Journal of Data Science 21 (1), 87–105
January 2023

DOI: 10.6339/22-JDS1052
Computing in Data Science

Variable Selection with Scalable Bootstrapping in Generalized Linear Model for Massive Data

Zhang Zhang[1], Zhibing He[2], Yichen Qin[3], Ye Shen[4], Ben-Chang Shia[5], and Yang Li[1,6,*]

[1]*Center for Applied Statistics and School of Statistics, Renmin University of China, Beijing, China*
[2]*School of Mathematical and Statistical Sciences, Arizona State University, AZ, USA*
[3]*Department of Operations, Business Analytics, and Information Systems, University of Cincinnati, OH, USA*
[4]*College of Public Health, University of Georgia, GA, USA*
[5]*Graduate Institute of Business Administration and College of Management, Fu Jen Catholic University, Taiwan*
[6]*RSS and China-Re Life Joint Lab on Public Health and Risk Management, Renmin University of China, Beijing, China*

Abstract

Bootstrapping is commonly used as a tool for non-parametric statistical inference to assess the quality of estimators in variable selection models. However, for a massive dataset, the computational requirement when using bootstrapping in variable selection models (BootVS) can be crucial. In this study, we propose a novel framework using a bag of little bootstraps variable selection (BLBVS) method with a ridge hybrid procedure to assess the quality of estimators in generalized linear models with a regularized term, such as lasso and group lasso penalties. The proposed method can be easily and naturally implemented with distributed computing, and thus has significant computational advantages for massive datasets. The simulation results show that our novel BLBVS method performs excellently in both accuracy and efficiency when compared with BootVS. Real data analyses including regression on a bike sharing dataset and classification of a lending club dataset are presented to illustrate the computational superiority of BLBVS in large-scale datasets.

Keywords *distributed computing; large-scale dataset; scalable bootstrap; variable selection*

1 Introduction

In a scenario of multi- and high-dimensional regression analysis, adding a penalty term is a popular method for identifying the attributes that actually affect the response. Commonly used penalty methods include lasso (Tibshirani, 1996), group lasso (Meier et al., 2008), adaptive lasso (Zou, 2006), and smoothly clipped absolute deviation (Fan and Li, 2001), among others. For instance, a peer-to-peer lending company, which matches borrowers with investors through an online platform, usually constructs a penalized logistic regression model to identify significant predictors that are associated with borrowers' default risk from a large number of attributes (e.g., personal information and loan records). Because the data contains both continuous and categorical predictors, the lasso penalty does not work well as it only selects individual dummy

*Corresponding author. Email: yang.li@ruc.edu.cn.

Received February 20, 2022; Accepted May 26, 2022

variables instead of whole factors. Moreover, the lasso solution depends on the dummy variable encoding algorithm. Choosing a different reference for the categorical predictor may lead to different solutions. Thus, the group lasso for logistic regression (Meier et al., 2008) is preferable in this case.

An interesting topic in variable selection models is assessing the quality of the estimators. The assessment method can be chosen according to one's inferential goals: for instance, standard deviation or confidence regions. Bootstrap variable selection (BootVS), a non-parametric method, is proposed in the literature (Shao, 1996; Chatterjee and Lahiri, 2011).The BootVS method first randomly generates bootstrap resamples. A variable selection method is applied to each bootstrap resample. Then the results from all bootstrap resamples are collated to assess the quality of the estimators.

The big data era has brought exponential growth in the size of datasets, which challenges statistical methods. When the size of dataset n is much larger than the number of covariates p ($n \gg p$), a natural solution is to use modern distributed computing. Indeed, traditional bootstrapping is ideally suited to distributed computing because it assigns different bootstrap resamples to different processors or compute nodes, and each bootstrap resample is independently computed in parallel. However, computing one bootstrap resample for massive datasets is still problematic because each bootstrap resample has the same order size as the original dataset. An example is building a robust prediction model for an online lending company, where the training dataset normally has millions of observations. Thus, traditional bootstrapping may overwhelm computational resources and be computationally inefficient when the original dataset is large.

Various methods have been proposed to reduce the sample size of each bootstrap resample. Subsampling, also known as delete-d jackknifing (Wu et al., 1986), is a "drawing without replacement" method. Meinshausen and Bühlmann (2010) introduced a stability selection framework for high-dimensional data, based on the subsampling algorithm. Liu et al. (2021) applied the subsampling strategy to lasso penalized regression to analysis microbiome data. A closely related method is the m out of n bootstrap proposed by (Bickel et al., 2012), which randomly samples $m < n$ observations from the original dataset. De Bin et al. (2016) provided a detailed comparison of subsampling and m out of n bootstrapping in the context of model selection for multivariable regression. Another popular method is "divide and conquer" (DAC), which randomly divides the original dataset into K disjoint and smaller subsets. For each subset, a penalized regression method is conducted. Then, the subset-specific estimates are integrated to obtain the final results. Chen and Xie (2014) applied the DAC strategy to generalized linear models with a penalty term, where a majority voting and averaging operator is used for subset-specific estimates. Wang et al. (2021b) extended the DAC strategy to fit a penalized Cox proportional hazard model. Hong et al. (2022) proposed a screening and one-step linearization infused DAC algorithm to fit sparse logistic regression to large-scale datasets. However, as discussed in Kleiner et al. (2014), although these methods are more generally consistent than traditional bootstrapping, their behavior is sensitive to the choice of resample (or subsample) size, and can be worse for finite samples. When the final objective is simultaneous inference for DAC regression parameters, Tang et al. (2020) proposed a strategy to combine bias-corrected lasso-type estimates by using confidence distributions.

Even for $n \gg p$, it's also useful to select the important variables (Wang et al., 2021a; Fan and Cheng, 2007). Because in real applications, we not only value the model's prediction ability but also the explanation ability. Take the loan business for example, we want to find out important characteristics of customs who are likely in default. In this case, we need inferen-

tial measures, such as confidence interval to help us decide which variables are important. For example, those variables whose confidence interval don't contain zero are important variables. Motivated by the need for a scalable and accurate method for assessing the quality of estimators in variable selection models for massive data, we propose a novel framework called the bag of little bootstraps for variable selection (BLBVS). The method is inspired by the idea of the bag of little bootstraps (BLB) method (Kleiner et al., 2014), which incorporates features of both subsampling and bootstrapping to yield a robust and computationally efficient assessment. Instead of directly resampling from the original dataset, the BLB method first generates subsets of reduced size from the original dataset without replacement and then generates resamples from the subsets with replacement. We combine BLB with variable selection by conducting a penalized optimization for each bootstrap resample of each subset. Further, a ridge hybrid procedure is followed after selecting variables to achieve a sparser and more accurate model. Finally, estimators on all resamples are integrated to give variable selection and assessment results. The computational cost of BLBVS is reduced because the number of distinct points in each bootstrap resample is the same as the reduced size of the subset. Moreover, because BLBVS estimates parameters and assesses the quality for each bootstrap resample independently, it can be easily implemented using distributed computing. Thus, BLBVS has significant computational advantages, especially for massive data. Simulations and applications show that BLBVS performs well in variable selection and is consistently more robust and faster than BootVS.

The remainder of this paper is organized as follows. In Section 2, we first formalize our statistical setting and notation, and propose the BLBVS method. Then we develop the BLBVS-ridge hybrid framework to achieve sparser and more accurate estimations in variable selection models. After that, we compare BLBVS with the existing BootVS algorithm in terms of computational characteristics. In Section 3, we elucidate the performance of BLBVS via various simulation studies and compare them with those of BootVS. The results show that BLBVS has a faster convergence rate with a better ability to select variables than BootVS does. Section 4 implements a large-scale BLBVS on a distributed computing system and presents results illustrating the superior computational performance of BLBVS on massive datasets. In Section 5, we apply BLBVS to two real datasets, one is an open dataset from UCI, and the other is the aforementioned default-risk predicting dataset. We conclude in Section 6.

2 Methodology

In this section, we first propose our novel BLBVS framework, followed by the ridge hybrid procedure. Then we compare our method with the existing BootVS algorithm in terms of their computational characteristics.

2.1 Variable Selection via Bag of Little Bootstraps

Assume that we have independent and identically distributed observations $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, where $\boldsymbol{x}_i$ is a p-dimensional vector and y_i is a response variable. The BLB method proposed by Kleiner et al. (2014) aims to assess the quality of estimators in generalized linear regression for massive datasets. Now, we propose a novel BLBVS framework by combining the BLB and variable selection models. Specifically, we first randomly select s subsets (also called bags or modules) of smaller size $b = n^\gamma$, where $\gamma \in (0, 1)$ is an exponential factor controlling the bag size, without replacement from $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$. For each subset, let $\mathcal{I} \subset \{1, \ldots, n\}$ be the corresponding index set such that $|\mathcal{I}| = b$ and $\hat{F}(t) = b^{-1} \sum_{i \in \mathcal{I}} I\{(\boldsymbol{x}_i^\top, y_i)^\top \leqslant t\}$ be the empirical distribution weighting

each data point with b^{-1}, where t is an arbitrary vector with length $p+1$ and $I\{\cdot\}$ is the indicator function. Then for each subset, generate r bootstraps with replacement resamples of size n (the same size as the original dataset) from $\hat{F}$. One can see that although the size of each bootstrap resample is n, it contains, at most, b distinct data points. Thus, fitting a variable selection model based on this weighted data directly requires much less computational time and storage.

The generalized linear model is formulated as $E(Y|\boldsymbol{X}=\boldsymbol{x}) = g^{-1}(\boldsymbol{\beta}^\top \boldsymbol{x})$, where $\boldsymbol{\beta} \in \mathbb{R}^p$ is a vector of parameters and $g(\cdot)$ is a link function. For example, in linear regression, the link function is chosen as $g(u) = u$, and in a logistic regression model for binary responses, the link function is $g(u) = \ln(u/(1-u))$.

For variable selection models, we consider the optimization problem

$$\min_{\boldsymbol{\beta}} \{\mathcal{L}(Y, \boldsymbol{X}; \boldsymbol{\beta}) + \lambda \cdot \mathcal{S}(\boldsymbol{\beta})\}, \tag{1}$$

where $\mathcal{L}(Y, \boldsymbol{X}; \boldsymbol{\beta})$ is a pre-defined loss function usually set as the negative log-likelihood, λ is the tuning parameter controlling the sparsity of the model, and $\mathcal{S}(\boldsymbol{\beta})$ is the corresponding regularization term, such as the lasso (Tibshirani, 1996) and group lasso (Yuan and Lin, 2006; Meier et al., 2008) penalties. For example, in linear regression, we have $\mathcal{L}(Y, \boldsymbol{X}; \boldsymbol{\beta}) = \sum_{i=1}^n (y_i - \boldsymbol{\beta}^\top \boldsymbol{x}_i)^2$, and for logistic regression models, it holds that $\mathcal{L}(Y, \boldsymbol{X}; \boldsymbol{\beta}) = \sum_{i=1}^n \{-y_i \boldsymbol{\beta}^\top \boldsymbol{x}_i + \log[1 + \exp(\boldsymbol{\beta}^\top \boldsymbol{x}_i)]\}$. Note that in (1), when λ increases, the penalization is intensified and fewer variables are selected.

Popular variable selection models include the lasso and group lasso. Considering the l_1 lasso penalty (Tibshirani, 1996), (1) can be written as

$$\min_{\boldsymbol{\beta}} \{\mathcal{L}(Y, \boldsymbol{X}; \boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_1\}.$$

However, the lasso model selects individual variables without any group structure.

To select or abandon a group of variables (e.g. dummy variables of a categorical predictor) simultaneously, the group lasso is proposed (Meier et al., 2008). Assuming there are G groups of predictors, the group lasso model is defined as

$$\min_{\boldsymbol{\beta}} \left\{ \mathcal{L}(Y, \boldsymbol{X}; \boldsymbol{\beta}) + \lambda \sum_{g=1}^{G} \|\boldsymbol{\beta}_{\mathcal{I}_g}\|_2 \right\}, \tag{2}$$

where $\mathcal{I}_g$ denotes the index of the g-th group of variables for $g = 1, \ldots, G$. Denote $\hat{\boldsymbol{\beta}}_{ij} = (\hat{\beta}_{ij}^{(1)}, \ldots, \hat{\beta}_{ij}^{(p)})^\top$, which is the of the optimization problem (1), as the vector of estimators of the j-th bootstrap resample in the i-th subset and $\hat{\boldsymbol{\xi}}_i = (\hat{\xi}_i^{(1)}, \ldots, \hat{\xi}_i^{(p)})^\top$ as the assessment of the quality of the estimators in the i-th subset, which consists of a summary of the distribution of $\hat{\boldsymbol{\beta}}_{ij}$, $j = 1, \ldots, r$ (i.e., standard deviation or confidence region) based on the bootstrap resamples. Then the overall assessment $\hat{\boldsymbol{\xi}} = (\hat{\xi}^{(1)}, \ldots, \hat{\xi}^{(p)})^\top$ of parameter estimation is calculated by averaging the results of s subsets, that is

$$\hat{\boldsymbol{\xi}} = s^{-1} \sum_{i=1}^{s} \hat{\boldsymbol{\xi}}_i. \tag{3}$$

Regarding variable selection, denote $\boldsymbol{\mathcal{Q}}(\hat{\boldsymbol{\beta}}_{ij}) = (I(\hat{\beta}_{ij}^{(1)} \neq 0), \ldots, I(\hat{\beta}_{ij}^{(p)} \neq 0))^\top$ as a binary vector of length p where the l-th coordinate indicates whether $\hat{\beta}_{ij}^{(l)}$ is non-zero. Note that $\boldsymbol{\mathcal{Q}}(\hat{\boldsymbol{\beta}}_{ij})$ is calculated to represent the selection results of $\hat{\boldsymbol{\beta}}_{ij}$ for the i-th bootstrap resample in the j-th

Algorithm 1: Bag of Little Bootstraps Variable Selection (BLBVS).

Input: $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$: original observed dataset
b: size of each subset
s: number of subsets
r: number of bootstrap resamples for each subset
Output: the assessment of quality $\hat{\boldsymbol{\xi}}$ and the selection proportion result $\boldsymbol{\mathcal{P}}$
for $i = 1$ *to* s **do**
  1. Randomly select a subset $\{(\boldsymbol{x}_l, y_l), l \in \mathcal{I}\}$ with size b from the original dataset without replacement where $\mathcal{I} \subset \{1, 2, \ldots, n\}$.
  for $j = 1$ *to* r **do**
    2. Let $\hat{F}$ be the empirical distribution, assigning a weight of b^{-1} to each pair $(\boldsymbol{x}_l, y_l), l \in \mathcal{I}$. Generate a bootstrap resample $\{(\boldsymbol{x}_l^*, y_l^*), l = 1, 2, \ldots, n\}$ of size n from $\hat{F}$ with replacement.
    3. Apply a variable selection model (e.g., lasso or group lasso) to $\{(\boldsymbol{x}_l^*, y_l^*), l = 1, 2, \ldots, n\}$, which contains at most b distinct points, to calculate the estimators of parameters $\hat{\boldsymbol{\beta}}_{ij}$.
  end
  4. Calculate the assessment of quality $\hat{\boldsymbol{\xi}}_i$ (e.g., standard deviation or confidence region) based on $\hat{\boldsymbol{\beta}}_{i1}, \ldots, \hat{\boldsymbol{\beta}}_{ir}$.
end
5. Compute the final assessment $\hat{\boldsymbol{\xi}}$ by (3).
6. Calculate the selection proportion $\boldsymbol{\mathcal{P}}$ for all predictors based on (4). Select the predictors with a selection proportion larger than the pre-defined cut-off c.

subset. To obtain the final variable selection result, a voted criterion is similarly applied. Suppose the voting weight of each bootstrap resample in any subset is identical, then the final selection proportion vector $\boldsymbol{\mathcal{P}}$ of all predictors is defined as

$$\boldsymbol{\mathcal{P}} = \frac{\sum_{i=1}^{s} \sum_{j=1}^{r} \mathcal{Q}(\hat{\boldsymbol{\beta}}_{ij})}{s \cdot r}. \tag{4}$$

The l-th predictor would be selected if it satisfies $\mathcal{P}^{(l)} > c$, where $\mathcal{P}^{(l)}$ is the l-th coordinate of $\boldsymbol{\mathcal{P}}$ for $l = 1, \ldots, p$ representing the corresponding selection proportion and the cut-off c can be determined similarly by the definition of majority in the specific study. Algorithm 1 gives the detailed BLBVS procedure and the workflow is shown in Figure 1.

2.2 BLBVS-Ridge Hybrid

As shown in the literature (Meinshausen, 2007; Meier et al., 2008), the models selected by lasso or group lasso are probably larger than the true model with redundant variables. Moreover, because of the shrinkage effect of the l_1 penalty, the estimators of significant predictors tend to be smaller than the true values. To solve these issues, Efron et al. (2004) proposed the lars-ordinary least squares hybrid, which uses lasso to select variables first and then fits ordinary linear regression to the selected predictors. To derive smaller and more robust models with good prediction performance, Meinshausen (2007); Meier et al. (2008) proposed the relaxed lasso (group lasso) model to include a l_2 penalty. Inspired by this idea, we propose the BLBVS-ridge hybrid model. We first use a variable selection model to select predictors and then fit ridge regression to the selected predictors to achieve a more accurate and robust model for each bootstrap resample of each subset.

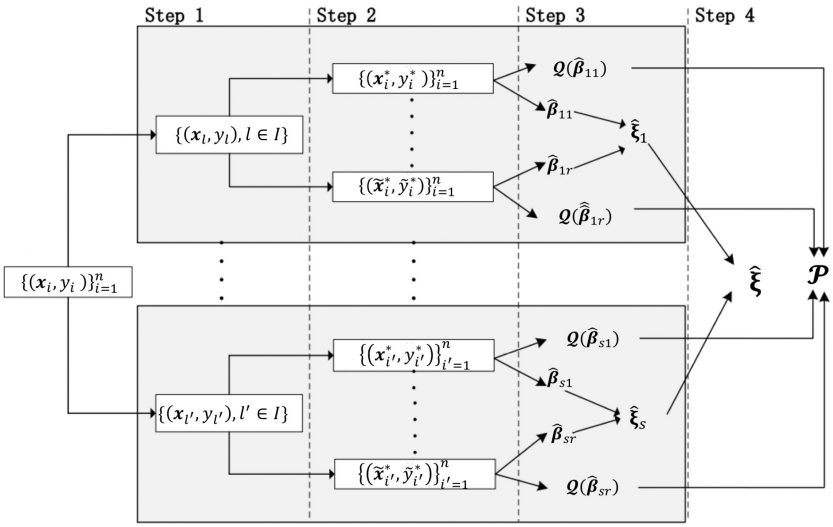


Figure 1: BLBVS workflow: the original dataset has n data points; (Step 1) s subsets of size b ($b < n$) are randomly generated without replacement; (Step 2) r bootstrap resamples of size n are generated from the subset with replacement, containing b distinct points at most; (Step 3) estimators and their corresponding indicators are calculated on each bootstrap resample in each subset, then the assessment of the quality of parameter estimation is computed in each subset; (Step 4) the overall assessment is derived by taking the average value across s subsets, and variables are selected according to the selection proportion $\mathcal{P}$ and a pre-defined threshold c.

Given λ, denote the set of active predictors by $\mathcal{I}_\lambda$ and let $\hat{\mathcal{M}}_\lambda = \{\boldsymbol{\beta} \in \mathbb{R}^p | \beta^{(i)} = 0 \text{ for } i \notin \mathcal{I}_\lambda\}$ be the set of possible parameter vectors of the corresponding submodel. The BLBVS-ridge hybrid estimator is defined as

$$\hat{\boldsymbol{\beta}}_{\lambda,\kappa} = \underset{\boldsymbol{\beta} \in \hat{\mathcal{M}}_\lambda}{\operatorname{argmin}} \left\{ \mathcal{L}(Y, \boldsymbol{X}, \boldsymbol{\beta}) + \kappa \|\boldsymbol{\beta}\|_2^2 \right\}, \tag{5}$$

where κ is the weight for the l_2 penalty. When $\kappa = 0$, the procedure is similar to the lars-ordinary least squares hybrid in (Efron et al., 2004). Optimization problem (5) can be solved with numerical algorithms such as the Newton method and the coordinatewise approach (Genkin et al., 2007) for massive data.

2.3 BLBVS Versus BootVS

As described in Section 1, Bootstrapping might be one of the most commonly used methods to assess the quality of the estimators in variable selection models (Shao, 1996; Chatterjee and Lahiri, 2011). Assume we randomly generate m bootstrap resamples with replacement from the original dataset $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$. Denote $\hat{\boldsymbol{\beta}}_i, i = 1, 2, \ldots, m$ as the vector of estimators for the i-th bootstrap resample. Then the assessment of the quality of the estimators, defined as $\hat{\boldsymbol{\xi}}$ (i.e., standard deviation or confidence regions), is calculated based on the empirical distribution of $\hat{\boldsymbol{\beta}}_i, i = 1, 2, \ldots, m$. As for variable selection, a voted criterion similar to the decision tree and random forest theories (Breiman, 2001; Lin and Jeon, 2006) is applied. Denote $\boldsymbol{\mathcal{Q}}(\hat{\boldsymbol{\beta}}_i) = (I(\hat{\beta}_i^{(1)} \neq 0), \ldots, I(\hat{\beta}_i^{(p)} \neq 0))^\top$ as the binary vector indicating whether the l-th coefficient is non-zero in

Algorithm 2: Bootstrap Variable Selection (BootVS).

Input: $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$: original observed dataset
m: number of bootstrap resamples
Output: the assessment of quality $\hat{\boldsymbol{\xi}}$ and the selection proportion result $\boldsymbol{\mathcal{P}}$
for $i = 1$ *to* m **do**
1. Let $\hat{F}$ be the empirical distribution assigning a weight of n^{-1} to each pair $(\boldsymbol{x}_l, y_l), l = 1, \ldots, n$. Generate a bootstrap resample with replacement $\{(\boldsymbol{x}_l^*, y_l^*)\}_{l=1}^n$ of size n from $\hat{F}$.
2. Apply the variable selection model (e.g., lasso or group lasso) (1) to $\{(\boldsymbol{x}_l^*, y_l^*)\}_{l=1}^n$ to calculate the estimators of parameters $\hat{\boldsymbol{\beta}}_i$.

end

3. Calculate the assessment of quality $\hat{\boldsymbol{\xi}}$ (e.g., standard deviation or confidence region) based on the empirical distribution of $\hat{\boldsymbol{\beta}}_1, \ldots, \hat{\boldsymbol{\beta}}_m$.
4. Calculate the selection proportion $\boldsymbol{\mathcal{P}}$ for all predictors based on (6). Select the predictors with a selection proportion larger than the pre-defined cut-off c.

the i-th bootstrap resample. Suppose the voting weight of each bootstrap resample is identical, then the final selection proportion vector $\boldsymbol{\mathcal{P}}$ of all predictors is defined as

$$\boldsymbol{\mathcal{P}} = \frac{\sum_{i=1}^m \boldsymbol{\mathcal{Q}}(\hat{\boldsymbol{\beta}}_i)}{m}. \tag{6}$$

The l-th predictor is selected if it satisfies $\mathcal{P}^{(l)} > c$, where $\mathcal{P}^{(l)}$ is the l-th coordinate of $\boldsymbol{\mathcal{P}}$ for $l = 1, \ldots, p$ representing the corresponding selection proportion, and the cut-off c can be determined by the definition of majority in specific studies. The detailed BootVS procedure is given in Algorithm 2. The difference between BLBVS and BootVS could be ideally described by computational complexity and storage occupation. For BootVS, as mentioned by (Tibshirani and Efron, 1993), the number of distinct points in each bootstrap resample from the original dataset is approximately $0.632n$. Thus, the computational complexity is $m \cdot O(f(0.632n, p))$. For BLBVS, as one can see, instead of n data points in the original dataset, the bootstrap resample from each subset contains, at most, b distinct points. And the computational complexity is $r \cdot s \cdot O(f(n^\gamma, p))$. Therefore, BLBVS involves much less computation by working directly on the weighted data. It's difficult to prove the computation superiority of BLBVS based on the computational complexity because the form of function f, which depends on the variable selection method and the optimization algorithm, is complicated. We will show computation superiority of BLBVS in different simulation scenario. As for storage occupation, we consider an original dataset of size $n = 10^6$. If we assume each data point occupies 1MB of storage space, then the original dataset would occupy 1TB. Each bootstrap resample in BootVS would occupy approximately 632GB. For BLBVS, the size of each subset is $b = n^\gamma = 3,981$ with $\gamma = 0.6$, and each bootstrap resample from the subset has 3, 981 distinct points. Thus, in BLBVS, a bootstrap resample from a subset only needs approximately 4GB of storage. It is obvious that our proposed method runs much faster and requires much fewer computational resources, especially when the original dataset is extremely large.

3 Simulations

3.1 Variable Selection and Convergence Property

In this section, we compare the statistical performance of BLBVS with that of BootVS via numerical simulations. Specifically, we present the variable selection ability and the convergence rate of both BootVS and BLBVS for the lasso and group lasso models. We also discuss the choice of the cut-off c in variable selection, the number of subsets s, and the number of bootstrap resamples for each subset r, with insightful suggestions.

We consider two different settings: linear regression and logistic regression. For both settings, the data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ are generated independently and identically, where $\boldsymbol{x}_i$ is the p-dimensional vector of predictors and y_i is the response. Note that $y_i \in \mathbb{R}$ is continuous in linear regression and $y_i \in \{0, 1\}$ is binary in logistic regression. All predictors are assumed to be continuous for the lasso model, whereas some categorical predictors are involved in the group lasso model. To assess the quality of estimators, we consider two criteria, standard deviation and a 95% confidence interval length.

To evaluate the convergence rate of the assessment $\hat{\boldsymbol{\xi}}$, we first compute the ground truth $\boldsymbol{\xi}$ of $\hat{\boldsymbol{\beta}}$ by generating 2000 realizations of datasets of size n from the true underlying distribution. For each dataset, we apply a variable selection model (e.g., lasso or group lasso) to estimate $\hat{\boldsymbol{\beta}}$ and then the estimators on all datasets form a high fidelity approximation of the true underlying distribution, which is used to approximate $\boldsymbol{\xi}$. Based on the approximation value of $\boldsymbol{\xi}$, we are able to define the relative error (RE) of $\hat{\boldsymbol{\xi}}$ obtained by BootVS or BLBVS. Denote $\mathcal{I}$ as the set of active predictors, of which the proportion being selected from the 2000 realizations is larger than the pre-defined cut-off c, then the RE is defined as

$$RE = |\mathcal{I}|^{-1} \sum_{i \in \mathcal{I}} \frac{\hat{\xi}^{(i)} - \xi^{(i)}}{\xi^{(i)}}, \tag{7}$$

where $|\cdot|$ represents the cardinality of the set and $\cdot^{(i)}$ is the i-th coordinate of the corresponding vector.

All experiments are implemented and executed using R software (http://www.r-project.org/) on a quad-core processor (MacOS High Sierra system; Intel Core i7 and 2.9GHz CPU; 16GB RAM). As for the tuning parameter λ in the variable selection model (1), we use a grid search in $\{\lambda_{\max}, 0.96\lambda_{\max}, \ldots, 0.96^{100}\lambda_{\max}\}$. For each candidate of λ, we first fit a variable selection model (1) to select significant predictors and then fit the ridge regression (5) to approximate the true underlying model better as stated in Section 2.3. In the ridge regression (5), the weight of the l_2 penalty κ is set as 10^{-5} as suggested by (Kleiner et al., 2014). Then, the Bayesian information criterion (BIC) for each fitted model is calculated and the model with the minimum BIC is finally selected.

For BootVS, we set the number of bootstrap resamples as $m = 500$. For BLBVS, we consider the size of each subset as $b = n^\gamma$ with $\gamma \in \{0.6, 0.7, 0.8, 0.9\}$. The choice of the number of subsets s and the number of bootstrap resamples r is an interesting problem. A basic rule is to set a larger s for a smaller γ as suggested by (Kleiner et al., 2014). From some preliminary experiments, we set $s = 30, 20, 10, 10$ for $\gamma = 0.6, 0.7, 0.8, 0.9$, respectively, and $r = 100$ in all BLBVS runs. More discussions about s and r are presented in Section 3.3.

We first consider the lasso model where the predictors are all continuous without any group structure for both regression and classification scenarios.

Example 1 (Linear Regression with Lasso)**.** *The dataset* $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ *is generated as follows. For* $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, p$, $x_{ij} \sim N(0, 1)$ *and* $y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta} + \epsilon_i$ *with* $\epsilon_i \sim N(0, 1)$. *Set* $p = 35$, $n = 20000$ *and* $\boldsymbol{\beta} = (\mathbf{1}_{12}, \mathbf{0}_3, \mathbf{1}_6, \mathbf{0}_4, \mathbf{1}_8, \mathbf{0}_2)^\top$.

Example 2 (Logistic Regression with Lasso)**.** *The dataset* $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ *is generated as follows. For* $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, p$, $x_{ij} \sim N(0, 1)$ *and* $y_i \sim$ *Bernoulli*$(1/\{1 + \exp(-\boldsymbol{x}_i^\top \boldsymbol{\beta})\})$. *Set* $p = 15$, $n = 20000$ *and* $\boldsymbol{\beta} = (\mathbf{1}_7, \mathbf{0}_3, \mathbf{1}_3, \mathbf{0}_2)^\top$.

Then we consider the group lasso model by involving categorical predictors. Both regression and classification scenarios are presented.

Example 3 (Linear Regression with Group Lasso)**.** *The dataset* $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ *is generated as follows. For* $i = 1, 2, \ldots, n$, $x_{ij} \sim N(0, 1)$, $j = 1, \ldots, p-5$ *and* x_{ij}, $j = p-4, \ldots, p$ *are dummy variables from two categorical predictors following a discrete uniform distribution with 4 levels and 3 levels, respectively. Note that* $y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta} + \epsilon_i$ *with* $\epsilon_i \sim N(0, 1)$. *Set* $p = 35$, $n = 20000$. *Divide* $\boldsymbol{\beta}$ *into 8 groups with* $\mathcal{I}_1 = \{1, \ldots, 5\}$, $\mathcal{I}_2 = \{6, \ldots, 12\}$, $\mathcal{I}_3 = \{13, \ldots, 15\}$, $\mathcal{I}_4 = \{16, \ldots, 21\}$, $\mathcal{I}_5 = \{22, \ldots, 25\}$, $\mathcal{I}_6 = \{26, \ldots, 30\}$, $\mathcal{I}_7 = \{31, \ldots, 33\}$, $\mathcal{I}_8 = \{34, 35\}$ *and set* $\boldsymbol{\beta} = (\mathbf{1}_5, \mathbf{1}_7, \mathbf{0}_3, \mathbf{1}_6, \mathbf{0}_4, \mathbf{1}_5, \mathbf{1}_3, \mathbf{0}_2)^\top$. *Thus, the predictors in* $\mathcal{I}_1$, $\mathcal{I}_2$, $\mathcal{I}_4$, $\mathcal{I}_6$, $\mathcal{I}_7$ *are significant and the remainder are inactive.*

Example 4 (Logistic Regression with Group Lasso)**.** *The dataset* $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ *is generated as follows. For* $i = 1, 2, \ldots, n$, $x_{ij} \sim N(0, 1)$, $j = 1, \ldots, p-5$ *and* x_{ij}, $j = p-4, \ldots, p$ *are dummy variables from two categorical predictors following a discrete uniform distribution with 4 levels and 3 levels, respectively. Note that* $y_i \sim$ *Bernoulli*$(1/\{1 + \exp(-\boldsymbol{x}_i^\top \boldsymbol{\beta})\})$. *Set* $p = 15$, $n = 20000$. *Divide* $\boldsymbol{\beta}$ *into 5 groups with* $\mathcal{I}_1 = \{1, \ldots, 3\}$, $\mathcal{I}_2 = \{4, \ldots, 7\}$, $\mathcal{I}_3 = \{8, \ldots, 10\}$, $\mathcal{I}_4 = \{11, \ldots, 13\}$, $\mathcal{I}_5 = \{14, 15\}$ *and set* $\boldsymbol{\beta} = (\mathbf{1}_3, \mathbf{1}_4, \mathbf{0}_3, \mathbf{1}_3, \mathbf{0}_2)^\top$. *Thus, the predictors in* $\mathcal{I}_1$, $\mathcal{I}_2$, $\mathcal{I}_4$ *are significant and the remainder are inactive.*

Figures 2 show the selection results for BLBVS with different values of $\gamma \in \{0.6, 0.7, 0.8, 0.9\}$ and for BootVS for Examples 1. For BLBVS, our results show that the selection becomes more and more accurate as γ increases. The BootVS selection performance is superior to that of BLBVS, especially when γ is small. As the selection proportion is computed based on the information of each bootstrap resample, it would be more accurate if the resample contains more distinct data points, which is the case for BootVS and BLBVS with a larger γ. Setting $c = 0.5$, the selection results are same for BootVS and BLBVS with $\gamma = 0.8, 0.9$ via the voted criterion. In Section 3.2, more discussions about c are presented. One can see that by choosing a proper cut-off c, BLBVS is a good method for selecting significant predictors.

The convergence properties of BootVS and BLBVS are shown in Figure 3. For BootVS method, let t_i, $1 \leqslant i \leqslant m$ be the computing time for the i-th bootstrapping resample. So we have time vector $T_{boot} = \{t_1, \sum_{i=1}^{2} t_i, \ldots, \sum_{i=1}^{m} t_i\}$ as x-axis. For BLBVS method, let t_{ij}, $1 \leqslant i \leqslant s$, $1 \leqslant j \leqslant r$ be the computing time for the i-th subset, j-th bootstraping resample. So we have time vector $T_{BLB} = \{\sum_{j=1}^{r} t_{1j}, \sum_{i=1}^{2}\sum_{j=1}^{r} t_{ij}, \ldots, \sum_{i=1}^{s}\sum_{j=1}^{r} t_{ij}\}$ as x-axis. Because BootVS takes much more time than BLBVS, we only show results for the first hundred seconds, when in fact BootVS continues running after this time point. One can see that for both linear regression and logistic regression, BLBVS converges to a lower RE significantly faster than BootVS in most situations. For BLBVS, although the differences are small for different values of γ, we can still see that BLBVS performs best with $\gamma = 0.7, 0.8$. For $\gamma = 0.6$, there are only $b = n^\gamma = 380$ distinct points

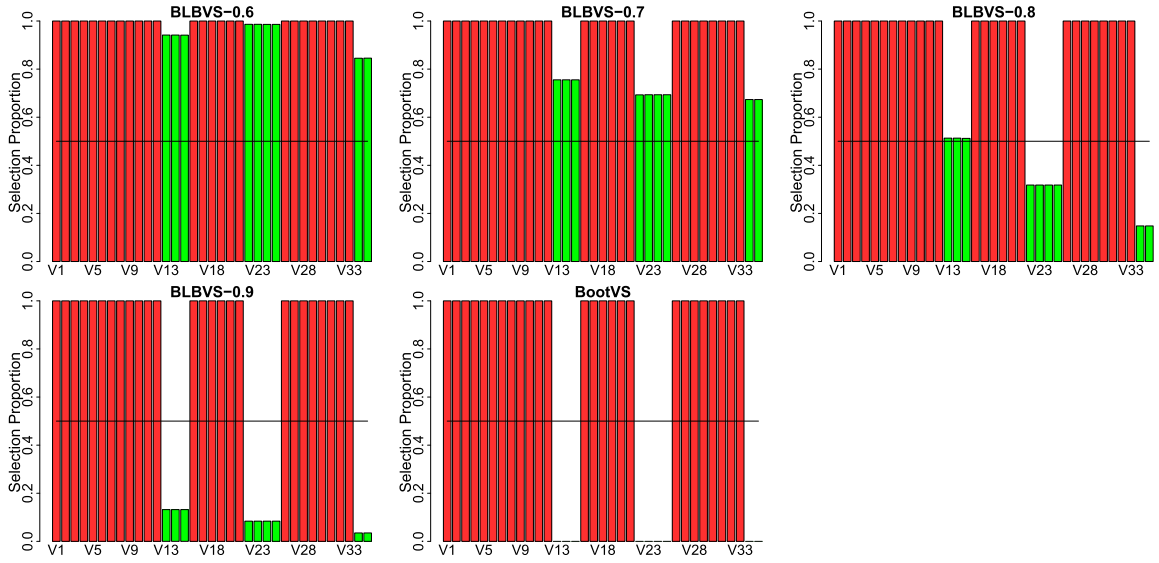


Figure 2: Results of variable selection in Example 1 for BLBVS with $\gamma = 0.6, 0.7, 0.8, 0.9$ and BootVS. The variables in red are active and those in green are inactive. Horizontal line refers to 50% selection proportion.

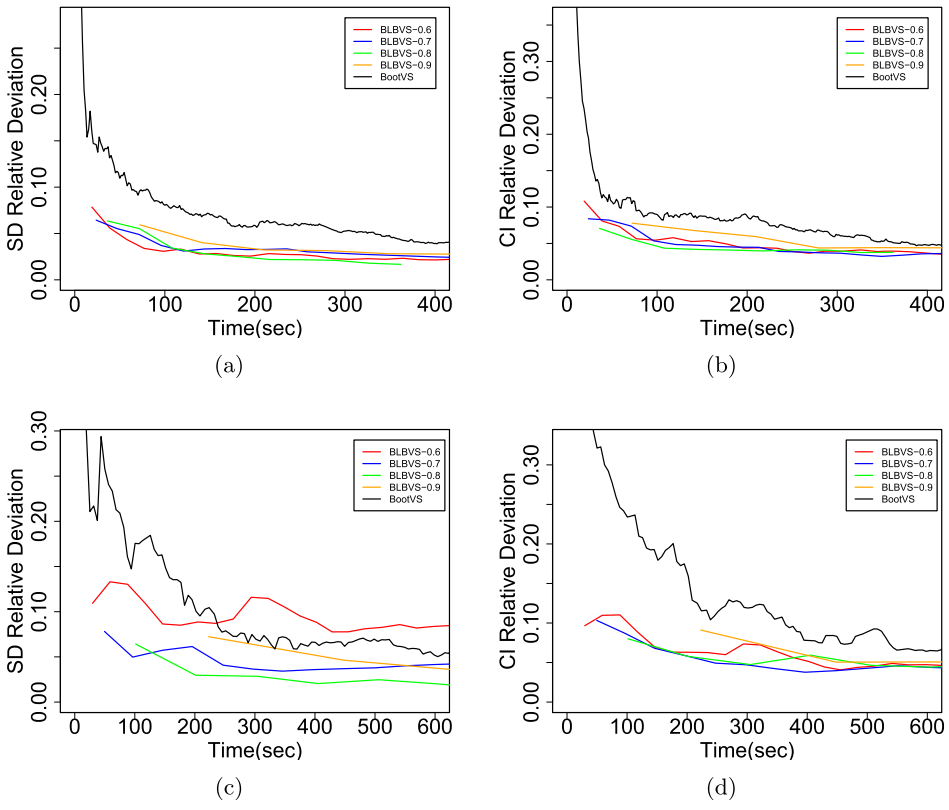


Figure 3: Relative error versus processing time for lasso: (a) standard deviations in Example 1; (b) 95% confidence interval lengths in Example 1; (c) standard deviations in Example 2; (d) 95% confidence interval lengths in Example 2.

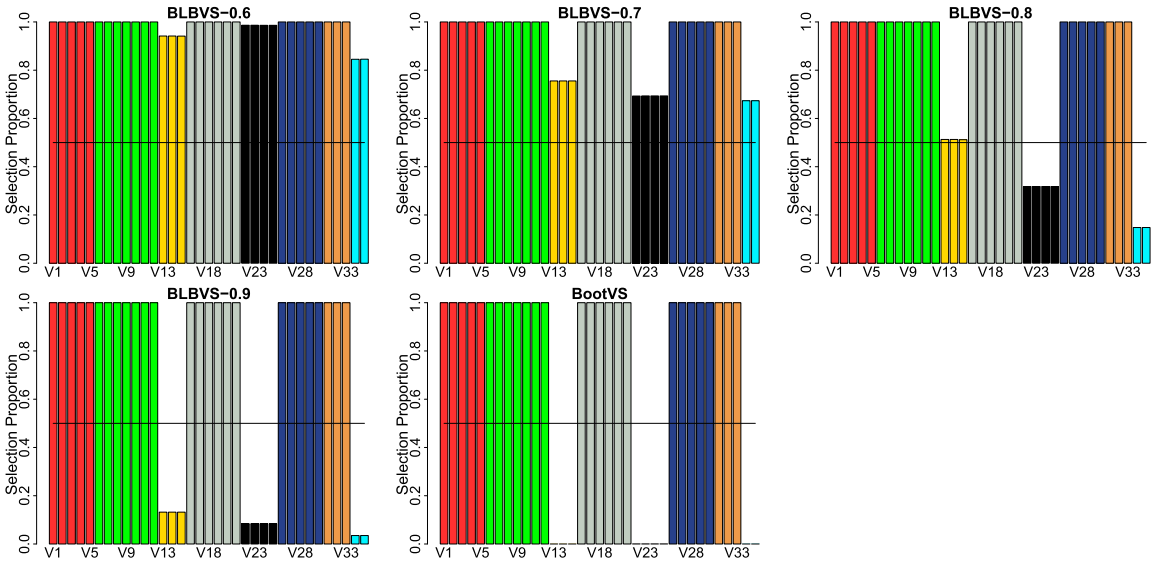


Figure 4: Results of variable selection in Example 3 for BLBVS with $\gamma = 0.6, 0.7, 0.8, 0.9$ and BootVS. Variables from the same group share the same color. Horizontal line refers to 50% selection proportion.

in a bootstrap resample, leading to the poor performance of the convergence rate. Whereas for $\gamma = 0.9$, b is large and thus the computation is consuming. BLBVS achieves a good balance between the diversity of the resample and computational efficiency with $\gamma = 0.7, 0.8$.

Figures 4 show the selection results for BLBVS with different values of $\gamma \in \{0.6, 0.7, 0.8, 0.9\}$ and BootVS for Examples 3. One can see that BLBVS fails in variable selection with $\gamma = 0.6$. As γ increases, the selection results become more accurate. The BootVS selection performance is superior to that of BLBVS, especially when γ is small. Setting $c = 0.5$, the selection results are same for BootVS and BLBVS with $\gamma = 0.9$. Selection results for Example 2 and Example 4 are in supplementary material. In Section 3.2, we further show that BLBVS selects variables well by choosing a proper cut-off c.

The convergence properties of BootVS and BLBVS are shown in Figure 5. One can see that for both linear regression and logistic regression with group lasso, BLBVS converges to a lower RE significantly faster than BootVS does, in most situations. For logistic regression with group lasso, BLBVS has an unsatisfying performance with $\gamma = 0.6$, whereas the performance is still stable for a larger γ. Combined with Figure 3, one can see that the problem is more difficult in classification than regression. Furthermore, involving categorical predictors, which is the case with group lasso, also brings some challenges to variable selection models. In Examples 1–4, we suggest setting $\gamma = 0.7, 0.8$ for our proposed BLBVS method, under which BLBVS selects variables well and converges much faster than BootVS does.

3.2 Choice of the Cut-off c

In this subsection, we discuss the choice of the cut-off c and show that by choosing a proper c, BLBVS selects significant variables very well.

After we have calculated the proportion vector $\boldsymbol{\mathcal{P}}$, we select those variables with $\mathcal{P}^{(l)} > c$, where c can be determined by the definition of majority in specific problems. It is obvious that

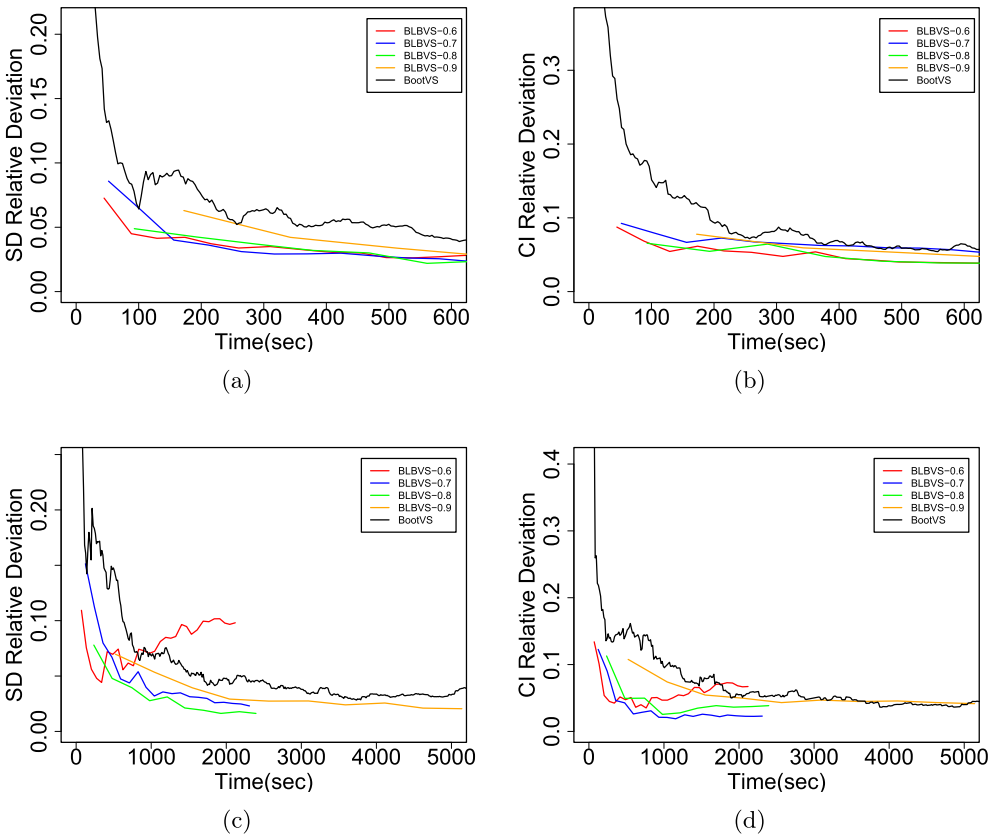


Figure 5: Relative error versus processing time for group lasso: (a) standard deviations in Example 3; (b) 95% confidence interval lengths in Example 3; (c) standard deviations in Example 4; (d) 95% confidence interval lengths in Example 4.

the choice of cut-off c will affect our selection results, and thus the accuracy of the selection procedure. The definition of majority can be different in different situations. Here, we propose a procedure to determine the value of c.

When the proportion vector $\boldsymbol{\mathcal{P}} \in \mathbb{R}^p$ is finally decided, we sort them in descending order, denoted as $\boldsymbol{\mathcal{P}}_{desc}$. And we plot $\boldsymbol{\mathcal{P}}_{desc}$ against the sequence of number of variables $\{1, 2, \ldots, p\}$. At certain point there is a sharp drop in the curve, giving an angle in the graph. Then the interval between the breakpoints is the proper range of cut-off c. Taking Example 1 where a linear regression with lasso is conducted for illustration, for BLBVS with $\gamma = 0.6$, when the number of variables increase from 26 to 27, the corresponding selection proportion decrease from nearly 1 to 0.8. So the proper range of c is $[0.8, 1.0)$ in this case. Similarly, the proper range of c is $[0.6, 1)$ for BLBVS with $\gamma = 0.7$ in Example 1. From Figure 6, one can see again that the range of c is broader with γ increasing. The selection results are more accurate for a larger γ as the bootstrap resample contains more distinct points. Moreover, BootVS is superior to BLBVS in variable selection by achieving a broader range of c. Moreover, it is more difficult to select variables in the classification scenario and the group lasso model, especially when γ is small. However, choosing a proper c value, BLBVS with a large γ is able to select the true model.

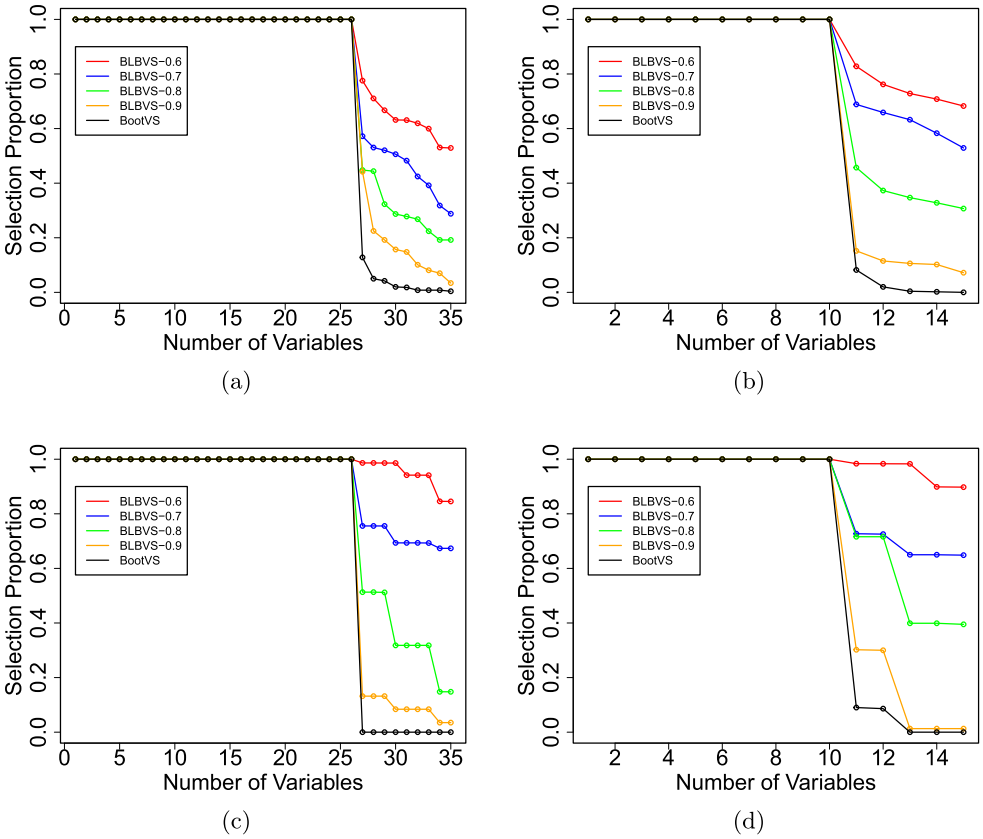


Figure 6: Proportion results sorted by descending order: (a) Example 1; (b) Example 2; (c) Example 3; (d) Example 4.

3.3 Choice of s and r

In the BLBVS framework, the number of subsets s and the number of bootstrap resamples r are given by the users. As stated above, one basic rule is to set a larger s for a smaller γ. As suggested by (Kleiner et al., 2014), one can also view s and r as tuning parameters, thus the value with best results may be chosen. In our simulations, we suggest setting $s = 30, 20, 10, 10$ for $\gamma = 0.6, 0.7, 0.8, 0.9$, respectively, and $r = 100$ in all runs.

To provide further insights into the influence of s and r, we apply a grid search for the linear regression with lasso described in Example 1 and the logistic regression with group lasso described in Example 4 as illustrations. Figure 7 gives the relative errors of both the standard deviations and 95% confidence region lengths achieved by BLBVS with $b = n^{0.7}$ for various values of s and r pairs. Note that, except for the smallest values of s and r, our proposed BLBVS achieves low relative errors for all pairs of s and r. One can see again that the classification problem with group lasso is much more difficult. In these two cases, selecting $s \geqslant 10$ and $r \geqslant 50$ is sufficient for the relative errors to converge.

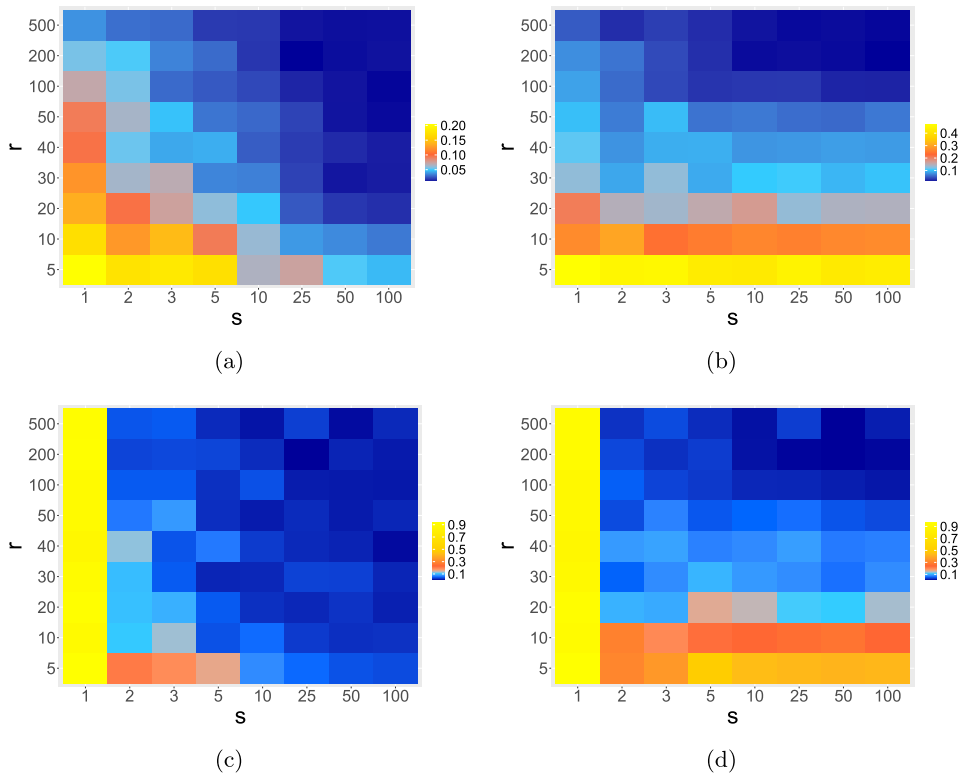


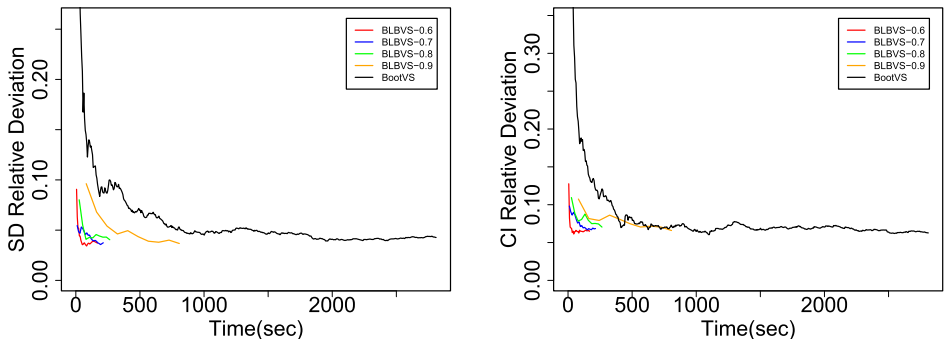
Figure 7: Relative errors for different s's and r's: (a) standard deviations in Example 1; (b) 95% confidence interval lengths in Example 1; (c) standard deviations in Example 4; (d) 95% confidence interval lengths in Example 4.

4 Scalability and Computational Analysis

In Section 3, we focus on the statistical performance of BLBVS and BootVS including variable selection and convergence properties, although the simulations also provide some insightful computational scalability results. One can see that when there is only one compute node involved, BLBVS converges much faster than BootVS does because there are only b distinct points in each bootstrap resample, which is much smaller than the size of the original dataset.

The processing and storage capabilities of individual processors or compute nodes bring great challenges to modern algorithms. One popular approach is the use of parallel and distributed computing. To use distributed computing in BootVS, the following procedure is usually followed: generating bootstrap resamples, assigning them to a cluster of compute nodes, and estimating parameters on each resample across the cluster simultaneously. This approach remains quite problematic. On one hand, the estimation of each resample requires the use of the entire cluster of compute nodes, and bootstrapping repeatedly incurs the cost of repeatedly communicating intermediate data among nodes. On the other hand, when the size of the bootstrap resample exceeds the storage capability of the available memory, BootVS needs to read the resampling set from disc, which is much slower than reading from memory.

By contrast, one notable advantage of BLBVS is the computational scalability for massive datasets. For BLBVS, the estimation procedures for subsets and resamples are independent and can be conducted by different individual compute nodes simultaneously. Specifically, one single



(a) (b)

Figure 8: Relative error versus processing time for the large dataset: (a) standard deviations; (b) 95% confidence intervals length.

compute node can be used for one subset and then intra-node parallelism could be applied across different resamples generated from that subset. Note that there are only b distinct points in each subset, and BLBVS only needs a single read of the original dataset from disc. After that, the generated subsets and resamples can be stored in memory. Therefore, BLBVS performs better on reducing the total computational cost and allowing better applications of parallel and distributed computing resources than BootVS does.

We now compare the performance of BootVS and BLBVS by simulating a much larger dataset on a distributed computing platform. The setting of this experiment is the same as Example 1 in Section 3 except for $n = 2 \times 10^5$. We use a compute machine with a cluster of 4 work nodes, each has 16 GB of memory and 6 CPU (AMD 6344) cores where the total memory of the cluster is 64 GB. The full dataset is partitioned between the four compute nodes. The results of the experiment are shown in Figure 8. For this dataset, we compare the performance of the proposed BLBVS method with different values of $\gamma \in \{0.6, 0.7, 0.8, 0.9\}$ and that of BootVS. From experiments in Section 3, we suggest setting $s = 30, 20, 10, 10$ for $\gamma = 0.6, 0.7, 0.8, 0.9$, respectively, and $r = 50$ in all runs. One can see that, especially for $\gamma = 0.6, 0.7, 0.8$, BLBVS converges much faster, and thus has a significant computational advantage over BootVS for this large dataset.

5 Real Data Analysis

In this section, we apply our proposed BLBVS method to two real datasets. The first dataset, named the Bike Sharing Dataset, is from the University of California at Irvine (https://archive.ics.uci.edu/ml/datasets/bike+sharing+dataset) and contains the hourly count of rental bikes between 2011 and 2012 in the Capital bikeshare system with the corresponding weather and seasonal information. There are 17389 samples and 15 covariates containing both continuous and categorical predictors. The response is the count of total rental bikes including both casual and registered. We first utilize a log transformation to the response variable to suit the linear regression. Moreover, we transform the categorical variable "weathersit" with four levels to a binary variable, taking 1 if the weather is clear and 0, otherwise. The linear regression is conducted for this dataset and the group lasso penalty is included to select important variables as there are categorical predictors.

The second dataset is the Lending Club Dataset. This dataset is available at the Lending Club (LC) website (https://www.lendingclub.com/info/download-data.action). We retrieve data for the period between January 2016 and the third quarter of 2019. The dataset contains personal information (e.g., home ownership, LC assigned loan grade) and financial information (e.g., number of tax liens, total collection amounts ever owed, number of mortgage accounts, number of bankcard accounts, credit balance, number of derogatory public records). We only include the observations whose loan status is "Fully Paid," "Default," or "Charged off." We merge "Charged off" and "Default," meaning that people in this category defaulted loans. After data preprocessing, there are 791332 samples and 40 covariates containing both continuous and categorical predictors. To predict loan defaults, logistic regression is conducted for this dataset and the group lasso penalty is included to select important variables as there are categorical predictors.

For these two datasets, we compare the performance of the proposed BLBVS method with different values of $\gamma \in \{0.6, 0.7, 0.8, 0.9\}$ and that of BootVS. From experiments in Section 3, we suggest setting $s = 30, 20, 10, 10$ for $\gamma = 0.6, 0.7, 0.8, 0.9$, respectively, and $r = 50$ in all runs. The selection results show that X_3, X_4, X_6, X_9, X_{12}, X_{13}, X_{15} (season, year, hr, workingday, atemp, hum, casual) are important variables for the first dataset. For the second dataset, we select 14 important variables, including the LC assigned loan grade, installment (the monthly payment owed by the borrower if the loan originates), home ownership, total credit revolving balance, number of mortgage accounts, and so on.

Because of the absence of the true underlying distributions of the estimators, it is impossible to calculate the relative error defined in (7). To compare the performance of the convergence rates, we calculate the standard deviation of the assessment (or the 95% confidence interval length) to evaluate the stability. Figure 9 shows the standard deviations of assessments $\hat{\boldsymbol{\xi}}$ versus the processing time of BootVS and BLBVS (with different values of γ) for these two datasets. One can see that our BLBVS method is much more stable than BootVS because it is faster at achieving a lower standard deviation of assessments regardless of the value of γ. Cut-off c graphs for bike sharing dataset and lending club dataset are in supplementary material.

6 Conclusions

This study proposes a novel BLBVS framework to obtain a computationally efficient method for assessing the quality of estimators and selecting significant predictors in generalized linear models. The method is to first generate subsets with reduced size from the original dataset without replacement and perform bootstrap resampling for each subset. Then, a variable selection model, such as lasso and group lasso, is applied for each bootstrap resample. Furthermore, to achieve a sparser and more accurate model, a ridge hybrid is followed. Various numerical studies and real data analyses including both regression and classification are presented to show that our proposed BLBVS has great accuracy and computational advantages over BootVS.

One of the remarkable characteristics of BLBVS is its computational scalability. On one hand, each bootstrap resample in BLBVS has a reduced size of distinct points and the estimation process can work on the weighted data directly. On the other hand, BLBVS is easily processed with parallel and distributed computing. Considering the limitation of processing and storage capabilities of individual processors or compute nodes, BLBVS is suitable for massive datasets in the big data era.

There are several possible extensions that can be considered. First, this study only discusses

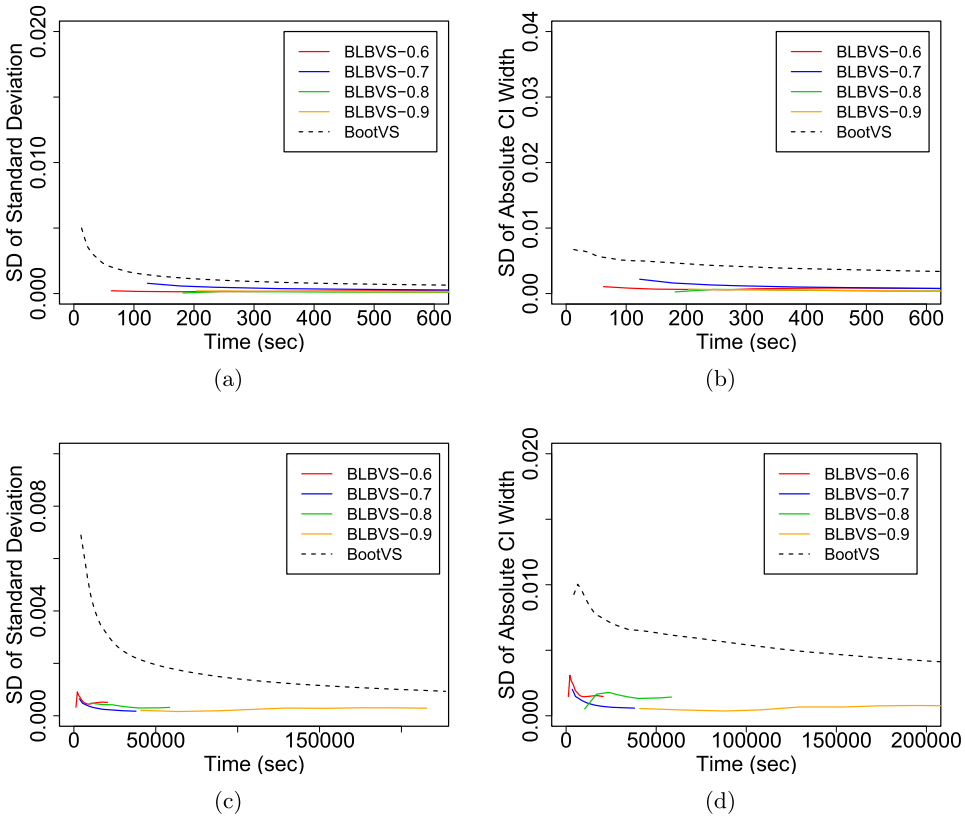


Figure 9: Standard deviation of estimators versus processing time for real data analyses: (a) SD of standard deviations for the Bike Sharing Dataset; (b) SD of the 95% confidence interval lengths for the Bike Sharing Dataset; (c) SD of standard deviations for the Lending Club Dataset; (d) SD of the 95% confidence interval lengths for the Lending Club Dataset.

two kinds of penalties in variable selection models. In fact, BLBVS can be naturally extended to other variable selection models with different penalties, such as the mcp (Zhang, 2010), adaptive lasso (Zou, 2006), and smoothly clipped absolute deviation (Fan and Li, 2001) models. Second, to enhance the computational efficiency of BLBVS, its hyperparameter γ can be viewed as a tuning parameter and an effective means of adaptively selecting γ is preferable. Third, when the dataset is high-dimensional (i.e., vast numbers of predictors), it is necessary to extend BLBVS to the feature screening scenario (Fan and Lv, 2008; Li et al., 2012; Xie et al., 2020). Fourth, when there are outliers in the variable selection (Yao and Wang, 2013; Xie et al., 2020), a procedure robust to outliers is required.

Supplementary Material

.zip contains the following files and/or directories:

- /code and data/: Directory that includes code and files necessary to reproduce the numerical results presented in this paper.
- supplementary.pdf: Online supplementary material.

Funding

Dr. Yang Li was supported by Platform of Public Health & Disease Control and Prevention, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China and National Natural Science Foundation of China (71771211).

References

Bickel PJ, Götze F, van Zwet WR (2012). Resampling fewer than n observations: gains, losses, and remedies for losses. In: *Selected Works of Willem van Zwet*, 267–297. Springer.

Breiman L (2001). Random forests. *Machine Learning*, 45(1): 5–32.

Chatterjee A, Lahiri SN (2011). Bootstrapping lasso estimators. *Journal of the American Statistical Association*, 106(494): 608–625.

Chen X, Xie Mg (2014). A split-and-conquer approach for analysis of extraordinarily large data. *Statistica Sinica*, 24: 1655–1684.

De Bin R, Janitza S, Sauerbrei W, Boulesteix AL (2016). Subsampling versus bootstrapping in resampling-based model selection for multivariable regression. *Biometrics*, 72(1): 272–280.

Efron B, Hastie T, Johnstone I, Tibshirani R, et al. (2004). Least angle regression. *Annals of Statistics*, 32(2): 407–499.

Fan J, Li R (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456): 1348–1360.

Fan J, Lv J (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5): 849–911.

Fan TH, Cheng KF (2007). Tests and variables selection on regression analysis for massive datasets. *Data & Knowledge Engineering*, 63(3): 811–819.

Genkin A, Lewis DD, Madigan D (2007). Large-scale bayesian logistic regression for text categorization. *Technometrics*, 49(3): 291–304.

Hong C, Wang Y, Cai T (2022). A divide-and-conquer method for sparse risk prediction and evaluation. *Biostatistics*, 23(2): 397–411.

Kleiner A, Talwalkar A, Sarkar P, Jordan MI (2014). A scalable bootstrap for massive data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(4): 795–816.

Li R, Zhong W, Zhu L (2012). Feature screening via distance correlation learning. *Journal of the American Statistical Association*, 107(499): 1129–1139.

Lin Y, Jeon Y (2006). Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association*, 101(474): 578–590.

Liu L, Gu H, Van Limbergen J, Kenney T (2021). Surf: A new method for sparse variable selection, with application in microbiome data analysis. *Statistics in Medicine*, 40(4): 897–919.

Meier L, Van De Geer S, Bühlmann P (2008). The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1): 53–71.

Meinshausen N (2007). Relaxed lasso. *Computational Statistics & Data Analysis*, 52(1): 374–393.

Meinshausen N, Bühlmann P (2010). Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4): 417–473.

Shao J (1996). Bootstrap model selection. *Journal of the American Statistical Association*, 91(434): 655–665.

Tang L, Zhou L, Song PXK (2020). Distributed simultaneous inference in generalized linear models via confidence distribution. *Journal of Multivariate Analysis*, 176: 104567.

Tibshirani R (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1): 267–288.

Tibshirani RJ, Efron B (1993). An introduction to the bootstrap. *Monographs on Statistics and Applied Probability*, 57: 1–436.

Wang K, Li S, Zhang B (2021a). Robust communication-efficient distributed composite quantile regression and variable selection for massive data. *Computational Statistics & Data Analysis*, 161: 107262.

Wang Y, Hong C, Palmer N, Di Q, Schwartz J, Kohane I, et al. (2021b). A fast divide-and-conquer sparse cox regression. *Biostatistics*, 22(2): 381–401.

Wu CFJ, et al. (1986). Jackknife, bootstrap and other resampling methods in regression analysis. *Annals of Statistics*, 14(4): 1261–1295.

Xie J, Lin Y, Yan X, Tang N (2020). Category-adaptive variable screening for ultra-high dimensional heterogeneous categorical data. *Journal of the American Statistical Association*, 115(530): 747–760.

Yao W, Wang Q (2013). Robust variable selection through mave. *Computational Statistics & Data Analysis*, 63: 42–49.

Yuan M, Lin Y (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1): 49–67.

Zhang CH (2010). Nearly unbiased variable selection under minimax concave penalty. *Annals of Statistics*, 38(2): 894–942.

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