Electronic Research Archive

http://www.aimspress.com/journal/era

ERA, 30(5): 1936–1953.
DOI: 10.3934/era.2022098
Received: 21 March 2022
Revised: 01 April 2022
Accepted: 06 April 2022
Published: 11 April 2022

Research article

Distributed Bayesian posterior voting strategy for massive data

Xuerui Li[1], Lican Kang[2], Yanyan Liu[1,*] and Yuanshan Wu[3]

[1] School of Mathematics and Statistics, Wuhan University, China
[2] Center for Quantitative Medicine Duke-NUS Medical School, Singapore
[3] School of Statistics and Mathematics, Zhongnan University of Economics and Law, China

* **Correspondence:** Email: liuyy@whu.edu.cn.

Abstract: The emergence of massive data has driven recent interest in developing statistical learning and large-scale algorithms for analysis on distributed platforms. One of the widely used statistical approaches is split-and-conquer (SaC), which was originally performed by aggregating all local solutions through a simple average to reduce the computational burden caused by communication costs. Aiming at lower computation cost and satisfactorily acceptable accuracy, this paper extends SaC to Bayesian variable selection for ultra-high dimensional linear regression and builds BVSaC for aggregation. Suppose ultrahigh-dimensional data are stored in a distributed manner across multiple computing nodes, with each computing resource containing a disjoint subset of data. On each node machine, we perform variable selection and coefficient estimation through a hierarchical Bayes formulation. Then, a weighted majority voting method BVSaC is used to combine the local results to retain good performance. The proposed approach only requires a small portion of computation cost on each local dataset and therefore eases the computational burden, especially in Bayesian computation, meanwhile, pays a little cost to receive accuracy, which in turn increases the feasibility of analyzing extraordinarily large datasets. Simulations and a real-world example show that the proposed approach performed as well as the whole sample hierarchical Bayes method in terms of the accuracy of variable selection and estimation.

Keywords: Hierarchical Bayes formulation; massive data; majority-voting; split-and-conquer; Shrinkage prior

1. Introduction

With the rapid growth of information collection, massive datasets with large sample sizes and ultra-high dimensions pose great challenges to data processing capabilities. Often, the data are stored across multiple computing resources. For example, this is often the case when the dataset becomes so large that it is infeasible to store or analyze the whole dataset in a single computing node. The data must be stored in multiple computing nodes, where each contains a subset of the data. Another example is in medical studies since health records are often scattered across many hospitals and it is not feasible to process the data in a central machine due to issues of privacy and ownership. In these situations, the computational efficiency of statistical processing and the cost of communication between computers are two critical problems to be considered. In recent years, various distributed statistical methods have been developed to solve these problems. Among these, split-and-conquer (SaC) [1] has become popular. The key idea of SaC is to first conduct statistical inference on each computing node and then aggregate the local results by simple averaging to produce a final solution. Because SaC is fast and easy to implement, it has been extended to various settings under massive data scenarios, such as distributed bootstrapping [2], M-estimation [1], partial linear regression [3], quantile regression [4], hypothesis testing [5], and principal component analysis [6]. In addition, considering that high-dimensional data are often encountered in practice, some researchers have focused on developing distributed regularized approaches. For example, Lee et al. [7] proposed solving distributed sparse regression-based on averaging the debiased lasso estimators [8]. Chen and Xie [9] proposed performing penalized regression analysis on each local computing resource and then aggregating the local solutions via an extended majority voting approach to improve the accuracy of variable selection. Zhang et al. [10] studied distributed kernel ridge regression by averaging the kernel ridge regression estimates over results from local machines. One critical issue for distributed regularized approaches is how to choose the tuning parameter for solving each of the high-dimensional subproblems. Zhang et al. [1] suggested using the m-fold averaging method to attain the optimal convergence rate.

Despite the aforementioned advancements, few studies have focused on extending SaC to Bayesian regressions. In the vertical data partition regime, some authors [11,12] proposed the so-called split-and-merge (SaM) Bayesian approach for selecting important variables for ultrahigh dimensional data. The idea of SaM is similar to SaC. Instead of implementing sample-wise splitting of the data, SaM vertically splits the ultrahigh dimensional data into a number of low dimensional subsets and conducts variable selection on each of the subsets, then aggregates the selected variables and conducts variable selection again on the aggregated data set. In this paper, we focus on the problem of sample-wise splitting of data or distributed stored data. We propose a distributed Bayesian variable selection algorithm applying SaC to hierarchical Bayes models when samples are scattered across multiple servers. Specifically, our algorithm consists of the following two stages:

- Splitting the large sample size data into several lower-sample size subsets and performing variable selection for each subset of data by Bayesian Lasso [13], which is a hierarchical Bayes model selection procedure under a specified prior.
- Aggregating the sub-solutions via a majority voting method.

The problem of variable selection for high-dimensional regression is often solved by penalized likelihood approaches. L_1-penalized least squares estimates (Lasso), developed by Tibshirani [14], is one of the most widely used penalized approaches. Tibshirani [14] also pointed out that Lasso estimates can be interpreted as Bayesian posterior mode estimates when the regression parameters have independent and identical Laplace priors. This connection encouraged some authors to develop Bayesian variable approaches by using Laplace priors in hierarchical Bayesian models [13, 15–18]. Casella et al. [19] extended the hierarchical Bayes model to a more general form that can represent the group lasso [20] and the elastic net [21]. In addition to the usual ease of interpretation and ease of implementation of hierarchical models, the Bayesian Lasso has some obvious advantages. First, using a conditional Laplace prior guarantees a unimodal full posterior, which is important for ensuring that the Gibbs sampler converges to a global solution. Second, the structure of the hierarchical model allows both Bayesian and likelihood-based methods for selecting tuning parameters. Third, the Bayesian Lasso automatically provides interval estimates for all parameters, including the error variance. The error variance does not need to be assumed to be known or fixed at a particular estimate; it can be estimated from the data. In addition to the above advantages, the Bayesian framework can more fully capture the uncertainty in estimated parameters and prevent overfitting. Therefore, it allows the use of larger-scale models. Kundu and Dunson [22] showed that Bayesian approaches can lead to a consistent selection of true predictors when the variable dimension p is larger than the sample size n. These aforementioned merits motivate us to use Bayesian Lasso to select variables and estimate parameters on each subset of the data.

On the other hand, the existence of discrepancies (e.g., sample size or other nonrandom patterns) among the subsets of the data may result in the estimated active sets being unmatched with each other or with the true nonzero set. To increase the selection accuracy, Meinshausen and B*ü*hlmann [23] proposed a stability selection procedure through a combination of subsampling and model selection procedures. Shah and Samworth [24] proposed a variant of stability selection with improved error control. However, stability selection and its variant are often computationally infeasible for massive data due to multiple rounds of subsampling and calculation. The majority voting approach developed by Chen and Xie [9] overcomes this drawback and only requires a single round of calculation for each data point. Therefore, the majority voting method reduces the computational cost substantially, especially when the number of machines is very large. Additionally, the majority voting method allows the number of splits to go to infinity, while the SaC and stability station methods require an upper bound, for the number of subdatasets. These merits motivated us to use a majority voting approach to aggregate the sub-solutions. Specifically, we first determine the estimated number of nonzero coefficients via a simple majority voting approach and then combine the estimators from the subdata through a weighted linear sum. In addition, when applying majority voting to penalized regression, as in Chen and Xie [9], the tuning parameters are chosen based on the usual criteria, such as CV and GCV, at each split and therefore can be varied across different subsets of the data, in turn, increases the computational cost. In our approach, we apply Bayesian Lasso to each subset of the data, and the structure of the hierarchical model accommodates both Bayesian and likelihood methods for estimating the tuning parameter. Therefore, as a byproduct, our approach lessens the computational burden raised by the choice of tuning parameters.

In summary, our contribution to this article is three-fold:

- We developed a distributed Bayesian approach for massive data. Our proposed approach contains two steps. First, we use hierarchical models and Gibbs sampling to estimate regression parameters from each subset of the data and then apply a simple majority voting method to estimate the number of selected variables. Second, the Bayesian Lasso estimates from subdatasets are combined via a weighted sum.
- Our method is fully adapted to parallel data collection procedures and thus significantly reduces the computational cost.
- We show through simulation studies that the combined result is asymptotically equivalent to the result of analyzing the entire dataset on one machine (assuming that there is a supercomputer that

could carry out such an analysis).

The rest of the article is organized as follows. In Section 2, we first briefly review Bayesian Lasso and motivated majority-voting sentiment. Section 3 describes the setup of Bayesian variable selection with SaC (BVSaC) and its implementation. Section 4 presents simulation studies to demonstrate the utilities of the proposed methodology. We also provide empirical guidance for the choice of threshold for voting. Section 5 presents an application of the methodology to data from human face images. Section 6 provides some discussions.

2. Related work: Bayesian Lasso

In this section, we first present a brief review of the Bayesian Lasso.

Consider the linear regression model,

$$\mathbf{y} = \mu \mathbf{1}_N + X\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{2.1}$$

where where μ is the overall mean, $\mathbf{y} = (y_1, \ldots, y_N)^{\mathrm{T}}$ is the response vector, $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)^{\mathrm{T}}$ is a sparse $N \times p$ design matrix with $\boldsymbol{x}_i = (x_{i,1}, \ldots, x_{i,p})^{\mathrm{T}}$, $i = 1, \ldots, N$, $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_p)^T$ is the unknown p-dimensional regression coefficient, $\boldsymbol{\epsilon}$ is the $N \times 1$ error vector. We focus on the case that p is large and the model is sparse in the sense that only a relatively small number of predictors are important. Denote the true parameters by $\boldsymbol{\beta}^0 = (\beta_1^0, \cdots, \beta_p^0)^T$.

For high-dimensional variable selection, numerous regularization methods have been developed in the literature. One of the popular methods is Lasso [14], which can be viewed as L_1-penalized least squares. The solution of Lasso is defined as follows:

$$\hat{\beta}_L = \min_{\beta} (\tilde{\mathbf{y}} - \mathbf{X}\beta)^{\mathrm{T}} (\tilde{\mathbf{y}} - \mathbf{X}\beta) + \lambda \sum_{j=1}^{p} |\beta_j|, \tag{2.2}$$

with some proper $\lambda \geq 0$, where $\tilde{\mathbf{y}} = \mathbf{y} - \bar{y}\mathbf{1}_n$. When $\lambda = 0$, $\hat{\beta}_L$ is the ordinary least-squares estimate.

As pointed out by Tibshirani [14], the Lasso estimator can be interpreted as the maximum a posteriori estimator when the regression parameters have independent and identical Laplace priors. Park and Casella [13] suggested Gibbs sampling for the Lasso with the Laplace prior in the hierarchical model. Specifically, they considered a fully Bayesian analysis using a conditional Laplace prior as follows.

$$\pi(\boldsymbol{\beta} \mid \sigma^2) = \prod_{j=1}^{p} \frac{\lambda}{2|\sigma|} e^{\frac{-\lambda|\beta_j|}{\sigma}}.$$

and a noninformative scale-invariant marginal prior $\pi(\sigma^2) = \sigma^{-2}$. Conditioning on σ^2 is important because it guarantees a unimodal full posterior. Lack of unimodality slows convergence of the Gibbs sampler and leads to less meaningful point estimates. In addition, any inverse-gamma prior to σ^2 could maintain the conjugacy in this model. To make Gibbs sampling possible, Park and Casella [13] proposed an expanded hierarchical Bayesian framework consisting of conjugate normal priors for the regression parameters and independent exponential priors for their variances. And they pointed that the full conditional distributions of $\boldsymbol{\beta}, \sigma^2$, and $\tau_1^2, \ldots, \tau_p^2$ are still easy to sample, and they depend on the centered response vector $\tilde{\mathbf{y}}$. Thus, to elucidate the algorithmic mechanism, we give the results of the centered response vector directly in the hierarchical representation. For convenience, we denote $\mathbf{y} = \tilde{\mathbf{y}}$. The specific priors result in the following hierarchical representation of the full model:

$$\begin{aligned}
&(\mathbf{y} \mid \boldsymbol{\beta}, \sigma^2; X) \sim \mathrm{N}(X\boldsymbol{\beta}, \sigma^2 \boldsymbol{I}_N),\\
&(\boldsymbol{\beta} \mid \sigma^2, \boldsymbol{\tau}^2) \sim \mathrm{N}(\mathbf{0}_p, \sigma^2 \mathrm{B}_{\tau}),\\
&(\sigma^2, \boldsymbol{\tau}^2) \sim \pi(\sigma^2)\mathrm{d}\sigma^2 \prod_{j=1}^{p} \frac{\lambda^2}{2} \exp\left\{-\frac{\lambda^2 \tau_j^2}{2}\right\} \mathrm{d}\tau_j^2,\\
&\sigma^2, \tau_1^2, \ldots, \tau_1^p > 0,
\end{aligned}$$

where $\mathrm{B}_{\tau} = \mathrm{diag}(\tau_1^2, \ldots, \tau_p^2)$ and $\tau_1^2, \ldots, \tau_p^2$ are conditionally independent $\boldsymbol{\tau}^2 = (\tau_1^2, \ldots, \tau_p^2)^{\mathrm{T}}$. By choosing a noninformative prior $\pi(\sigma^2) = \sigma^{-2}$ for σ^2, the full conditional distributions of $\boldsymbol{\beta}$, σ^2 and $\tau_1^2, \ldots, \tau_p^2$ are as follows.

$$\boldsymbol{\beta} \mid \mathbf{y}, \sigma^2, \boldsymbol{\tau}; X \sim \mathrm{N}\left(\left(X^{\mathrm{T}}X + \mathrm{B}_{\tau}^{-1}\right)^{-1} X^{\mathrm{T}}\mathbf{y}, \sigma^2 \left(X^{\mathrm{T}}X + \mathrm{B}_{\tau}^{-1}\right)^{-1}\right), \tag{2.3}$$

$$\sigma^2 \mid \mathbf{y}, \boldsymbol{\beta}, \boldsymbol{\tau}^2; X \sim \text{inverse-gamma}\left(\frac{n+p-1}{2}, d_{\sigma}\right), \tag{2.4}$$

$$\frac{1}{\tau_j^2} \sim \text{inverse-Gaussian}\left(\left[\lambda^2 \sigma^2 \beta_j^{-2}\right]^{1/2}, \lambda^2\right),\ j = 1, \ldots, p, \tag{2.5}$$

where $d_{\sigma} = 3$ is the scale parameter of the inverse-gamma distribution. The density for inverse Gaussian (u, v) is

$$f(x) = \left[\frac{v}{2\pi x^2}\right]^{1/2} \exp\left(-\frac{v(x-u)^2}{2u^2 x}\right)^2, x > 0.$$

Based on the above discussions, the true regression parameters $\boldsymbol{\beta}^0$ can be estimated by a Gibbs sampler with block updating of $\boldsymbol{\beta}$ and $(\tau_1^2, \ldots, \tau_p^2)$. Each block update involves the choice of tuning parameter λ, Park and Casella [13] suggest two simple methods for choosing λ: the empirical Bayes by hyperprior diffusion and marginal maximum likelihood using Monte Carlo EM(MCEM) algorithm proposed by Casella [25]. In this article, we adapt the marginal maximum likelihood method to reduce the computational burden, which in turn increases the feasibility of handling extraordinarily large data. Each iteration of the algorithm involves running the Gibbs sampler using a value of tuning parameter λ, which is estimated from the sample of the previous iteration. The iterations are described as follows.

(a) The initial value is given by $\lambda^{(0)} = p\sqrt{\hat{\sigma}_{OLS}^2}/\|\hat{\boldsymbol{\beta}}_{OLS}\|_1$, where $\hat{\boldsymbol{\beta}}_{OLS}$ and $\hat{\sigma}_{OLS}^2$ denote the ordinary least square estimates of $\boldsymbol{\beta}$ and σ^2, respectively. $\|\cdot\|$ is the L_1 norm.

(b) The updated estimate of λ at the t-th iteration is

$$\lambda^{(t)} = \sqrt{2p/\sum_{j=1}^{p} \mathrm{E}_{\lambda^{(t-1)}}[\tau_j^2 \mid \mathbf{y}]}. \tag{2.6}$$

By using MCEM to update λ, the conditional expectation is replaced by the sample averages from one run of the Gibbs sampler.

3. SaC based Bayesian variable selection

Suppose that the entire dataset $\{y_i, \boldsymbol{x}_i, i = 1, \cdots, N\}$ is spread across K computing nodes. Let S_k be a partition of the data indices; therefore, each computing node k can store the corresponding subdataset $D_k = \{y_i, \boldsymbol{x}_i, i \in S_k\}$ with n_k observations. The Bayesian Lasso approach described in Section 2 is used to analyze each subdataset. Let $\tilde{\boldsymbol{\beta}}_k = (\tilde{\beta}_{k,1}, \cdots, \tilde{\beta}_{k,p})^T$ and $\hat{\mathcal{A}}_k = \{j : \tilde{\beta}_{k,j} \neq 0\}$ be the Bayesian Lasso estimate and estimated active set from the kth subdataset, respectively. Following the SaC approach, the final estimate of $\boldsymbol{\beta}^0$ can be obtained by using an arithmetic average combining local estimates, i.e.,

$$\tilde{\boldsymbol{\beta}} = \sum_{k=1}^{K} \tilde{\boldsymbol{\beta}}_k.$$

However, due to discrepancies in subdatasets, the selected variables may be different across local subdatasets. Therefore, this simple combination would result in selecting many noise variables in addition to the true nonzero variables and decreasing the selection accuracy. To improve the selection performance, we use the majority voting approach to aggregate the local results.

First, we estimate the set of selected variables as follows.

$$\hat{\mathcal{A}}^{(c)} \triangleq \{j : \sum_{k=1}^{K} \mathbf{I}(\tilde{\beta}_{k,j} \neq 0) > \omega,\ j = 1, \ldots, p\}, \tag{3.1}$$

where $\omega \in [0, K)$ is a prespecified threshold and $\mathbf{I}(\cdot)$ is the indicator function. It is clear that the cardinality of $\hat{\mathcal{A}}^{(c)}$ is much smaller than p. In the extreme case in which $\omega \geq K - 1$, the estimated active set contains only the variables that are selected by all subset analyses. In another extreme case in which $\omega = 0$, the estimated active set contains the variables selected by one or more local analyses. The choice of ω is a key problem in practice. Chen and Xie [9] suggests $\omega < K/2$ with theoretical and simulation supports.

For our proposed method, we provide a simulation result for the choice of ω. We found that a larger value of ω should be selected to obtain a small expected number of falsely selected variables for low sparse data. More details can be found in the next section.

Second, we aggregate the local estimates through weighted majority voting. Let $\hat{\boldsymbol{\beta}}_{k,\hat{\mathcal{A}}^{(c)}}$ be the sub-vector of $\tilde{\boldsymbol{\beta}}_k$ under the active set $\hat{\mathcal{A}}^{(c)}$. Define $V = \operatorname{diag}(v_1, \ldots, v_p)$ as a $p \times p$ voting matrix with elements

$$v_j = \begin{cases} 1, & \text{if } j \in \hat{\mathcal{A}}^{(c)}, \\ 0, & \text{otherwise.} \end{cases} \tag{3.2}$$

Let $V_{\hat{\mathcal{A}}^{(c)}}$ be a $p \times |\hat{\mathcal{A}}^{(c)}|$ matrix consisting of the jth column in V that satisfies $j \in \hat{\mathcal{A}}^{(c)}$. To extend the above thinking to SaC avoid possible discrepancies, Chen and Xie [9] proposed a weighted summation for final aggregation under a generalized linear model $E(y_i) = g(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta})$. By assuming that, given $\boldsymbol{X}$, the conditional distribution of response is the canonical exponential distribution

$$f(\mathbf{y}; \boldsymbol{X}, \boldsymbol{\beta}) = \prod_{i=1}^{N} f_0(y_i; \theta_i) = \prod_{i=1}^{N} \left\{ c(y_i) \exp\left[\frac{y_i\theta_i - b(\theta_i)}{\phi} \right] \right\},$$

where $\theta_i = \boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{\beta}$, $i = 1, \ldots, N$, and ϕ is a nuisance dispersion parameter. It is known that $b(\theta_i) = \theta_i^2/2$ under the linear model with identity and independent Gaussian error. The aggregation estimator is a weighted average of solutions $\hat{\boldsymbol{\beta}}_{k,\hat{\mathcal{A}}^{(c)}}$ among K computing nodes, that is,

$$V_{\hat{\mathcal{A}}^{(c)}}\left(\sum_{k=1}^{K} V_{\hat{\mathcal{A}}^{(c)}}^{\mathrm{T}}\{X_k^{\mathrm{T}}\Sigma(\hat{\boldsymbol{\theta}})X_k\}V_{\hat{\mathcal{A}}^{(c)}}\right)^{-1}\sum_{k=1}^{K} V_{\hat{\mathcal{A}}^{(c)}}^{\mathrm{T}}\{X_k^{\mathrm{T}}\Sigma(\hat{\boldsymbol{\theta}})X_k\}V_{\hat{\mathcal{A}}^{(c)}}\hat{\boldsymbol{\beta}}^{(k)}_{\hat{\mathcal{A}}^{(c)}},$$

where $\Sigma(\hat{\boldsymbol{\theta}}) = \operatorname{diag}\left(\frac{\partial^2 b(\theta_1)}{\partial\theta_1^2},\ldots,\frac{\partial^2 b(\theta_{n_k})}{\partial\theta_{n_k}^2}\right)$. In the linear model, we have $\Sigma(\hat{\boldsymbol{\theta}}) = \mathbf{I}_{n_k}$. We use the weighted combination to obtain the final aggregated estimate for $\boldsymbol{\beta}^0$.

$$\hat{\boldsymbol{\beta}}^{(c)} \triangleq V_{\hat{\mathcal{A}}^{(c)}}\left[\sum_{k=1}^{K} M^{(k)}_{\hat{\mathcal{A}}^{(c)}}\right]^{-1}\sum_{k=1}^{K} M^{(k)}_{\hat{\mathcal{A}}^{(c)}}\hat{\boldsymbol{\beta}}^{(k)}_{\hat{\mathcal{A}}^{(c)}}. \tag{3.3}$$

where $M^{(k)}_{\hat{\mathcal{A}}^{(c)}} = V_{\hat{\mathcal{A}}^{(c)}}^{\mathrm{T}}\{X_k^{\mathrm{T}}X_k\}V_{\hat{\mathcal{A}}^{(c)}}$. We describe the proposed Bayesian majority-voting SaC (BVSaC) algorithm as follows.

Algorithm 1 Bayesian majority-voting SaC(BVSaC) algorithm

```
Input: Data $D_k = \{y_i, \boldsymbol{x}_i, i \in S_k\}$.
    Initialization: The pre-specified threshold $\omega$.
    Step 1: Receive $\tilde{\boldsymbol{\beta}}_k$, $k = 1,\ldots,K$, through Gibbs sampler;
    Step 2: Receive $\hat{\mathcal{A}}^{(c)}$ through (3.1), and obtain $\hat{\boldsymbol{\beta}}^{(k)}_{\hat{\mathcal{A}}^{(c)}}$;
    Step 3: Receive $V_{\hat{\mathcal{A}}^{(c)}}$ from (3.2);
Output: Output $\hat{\boldsymbol{\beta}}^{(c)}$ according (3.3).
```

4. Simulation studies

We ran extensive simulations to evaluate the performance of the proposed approach. We compared the proposed estimates with their full sample counterparts in terms of selection and estimation accuracy based on the following criteria:

- The estimated model size ($|\hat{\mathcal{A}}| = |\{j : \beta_j^0 \neq 0\}|$): the mean of the number of selected nonzero coefficients;
- Sensitivity (%): the model selection sensitivity, which is defined as the number of truly selected variables divided by the true model size;
- Specificity (%): the model selection specificity, which is defined as the number of truly removed variables divided by the number of noise variables;
- MSE: The predictive mean squared error defined as $\|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}^{(c)}\|_2$.

The data were generated from the linear model (2.1) with i.i.d. ϵ_i from N(0, 1). The explanatory variables $\boldsymbol{x}_i$ followed $\mathrm{N}_p(\mathbf{0}_p, \Sigma)$ with autoregressive correlated design, where the $p \times p$ matrix is $\Sigma = \rho^{|j-j'|}$ for all j, $j' = 1,\ldots,p$. Three settings of ρ were considered: $\rho = 0.1$, 0.5 and 0.9. The sample size was fixed as $N = 10^4$ across models. The number of subdatasets K was set to be 1, 10, 20, and 50. $K = 1$ means that all data are used at once to obtain the Bayesian Lasso estimate. For each value of K, three values of $\omega \in [0, K)$ were considered.

The Bayesian Lasso approach described in Section 2 was used to obtain local estimates for regression parameters. For each subdataset, the length of the Gibbs sampler was 10000, where the first 1000 were discarded in the burn-in process, and one out of every 10 of the remaining samples was chosen to reduce the dependence among samples. When the value of the Lasso estimate was less than 0.1, we set it to 0. The simulation was repeated 100 times.

4.1. Equally valued nonzero signals

We first considered the case that the true model contains $10\% \times p$ nonzero coefficients with values of 1. We investigated how the number of data splits K and the values of ω affected the performance of the proposed approach. The whole data set was divided evenly into K subdatasets with $n_1 = \cdots = n_K = N/K$.

Tables 1 and 2 report the simulation results for $p = 200$ and 500, respectively. If the number of subdata K is not very large, e.g., $K = 10$, 20, when the sample size in each subdata is not larger than the covariate dimension p. In these situations, the proposed estimators exhibited good model selection results with high model selection sensitivity and specificity similar to those of the full sample Bayesian Lasso estimator (corresponding to $K = 1$). The MSEs of the combined estimator were similar to those of the corresponding Bayesian Lasso estimate obtained from the full dataset. In addition, all the considered criteria were robust to variations in ω and ρ. However, when the number of subsamples was large, e.g., $K = 50$, or equivalently, the sample size of each subsample was small but not smaller than p (e.g., the sample size of a subdataset is 200 when $p = 200$), the proposed method performed well enough with large values of ω (e.g., $\omega = 10$, 15). When $(K, p) = (50, 500)$, the sample size of each subdataset ($n_k = 200$) was much smaller than p, and the proposed approach performed well when the within covariate correlation was weak (e.g., $\rho = 0.1$). With the increase of ρ, the number of selected noise variables increased, and the specificity of the model selection decreased. When the correlation between covariates was very strong (e.g., $\rho = 0.9$), the specificity of the model selection increased, but the model sensitivity decreased with increasing values of ω. These results are consistent with the trade-offs of the two types of errors discussed by Chen and Xie [9].

In summary, our proposed method had good performance even though there are $K = 50$ subdatasets. For ultrahigh dimensional data, the sample size in each subdataset may be larger than the covariate dimension. For sparse and simple signals, even if there is a requirement to greatly save the cost of a single machine, which leads to a large K, we recommend a choice of ω no more than $K/2$ is enough. Besides, under a simple signal, large K can still provide accuracy. Thus, one can choose K by computation limitation.

4.2. Complex signals

Suppose the nonzero coefficients in the true model $\mathcal{A} = \{j : \beta_j^0 \neq 0\}$ take different values for different important features. Data were generated from the model with $p = 200$ and $\rho = 0.5$. We considered two sparsity levels $|\mathcal{A}| = 0.1p = 20$ and $|\mathcal{A}| = 0.33p = 66$, with the values of nonzero coefficients being $\{\pm 0.5s, s = 1, \ldots, 10\}$ and $\{-5 - 0.15s, s = 0, \ldots, 66\} \backslash \{-0.05\}$, respectively. Table 3 summarizes the numerical results.

For the sparse model, i.e., the true model size $|\mathcal{A}| = 20$ was relatively small compared to p, the proposed method performed similarly to the full sample Bayesian Lasso approach for $K = 10$ for

Table 1. Comparison of the combined estimator and the complete estimator on the selection and estimation accuracy (with standard deviation in the parenthesis) when the nonzero coefficients take values 1. $N = 10000$, $p = 200$ and the true model size is 20.

ρ	K	ω	Selected	Sensitivity(%)	Specificity(%)	MSE
0.1	1	-	20	100	100	0.33(0.003)
	10	1	20	100	100	0.33(0.003)
		3	20	100	100	0.33(0.003)
		5	20	100	100	0.33(0.003)
	20	1	20	100	100	0.33(0.003)
		4	20	100	100	0.33(0.003)
		6	20	100	100	0.33(0.003)
	50	1	20.02(0.141)	100	99.99(0.001)	0.34(0.004)
		10	20	100	100	0.34(0.004)
		15	20	100	100	0.34(0.003)
0.5	1	-	20	100	100	0.33(0.003)
	10	1	20	100	100	0.33(0.003)
		3	20	100	100	0.33(0.003)
		5	20	100	100	0.33(0.003)
	20	1	20	100	100	0.33(0.003)
		4	20	100	100	0.33(0.003)
		6	20	100	100	0.33(0.003)
	50	1	22.39(1.740)	100	98.67(0.010)	0.35(0.004)
		10	20	100	100	0.35(0.004)
		15	20	100	100	0.35(0.003)
0.9	1	-	20	100	100	0.33(0.003)
	10	1	20	100	100	0.33(0.003)
		3	20	100	100	0.33(0.003)
		5	20	100	100	0.33(0.003)
	20	1	20	100	100	0.33(0.003)
		4	20	100	100	0.33(0.003)
		6	20	100	100	0.33(0.003)
	50	1	58.18(1.604)	100	78.79(0.009)	0.34(0.004)
		10	20	100	100	0.51(0.014)
		15	20	100	100	0.51(0.015)

For simplicity, we omit the zero sample standard deviation.
$K = 1$ stands for the entire dataset analysis.
The numbers in the parenthesis are the standard deviations through 100 repetitions.

Table 2. Comparison of the combined estimator and the complete estimator on the selection and estimation accuracy (with standard deviation in the parenthesis) when the nonzero coefficients take values 1. $N = 10000$, $p = 500$ and the true model size is 50.

ρ	K	ω	Selected	Sensitivity(%)	Specificity(%)	MSE
0.1	1	-	50	100	100	0.33(0.003)
	10	1	50	100	100	0.33(0.003)
		3	50	100	100	0.33(0.003)
		5	50	100	100	0.33(0.003)
	20	1	50	100	100	0.34(0.003)
		4	50	100	100	0.34(0.004)
		6	50	100	100	0.34(0.004)
	50	25	52.48(1.691)	100	99.45(0.004)	18.46(0.364)
		30	50.03(0.180)	100	99.99(0.004)	18.49(0.317)
		35	50	100	100	18.40(0.351)
0.5	1	-	50	100	100	0.33(0.003)
	10	1	50	100	100	0.33(0.003)
		3	50	100	100	0.33(0.003)
		5	50	100	100	0.33(0.003)
	20	1	20	100	100	0.35(0.004)
		4	20	100	100	0.35(0.005)
		6	20	100	100	0.35(0.004)
	50	25	297.88(8.741)	100	44.92(0.019)	2.68(0.197)
		30	122.58(6.392)	100	83.87(0.014)	9.44(0.464)
		35	50.13(0.393)	100	99.97(0.001)	15.67(0.211)
0.9	1	-	50	100	100	0.33(0.003)
	10	1	50	100	100	0.33(0.003)
		3	50	100	100	0.33(0.003)
		5	50	100	100	0.33(0.003)
	20	1	50.20(0.402)	100	100	0.52(0.016)
		4	20	100	100	0.52(0.019)
		6	20	100	100	0.52(0.020)
	50	25	499.88(0.327)	100	0.03(0.001)	0.56(0.764)
		30	172.78(7.275)	100	72.72(0.016)	8.26(0.974)
		35	49.77(5.477)	10.43(0.018)	90.10(0.011)	44.20(5.028)

For simplicity, we omit the zero sample standard deviation.
$K = 1$ stands for the entire dataset analysis.
The numbers in the parenthesis are the standard deviations through 100 repetitions.

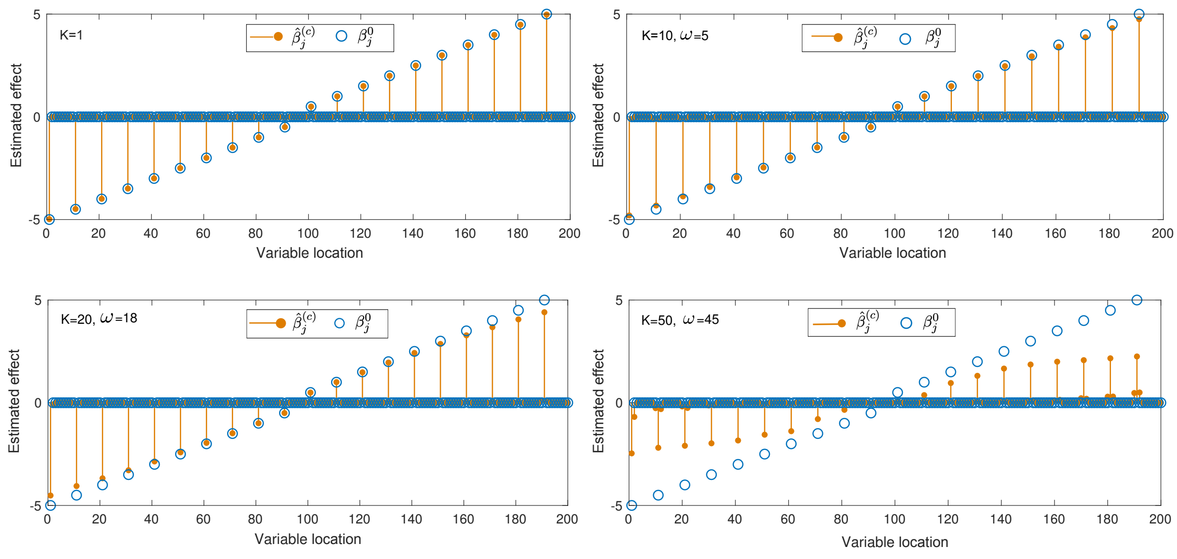

Figure 1. BVSaC estimates from data simulated in Section 4.2($|\mathcal{A}| = 20$). The height of each vertical line refers the combined estimate of each coefficient.

every choice of ω and for $K = 20$ at a large ω close to K. However, when $K = 50$, the proposed method selected many noise variables, which resulted in low specificity, but the sensitivity was larger than 80%. This means that the proposed method selected most of the true variables even though it selected some noise variables. For the model with a moderate sparsity level, $|\mathcal{A}| = 66$, we observed a phenomenon similar to that of the sparse model with $|\mathcal{A}| = 20$.

Figures 1 and 2 display the estimated coefficients under different values of K for the sparse model and moderately sparse model, respectively. $K = 1$ represents the Bayesian Lasso estimate obtained by analyzing all the data together. Within each figure, estimates of nonzero signals are plotted for comparison with the true signals. From these figures, we can clearly see that the combined estimates had almost the same mean as the estimates obtained using the entire dataset when $K = 10, 20$. However, when $K = 50$, the proposed method underestimated the true effect. We speculate this may be because the sample size of subdataset n was the same as p. Additional simulation studies show that the problem of underestimating can be improved when the subdataset sample size is larger than p. Due to the space limit, we omit the relevant simulation results.

In summary, our proposed method stays efficient performance under a suitable choice of ω even for large K. For ultrahigh dimensional data, the sample size in each subdataset may be larger than the covariate dimension. For complex and middle sparse signals, we recommend setting ω to a slightly larger value than $K/2$ to optimize the trade-off of the estimation of the model size and sensitivity/specificity. When K is larger, a larger ω may be required. The number of machines K is a fixed choice for the consideration of computation cost. To save the cost, smaller ω might be chosen when more local machines are added. Besides, under a complex and middle sparse signal, not too large K provides more accuracy.

Table 3. Comparison of the combined estimator and the complete estimator on the selection and estimation accuracy (with standard deviation in the parenthesis) when the nonzero coefficients are not equally valued with $p = 200$ and $\rho = 0.5$.

Model size	K	ω	Selected	Sensitivity(%)	Specificity(%)
20	1	-	20(0)	100(0)	100(0)
	10	2	20.04(0.197)	100(0)	99.98(0.001)
		5	20(0)	100(0)	100(0)
		8	20(0)	100(0)	100(0)
	20	10	27.04(1.109)	100(0)	96.089(0.006)
		15	23.15(0.657)	100(0)	98.250(0.004)
		18	21.06(0.664)	100(0)	99.411(0.004)
	50	41	131.1(28.495)	96.25(0.04)	37.86(0.156)
		43	76.73(25.704)	90.6(0.052)	67.44(0.14)
		45	35.52(13.167)	81.1(0.075)	89.28(0.068)
66	1	-	65.470(0.502)	97.72(0.007)	100(0)
	10	2	65.23(0.446)	97.10(0.004)	99.87(0.003)
		5	65(0)	97.02(0)	100(0)
		8	64.81(0.394)	96.93(0.006)	100(0)
	20	10	84.87(2.658)	96.93(0.004)	85.02(0.020)
		15	67.68(1.449)	95.27(0.009)	97.11(0.009)
		18	62.99(0.643)	93.6(0.009)	99.79(0.004)
	50	25	200(0)	100(0)	0(0)
		38	199.02(1.287)	99.92(0.003)	0.69(0.009)
		44	131.45(18.284)	86.20(0.051)	44.36(0.118)
		45	102.93(18.758)	77.66(0.063)	61.73(0.116)
		46	74.31(15.574)	68.35(0.071)	78.21(0.089)

The numbers in the parenthesis are the standard deviations through 100 repetitions.
For simplicity, we omit the zero sample standard deviation.

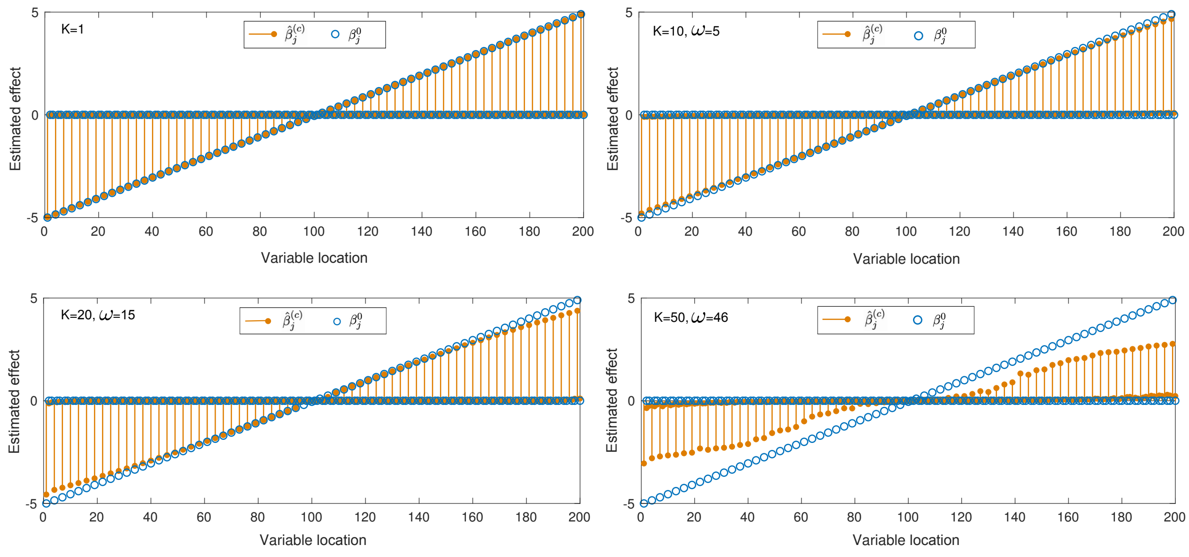

Figure 2. BVSaC estimates from data simulated in Section 4.2($|\mathcal{A}| = 66$). The height of each vertical line refers the combined estimate of each coefficient.

5. Application to age prediction from images of the human face

In this section, we analyze the face image data to illustrate the proposed BVSaC approach for massive data. The relevant dataset is a public resource for the study of face recognition (`https://github.com/cetinsamet/age-estimation`), which consists of 5000 face images of people aged 14 to 62 years old. In total, 512 features were extracted through images by resNet18 (Deep Residual Learning for Image Recognition) with PyTorch in ImageNet Classification in Deep Convolutional Neural Networks. The study goal was to predict people's age based on facial image information. In the process of aging, there are many features apparent on the human face, such as a nasolabial fold, wrinkles, and the shape of cheekbones, which are helpful for identifying the faces, especially of older and younger people. The recognition of middle-aged people is more challenging due to similar features. With this in mind, we restricted our study to 1000 subjects aged from 35 to 45 years old. Although there could be multiple images for each person, we labeled them as different samples, resulting in a total of 50272 facial images.

The selected dataset included 50272 digital records with 512 features for 1000 middle-aged people. Some of these people had more than one image at a certain age. All the experiments are performed in R 3.5.2 on a 32GB workstation with Intel Xeon CPU E5-2603 (1.80 GHz) running 64-bit Linux 2.6.32. We fit model (2.1) in Section 2, with $N = 50272$ age records $\mathbf{y} = (y_1, \dots, y_N)^{\mathrm{T}}$, and $p = 512$ features $\boldsymbol{x}_i = (x_{i,1}, \dots, x_{i,p})^{\mathrm{T}}$.

To illustrate the proposed method, we randomly sampled 700 subjects out of 1000 as the training dataset and left the remaining as the test set. The training set consisted of the selected 700 individuals' 34637 records, and the test set consisted of 15635 records from 300 individuals. Due to the application setting of the SaC framework, we assumed that a person's image data will not be collected and stored separately in several locations. We randomly split the 700 people in the training set, which did not split

Table 4. Age prediction of human face images.

Method	ω	Computing time (in second)	Selected variables	RMPE
Entire data	-	40033.58(3753.5)	504.7(1.494)	3.531(0.003)
SaC	-	1676.7(49.3)	512(0)	3.55(0.011)
BVSaC	20	1709.2(48.7)	512(0)	3.53(0)
	36	1707.7(49.3)	511.3(0.674)	3.54(0.011)
	37	1707.5(49.5)	506.3(1.947)	3.58(0.047)
	38	1704.5(49.0)	478.1(0.674)	3.96(0.432)

RMPE: Square root mean prediction error.
The numbers in the parenthesis are the standard deviations through 10 segmentations.

the records evenly. With this in mind, the training set was randomly split into 40 subsets (assuming there are 40 locations for the whole dataset) by randomly assigning the 700 people along with their records. The split was not balanced due to the different numbers of records for each person.

For $k = 1, \cdots, 40$, let n_k denote the sample size for kth local dataset. During the operation of Algorithm 1 on each local subdataset, we applied the following setup: the burn-in period 5000 of 10000 lengths, thinning every 10 in one run of a Gibbs sampler. In addition, we set $\tilde{\beta}_{k,j} = 0$ if $|\tilde{\beta}_{k,j}| < 0.01$, $j = 1, \ldots, p$. Additionally, we propose to assess the other possible effects on the performance of BVSaC through 10 different segmentations of the entire data set to verify the stability.

Table 4 shows the number of selected variables and the square root prediction error. Aiming at further variable selection among the 512 selected features by using resNet18, the entire dataset variable selection through Bayes Lasso estimation yielded 504.7 selected variables on average(10 different segmentations) with RMPE = 3.53. It is obvious that the original SaC selected all 512 variables and did not serve the purpose of variable selection, however, it evidently requires less computing cost on one machine. Based on a low sparsity dataset, a similar result was found for BVSaC with $\omega = 20$. However, following the ω selection suggestion in Section 4: a larger threshold under complex signals. We set $\omega = 37$, 506.3 features were selected with RMPE = 3.58, which is very close to the result based on the analysis of the entire data with less computing time.

Figure 3 shows the gap between the real age, SaC estimates, and BVSaC estimates (with $\omega = 38$). To assess the validity of the estimation completely, we selected and printed some peoples' age predictions based on individuals with different skin tones and genders: Brendan Fraser, Naveen Andrews, Jada Pinkett Smith, and Naomi Watts. Each of them had various records at different ages. The proposed BVSaC yielded more accurate estimates compared with the original SaC.

6. Conclusions

We propose a distributed Bayesian variable selection approach for massive datasets with large sample sizes and high-dimensional covariates. The main architecture is split-and-conquer to maintain local privacy and limit computation costs. To demonstrate the high-dimensional distributed data on the local computing modes, we propose using Bayesian Lasso with a conditional Laplace prior to achieving variable selection in Bayesian framework. To improve selection and estimation accuracy, we modify

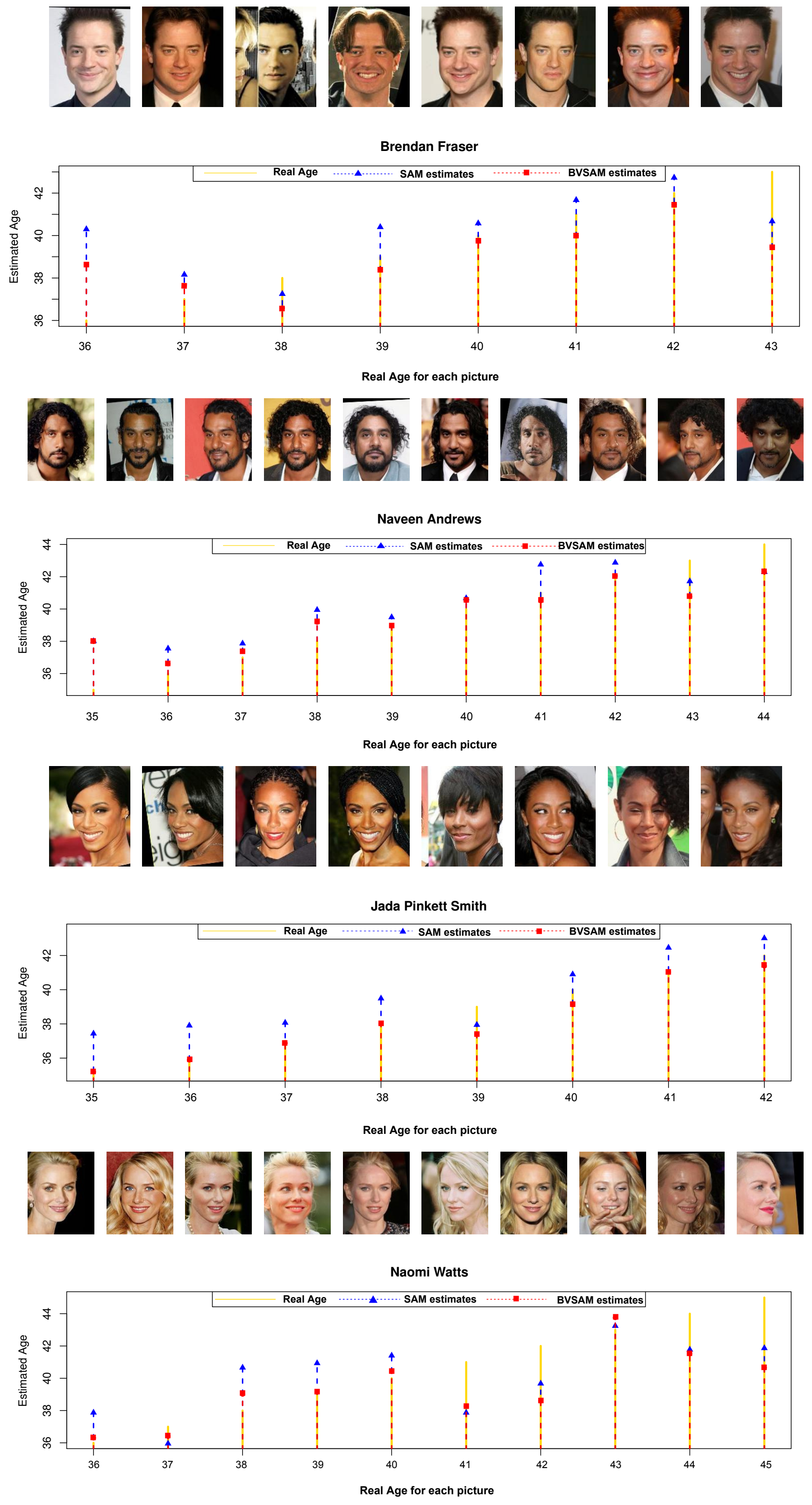

Figure 3. Age estimation by SaC, BVSaC and the real age for each image of each person.

the simple SaC to a Bayesian majority-voting SaC without gaining much computing cost. The proposed BVSaC works to shrink the combined results from local Bayesian variable selections through an adjustable threshold. Simulation studies demonstrated the good performance of BVSaC, which delivers almost the same performance as the result of the analysis of the entire dataset. When the signals were not very sparse, the proposed approach still performed quite well. The computing cost has been greatly reduced by SaC framework to solve the high communication cost for Bayesian estimation. In addition, a little more computing time has been gained for the proposed BVSaC than SaC but much higher accuracy, which is a key consideration in a distributed framework for massive data. In conclusion, the proposed BVSaC is a plausible and efficient method in practice for conducting variable selection and estimation simultaneously for massive data. The framework can be extended to the other penalized Bayesian method e.g., [19, 26–28].

Acknowledgments

The authors are grateful to the Editor, Associate Editor, and reviewers for their valuable comments and suggestions that greatly improved this work. This research is partly supported by the National Nature Science Foundation of China (No. 11971362), and the China Scholarship Council (CSC, No. 201906270074).

Conflict of interest

The authors declare there is no conflicts of interest.

References

1. Y. Zhang, M. J. Wainwright, J. C. Duchi, Communication-efficient algorithms for statistical optimization, *Adv. Neural Inf. Process. Syst.*, **25** (2012). https://doi.org/10.1109/CDC.2012.6426691
2. A. Kleiner, A. Talwalkar, P. Sarkar, M. Jordan, The big data bootstrap, *arXiv preprint*, (2012), arXiv:1206.6415.
3. T. Zhao, G. Cheng, H. Liu, A partially linear framework for massive heterogeneous data, *Ann. Stat.*, **44** (2016), 1400–1437. https://doi.org/10.1214/15-AOS1410
4. Q. Xu, C. Cai, C. Jiang, F. Sun, X. Huang, Block average quantile regression for massive dataset, *Stat. Pap. (Berl)*, **61** (2020), 141–165. https://doi.org/10.1007/s00362-017-0932-6
5. H. Battey, J. Fan, H. Liu, J. Lu, Z. Zhu, Distributed testing and estimation under sparse high dimensional models, *Ann. Stat.*, **46** (2018), 1352. https://doi.org/10.1214/17-AOS1587
6. J. Fan, D. Wang, K. Wang, Z. Zhu, Distributed estimation of principal eigenspaces, *Ann. Stat.*, **47** (2019), 3009–3031. https://doi.org/10.1214/18-AOS1713
7. J. D. Lee, Q. Liu, Y. Sun, J. E. Taylor, Communication-efficient sparse regression, *J. Mach. Learn. Res.*, **18** (2017), 115–144.
8. A. Javanmard, A. Montanari, Confidence intervals and hypothesis testing for high-dimensional regression, *J. Mach. Learn. Res.*, **15** (2014), 2869–2909.

9. X. Chen, M.-g. Xie, A split-and-conquer approach for analysis of extraordinarily large data, *Stat. Sin.*, (2014), 1655–1684.

10. Y. Zhang, J. Duchi, M. Wainwright, Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates, *J. Mach. Learn. Res.*, **16** (2015), 3299–3340.

11. F. Liang, Q. Song, K. Yu, Bayesian subset modeling for high-dimensional generalized linear models, *J. Am. Stat. Assoc.*, **108** (2013), 589–606. https://doi.org/10.1080/01621459.2012.761942

12. Q. Song, F. Liang, A split-and-merge bayesian variable selection approach for ultrahigh dimensional regression, *J. R. Stat. Soc. Series B Stat. Methodol.*, **77** (2015), 947–972. https://doi.org/10.1111/rssb.12095

13. T. Park, G. Casella, The bayesian lasso, *J. Am. Stat. Assoc.*, **103** (2008), 681–686. https://doi.org/10.1198/016214508000000337

14. R. Tibshirani, Regression shrinkage and selection via the lasso, *J. R. Stat. Soc. Series B Stat. Methodol.*, **58** (1996), 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

15. M. Yuan, Y. Lin, Efficient empirical bayes variable selection and estimation in linear models, *J. Am. Stat. Assoc.*, **100** (2005), 1215–1225. https://doi.org/10.1198/016214505000000367

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

17. H. Mallick, N. Yi, A new bayesian lasso, *Stat. Interface*, **7** (2014), 571–582. https://doi.org/10.4310/SII.2014.v7.n4.a12

18. F. Liang, Y. K. Truong, W. H. Wong, Automatic bayesian model averaging for linear regression and applications in bayesian curve fitting, *Sta. Sin.*, 1005–1029. http://www.jstor.org/stable/24306895

19. G. Casella, M. Ghosh, J. Gill, M. Kyung, Penalized regression, standard errors, and bayesian lassos, *Bayesian Anal.*, **5** (2010), 369–411. https://doi.org/10.1214/10-BA607

20. M. Yuan, Y. Lin, Model selection and estimation in regression with grouped variables, *J. R. Stat. Soc. Series B Stat. Methodol.*, **68** (2006), 49–67. https://doi.org/10.1111/j.1467-9868.2005.00532.x

21. H. Zou, T. Hastie, Regularization and variable selection via the elastic net, *J. R. Stat. Soc. Series B Stat. Methodol.*, **67** (2005), 301–320. https://doi.org/10.1080/01621459.2014.881153

22. S. Kundu, D. B. Dunson, Bayes variable selection in semiparametric linear models, *J. Am. Stat. Assoc.*, **109** (2014), 437–447. https://doi.org/10.1080/01621459.2014.881153

23. N. Meinshausen, P. Bühlmann, Stability selection, *J. R. Stat. Soc. Series B Stat. Methodol.*, **72** (2010), 417–473. https://doi.org/10.1111/j.1467-9868.2010.00740.x

24. R. D. Shah, R. J. Samworth, Variable selection with error control: another look at stability selection, *J. R. Stat. Soc. Series B Stat. Methodol.*, **75** (2013), 55–80. https://doi.org/10.1111/j.1467-9868.2011.01034.x

25. G. Casella, Empirical bayes gibbs sampling, *Biostatistics*, **2** (2001), 485–500. https://doi.org/10.1093/biostatistics/2.4.485

26. A. Bhattacharya, D. Pati, N. S. Pillai, D. B. Dunson, Dirichlet-laplace priors for optimal shrinkage, *J. Am. Stat. Assoc.*, **110** (2015), 1479–1490. https://doi.org/10.1080/01621459.2014.960967

27. C. Leng, M.-N. Tran, D. Nott, Bayesian adaptive lasso, *Ann. Inst. Stat. Math.*, **66** (2014), 221–244. https://doi.org/10.1007/s10463-013-0429-6

28. H. Mallick, N. Yi, Bayesian methods for high dimensional linear models, *J. Biometrics Biostatistics*, **1** (2013), 005. https://doi.org/10.4172/2155-6180.S1-005