439.531

Gordon M, Beiser J, Brandt J, et al. (2002). The ocular hypertension treatment study: Baseline factors that predict the onset of primary open-angle glaucoma. *Archives of Ophthalmology*, 120: 714–34. https://doi.org/10.1001/archopht.120.6.714

Guhaniyogi R, Dunson DB (2015). Bayesian compressed regression. *Journal of the American Statistical Association*, 110(512): 1500–1514. https://doi.org/10.1080/01621459.2014.969425

Hans C (2009). Bayesian lasso regression. *Biometrika*, 96(4): 835–845. https://doi.org/10.1093/biomet/asp047

Held L, Holmes CC (2006). Bayesian auxiliary variable models for binary and multinomial regression. *Bayesian Analysis*, 1(1): 145–168. https://doi.org/10.1214/06-BA105

Hinton GE, Roweis S (2002). Stochastic neighbor embedding. In: *Advances in Neural Information Processing Systems* (S Becker, S Thrun, K Obermayer, eds.), volume 15. MIT Press.

Hoeting JA, Madigan D, Raftery AE, Volinsky CT (1999). Bayesian model averaging: A tutorial. *Statistical Science*, 14(4): 382–401. https://doi.org/10.1214/ss/1009212519

Hotelling H (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 6: 417–441. https://doi.org/10.1037/h0071325

Johnson WB, Lindenstraus J (1984). Extensions of lipschitz mappings into hilbert space. *Contemporary Mathematics*, 26: 189–206. https://doi.org/10.1090/conm/026/737400

Jolliffe I, Cadima J (2016). Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A*, 374: 1–16.

Kim HC, Ghahramani Z (2012). Bayesian classifier combination. In: *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics* (ND Lawrence, M Girolami, eds.), volume 22 of *Proceedings of Machine Learning Research*, 619–627. PMLR, La, Palma, Canary Islands.

Lee HKH, Taddy M, Gray GA (2010). Selection of a representative sample. *Journal of Classification*, 27: 41–53. https://doi.org/10.1007/s00357-010-9044-x

Li G, Japkowicz N, Stocki TJ, Ungar RK (2010). Cascading Customized Naïve Bayes Couple. 147–160. Springer, Berlin Heidelberg, Berlin, Heidelberg.

Li P, Hastie T, Church K (2006a). Improving random projections using marginal information. In: Conference on Learning Theory. 635–649. 2006.

Li P, Hastie T, Church K (2006b). Very sparse random projections. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. 287–296.

2006.

Loaiza-Maya R, Nibbering D (2022). Fast variational bayes methods for multinomial probit models. *Journal of Business & Economic Statistics*, https://doi.org/10.1080/07350015.2022.2139267.

Lorbert A, Blei DM, Schapire RE, Ramadge PJ (2012). A bayesian boosting model. arXiv preprint: https://arxiv.org/abs/1209.1996.

Madigan (2004). Likelihood-based data squashing: A modeling approach to instance construction. *Data Mining and Knowledge Discovery*, 6: 173–190. https://doi.org/10.1023/A:1014095614948

Mika S, Schölkopf B, Smola A, Müller KR, Scholz M, Rätsch G (1998). Kernel pca and denoising in feature spaces. In: *Advances in Neural Information Processing Systems* (M Kearns, S Solla, D Cohn, eds.), volume 1, 8. MIT Press.

Mitchell TJ, Beauchamp JJ (1988). Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404): 1023–1032. https://doi.org/10.1080/01621459.1988.10478694

Mukherjee S, Sen S (2021). Variational inference in high-dimensional linear regression. arXiv preprint: https://arxiv.org/abs/2104.12232.

Owen A (2003). Data squashing empirical likelihood. *Data Mining and Knowledge Discovery*, 7: 101–113. https://doi.org/10.1023/A:1021568920107

Park T, Casella G (2008). The bayesian lasso. *Journal of the American Statistical Association*, 103(482): 681–686. https://doi.org/10.1198/016214508000000337

Piironen J, Vehtari A (2017). Sparsity information and regularization in the horseshoe and other shrinkage priors. *Electronic Journal of Statistics*, 11(2): 5018–5051. https://doi.org/10.1214/17-EJS1337SI

Polson NG, Scott JG (2011). *Shrink Globally, Act Locally: Sparse Bayesian Regularization and Prediction*. Oxford University Press.

Polson NG, Scott JG, Windle J (2013). Bayesian inference for logistic models using pólya–gamma latent variables. *Journal of the American Statistical Association*, 108(504): 1339–1349. https://doi.org/10.1080/01621459.2013.829001

Roweis ST, Saul LK (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290: 2323–2326. https://doi.org/10.1126/science.290.5500.2323

Shin M, Bhattacharya A, Johnson VE (2015). Scalable bayesian variable selection using nonlocal prior densities in ultrahigh-dimensional settings. *Statistica Sinica*, 28: 1053–1078.

Singh D, Febbo P, Ross K, et al. (2002). Gene expression correlates of clinical prostate cancer behavior. *Genome Biology*, 1: 203–212.

Sra S, Dhillon I (2005). Generalized nonnegative matrix approximations with bregman divergences. In: *Advances in Neural Information Processing Systems* (Y Weiss, B Schölkopf, J Platt, eds.), volume 18. MIT Press.

Tanner MA, Wong WH (1987). The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82(398): 528–540. https://doi.org/10.1080/01621459.1987.10478458

Tibshirani R (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B, Methodological*, 58(1): 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Titsias M, Lawrence ND (2010). Bayesian gaussian process latent variable model. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (YW Teh,

M Titterington, eds.), volume 9 of *Proceedings of Machine Learning Research*, 844–851. PMLR, Chia, Laguna Resort, Sardinia, Italy.

van der Maaten L, Hinton G (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9: 1–27.

Xie H, Huang J (2009). SCAD-penalized regression in high-dimensional partially linear models. *The Annals of Statistics*, 37(2): 673–696. https://doi.org/10.1214/07-AOS580

Zhang CH (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2): 894–942. https://doi.org/10.1214/09-AOS729

Zou H (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476): 1418–1429. https://doi.org/10.1198/016214506000000735

Zou H, Hastie T (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B, Statistical Methodology*, 67(2): 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x