Least Squares Approximation for a Distributed System

Xuening Zhu[1], Feng Li[2], and Hansheng Wang[3]

[1]*School of Data Science, Fudan University, Shanghai, China*

[2]*School of Statistics and Mathematics, Central University of Finance and Economics, Beijing, China*

[3]*Guanghua School of Management, Peking University, Beijing, China*

Abstract

In this work, we develop a distributed least squares approximation (DLSA) method that is able to solve a large family of regression problems (e.g., linear regression, logistic regression, and Cox's model) on a distributed system. By approximating the local objective function using a local quadratic form, we are able to obtain a combined estimator by taking a weighted average of local estimators. The resulting estimator is proved to be statistically as efficient as the global estimator. Moreover, it requires only one round of communication. We further conduct shrinkage estimation based on the DLSA estimation using an adaptive Lasso approach. The solution can be easily obtained by using the LARS algorithm on the master node. It is theoretically shown that the resulting estimator possesses the oracle property and is selection consistent by using a newly designed distributed Bayesian information criterion (DBIC). The finite sample performance and the computational efficiency are further illustrated by an extensive numerical study and an airline dataset. The airline dataset is 52 GB in size. The entire methodology has been implemented in Python for a *de-facto* standard Spark system. The proposed DLSA algorithm on the Spark system takes 26 minutes to obtain a logistic regression estimator, whereas a full likelihood algorithm takes 15 hours to obtain an inferior result.

KEY WORDS: Distributed System; Least Squares Approximation; Shrinkage Estimation; Distributed BIC.

*Xuening Zhu is supported by the National Natural Science Foundation of China (nos. 11901105, U1811461), the Shanghai Sailing Program for Youth Science and Technology Excellence (19YF1402700), and the Fudan-Xinzailing Joint Research Centre for Big Data, School of Data Science, Fudan University. Feng Li's research is supported by the National Natural Science Foundation of China (no. 11501587). Hansheng Wang's research is partially supported by the National Natural Science Foundation of China (nos. 11831008, 11525101, 71532001, and 71332006). It is also supported in part by China's National Key Research Special Program (no. 2016YFC0207704). Feng Li is the corresponding author.

1. INTRODUCTION

Modern data analysis often needs to address huge datasets. In many cases, the size of the dataset could be too large to be conveniently handled by a single computer. Consequently, the dataset must be stored and processed on many connected computer nodes, which thereafter are referred to as a distributed system. More precisely, a distributed system refers to a large cluster of computers, which are typically connected with each other via wire protocols such as RPC and HTTP (Zaharia et al., 2012). Consequently, they are able to communicate with each other and accomplish the intended data analysis tasks at huge scales in a collective manner.

By using a distributed system, we are able to break a large-scale computation problem into many small pieces and then solve them in a distributed manner. A key challenge faced by statistical computation on a distributed system is the communication cost. The communication cost refers to the wall-clock time cost needed for data communication between different computer nodes, which could be expensive in distributed systems (Zhang et al., 2013; Shamir et al., 2014; Jordan et al., 2018). In this work, we consider a "master-and-worker"-type distributed system with strong workers. We assume that the workers are strong in the sense they are modern computers with reasonable storage and computing capacities. For example, a worker with 32 CPU cores, 128 GB RAM, and 512 GB SSD hard disk could be a very strong worker. As one will see later, the most widely used systems, Hadoop (Apache Software Foundation, 2019a) and Spark (Apache Software Foundation, 2019b), belong to this category. Typically, the workers do not communicate with each other directly. However, they should be connected to a common master node, which is another computer with outstanding capacities. Consequently, most data should be distributed to workers, and most computations should be conducted by the workers. This enables us to solve a large-scale

computation problem in a distributed manner. In contrast, the master should take the responsibility to coordinate with different workers.

For this "master-and-worker"-type distributed system, the communication cost is mostly between the master and workers. One can easily verify that good algorithms for some simple moment estimates (e.g., sample mean) can be easily developed using this type of distributed system. For example, to compute the sample mean on a distributed system, one can first compute the sample mean on each worker, which is known as a map process. Then, each worker reports to the master the resulting sample mean and the associated sample size. Thereafter, the master can compute the overall sample mean by a weighted average of the sample means from each worker, which is known as a reduce process. Such a "MapReduce" algorithm requires only one "master-and-worker" communication for each worker. It requires no direct communication between workers. Because most computations are accomplished by the workers, it also makes good use of the strong worker capacities. As a result, the algorithm can be considered effective. Unfortunately, cases such as the sample mean are rather rare in statistical analysis. Most statistical algorithms do not have an analytical solution (e.g., the maximum likelihood estimation of a logistic regression model) and thus require multiple iterations (e.g., Newton-Raphson iteration or stochastic-gradient-descent-type algorithms). These iterations unfortunately lead to substantial "master-and-worker" communication, which is communicationally expensive. Therefore, developing algorithms that are highly efficient computationally, communicationally and statistically for distributed systems has become a problem of great interest.

In the literature, the common wisdom for addressing a distributed statistical problem can be classified into two categories. The first category is the "one-shot" (OS) or "embarrassingly parallel" approach, which requires only one round of communication.

Specifically, the local worker computes the estimators in parallel and then communicate to the master to obtain an average global estimator (Zhang et al., 2013; Liu and Ihler, 2014; Lee et al., 2015; Battey et al., 2015; Fan et al., 2017; Chang et al., 2017b,a). Although this approach is highly efficient in terms of communication, it might not achieve the best efficiency in statistical estimation in most occasions (Shamir et al., 2014; Jordan et al., 2018). The second approach includes iterative algorithms, which require multiple rounds of communication between the master and the workers. This approach typically requires multiple iterations to be taken so that the estimation efficiency can be refined to match the global (or centralized) estimator (Shamir et al., 2014; Wang et al., 2017a,b; Jordan et al., 2018). In addition, see Yang et al. (2016); Heinze et al. (2016); Smith et al. (2018); Li et al. (2019) for distributed statistical modelling methods when the data are distributed according to features rather than samples.

The aforementioned two approaches are also studied for the sparse learning problem using ℓ_1 shrinkage estimation. For the first approach, Lee et al. (2015) investigated the distributed high-dimensional sparse regression using the OS approach by combining local debiased ℓ_1 estimates. Battey et al. (2015) revisited the same problem but further considered distributed testing and estimation methods in a unified likelihood framework, in which a refitted estimation is used to obtain an oracle convergence rate. For the second approach, both Wang et al. (2017a) and Jordan et al. (2018) have developed iterative algorithms to solve the sparse estimation problem, and they theoretically proved that the error bounds match the centralized estimator. Beyond the ℓ_1 shrinkage estimation, Chen and Xie (2014) studied a penalized likelihood estimator with more general penalty function forms in a high-dimensional setting. However, to the best of our knowledge, there are no guarantees that simultaneously ensure the

model selection consistency (Fan and Li, 2001) and establish a criterion for consistent tuning parameter selection (Wang et al., 2007). In addition, all of the above methods assume independent and identical samples stored by each worker, which is questionable in practice because the distributed dataset might experience great heterogeneity from worker to worker. We would like to remark that the heterogeneity cannot be avoided because it is mainly due to the practical need to record data across time or space (for example).

In this work, we aim to develop a novel methodology to address a sparse estimation problem with low dimensions ($p < n$, where n is the local sample size). Under this setting, we pay greater attention to the communication cost caused by the iterations rather than transmitting digits. The data possessed by different workers are allowed to be heterogeneous but share the same regression relationship. The proposed method borrows the idea of the least squares approximation (LSA, Wang and Leng, 2007) and can be used to handle a large class of parametric regression models on a distributed system. Specifically, let $Y \in \mathbb{R}$ be the response of interest, let X be the associated predictor with finite dimension, and let $\theta \in \mathbb{R}^p$ be the corresponding regression coefficient. The objective is to estimate the regression parameter θ and conduct a variable selection on a distributed system that has one master and many strong workers. Assume data, denoted by (Y_i, X_i), with $1 \leq i \leq N$, that are distributed across different workers. Further, assume that the sample size on each worker is sufficiently large and of the same order. Under this setting, we propose a distributed LSA (DLSA) method. The key idea is as follows:

(1) First, we estimate the parameter θ on each worker separately by using local data on distributed workers. This can be done efficiently by using standard statistical estimation methods (e.g., maximum likelihood estimation). By assuming that

the sample size of each worker is sufficiently large, the resulting estimator and its asymptotic covariance estimate should be consistent but not statistically efficient, as compared with the global estimates.

(2) Next, each worker passes the local estimator of θ and its asymptotic covariance estimate to the master. Because we do not consider a high-dimensional model setting, the communication cost in this regard should be negligible.

(3) Once the master receives all the local estimators from the workers, a weighted least squares-type objective function can be constructed. This can be viewed as a local quadratic approximation of the global log-likelihood functions. As one can expect, the resulting estimator shares the same asymptotic covariance with the full-size MLE method (i.e., the global estimator) under appropriate regularity conditions.

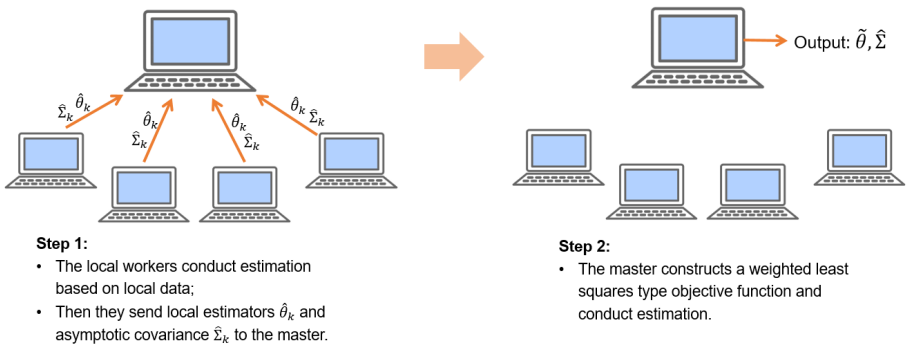

Figure 1: Illustration of the DLSA method.

The major steps of the DLSA method are further illustrated in Figure 1. As one can see, the DLSA method reduces communication costs mainly by using only one round of communication and avoids further iterative steps. Given the DLSA objective function on the master node, we can further conduct shrinkage estimation on the

master. This is done by formulating an adaptive Lasso-type (Zou, 2006; Zhang and Lu, 2007) objective function. The objective functions can be easily solved by the LARS algorithm (Efron et al., 2004) with minimal computation cost on the master. Thus, no communication is required. Accordingly, a solution path can be obtained on the master node. Thereafter, the best estimator can be selected from the solution path in conjunction with the proposed distributed Bayesian information criterion (DBIC). We theoretically show that the resulting estimation is selection consistent and as efficient as the oracle estimator, which is the global estimator obtained under the true model.

To summarize, we aim to make the following important contributions to the existing literature. First, we propose a master with strong workers (MSW) distributed system framework, which solves a large-scale computation problem in a communication-efficient way. Second, given this MSW system, we propose a novel DLSA method, which easily handles a large class of classical regression models such as linear regression, generalized linear regression, and Cox's model. Third, due to the simple quadratic form of the objective function, the analytical solution path can be readily obtained using the LARS algorithm on the master. Then, the best model can be easily selected by the DBIC criterion. Finally, but also most importantly, the proposed DLSA method fully takes advantage of the specialty of the MSW system, which pushes the intensive computation to the workers and therefore is as computationally, communicationally and statistically efficient as possible. Furthermore, we would like to make a remark here that although the proposed DLSA is designed for a distributed system, it can also be applied to a single computer when there are memory constraints (Chen et al., 2018; Wang et al., 2018).

The remainder of this article is organized as follows. Section 2 introduces the model setting and the least squares approximation method. Section 3 presents a

communication-efficient shrinkage estimation and a distributed BIC criterion. Numerical studies are given in Section 4. An application to U.S. Airline data with datasets greater than 52 GB is illustrated using the DLSA method on the Spark system in Section 5. The article concludes with a brief discussion in Section 6. All technical details are delegated to the Appendix.

2. STATISTICAL MODELLING ON DISTRIBUTED SYSTEMS

2.1. Model and Notations

Suppose in the distributed system that there are in total N observations, which are indexed as $i = 1, \cdots, N$. The ith observation is denoted as $Z_i = (X_i^\top, Y_i)^\top \in \mathbb{R}^{p+1}$, where $Y_i \in \mathbb{R}$ is the response of interest and $X_i \in \mathbb{R}^p$ is the corresponding covariate vector. Specifically, the observations are distributed across K local workers. Define $\mathcal{S} = \{1, \cdots, N\}$ to be all sample observations. Decompose $\mathcal{S} = \cup_{k=1}^K \mathcal{S}_k$, where $\mathcal{S}_k$ collects the observations distributed to the kth worker. Obviously, we should have $\mathcal{S}_{k_1} \cap \mathcal{S}_{k_2} = \emptyset$ for any $k_1 \neq k_2$. Define $n = N/K$ as the average sample size for each worker. Then, we assume $|\mathcal{S}_k| = n_k$ and that all n_k diverge in the same order $O(n)$. Specifically, $c_1 \leq \min_k n_k/n \leq \max_k n_k/n \leq c_2$ for some positive constants c_1 and c_2. We know immediately that $N = \sum n_k$. In practice, due to the data storing strategy, the data in different workers could be quite heterogeneous, e.g., they might be collected according to spatial regions. Despite the heterogeneity here, we assume they share the same regression relationship, and the parameter of interest is given by $\theta_0 \in \mathbb{R}^p$. We focus on the case in which p is fixed.

Let $\mathcal{L}(\theta; Z)$ be a plausible twice-differentiable loss function. Define the global loss function as $\mathcal{L}(\theta) = N^{-1} \sum_{i=1}^N \mathcal{L}(\theta; Z_i)$, whose global minimizer is $\widehat{\theta} = \arg\min \mathcal{L}(\theta)$ and

the true value is θ_0. It is assumed that $\widehat{\theta}$ admits the following asymptotic rule

$$\sqrt{N}(\widehat{\theta} - \theta_0) \to_d N(0, \Sigma)$$

for some positive definite matrix $\Sigma \in \mathbb{R}^{p\times p}$ as $N \to \infty$. If $\mathcal{L}(\theta; Z)$ is the negative log-likelihood function, then $\widehat{\theta}$ is the global MLE estimator. Correspondingly, define the local loss function in the kth worker as $\mathcal{L}_k(\theta) = n_k^{-1} \sum_{i\in\mathcal{S}_k} \mathcal{L}(\theta; Z_i)$, whose minimizer is $\widehat{\theta}_k = \arg\min_\theta \mathcal{L}_k(\theta)$. We assume that

$$\sqrt{n_k}(\widehat{\theta}_k - \theta_0) \to_d N(0, \Sigma_k)$$

as $n_k \to \infty$ for a positive definite matrix Σ_k. The goal is to conduct statistical analysis based on the data on the local worker and minimize the communication cost as much as possible.

2.2. Least Squares Approximation and Variance Optimality

In this section, we motivate our approach through least squares approximation to the global loss function, which takes a local quadratic form. To motivate this idea, we begin by decomposing and approximating the global loss function using Taylor's expansion techniques as follows:

$$\begin{aligned}\mathcal{L}(\theta) &= N^{-1}\sum_{k=1}^{K}\sum_{i\in\mathcal{S}_k}\mathcal{L}(\theta; Z_i) = N^{-1}\sum_{k=1}^{K}\sum_{i\in\mathcal{S}_k}\Big\{\mathcal{L}(\theta; Z_i) - \mathcal{L}(\widehat{\theta}_k; Z_i)\Big\} + C_1 \\ &\approx N^{-1}\sum_{k=1}^{K}\sum_{i\in\mathcal{S}_k}(\theta - \widehat{\theta}_k)^\top \ddot{\mathcal{L}}(\widehat{\theta}_k; Z_i)(\theta - \widehat{\theta}_k) + C_2, \end{aligned} \tag{2.1}$$

where the last equation uses the fact that $\dot{\mathcal{L}}_k(\widehat{\theta}_k) = 0$, and C_1 and C_2 are some constants. Typically, the minimizer $\widehat{\theta}_k$ will achieve the convergence rate $\sqrt{n_k}$. Intuitively,

the quadratic form in (2.1) should be a good local approximation of the global loss function. This inspires us to consider the following weighted least squares objective function:

$$\begin{aligned}\widetilde{\mathcal{L}}(\theta) &= N^{-1}\sum_{k}(\theta-\widehat{\theta}_k)^\top\Big\{\sum_{i\in\mathcal{S}_k}\ddot{\mathcal{L}}(\widehat{\theta}_k;Z_i)\Big\}(\theta-\widehat{\theta}_k),\\ &\stackrel{\text{def}}{=}\sum_{k}(\theta-\widehat{\theta}_k)^\top\alpha_k\widehat{\Sigma}_k^{-1}(\theta-\widehat{\theta}_k),\end{aligned}$$

where $\alpha_k = n_k/N$. This leads to a weighted least squares estimator (WLSE), which takes an analytical form as follows:

$$\widetilde{\theta} = \arg\min_{\theta}\widetilde{\mathcal{L}}(\theta) = \Big(\sum_{k}\alpha_k\widehat{\Sigma}_k^{-1}\Big)^{-1}\Big(\sum_{k}\alpha_k\widehat{\Sigma}_k^{-1}\widehat{\theta}_k\Big). \tag{2.2}$$

It is remarkable that the estimator $\tilde{\theta}$ in (2.2) can be easily computed on a distributed system. Specifically, the local worker sends $\widehat{\theta}_k$ and $\widehat{\Sigma}_k$ to the master node, and then, the master node produces the WLSE by (2.2). As a result, the above WLSE requires only one round of communication. Hence, it is highly efficient in terms of communication.

Note that instead of taking a simple average of local estimators $\widehat{\theta}_k$ in the literature, the analytical solution in (2.2) takes a weighted average of $\widehat{\theta}_k$ using weights $\widehat{\Sigma}_k^{-1}$. This will result in a higher statistical efficiency if the data are stored heterogeneously. To investigate the asymptotic properties of the WLSE, we assume the following conditions.

(C1) (Parameter Space) The parameter space Θ is a compact and convex subset of $\mathbb{R}^p$. In addition, the true value θ_0 lies in the interior of Θ.

(C2) (Covariates Distribution) Assume the covariates X_i $(i \in \mathcal{S}_k)$ from the kth worker are independently and identically distributed from the distribution $F_k(x)$.

(C3) (IDENTIFIABILITY) For any $\delta > 0$, there exists $\epsilon > 0$ such that

$$\lim_{n\to\infty} \inf P\Big\{ \inf_{\|\theta^* - \theta_0\| \geq \delta, 1 \leq k \leq K} \big(\mathcal{L}_k(\theta^*) - \mathcal{L}_k(\theta_0)\big) \geq \epsilon \Big\} = 1,$$
$$\text{and} \quad E\Big\{ \frac{\partial \mathcal{L}_k(\theta)}{\partial \theta}\Big|_{\theta=\theta_0} \Big\} = \mathbf{0}.$$

(C4) (LOCAL CONVEXITY) Define

$$\Omega_k(\theta) = E\Big\{ \frac{\partial \mathcal{L}(\theta; Z_i)}{\partial \theta} \frac{\partial \mathcal{L}(\theta; Z_i)}{\partial \theta^\top} \Big| i \in \mathcal{S}_k \Big\} = E\Big\{ \frac{\partial^2 \mathcal{L}_k(\theta; Z_i)}{\partial \theta \partial \theta^\top} \Big| i \in \mathcal{S}_k \Big\}$$

for $i \in \mathcal{S}_k$. Assume that $\Omega_k(\theta)$ is nonsingular at the true value θ_0. In addition, let $\Sigma_k = \{\Omega_k(\theta_0)\}^{-1}$, and $\Sigma = \{\sum_k \alpha_k \Omega(\theta_0)\}^{-1}$.

(C5) (SMOOTHNESS) Define $B(\delta) = \{\theta^* \in \Theta | \|\theta^* - \theta_0\| \leq \delta\}$ as a ball around the true value θ_0 with radius $\delta > 0$. Assume for almost all $Z \in \mathbb{R}^p$ that the loss function $\mathcal{L}(\theta; Z)$ admits all third derivatives $\partial^3 \mathcal{L}(\theta; Z)/(\partial\theta_i \partial\theta_j \partial\theta_l)$ for all $\theta \in B(\delta)$. In addition, assume that there exist functions $M_{ijl}(Z)$ and $\delta > 0$ such that

$$\Big| \frac{\partial^3}{\partial\theta_i \partial\theta_j \partial\theta_l} \mathcal{L}(\theta^*; Z) \Big| \leq M_{ijl}(Z), \quad \text{for all} \quad \theta^* \in B(\delta), \tag{2.3}$$

where $E\{M_{ijl}(Z_m) | m \in \mathcal{S}_k\} < \infty$ for all $1 \leq i, j, l \leq p$ and $1 \leq k \leq K$.

The above conditions are standard conditions to establish the asymptotic properties for M-estimators. First, Condition (C1) assumes the parameter space to be convex (Jordan et al., 2018). Next, Condition (C2) concerns the distribution of the covariates $\{X_i : i \in \mathcal{S}_k\}$. Specifically, there are different $F_k(x)$ for different $1 \leq k \leq K$. In particular, it allows for the heterogeneous distribution of covariates across workers. We would like to remark that heterogeneity is a common phenomenon in distributed

systems, and it has been ignored in much of the literature. Condition (C3) assures the identifiability of the local loss functions across all workers. Finally, Conditions (C4) and (C5) are standard regularity conditions of the loss functions, which require certain degrees of local convexity and smoothness of the loss functions. These conditions are widely assumed in the literature to guarantee asymptotic convergence of the estimators (Fan and Li, 2001; Lehmann and Casella, 2006; Jordan et al., 2018).

Given the conditions, we can establish the asymptotic properties of WLSE in the following Proposition 1 and Theorem 1.

Proposition 1. *Assume Conditions (C1)–(C5). Then, we have*

$$\sqrt{N}(\widetilde{\theta} - \theta_0) = V(\theta_0) + B(\theta_0) \tag{2.4}$$

with $cov\{V(\theta_0)\} = \Sigma$ *and* $B(\theta_0) = O_p(K/\sqrt{N})$, *where* $\Sigma = (\sum_{k=1}^{K} \alpha_k \Sigma_k^{-1})^{-1}$.

The proof of Proposition 1 is given in Appendix A.1. Proposition 1 separates $\sqrt{N}(\widetilde{\theta} - \theta_0)$ into two parts, namely, the variance part and the bias part. Particularly, one should note that the variance order is the same as the global estimator $\widehat{\theta}$, which is $O(N^{-1})$, while the bias order is related to the number of local workers K. Consequently, if the local sample size is sufficiently large, the bias should be sufficiently small, and thus, the global statistical efficiency can be achieved. We state this result in the following theorem.

Theorem 1. (Global Asymptotic Normality) *Assume conditions (C1)–(C5) and further assume* $n/N^{1/2} \to \infty$. *Then, we have* $\sqrt{N}(\widetilde{\theta} - \theta_0) \to_d N(0, \Sigma)$, *which achieves the same asymptotic normality as the global estimator* $\widehat{\theta}$.

The proof of Theorem 1 is given in Appendix A.2. It can be concluded that we should require the local sample size to be of an order larger than $\sqrt{N}$, which is easy

to satisfy in practice. Otherwise, we should have $N/n^2 = K/n \to \infty$. This implies that the number of workers is even larger than the average local sample size n. This is obviously not the case in practice. In the next section, we further discuss the shrinkage estimation based on the DLSA method.

3. COMMUNICATION-EFFICIENT SHRINKAGE ESTIMATION

3.1. Distributed Adaptive Lasso Estimation and Oracle Property

Variable selection is a classical but critically important problem. That is because in practice, the number of available covariates is typically large, but only a small number of covariates are related to the response. Given an appropriate variable selection technique, one can discover the important variables with high probability. In recent decades, various variable selection techniques have been well studied (Tibshirani, 1996; Fan and Li, 2001; Zou, 2006; Wang and Leng, 2007; Zhang, 2010). However, how to conduct variable selection on a distributed system has not been sufficiently investigated. Existing approaches mostly focus on the ℓ_1 shrinkage estimation and develop corresponding algorithms (Lee et al., 2015; Battey et al., 2015; Wang et al., 2017a; Jordan et al., 2018). However, to the best of our knowledge, there are three problems that remain unsolved on a distributed system: (a) most works do not establish the oracle properties of the shrinkage estimators, (b) no consistent tuning parameter selection criterion is given or investigated, and (c) the computation will be heavy if one needs to conduct estimation and select the tuning parameters simultaneously.

To solve the above problems, we first define some notations. Without loss of generality, we assume the first d_0 $(0 < d_0 < p)$ to be nonzero, i.e., $\theta_j \neq 0$ for $1 \leq j \leq d_0$ and $\theta_j = 0$ for $j > d_0$. Correspondingly, we denote $\mathcal{M}_T = \{1, \cdots, d_0\}$ to be true

model. In addition, let $\mathcal{M} = \{i_1, \cdots, i_d\}$ be an arbitrary candidate model with size $|\mathcal{M}| = d$. In addition, for an arbitrary vector $v = (v_j : 1 \leq j \leq p)^\top$, define $v^{(\mathcal{M})} = (v_i : i \in \mathcal{M})^\top \in \mathbb{R}^{|\mathcal{M}|}$ and $v^{(-\mathcal{M})} = (v_i : i \notin \mathcal{M})^\top \in \mathbb{R}^{p-|\mathcal{M}|}$. For an arbitrary matrix $M = (m_{ij})$, define $M^{(\mathcal{M})} = (m_{j_1 j_2} : j_1, j_2 \in \mathcal{M}) \in \mathbb{R}^{|\mathcal{M}| \times |\mathcal{M}|}$

For simultaneous variable selection and parameter estimation, we follow the idea of Wang and Leng (2007) and consider the adaptive Lasso objective function on the master (Zou, 2006; Zhang and Lu, 2007),

$$Q_\lambda(\theta) = \widetilde{\mathcal{L}}(\theta) + \sum_j \lambda_j |\theta_j|. \tag{3.1}$$

By the adaptive Lasso method, different amounts of shrinkage λ_j are imposed on each estimator to improve the estimation efficiency (Zou, 2006; Zou and Li, 2008). Compared to the LSA approach of Wang and Leng (2007), we have the following key differences. First, $\widetilde{\theta}$ is the combined WLSE from local workers. Second, $\widehat{\Sigma}$ is constructed by the local asymptotic covariance estimators $\widehat{\Sigma}_k$. Consequently, to achieve a global convergence rate, one needs to carefully balance the local convergence rate of $\widehat{\theta}_k$ and $\widehat{\Sigma}_k$ with that of the global ones.

Define $\widetilde{\theta}_\lambda = \arg\min_\theta Q_\lambda(\theta)$. Then, we can establish the $\sqrt{N}$-consistency as well as the selection consistency result of $\widetilde{\theta}_\lambda$ under certain conditions of $\lambda = (\lambda_1, \cdots, \lambda_p)^\top$.

Theorem 2. *Assume the conditions (C1)–(C5). Let* $a_\lambda = \max\{\lambda_j, j \leq d_0\}$ *and* $b_\lambda = \min\{\lambda_j, j > d_0\}$. *Then, the following result holds.*

a.a. ($\sqrt{N}$-CONSISTENCY). *If* $\sqrt{N} a_\lambda \to_p 0$, *then* $\widetilde{\theta}_\lambda - \theta = O_p(N^{-1/2})$.

b.b. (SELECTION CONSISTENCY). *If* $\sqrt{N} a_\lambda \to_p 0$ *and* $\sqrt{N} b_\lambda \to_p \infty$, *then*

$$P(\widetilde{\theta}_\lambda^{(-\mathcal{M}_T)} = 0) \to 1 \tag{3.2}$$

The proof of Theorem 2 is given in Appendix A.3. Note here that a_λ controls the largest amount of penalty on the true nonzero parameters. Consequently, this amount cannot be too large; otherwise, it will result in a highly biased estimator. In contrast, b_λ is responsible for producing sparse solutions of irrelevant covariates. Therefore, b_λ should be sufficiently large to produce an effective amount of shrinkage.

By Theorem 2, we know that with probability tending to one, we have $\widetilde{\theta}_\lambda^{(-\mathcal{M}_T)} = \mathbf{0}$. Meanwhile, $\widetilde{\theta}_\lambda^{(\mathcal{M}_T)} - \theta_0^{(\mathcal{M}_T)} = O_p(N^{-1/2})$. It is then natural to ask whether the statistical efficiency of $\widetilde{\theta}_\lambda^{(\mathcal{M}_T)}$ can be as good as the oracle estimator, which is the global estimator obtained under the true model, i.e., $\widehat{\theta}_{oracle} = \arg\min_{\theta \in \mathbb{R}^p, \theta_j = 0, \forall j \notin \mathcal{M}_T} \mathcal{L}(\theta)$. To this end, we require the following technical condition.

(C6) (Covariance Assumption) Define the global unpenalized estimator as $\widehat{\theta}_{\mathcal{M}} = \arg\min_{\{\theta \in \mathbb{R}^p : \theta_j = 0, \forall j \notin \mathcal{M}\}} \mathcal{L}(\theta)$. Assume for the global estimator $\widehat{\theta}_{\mathcal{M}}$ with $\mathcal{M} \supset \mathcal{M}_T$ that $\sqrt{N}(\widehat{\theta}_{\mathcal{M}}^{(\mathcal{M})} - \theta_{\mathcal{M}}^{(\mathcal{M})}) \to_d N(0, \Sigma_{\mathcal{M}}) = N(0, \Omega_{\mathcal{M}}^{-1})$, where $\Sigma_{\mathcal{M}} \in \mathbb{R}^{|\mathcal{M}| \times |\mathcal{M}|}$ is a positive-definite matrix. Further assume for any $\mathcal{M} \supset \mathcal{M}_T$ that $\Omega_{\mathcal{M}} = \Omega_{\mathcal{M}_F}^{(\mathcal{M})}$ holds, where $\mathcal{M}_F = \{1, \cdots, p\}$ denotes the whole set.

Condition (C6) does not seem very intuitive. Nevertheless, it is a condition that is well satisfied by most maximum likelihood estimators. A more detailed discussion has been provided by Wang and Leng (2007). We then have the oracle property in the following theorem.

Theorem 3. (Oracle Property) *Assume Conditions (C1)–(C6). Let* $\sqrt{N}a_\lambda \to_p 0$ *and* $\sqrt{N}b_\lambda \to_p \infty$*; then, it holds that*

$$\sqrt{N}\Big(\widetilde{\theta}_\lambda^{(\mathcal{M}_T)} - \theta^{(\mathcal{M}_T)}\Big) \to_d N\Big(0, \Sigma_{\mathcal{M}_T}\Big). \tag{3.3}$$

By Theorem 2 and 3, we know that as long as the tuning parameters are approximately selected, the resulting estimator is selection consistent and as efficient as the oracle estimator. It is remarkable that tuning a total of p parameters simultaneously is not feasible in practice. To fix this problem, we follow the tradition of Zou (2006) and Wang et al. (2007) to specify $\lambda_j = \lambda_0 |\widetilde{\theta}_j|^{-1}$. Since $\widetilde{\theta}_j$ is $\sqrt{N}$-consistent, then as long as as λ_0 satisfies the condition $\lambda_0 \sqrt{N} \to 0$ and $\lambda_0 N \to \infty$, then the conditions $\sqrt{N} a_\lambda \to_p 0$ and $\sqrt{N} b_\lambda \to_p \infty$ are satisfied. Thereafter, the original problem of tuning parameter selection for λ can be replaced by selection for λ_0.

3.2. The Distributed Bayes Information Criterion

Although it has been shown that asymptotically, the oracle property can be guaranteed as long as the tuning parameters are approximately selected, it is still unclear how to conduct variable selection in practice. That motivates us to design a BIC-type criterion that can select the true model consistently in a completely data-driven manner (Zhang and Lu, 2007; Chen and Chen, 2008; Zou and Zhang, 2009; Wang et al., 2013). Specifically, to consistently recover the sparsity pattern, we consider a distributed Bayesian information criterion (DBIC)-based criterion as follows:

$$\text{DBIC}_\lambda = (\widetilde{\theta}_\lambda - \widetilde{\theta})^\top \widehat{\Sigma}^{-1} (\widetilde{\theta}_\lambda - \widetilde{\theta}) + \log N \times df_\lambda / N, \tag{3.4}$$

where df_λ is the number of nonzero elements in $\widetilde{\theta}_\lambda$.

The design of the DBIC criterion is in the spirit of the BIC criterion used in Wang and Leng (2007). The difference is that the DBIC uses the WLSE estimator $\widetilde{\theta}$ and the average of distributed covariance estimators $\widehat{\Sigma}$ to construct the least squares objective function. Intuitively, if $\widetilde{\theta}$ and $\widehat{\Sigma}$ approximate the global estimator $\widehat{\theta} = \arg\max \mathcal{L}(\theta)$ and asymptotic covariance very well, then the DBIC criterion should be able to facili-

tate consistent tuning parameter selection. Specifically, the resulting model should be selection consistent (Shao, 1997).

To formally investigate the theoretical performance of DBIC, we first define some notations. First, we define the set of nonzero elements of $\widehat{\theta}_\lambda$ by $\mathcal{M}_\lambda$. Given a tuning parameter λ, $\mathcal{M}_\lambda$ could be underfitted, correctly fitted or overfitted. We could then have the following partition:

$$\mathbb{R}_- = \{\lambda \in \mathbb{R}^p : \mathcal{M}_\lambda \not\supset \mathcal{M}_T\}, \quad \mathbb{R}_0 = \{\lambda \in \mathbb{R}^p : \mathcal{M}_\lambda = \mathcal{M}_T\},$$
$$\mathbb{R}_+ = \{\lambda \in \mathbb{R}^p : \mathcal{M}_\lambda \supset \mathcal{M}_T, \mathcal{M}_\lambda \neq \mathcal{M}_T\},$$

where $\mathbb{R}_-$ denotes the underfitted model, and $\mathbb{R}_+$ denotes an overfitted model. We show in the following Theorem that the DBIC can consistently identify the true model.

Theorem 4. *Assume Conditions (C1)–(C6). Define a reference tuning parameter sequence* $\{\lambda_N \in \mathbb{R}^p\}$*, where the first* d_0 *elements of* λ_N *are* $1/N$ *and the remaining elements are* $\log N/N$*. Then, we have*

$$P\Big(\inf_{\lambda \in \mathbb{R}_- \cup \mathbb{R}_+} DBIC_\lambda > DBIC_{\lambda_N}\Big) \to 1.$$

By Theorem 2 and 3, we know that with probability tending to one, we should have $\mathcal{M}_{\lambda_N} = \mathcal{M}_T$. Consequently, the sequence λ_N here plays a role as a reference sequence that leads to the true model. Accordingly, Theorem 4 implies that the optimal λ selected by the DBIC will consistently identify the true model. This is because any λ leading to an inconsistent model selection result should perform worse than λ_N in terms of DBIC values. This result does not imply that the tuning parameter selected by the DBIC is λ_N; it only states that any inconsistent λ should be at least worse than λ_N, but λ_N leads to consistent model selection results.

4. NUMERICAL STUDIES

4.1. Simulation Models and Settings

To demonstrate the finite sample performance of the DLSA method, we conduct a number of simulation studies in this section. Five classical regression models are presented, and the corresponding DLSA algorithms are implemented. For each model, we consider two typical settings to verify the numerical performance of the proposed method. They represent two different data storing strategies together with competing methods. The first strategy is to distribute data in a complete random manner. Thus, the covariates on different workers are independent and identically distributed (i.i.d). In contrast, the second strategy allows for covariate distribution on different workers to be heterogeneous. The estimation efficiency as well as the as the variable selection accuracy are evaluated. Examples are given as follows.

Example 1. (Linear Regression). We first consider one of the most popular regression analysis tools, i.e., linear regression. In particular, we generate the continuous response Y_i by a linear relationship with the covariates X_i as follows:

$$Y_i = X_i^\top \theta_0 + \epsilon_i,$$

, where the noise term ε_i is independently generated using a standard normal distribution $N(0,1)$. Following Fan and Li (2001), the true parameter is set as $\theta_0 = (3, 1.5, 0, 0, 2, 0, 0, 0)^\top$.

Example 2. (Logistic Regression). The logistic regression is a classical model that addresses binary responses (Hosmer Jr et al., 2013). In this example, we generate

the response Y_i independently by the Bernoulli distribution given the covariate X_i as

$$P(Y_i = 1|X_i) = \frac{\exp(X_i^\top \theta_0)}{1 + \exp(X_i^\top \theta_0)}.$$

We follow Wang and Leng (2007) to set the true parameter $\theta_0 = (3, 0, 0, 1.5, 0, 0, 2, 0)^\top$.

Example 3. (Poisson Regression). In this example, we consider the Poisson regression, which is used to model counted responses (Cameron and Trivedi, 2013). The responses are generated according to the Poisson distribution as

$$P(Y|X_i, \theta_0) = \frac{\lambda^{Y_i}}{Y_i!} \exp(-\lambda), \tag{4.1}$$

where $\lambda = \exp(X_i^\top \theta_0)$. The true parameter θ_0 is set to $(0.8, 0, 0, 1, 0, 0, -0.4, 0, 0)^\top$.

Example 4. (Cox Model). Next, we consider Cox's model in the survival analysis. (Fan and Li, 2002). Specifically, we set the hazard function to be

$$h(t_i|X_i) = \exp(X_i^\top \theta_0),$$

, where t_i is the survival time from the ith subject. In practice, we generate the survival time from an exponential distribution with mean $\exp(-X_i^\top \theta_0)$. The true parameter θ_0 is set as $\theta_0 = (0.8, 0, 0, 1, 0, 0, 0.6, 0)^\top$. Next, the censoring time is generated independently from an exponential distribution with a mean $u_i \exp(X_i^\top \beta)$, where u_i is sampled from a uniform distribution $U[1, 3]$. This censoring mechanism produces approximately 30% censored data (Fan and Li, 2002).

Example 5. (Ordered Probit Regression). Finally, we consider the ordered Probit regression model, which is widely used to model ordinal responses (Harrell Jr, 2015). Specifically, the responses take the value $Y = 1, \cdots, L$, which represents natural

orders. Given the covariates X_i, the ordinal responses are independently generated as follows:

$$P(Y_i = l|X_i, \theta_0) = \begin{cases} \Phi(c_1 - X^\top \theta_0) & l = 1 \\ \Phi(c_l - X^\top \theta_0) - \Phi(c_{l-1} - X^\top \theta_0) & 2 \leq l \leq L-1 \\ 1 - \Phi(c_{L-1} - X^\top \theta_0) & l = L \end{cases}$$

where $\Phi(\cdot)$ denotes the distribution function for the standard normal distribution, and $c_1, \cdots, c_{L-1}$ are the cut points. We set $\theta_0 = (0.8, 0, 0, 1, 0, 0, 0.6, 0)^\top$ and $(c_1, \cdots, c_{L-1})^\top = (-1, 0, 0.8)^\top$ with $L = 4$.

For each example, two different data storage strategies are considered. They lead to different covariate distributions $F_x(x)$. Specifically, the following two settings are investigated.

- Setting 1 (i.i.d Covariates). We first consider the setting in which the data are distributed independently and identically across the workers. Specifically, the covariates X_{ij} $(1 \leq i \leq N, 1 \leq j \leq p)$ are sampled from the standard normal distribution $N(0, 1)$.

- Setting 2 (Heterogeneous Covariates). Next, we look at the case whereby the covariates distributed across each worker are heterogeneous. This is a common case in practice. Specifically, on the kth worker, the covariates are sampled from the multivariate normal distribution $N(\mu_k, \Sigma_k)$, where μ_k is generated from the uniform distribution $U[-1, 1]$, and $\Sigma_k = (\sigma_{k,ij}) = (\rho_k^{|j_1 - j_2|})$ with ρ_k sampled from $U[0.3, 0.4]$.

4.2. Performance Measurements

In this section, we give performance measurements and summarize simulation results with respect to the estimation efficiency as well as the variable selection accuracy. The sample sizes are set as $N = (10, 20, 100) \times 10^3$. Correspondingly, the number of workers is set to $K = (5, 5, 10)$.

For a reliable evaluation, we repeat the experiment $R = 500$ times. For the rth replication, denote $\widehat{\theta}^{(r)}$ and $\widetilde{\theta}^{(r)}$ as the global estimator and WLSE, respectively. To measure the estimation efficiency, we calculate the root mean square error (RMSE) for the jth estimator as $\text{RMSE}_{\widetilde{\theta},j} = \{R^{-1}\sum_r \|\widetilde{\theta}_j^{(r)} - \theta_{0j}\|^2\}^{1/2}$. The RMSE for the global estimator $\widehat{\theta}$ can be defined similarly. Then, the relative estimation efficiency (REE) with respect to the global estimator is given by $\text{REE}_j = \text{RMSE}_{\widehat{\theta},j}/\text{RMSE}_{\widetilde{\theta},j}$ for $j = 1, \cdots, p$.

Next, based on the WLSE, we further conduct shrinkage estimation on the master node. Let $\widehat{\mathcal{M}}^{(r)}$ be the set of selected variables in the rth replication using the DBIC. Correspondingly, $\widetilde{\theta}_\lambda^{(r)}$ is the shrinkage estimator. To measure the sparse discovery accuracy, we calculate the average model size (MS) as $\text{MS} = R^{-1}\sum_r |\widehat{\mathcal{M}}^{(r)}|$. Next, the percentage of the true model being correctly identified is given by CM= $R^{-1}\sum_r I(\widehat{\mathcal{M}}^{(r)} = \mathcal{M}_T)$. In addition, to further investigate the estimation accuracy, we calculate the REE of the shrinkage estimation with respect to the global estimator as $\text{REE}_j^s = \text{RMSE}_{\widehat{\theta},j}/\text{RMSE}_{\widetilde{\theta}_\lambda,j}$ for $j \in \mathcal{M}_T$.

4.3. Simulation Results

We compare the proposed DLSA method with (a) the OS estimator (Zhang et al., 2013), and (b) the CSL estimator (Jordan et al., 2018). The simulation results are summarized in Table 2–6. First, in the i.i.d case, one can observe that all three methods are as efficient as the global estimator when N is increased. For example, for the Poisson

regression (i.e., Table 4), all the methods could achieve REE ≈ 1 when $N = 100,000$ and $K = 10$. However, in the heterogeneous setting (i.e., Setting 2), the finite sample performances of the three competing methods are quite different. The proposed DLSA method achieves the highest efficiency of all the methods, which is also asymptotically efficient as the global estimator. For instance, the REE of the DLSA estimation for the logistic regression (i.e., Table 3) is near 1 in the second setting with $N = 20,000$ and $K = 5$, while the REEs of the OS and CSL methods are approximately 0.88 and 0.37, respectively. Although the OS estimator is less efficient than the DLSA estimator, it is still consistent as N increases. The CSL method behaves worst under this situation. That is because it only uses the local Hessian matrix; this could result in a highly biased estimator.

With respect to the shrinkage estimation, one can observe that the adaptive Lasso estimator is able to achieve higher estimation efficiency in the second setting than the global estimator. For example, the REE for the shrinkage DLSA (SDLSA) method could be even higher than 1 for Cox's model in Setting 2 (i.e., Table 5). Finally, the newly designed DBIC method does a great job at identifying the nonzero variables with high accuracy. To see this, one could observe, for example, in the ordered Probit regression (i.e., Table 6), that the MS is controlled well to be approximately 3 and, CM is near 1.

5. APPLICATION TO AIRLINE DATA

For illustration purposes, we study a large real-world dataset. Specifically, the dataset considered here is the U.S. Airline Dataset. The dataset is available at http://stat-computing.org/dataexpo/2009. It contains detailed flight information about U.S. airlines from 1987 to 2008. The task is to predict the delayed status of a flight given

all other flight information. Each sample in the data corresponds to one flight record, which consists of delayed status, departure time, arrival time, distance of the flight, flight date, delay status at departures, carrier information, origin and destination. The complete variable information is described in Table 7. The data contain six continuous variables and five categorical variables. The categorical variables are converted to dummies with appropriate dimensions. We treat the Year and DayofMonth variables as numerical to capture the time effects. To capture possible seasonal patterns, we also convert the time variables Month and *DayofWeek* to dummies. Ultimately, a total of 181 variables are used in the model. The total sample size is 113.9 million observations. This leads to the raw dataset being 12 GB on a hard drive. After the dummy transformation described in Table 7, the overall in-memory size is over 52 GB, even if all the dummies are stored in a sparse matrix format. Thus, this dataset can hardly be handled by a single computer. All the numerical variables are standardized to have a mean of zero and a variance of one.

5.1. The Spark System and MLE

To demonstrate our method, we set up a Spark-on-YARN cluster on the Alibaba cloud server (https://www.alibabacloud.com/products/emapreduce). This is a standard industrial-level architecture setup for a distributed system. The system consists of one master node and two worker nodes. Each node contains 64 virtual cores, 64 GB of RAM and two 80 GB SSD local hard drives. The dataset is stored on the Hadoop data file system (HDFS).

Because the RAM is larger than the raw data size, one may wonder whether the logistic regression task can be run on a single node. Unfortunately, this is infeasible in practice. This is because much more memory (typically $>$ 128 GB) is needed for

operating on matrices of such a huge size. Even for the installed Spark system, a task of this magnitude cannot be directly performed using an existing algorithm library (e.g., Spark ML). This is because Spark is a very memory-intensive system. For example, to compute a single distributed matrix with a size of approximately 1 GB in memory, one might need each worker to have 2 GB of memory in practice. This overhead memory consumption grows significantly as the size of the data matrix increases. For discussions, see Chen and Guestrin (2016).

If one insists on computing the traditional MLE based on the entire dataset under memory constraints, then a stochastic gradient decent (SGD) algorithm (Zhang, 2004) has to be used. However, the built distributed SGD algorithm simply fails with the aforementioned cluster due to the well-known out-of-memory problem in Spark. To this end, we have to use a serialized SGD algorithm; however, we run it on the same hardware for comparison purposes. The entire dataset needs to be randomly shuffled first. Then, a serialized SGD algorithm optimizes the log-likelihood function in an iterative manner. For each iteration, a batch of the data is randomly sampled and then used for calculating the log likelihood and its gradient. A heuristic batch size and learning rate are used. The total computational time is 15.4 hours, and the resulting log-likelihood is approximately -1.71×10^8. We would like to remark that the computing time for the MLE does not include the data shuffling time. This serves as an important benchmark to gauge the performance of the other competing methods (e.g., DLSA and OS methods).

5.2. The DLSA and OS Method

Fortunately, both the proposed DLSA and OS methods allow us to develop a user-friendly Spark algorithm with very limited computer resources. As the algorithm is

designed in a batch manner, it is highly efficient under memory constraints. The algorithm is developed with the Spark Python API (PySpark) and run on a Spark system (version > 2.3) (see Algorithm 1 for details). It can be freely downloaded from https://github.com/feng-li/dlsa. We then use our algorithm to fit the model.

To this end, the entire dataset is randomly partitioned into 1139 subgroups. The sample size for each subgroup is approximately $100,000$. Next, for each subgroup of data, we create a virtual worker (i.e., an executor in the Spark system) so that the computation for each worker can be conducted in a parallel manner. By doing so, the computation power of the entire Spark system can be maximized for both DLSA and OS methods. Finally, we find that 26.2 minutes are needed for DLSA, and 25.3 minutes are needed for the OS method. We remark that the computing time for the MLE includes the data shuffling time with our DLSA and OS algorithms. The corresponding log-likelihood values are -1.62×10^8 and -1.65×10^8, respectively. Comparing these results with that of the traditional MLE, we find that the traditional MLE is extremely difficult to compute. It takes more that 15 hours and obtains an inferior result (i.e., smaller log-likelihood value). In contrast, the log-likelihood value of the DLSA is the best.

5.3. Variable Selection Results with BIC

We next apply the proposed shrinkage DLSA method (referred to as SDLSA) with the BIC criterion to conduct variable selection. It is remarkable that this can be fully conducted on the master, and no further communication is needed. It takes only 0.2 seconds to accomplish the task. After the shrinkage estimation, we are able to reduce the 181 variables to 157 variables.

The detailed results are summarized in Table 1. First, with respect to time effects,

both yearly and seasonal trends are found. The coefficient for the Year is 6.12, which implies that as the year proceeds, the airline delays become more severe. Next, the passengers are likely to encounter delays in May, June, October, November, and December (with coefficients of 6.03, 0.28, 0.8, 0.4, and 0.19, respectively). In terms of days of the week, more delays are expected for certain working days, i.e., Tuesday, Wednesday and Friday (with coefficients of 0.6, 0.85 and 0.39, respectively) compared to weekends. Finally, within a given day, we find the coefficients for both the scheduled departure time (CRSDepTime, -0.12) and the scheduled arrival time (CRSArrTime, -0.13) are negative, indicating that late departing and arriving flights suffer less so from delays.

Next, with respect to the airline carriers, AA (American Airlines), DL (Delta Air Lines), NW (Northwest Airlines), and UA (United Airlines) have estimated coefficients of 0.49, 0.18, 0.39, and 0.70, which indicates how more likely they are to be delayed compared with other airlines. In addition, one may be interested in with which airports are more likely to have delayed flights. According to our estimation results, the top five origin airports that cause delays are IAH (George Bush Intercontinental Airport), LGA (LaGuardia Airport), PHL (Philadelphia International Airport), RDU (Raleigh-Durham International Airport), ONT (Ontario International Airport), and SMF (Sacramento International Airport), with coefficients of 0.82, 0.87, 0.94, 1.58, and 1.59, respectively. The top five destination airports that cause delays are PBI (Palm Beach International Airport), MCI (Kansas City International Airport), DCA (Ronald Reagan Washington National Airport), SAN (San Diego International Airport), and MEM (Memphis International Airport), with coefficients of 0.99, 1.00, 1.07, 1.15, and 1.16, respectively.

6. CONCLUDING REMARKS

Table 1: Coefficients for the logistic model estimated with SDLSA using the BIC. The carrier and airport abbreviations are assigned by the International Air Transport Association. We denote "Airport", "Origin" and "Destination" by "A", "O" and "D", respectively.

	Intercept	Year	CRSArrTime	Distance	CRSDepTime	DayofMonth	ElapsedTime	DepTime			
	−0.30	6.12	−0.13	−5.68	−0.12	−0.06	0.08	0.15			
Month	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
	−0.13	−0.01	0.03	6.03	−0.28	−0.09	−0.02	−0.07	0.8	0.4	0.19
Day of Week	Tue	Wed	Thu	Fri	Sat	Sun					
	0.6	0.85	−0.57	0.39	0.22	0.26					
Carrier	AA	CO	DL	NW	UA	US	WN				
	0.49	−0.16	0.18	0.39	0.7	−0.43	−0.6				
A	ABQ	ANC	ATL	AUS	BDL	BHM	BNA	BOS	BUF	BUR	BWI
O	−0.46	−0.14	−0.02	0.39	0.38	0.13	−0.17	0.13	−0.55	−0.01	0.54
D	−0.61	0.78	−0.84	−0.73	−0.74	−0.46	−0.28	0.51	−0.50	−0.90	0.69
A	CLE	CLT	CMH	CVG	DAL	DAY	DCA	DEN	DFW	DTW	ELP
O	0.49	−0.87	−0.80	0.29	−0.08	0.40	−0.07	−0.22	0.53	0.32	−0.56
D	0.14	−0.46	0.64	−1.60	−0.72	−0.79	1.07	0.37	−1.04	−0.19	−0.54
A	EWR	FLL	GSO	HNL	HOU	IAD	IAH	IND	JAX	JFK	LAS
O	0.21	−0.99	0.11	0.19	−0.51	−0.29	0.82	0.46	−0.71	0.07	−0.94
D	−0.44	−1.70	0.77	0.79	−1.62	0.00	−0.54	−0.27	−0.12	−0.70	−0.29
A	LAX	LGA	MCI	MCO	MDW	MEM	MIA	MKE	MSP	MSY	OAK
O	0.38	0.87	0.61	0.52	−0.48	0.71	0.39	−0.64	−0.10	0.21	−1.04
D	−0.20	0.13	1.00	0.30	0.10	1.16	−0.11	−0.83	−0.61	0.97	−1.47
A	OKC	OMA	ONT	ORD	ORF	PBI	PDX	PHL	PHX	PIT	PVD
O	−0.71	−0.85	1.58	−0.41	−0.67	−1.12	−0.13	0.94	0.37	0.18	−0.42
D	0.69	−0.74	−1.49	−0.72	−0.79	0.99	−0.64	−0.54	0.83	0.33	−0.40
A	RDU	RIC	RNO	ROC	RSW	SAN	SAT	SDF	SEA	SFO	SJC
O	−5.80	0.02	−0.06	0.20	0.05	0.00	−0.04	0.00	0.24	0.24	0.14
D	0.97	0.83	0.71	−1.07	0.37	1.15	0.78	−0.35	−0.79	−0.51	0.21
A	SJU	SLC	SMF	SNA	STL	SYR	TPA	TUL	TUS		
O	−0.14	0.59	1.59	−0.15	0.14	−1.21	−0.28	−0.36	−1.51		
D	0.84	0.09	−0.03	0.02	0.38	−0.06	0.31	0.43	0.03		

In this article, we develop a novel DLSA algorithm that is able to perform large-scale statistical estimation and inference on a distributed system. The DLSA method can be applied to a large family of regression models (e.g., logistic regression, Poisson regression, and Cox's model). First, it is shown that the DLSA estimator is as statistically optimal as the global estimator. Moreover, it is computationally efficient and only requires one round of communication.

Furthermore, we develop the corresponding shrinkage estimation by using an adaptive Lasso approach. The oracle property is theoretically proven. A new DBIC measure for distributed variable selection, which only needs to be performed on the master and requires no further communication, is designed. We prove the DBIC measure to be selection consistent. Finally, numerical studies are conducted with five classical regression examples. In addition, a Spark toolbox is developed, which is shown to be computationally efficient both through simulation and in airline data analysis.

To facilitate future research, we now discuss several interesting topics. First, the DLSA method requires the objective function to have continuous second-order derivatives. This assumption might be restrictive and cannot be satisfied for certain regression models, e.g., the quantile regression. Consequently, the relaxation of this assumption can be investigated, and corresponding distributed algorithms should be designed for such regression models. Second, the dimension considered in our framework is finite. As a natural extension, one could study the shrinkage estimation properties in high-dimensional settings. Finally, the algorithm is designed for independent data. In practice, dependent data (e.g., time series data and network data) are frequently encountered. It is thus interesting to develop corresponding algorithms by considering the dependency structure.

APPENDIX A

Appendix A.1: Proof of Proposition 1

Note that $\widetilde{\theta} - \theta_0$ takes the form

$$\widetilde{\theta} - \theta_0 = \{ \sum_k \alpha_k \widehat{\Sigma}_k^{-1} \}^{-1} \{ \sum_k \alpha_k \widehat{\Sigma}_k^{-1} (\widehat{\theta}_k - \theta_0) \}.$$

Define $\widehat{\Sigma}_k(\theta) = \partial^2 \mathcal{L}_k(\theta)/\partial\theta\partial\theta^\top$. In the following section, we denote $\widehat{\Sigma}_k$ by $\widehat{\Sigma}_k(\widehat{\theta}_k)$ to make it clearer. By Slutsky's Theorem, to prove (2.4), it suffices to verify that

$$\sum_k \alpha_k \{\widehat{\Sigma}_k(\widehat{\theta}_k)\}^{-1} \to_p \Sigma^{-1}, \tag{A.1}$$

$$\sqrt{N}\Big[\sum_k \alpha_k \{\widehat{\Sigma}_k(\widehat{\theta}_k)\}^{-1} (\widehat{\theta}_k - \theta_0) \Big] = V^*(\theta_0) + B^*(\theta_0), \tag{A.2}$$

where $\mathrm{cov}\{V^*(\theta_0)\} = \Sigma^{-1}$ and $B^*(\theta_0) = O_p(K/\sqrt{N})$. We prove them in the following.

1. Proof of (A.1). Recall that $\widehat{\theta}_k$ is a $\sqrt{n}$-consistent estimator of θ_0. This enables us to conduct a Taylor's expansion of $\widehat{\Sigma}_k^{-1}(\widehat{\theta}_k)$ at θ_0, which yields

$$\begin{aligned}
&\widehat{\Sigma}_k^{-1}(\widehat{\theta}_k) - \Sigma_k^{-1} = \widehat{\Sigma}_k^{-1}(\widehat{\theta}_k) - \widehat{\Sigma}_k^{-1}(\theta_0) + \widehat{\Sigma}_k^{-1}(\theta_0) - \Sigma_k^{-1} \\
&= \sum_j \frac{\partial^3 \mathcal{L}_k(\theta)}{\partial\theta_j \partial\theta\partial\theta^\top}\Big|_{\theta=\theta^*} (\theta_j^* - \theta_j) + \widehat{\Sigma}_k^{-1}(\theta_0) - \Sigma_k^{-1}
\end{aligned}$$

where θ^* lies on the line joining θ_0 and $\widehat{\theta}_k$. By Condition (C5), we have $\frac{\partial^3 \mathcal{L}_k(\theta)}{\partial\theta_j \partial\theta\partial\theta^\top}\Big|_{\theta=\theta^*} = O_p(1)$. Therefore, the order of the first term is $O_p(1/\sqrt{n_k})$. In addition, we have $\widehat{\Sigma}_k^{-1}(\theta_0) - \Sigma_k^{-1} = \widehat{\Sigma}_k^{-1}(\theta_0) - E\{\widehat{\Sigma}_k^{-1}(\theta_0)\} = O_p(n_k^{-1/2})$. Consequently, it can be derived that $\widehat{\Sigma}_k^{-1}(\widehat{\theta}_k) - \Sigma_k^{-1} = O_p(n_k^{-1/2})$. Further note that $\alpha_k = n_k/N$ and $\sum_k \alpha_k = 1$. Then, we have

$$\widehat{\Sigma} - \Sigma = \sum_k \alpha_k [\{\widehat{\Sigma}_k(\widehat{\theta}_k)\}^{-1} - \Sigma_k^{-1}] = O_p(n^{-1/2}) = o_p(1). \tag{A.3}$$

Hence (A.1) is proven.

2. Proof of (A.2). Recall that $\widehat{\theta}_k$ is the local minimizer of $\mathcal{L}_k(\theta)$. Therefore, it holds that

$$0 = \frac{\partial \mathcal{L}_k(\theta)}{\partial \theta}\Big|_{\theta=\widehat{\theta}_k} = \frac{\partial \mathcal{L}_k(\theta)}{\partial \theta}\Big|_{\theta=\theta_0} + \frac{1}{n_k}\sum_{i\in\mathcal{S}_k} \frac{\partial^2 \mathcal{L}(\theta; Z_i)}{\partial\theta\partial\theta^\top}\Big|_{\theta=\theta_0}(\widehat{\theta}_k - \theta_0) + \frac{1}{2n_k}\sum_{i\in\mathcal{S}_k}\sum_{j=1}^{p}(\theta^* - \theta_0)^\top \frac{\partial^3 \mathcal{L}(\theta; Z_i)}{\partial\theta_j\partial\theta\partial\theta^\top}\Big|_{\theta=\theta^*}(\theta^* - \theta_0),$$

where θ^* lies between θ_0 and $\widehat{\theta}_k$. By standard arguments,

$$\begin{aligned}
&\frac{\partial \mathcal{L}_k(\theta)}{\partial \theta}\Big|_{\theta=\theta_0} = O_p(n_k^{-1/2}),\\
&\widehat{\Sigma}_k^{-1}(\theta_0) = E\Big\{\frac{\partial^2 \mathcal{L}(\theta; Z_i, i\in\mathcal{S}_k)}{\partial\theta\partial\theta^\top}\Big\}\Big|_{\theta=\theta_0} + O_p(n_k^{-1/2}) = \Sigma_k^{-1} + O_p(n_k^{-1/2}).\\
&\frac{\partial^3 \mathcal{L}_k(\theta)}{\partial\theta_j\partial\theta\partial\theta^\top}\Big|_{\theta=\theta^*} = E\Big\{\frac{\partial^3 \mathcal{L}(\theta; Z_i, i\in\mathcal{S}_k)}{\partial\theta\partial\theta^\top}\Big|_{\theta=\theta^*}\Big\} + o_p(1).
\end{aligned}$$

Further note that $\widehat{\theta}_k - \theta_0 = O_p(n_k^{-1/2})$. Then, we have

$$\widehat{\theta}_k - \theta_0 = -\Sigma_k \frac{\partial \mathcal{L}_k(\theta)}{\partial\theta}\Big|_{\theta=\theta_0} + \frac{B_k(\theta_0)}{n_k} + O_p\Big(\frac{1}{n_k}\Big),$$

where $B_k(\theta_0) = O(1)$ is the bias term. Then, it holds that

$$\begin{aligned}
&\sqrt{N}\sum_k \alpha_k \widehat{\Sigma}_k^{-1}(\widehat{\theta}_k)(\widehat{\theta}_k - \theta_0)\\
&= \sqrt{N}\sum_k \alpha_k \Sigma_k^{-1}\Big\{ - \Sigma_k \frac{\partial \mathcal{L}_k(\theta)}{\partial\theta}\Big|_{\theta=\theta_0} + \frac{B_k(\theta_0)}{n_k} + O_p(n_k^{-1})\Big\}\\
&+ \sqrt{N}\sum_k \alpha_k \{\widehat{\Sigma}_k^{-1}(\widehat{\theta}_k) - \Sigma_k^{-1}\}(\widehat{\theta}_k - \theta_0)\\
&= -\frac{1}{\sqrt{N}}\sum_k n_k \frac{\partial \mathcal{L}_k(\theta)}{\partial\theta}\Big|_{\theta=\theta_0} + \frac{1}{\sqrt{N}}\sum_k \Sigma_k^{-1} B_k(\theta_0) + O_p\Big(\frac{K}{\sqrt{N}}\Big), \qquad \text{(A.4)}
\end{aligned}$$

where the second equation is implied by $\sqrt{N}\sum_k \alpha_k\{\widehat{\Sigma}_k^{-1}(\widehat{\theta}_k)-\Sigma_k^{-1}\}(\widehat{\theta}_k-\theta_0) = O_p(K/\sqrt{N})$. By condition (C4), it can be concluded that $\mathrm{cov}\{\frac{1}{\sqrt{N}}\sum_k n_k \frac{\partial \mathcal{L}_k(\theta)}{\partial \theta}|_{\theta=\theta_0}\} = \Sigma^{-1}$. Consequently, (A.2) can be proven.

Appendix A.2: Proof of Theorem 1

By Slutsky's Theorem, the asymptotic normality is directly implied by (A.1) and (A.2) with $V^*(\theta_0) \to_d N(0, \Sigma^{-1})$ and $B^*(\theta_0) = o_p(1)$. First, by Condition (C6) and the Lyapunov central limit theorem, we have $V^*(\theta_0) \to_d N(0, \Sigma^{-1})$. Next, by the condition that $n \gg \sqrt{N}$, we have $K \gg \sqrt{N}$, and thus, $B^*(\theta_0) = o_p(1)$.

Appendix A.3: Proof of Theorem 2

1. Proof of $\sqrt{N}$-consistency.

Note that the objective function $Q_\lambda(\theta)$ in (3.1) is a strictly convex function. Then, the local minimizer is also a global minimizer. To establish $\sqrt{N}$-consistency results, it suffices to verify the following result (Fan and Li, 2001), i.e., for an arbitrarily small $\epsilon > 0$, there exists a sufficiently large constant C such that

$$\liminf_N P\Big\{\inf_{u\in\mathbb{R}^p:\|u\|=C} Q_\lambda(\theta_0 + N^{-1/2}u) > Q(\theta_0)\Big\} > 1-\epsilon. \tag{A.5}$$

Let $u = (u_1, \cdots, u_p)^\top$ and $\widehat{\Delta}_N = \sum_k \alpha_k \widehat{\Sigma}_k^{-1}(\widehat{\theta}_k)\{\sqrt{N}(\theta_0 - \widehat{\theta}_k)\}$, Then, we have

$$
\begin{aligned}
& N\Big\{Q_\lambda(\theta_0 + N^{-1/2}u) - Q_\lambda(\theta_0)\Big\} \\
& = u^\top \widehat{\Sigma}^{-1} u + 2u^\top \widehat{\Delta}_N + N\sum_{j=1}^{p} \lambda_j |\theta_{0j} + N^{-1/2}u_j| - N\sum_{j=1}^{p} \lambda_j |\theta_{0j}| \\
& \geq u^\top \widehat{\Sigma}^{-1} u + 2u^\top \widehat{\Delta}_N + N\sum_{j=1}^{d_0} \lambda_j \big(|\theta_{0j} + N^{-1/2}u_j| - |\theta_{0j}|\big) \\
& \geq u^\top \widehat{\Sigma}^{-1} u + 2u^\top \widehat{\Delta}_N - d_0(\sqrt{N}a_N)\|u\|. \qquad \text{(A.6)}
\end{aligned}
$$

where the second equality holds because we assume $\theta_{0j} = 0$ for $j > d_0$. Further note that we assume that $\sqrt{N}a_N \to_p 0$. Consequently, the last term (A.6) is $o_p(1)$. Next, note that the first term of (A.6) is lower bounded by $\lambda_{\max}^{-1}(\widehat{\Sigma})C^2$ because $\|u\| = C$. By (A.3), we have $\lambda_{\max}(\widehat{\Sigma}) \to_p \lambda_{\max}(\Sigma)$. Consequently, with probability tending to 1, we have the first term uniformly larger than $0.5\lambda_{\max}^{-1}(\Sigma)C^2$, which is positive due to Condition (C4). In addition, by $K/\sqrt{N} \to 0$, we have $\widehat{\Delta}_N = O_p(1)$. Consequently, as long as C is sufficiently large, the first term will dominate the last two terms. Then, the result of (A.5) is proven.

2. Proof of Selection Consistency.

It suffices to verify that $P(\widetilde{\theta}_{\lambda,j} = 0) \to 1$ for any $d_0 < j \leq p$. Note that $Q_\lambda(\theta)$ can be rewritten as

$$
Q_\lambda(\theta) = (\theta - \widetilde{\theta})^\top \widehat{\Sigma}^{-1}(\theta - \widetilde{\theta}) + \sum_j \lambda_j |\theta_j| + C,
$$

where C is a constant. Define $\widehat{\Omega} = \widehat{\Sigma}^{-1}$, and $\widehat{\Omega}^{(j)}$ denotes the jth row of the matrix $\widehat{\Omega}$. If $\widetilde{\theta}_{\lambda,j} \neq 0$ for some $j > d_0$, then the partial derivative can be calculated as

$$
\sqrt{N}\frac{\partial Q_\lambda(\theta)}{\partial \theta_j}\Big|_{\theta=\widetilde{\theta}_\lambda} = 2\widehat{\Omega}^{(j)\top}\sqrt{N}(\widetilde{\theta}_\lambda - \widetilde{\theta}) + \sqrt{N}\lambda_j \text{sign}(\widetilde{\theta}_{\lambda,j}). \qquad \text{(A.7)}
$$

Note that $\widehat{\Omega} \to_p \Sigma^{-1}$ and $\sqrt{N}(\widetilde{\theta}_\lambda - \widetilde{\theta}) = \sqrt{N}(\widetilde{\theta}_\lambda - \theta_0) - \sqrt{N}(\widetilde{\theta} - \theta_0) = O_p(1)$, by (A.1), Theorem 1, and Theorem 2 (a). Consequently, the first term (A.7) is $O_p(1)$. Next, by this condition, we know that $\sqrt{N}\lambda_j \geq \sqrt{N}b_\lambda \to \infty$ for $j > d_0$. Because $\widetilde{\theta}_{\lambda,j} \neq 0$, we have $\text{sign}(\widetilde{\theta}_{\lambda,j}) = 1$ or -1; thus, the second term (A.7) goes to infinity. Obviously, the equation will not be equal to zero. This implies $P(\widetilde{\theta}_{\lambda,j} = 0) \to 1$ as a result.

Appendix A.3: Proof of Theorem 3

We first rewrite the asymptotic covariance Σ into the following block matrix form:

$$\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix},$$

where $\Sigma_{11} \in \mathbb{R}^{d_0 \times d_0}$. Similarly, we partition its inverse matrix Ω into 4 corresponding parts, $\Omega = (\Omega_{11}, \Omega_{12}; \Omega_{21}, \Omega_{22})$. By Theorem 2, with probability tending to 1, we have $\widetilde{\theta}_\lambda^{(-\mathcal{M}_T)} = 0$. Therefore, $\widetilde{\theta}_\lambda^{(\mathcal{M}_T)}$ should be the global minimizer of the objective function,

$$\begin{aligned} Q_{\lambda,0}(\theta^{(\mathcal{M}_T)}) =& (\theta^{(\mathcal{M}_T)} - \widetilde{\theta}^{(\mathcal{M}_T)})^\top \widehat{\Omega}_{11}(\theta^{(\mathcal{M}_T)} - \widetilde{\theta}^{(\mathcal{M}_T)}) - 2(\theta^{(\mathcal{M}_T)} - \widetilde{\theta}^{(\mathcal{M}_T)})^\top \widehat{\Omega}_{12}\widetilde{\theta}^{(-\mathcal{M}_T)} \\ &+ \widetilde{\theta}^{(-\mathcal{M}_T)\top}\widehat{\Omega}_{22}\widetilde{\theta}^{(-\mathcal{M}_T)} + \sum_{j=1}^{d_0} \lambda_j |\theta_j| \end{aligned}$$

By Theorem 2, it can be concluded that with probability tending to 1, $\widetilde{\theta}_\lambda^{(\mathcal{M}_T)}$ should be nonzero (otherwise, the $\sqrt{N}$-consistency result in Theorem 2 will not hold). As a result, The partial derivative $\partial Q_\lambda(\theta)/\partial\theta_j$ should exist for $1 \leq j \leq d_0$, which yields

$$0 = \frac{1}{2}\frac{\partial Q_{\lambda,0}(\theta^{(\mathcal{M}_T)})}{\partial \theta^{(\mathcal{M}_T)}}\Big|_{\theta^{(\mathcal{M}_T)} = \widetilde{\theta}_\lambda^{(\mathcal{M}_T)}} = \widehat{\Omega}_{11}(\widetilde{\theta}_\lambda^{(\mathcal{M}_T)} - \widetilde{\theta}^{(\mathcal{M}_T)}) - \widehat{\Omega}_{12}\widetilde{\theta}^{(-\mathcal{M}_T)} + D(\widetilde{\theta}_\lambda^{(\mathcal{M}_T)}). \tag{A.8}$$

where $D(\widetilde{\theta}_{\lambda}^{(\mathcal{M}_T)})$ is a d_0-dimensional vector, with its jth component given by $0.5\lambda_j \text{sign}(\widetilde{\theta}_{\lambda,j})$. By (A.8), it can be derived that

$$\begin{aligned}\sqrt{N}(\widetilde{\theta}_{\lambda}^{(\mathcal{M}_T)} - \theta_0^{(\mathcal{M}_T)}) &= \sqrt{N}(\widetilde{\theta}^{(\mathcal{M}_T)} - \theta_0^{(\mathcal{M}_T)}) + \widehat{\Omega}_{11}^{-1}\widehat{\Omega}_{12}(\sqrt{N}\widetilde{\theta}^{(-\mathcal{M}_T)}) - \widehat{\Omega}_{11}^{-1}\sqrt{N}D(\widetilde{\theta}_{\lambda}^{(\mathcal{M}_T)}) \\ &= \sqrt{N}(\widetilde{\theta}^{(\mathcal{M}_T)} - \theta_0^{(\mathcal{M}_T)}) + \Omega_{11}^{-1}\Omega_{12}(\sqrt{N}\widetilde{\theta}^{(-\mathcal{M}_T)}) + o_p(1), \qquad \text{(A.9)}\end{aligned}$$

where the second equality follows because $\sqrt{N}\widetilde{\theta}^{(-\mathcal{M}_T)} = O_p(1)$ by Theorem 1, $\widehat{\Omega}_{11} \to_p \Omega_{11}$ and $\widehat{\Omega}_{12} \to_p \Omega_{12}$ by (A.1), and $\sqrt{N}\lambda_j = o_p(1)$ $(1 \le j \le d_0)$ by Theorem 2. Furthermore, by the matrix inverse formula, it holds that $\Omega_{11}^{-1}\Omega_{12} = -\Sigma_{12}\Sigma_{22}^{-1}$. Consequently, we have

$$\sqrt{N}(\widetilde{\theta}_{\lambda}^{(\mathcal{M}_T)} - \theta_0^{(\mathcal{M}_T)}) = \sqrt{N}(\widetilde{\theta}^{(\mathcal{M}_T)} - \theta_0^{(\mathcal{M}_T)}) - \Sigma_{12}\Sigma_{22}^{-1}(\sqrt{N}\widetilde{\theta}^{(-\mathcal{M}_T)}) + o_p(1).$$

By Theorem 1, we have that the above is asymptotically normal with a mean of 0, and the inverse asymptotic covariance matrix is given by $(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})^{-1} = \Omega_{11}$. By condition (C6), we have $\Omega_{11} = \Omega_{\mathcal{M}_T}$. Consequently, the estimator $\widetilde{\theta}_{\lambda}^{(\mathcal{M})}$ shares the same asymptotic distribution with the oracle estimator $\widehat{\theta}_{\mathcal{M}_T}^{(\mathcal{M}_T)}$.

Appendix A.4: Proof of Theorem 4

To establish the selection consistency property of the DBIC, we consider the following two cases for any $\mathcal{M}_\lambda \neq \mathcal{M}_T$. The first case is the underfitted case, and the second case is the overfitted case.

1. Underfitted Model. Note that λ_N satisfies the condition in Theorem 2. Consequently, we have that $\widetilde{\theta}_{\lambda_N}$ is $\sqrt{N}$-consistent. We thus also have $\text{DBIC}_{\lambda_N} = o_p(1)$.

For $\mathcal{M} \not\supset \mathcal{M}_T$, it can be derived that

$$\begin{aligned}\text{DBIC}_\lambda &= (\widetilde{\theta}_\lambda - \widetilde{\theta})^\top \widehat{\Sigma}^{-1} (\widetilde{\theta}_\lambda - \widetilde{\theta}) + df_\lambda (\log N)/N \\ &\geq (\widetilde{\theta}_\lambda - \widetilde{\theta})^\top \widehat{\Sigma}^{-1} (\widetilde{\theta}_\lambda - \widetilde{\theta})^\top\end{aligned}$$

Define $\widetilde{\theta}_{\mathcal{M}} = \arg\min_{\theta \in \mathbb{R}^p : \theta_j = 0, \forall j \notin \mathcal{M}} (\theta - \widetilde{\theta})^\top \widehat{\Sigma}^{-1} (\theta - \widetilde{\theta})$ as the unpenalized estimator given the model identified by $\mathcal{M}$. Consequently, by definition, we should have

$$\begin{aligned}&(\widetilde{\theta}_\lambda - \widetilde{\theta})^\top \widehat{\Sigma}^{-1} (\widetilde{\theta}_\lambda - \widetilde{\theta})^\top \geq (\widetilde{\theta}_{\mathcal{M}_\lambda} - \widetilde{\theta})^\top \widehat{\Sigma}^{-1} (\widetilde{\theta}_{\mathcal{M}_\lambda} - \widetilde{\theta}) \\ &\geq \min_{\mathcal{M} \not\supset \mathcal{M}_T} (\widetilde{\theta}_{\mathcal{M}} - \widetilde{\theta})^\top \widehat{\Sigma}^{-1} (\widetilde{\theta}_{\mathcal{M}} - \widetilde{\theta}) \to_p \min_{\mathcal{M} \not\supset \mathcal{M}_T} (\widetilde{\theta}_{\mathcal{M}} - \widetilde{\theta})^\top \Sigma^{-1} (\widetilde{\theta}_{\mathcal{M}} - \widetilde{\theta}),\end{aligned}$$

where the last convergence is due to (A.1). Because Σ is positive definite by Condition (C4), we have $\min_{\mathcal{M} \not\supset \mathcal{M}_T} (\widetilde{\theta}_{\mathcal{M}} - \widetilde{\theta})^\top \Sigma^{-1} (\widetilde{\theta}_{\mathcal{M}} - \widetilde{\theta}) > 0$ with probability tending to 1. One could then conclude immediately that $P(\inf_{\lambda \in \mathbb{R}_-} \text{DBIC}_\lambda > \text{DBIC}_{\lambda_N}) \to 1$.

2. Overfitted Model. We next consider the overfitted model. In contrast, let λ be an arbitrary tuning parameter that over selects the parameters. We then have $df_\lambda - d_0 \geq 1$. It can then be concluded that $N(\text{DBIC}_\lambda - \text{DBIC}_{\lambda_N}) =$

$$\begin{aligned}&N(\widetilde{\theta}_\lambda - \widetilde{\theta})^\top \widehat{\Sigma}^{-1} (\widetilde{\theta}_\lambda - \widetilde{\theta}) - N(\widetilde{\theta}_{\lambda_N} - \widetilde{\theta})^\top \widehat{\Sigma}^{-1} (\widetilde{\theta}_{\lambda_N} - \widetilde{\theta}) + (df_\lambda - d_0) \log N \\ &\geq N(\widetilde{\theta}_{\mathcal{M}_\lambda} - \widetilde{\theta})^\top \widehat{\Sigma}^{-1} (\widetilde{\theta}_{\mathcal{M}_\lambda} - \widetilde{\theta}) - N(\widetilde{\theta}_{\lambda_N} - \widetilde{\theta})^\top \widehat{\Sigma}^{-1} (\widetilde{\theta}_{\lambda_N} - \widetilde{\theta}) + \log N \\ &\geq \inf_{\mathcal{M} \supset \mathcal{M}_T} N(\widetilde{\theta}_{\mathcal{M}} - \widetilde{\theta})^\top \widehat{\Sigma}^{-1} (\widetilde{\theta}_{\mathcal{M}} - \widetilde{\theta}) - N(\widetilde{\theta}_{\lambda_N} - \widetilde{\theta})^\top \widehat{\Sigma}^{-1} (\widetilde{\theta}_{\lambda_N} - \widetilde{\theta}) + \log N. \qquad \text{(A.10)}\end{aligned}$$

First note that for $\mathcal{M} \supset \mathcal{M}_T$, we have that $\widetilde{\theta}_{\mathcal{M}}$ is $\sqrt{N}$-consistent. As a result, the first term of (A.10) is $O_p(1)$. Similarly, by Theorem 2, $\widetilde{\theta}_{\lambda_N}$ is $\sqrt{N}$-consistent, Thus, the second term (A.10) is also $O_p(1)$. As a result, (A.10) diverges to infinity as $N \to \infty$. This implies that $P(\inf_{\lambda \in \mathbb{R}_+} \text{DBIC}_\lambda > \text{DBIC}_{\lambda_N}) \to 1$. This completes the proof.

Algorithm 1: Spark implementation

Input: The model function for modelling each partitioned dataset

Output: The weighted least squares estimator $\tilde{\theta}$, covariance matrix $\widehat{\Sigma}$, DBIC DBIC$_\lambda$

Steps:

Step (1). Pre-determine the overall cluster available memory as M_{ram}, the total number of CPU cores as C_{cores}, and the total data size to be processed as D_{total};

Step (2) Define the number of batched chunks N_{chunks} to allow for out-of-memory data processing. We recommend that N_{chunks} be at least greater than $3 \times D_{total}/M_{ram}$ in a Spark system.

Step (3). Define the number of partitions $P_{partition} = D_{total}/(N_{chunks} \times C_{cores})$.

Step (4). Define a *model function* whereby the input is an $n \times (p+2)$ Python Pandas DataFrame containing the response variable, covariates and partition id, and the output is a $p \times (p+1)$ Pandas DataFrame whereby the first column is $\widehat{\theta}_k$ and the remaining columns store $\widehat{\Sigma}_k^{-1}$.

Step (5).

for *i in 1:*N_{chunks} **do**

- **(a)**. Transfer the data chunk to Spark's distributed DataFrame if the data are stored in another format.
- **(b)**. Randomly assign an integer partition label from $\{1, ..., P_{partition}\}$ to each row of the Spark DataFrame.
- **(c)**. Repartition the DataFrame in the distributed system if the data are not partitioned by the partition label.
- **(d)**. Group the Spark DataFrames by the assigned partition label.
- **(e)**. Apply the model function to each grouped dataset with Spark's *Grouped map Pandas UDFs* API and obtain a $(pP_{partition}) \times (p+1)$ distributed Spark DataFrame R_i.

end

Step (6). Aggregate R_i over both partitions and chunks and return the $p \times (p+1)$ matrix R_{final}.

Step (7). Return $\tilde{\theta}$ by Equation (2.2), $\widehat{\Sigma}$, and DBIC$_\lambda$ by Equation (3.4).

- Because the final step in the DLSA algorithm is carried out on the master node and because data transformation from worker nodes to the master node is required, a special tool called "Apache Arrow" (`https://arrow.apache.org/`) is plugged-in to our system to allow efficient data transformation between Sparks distributed DataFrame and Python's Pandas DataFrame.

References

Apache Software Foundation (2019a), "Apache Hadoop," .

— (2019b), "Apache Spark," .

Battey, H., Fan, J., Liu, H., Lu, J., and Zhu, Z. (2015), "Distributed estimation and inference with statistical guarantees," *arXiv preprint arXiv:1509.05457*.

Cameron, A. C. and Trivedi, P. K. (2013), *Regression analysis of count data*, vol. 53, Cambridge university press.

Chang, X., Lin, S.-B., and Wang, Y. (2017a), "Divide and conquer local average regression," *Electronic Journal of Statistics*, 11, 1326–1350.

Chang, X., Lin, S.-B., and Zhou, D.-X. (2017b), "Distributed semi-supervised learning with kernel ridge regression," *The Journal of Machine Learning Research*, 18, 1493–1514.

Chen, J. and Chen, Z. (2008), "Extended Bayesian information criteria for model selection with large model spaces," *Biometrika*, 95, 759–771.

Chen, T. and Guestrin, C. (2016), "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, ACM, pp. 785–794.

Chen, X., Liu, W., and Zhang, Y. (2018), "Quantile regression under memory constraint," *arXiv preprint arXiv:1810.08264*.

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

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

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

— (2002), "Variable selection for Cox's proportional hazards model and frailty model," *The Annals of Statistics*, 30, 74–99.

Fan, J., Wang, D., Wang, K., and Zhu, Z. (2017), "Distributed estimation of principal eigenspaces," *arXiv preprint arXiv:1702.06488*.

Harrell Jr, F. E. (2015), *Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis*, Springer.

Heinze, C., McWilliams, B., and Meinshausen, N. (2016), "Dual-loco: Distributing statistical estimation using random projections," in *Artificial Intelligence and Statistics*, pp. 875–883.

Hosmer Jr, D. W., Lemeshow, S., and Sturdivant, R. X. (2013), *Applied logistic regression*, vol. 398, John Wiley & Sons.

Jordan, M. I., Lee, J. D., and Yang, Y. (2018), "Communication-efficient distributed statistical inference," *Journal of the American Statistical Association*, 1–14.

Lee, J. D., Sun, Y., Liu, Q., and Taylor, J. E. (2015), "Communication-efficient sparse regression: a one-shot approach," *arXiv preprint arXiv:1503.04337*.

Lehmann, E. L. and Casella, G. (2006), *Theory of point estimation*, Springer Science & Business Media.

Li, X., Li, R., Xia, Z., and Xu, C. (2019), "Distributed feature screening via componentwise debiasing," *arXiv preprint arXiv:1903.03810*.

Liu, Q. and Ihler, A. T. (2014), "Distributed estimation, information loss and exponential families," in *Advances in neural information processing systems*, pp. 1098–1106.

Shamir, O., Srebro, N., and Zhang, T. (2014), "Communication-efficient distributed optimization using an approximate newton-type method," in *International conference on machine learning*, pp. 1000–1008.

Shao, J. (1997), "An asymptotic theory for linear model selection," *Statistica Sinica*, 221–242.

Smith, V., Forte, S., Ma, C., Takáč, M., Jordan, M. I., and Jaggi, M. (2018), "CoCoA: A General Framework for Communication-Efficient Distributed Optimization," *Journal of Machine Learning Research*, 18, 1–49.

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

Wang, H. and Leng, C. (2007), "Unified LASSO estimation by least squares approximation," *Journal of the American Statistical Association*, 102, 1039–1048.

Wang, H., Li, R., and Tsai, C.-L. (2007), "Tuning parameter selectors for the smoothly clipped absolute deviation method," *Biometrika*, 94, 553–568.

Wang, H., Zhu, R., and Ma, P. (2018), "Optimal subsampling for large sample logistic regression," *Journal of the American Statistical Association*, 113, 829–844.

Wang, J., Kolar, M., Srebro, N., and Zhang, T. (2017a), "Efficient distributed learning with sparsity," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, JMLR. org, pp. 3636–3645.

Wang, J., Wang, W., and Srebro, N. (2017b), "Memory and communication efficient distributed stochastic optimization with minibatch prox," in *Proceedings of the 2017 Conference on Learning Theory*, eds. Kale, S. and Shamir, O., Amsterdam, Netherlands: PMLR, vol. 65 of *Proceedings of Machine Learning Research*, pp. 1882–1919.

Wang, L., Kim, Y., and Li, R. (2013), "Calibrating non-convex penalized regression in ultra-high dimension," *Annals of statistics*, 41, 2505.

Yang, J., Mahoney, M. W., Saunders, M., and Sun, Y. (2016), "Feature-distributed sparse regression: a screen-and-clean approach," in *Advances in Neural Information Processing Systems*, pp. 2712–2720.

Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., Franklin, M. J., Shenker, S., and Stoica, I. (2012), "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," in *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, USENIX Association, pp. 2–2.

Zhang, C.-H. (2010), "Nearly unbiased variable selection under minimax concave penalty," *Annals of Statistics*, 38, 894–942.

Zhang, H. H. and Lu, W. (2007), "Adaptive Lasso for Cox's proportional hazards model," *Biometrika*, 94, 691–703.

Zhang, T. (2004), "Solving large scale linear prediction problems using stochastic gradient descent algorithms," in *Proceedings of the twenty-first international conference on Machine learning*, ACM, p. 116.

Zhang, Y., Duchi, J. C., and Wainwright, M. J. (2013), "Communication-efficient

algorithms for statistical optimization," *The Journal of Machine Learning Research*, 14, 3321–3363.

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

Zou, H. and Li, R. (2008), "One-step sparse estimates in nonconcave penalized likelihood models," *Annals of statistics*, 36, 1509.

Zou, H. and Zhang, H. H. (2009), "On the adaptive elastic-net with a diverging number of parameters," *Annals of statistics*, 37, 1733.

Table 2: Simulation results for Example 1 with 500 replications. The numerical performances are evaluated for different sample sizes N ($\times 10^3$) and numbers of workers K. For the OS, CSL, and DLSA method, the REE_j is reported. For the shrinkage DLSA (SDLSA) method, the REE_j^s is reported. Finally, the MS and CM are reported for the SDLSA method to evaluate the variable selection accuracy.

N	K	Est.	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	MS	CM
Setting 1: i.i.d Covariates												
10	5	OS	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00		
		CSL	1.00	0.99	1.00	0.98	0.99	0.99	0.99	0.99		
		DLSA	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00		
		SDLSA	1.00	-	-	1.00	-	-	1.00	-	3.01	1.00
20	5	OS	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00		
		CSL	0.99	1.00	0.99	0.99	0.99	1.00	0.99	1.00		
		DLSA	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00		
		SDLSA	1.00	-	-	1.00	-	-	1.00	-	3.01	1.00
100	10	OS	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00		
		CSL	0.99	0.99	1.00	0.99	0.99	0.99	0.99	0.99		
		DLSA	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00		
		SDLSA	1.00	-	-	1.00	-	-	1.00	-	3.01	1.00
Setting 2: Heterogeneous Covariates												
10	5	OS	1.00	0.99	0.99	1.00	0.99	0.98	0.99	0.99		
		CSL	0.97	0.97	0.98	0.98	0.98	1.00	0.98	0.98		
		DLSA	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00		
		SDLSA	1.16	-	-	1.25	-	-	1.25	-	3.01	1.00
20	5	OS	0.99	0.99	0.99	1.01	0.99	0.99	0.99	0.99		
		CSL	0.99	0.99	0.98	0.97	0.98	1.00	1.00	0.99		
		DLSA	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00		
		SDLSA	1.15	-	-	1.23	-	-	1.20	-	3.01	1.00
100	10	OS	0.99	0.99	1.00	1.00	0.99	0.99	0.99	0.99		
		CSL	0.98	0.97	0.95	0.98	0.96	0.97	0.96	0.95		
		DLSA	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00		
		SDLSA	1.15	-	-	1.22	-	-	1.28	-	3.00	1.00

Table 3: Simulation results for Example 2 with 500 replications. The numerical performances are evaluated for different sample sizes N ($\times 10^3$) and numbers of workers K. For the OS, CSL, and DLSA method, the REE_j is reported. For the shrinkage DLSA (SDLSA) method, the REE_j^s is reported. Finally, the MS and CM are reported for the SDLSA method to evaluate the variable selection accuracy.

N	K	Est.	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	MS	CM
Setting 1: i.i.d Covariates												
10	5	OS	0.92	0.98	0.99	0.95	0.98	0.98	0.94	0.99		
		CSL	0.94	1.00	0.99	0.96	0.97	0.99	0.94	0.98		
		DLSA	0.98	1.01	1.01	0.99	1.01	1.01	0.98	1.01		
		SDLSA	0.91	-	-	0.92	-	-	0.91	-	3.01	1.00
20	5	OS	0.95	1.00	1.00	0.96	0.99	0.99	0.96	0.99		
		CSL	0.99	0.99	0.99	1.00	1.00	0.99	0.99	0.99		
		DLSA	1.00	1.00	1.00	1.01	1.00	1.00	1.00	1.00		
		SDLSA	0.96	-	-	1.00	-	-	0.97	-	3.01	1.00
100	10	OS	0.95	1.00	1.00	0.97	1.00	1.00	0.97	1.00		
		CSL	0.99	0.99	1.00	0.99	1.00	1.00	0.98	0.99		
		DLSA	1.01	1.00	1.00	1.00	1.00	1.00	0.99	1.00		
		SDLSA	1.00	-	-	1.00	-	-	0.98	-	3.01	1.00
Setting 2: Heterogeneous Covariates												
10	5	OS	0.79	0.86	0.89	0.83	0.88	0.86	0.82	0.90		
		CSL	0.24	0.31	0.35	0.27	0.33	0.30	0.26	0.28		
		DLSA	0.96	1.01	1.01	0.98	1.01	1.01	0.99	1.01		
		SDLSA	0.86	-	-	1.00	-	-	1.00	-	3.00	1.00
20	5	OS	0.86	0.87	0.88	0.87	0.90	0.94	0.89	0.89		
		CSL	0.33	0.37	0.41	0.32	0.36	0.37	0.32	0.32		
		DLSA	0.98	1.01	1.00	0.99	1.00	1.00	0.97	1.01		
		SDLSA	0.96	-	-	1.07	-	-	1.03	-	3.01	1.00
100	10	OS	0.92	0.94	0.92	0.93	0.91	0.92	0.91	0.92		
		CSL	0.21	0.24	0.24	0.21	0.23	0.25	0.21	0.21		
		DLSA	0.96	1.00	1.00	1.00	1.00	1.00	0.97	1.00		
		SDLSA	0.97	-	-	1.09	-	-	1.07	-	3.00	1.00

Table 4: Simulation results for Example 3 with 500 replications. The numerical performances are evaluated for different sample sizes N ($\times 10^3$) and numbers of workers K. For the OS, CSL, and DLSA method, the REE_j is reported. For the shrinkage DLSA (SDLSA) method, the REE_j^s is reported. Finally, the MS and CM are reported for the SDLSA method to evaluate the variable selection accuracy.

N	K	Est.	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	MS	CM
Setting 1: i.i.d Covariates												
10	5	OS	0.98	0.99	1.00	0.99	1.00	1.00	0.99	0.99		
		CSL	0.93	0.95	0.94	0.89	0.93	0.95	0.94	0.93		
		DLSA	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00		
		SDLSA	1.00	-	-	1.00	-	-	1.00	-	3.01	1.00
20	5	OS	1.00	0.99	1.00	0.99	0.99	0.99	0.99	1.00		
		CSL	0.94	0.98	0.97	0.96	0.98	0.96	0.96	0.99		
		DLSA	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00		
		SDLSA	1.00	-	-	1.01	-	-	1.00	-	3.00	1.00
100	10	OS	1.00	1.00	1.00	0.99	1.00	1.00	0.99	0.99		
		CSL	0.97	0.98	0.97	0.95	0.96	0.98	0.98	0.97		
		DLSA	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00		
		SDLSA	1.00	-	-	1.00	-	-	1.00	-	3.00	1.00
Setting 2: Heterogeneous Covariates												
10	5	OS	0.59	0.62	0.68	0.59	0.65	0.64	0.68	0.68		
		CSL	0.01	0.02	0.03	0.01	0.03	0.03	0.02	0.03		
		DLSA	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00		
		SDLSA	1.14	-	-	1.27	-	-	1.26	-	3.00	1.00
20	5	OS	0.59	0.67	0.65	0.62	0.63	0.63	0.70	0.64		
		CSL	0.01	0.03	0.03	0.01	0.03	0.03	0.02	0.03		
		DLSA	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00		
		SDLSA	1.18	-	-	1.28	-	-	1.31	-	3.00	1.00
100	10	OS	0.62	0.68	0.75	0.67	0.69	0.69	0.75	0.69		
		CSL	0.01	0.02	0.02	0.01	0.03	0.03	0.02	0.03		
		DLSA	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00		
		SDLSA	1.15	-	-	1.28	-	-	1.23	-	3.01	1.00

Table 5: Simulation results for Example 4 with 500 replications. The numerical performances are evaluated for different sample sizes N ($\times 10^3$) and numbers of workers K. For the OS, CSL, and DLSA method, the REE_j is reported. For the shrinkage DLSA (SDLSA) method, the REE_j^s is reported. Finally, the MS and CM are reported for the SDLSA method to evaluate the variable selection accuracy.

N	K	Est.	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	MS	CM
Setting 1: i.i.d Covariates												
10	5	OS	0.97	0.99	0.99	0.95	0.99	0.98	0.99	0.99		
		CSL	0.97	0.99	0.98	0.99	0.99	0.98	0.96	0.99		
		DLSA	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00		
		SDLSA	1.00	-	-	1.00	-	-	1.00	-	3.01	1.00
20	5	OS	0.98	1.00	1.00	0.97	1.00	0.99	0.99	1.00		
		CSL	0.99	0.99	0.98	1.00	0.99	0.97	0.99	0.98		
		DLSA	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00		
		SDLSA	0.99	-	-	1.00	-	-	1.00	-	3.01	1.00
100	10	OS	1.00	1.00	0.99	0.99	1.00	1.00	1.00	1.00		
		CSL	0.98	0.98	1.00	0.99	0.99	0.99	0.99	1.00		
		DLSA	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00		
		SDLSA	1.00	-	-	1.00	-	-	1.00	-	3.00	1.00
Setting 2: Heterogeneous Covariates												
10	5	OS	0.65	0.75	0.71	0.72	0.74	0.75	0.70	0.70		
		CSL	0.04	0.05	0.05	0.04	0.05	0.05	0.05	0.04		
		DLSA	0.95	0.98	0.99	0.98	0.99	0.98	0.97	0.95		
		SDLSA	1.01	-	-	1.04	-	-	1.05	-	3.00	1.00
20	5	OS	0.70	0.80	0.77	0.73	0.75	0.74	0.75	0.70		
		CSL	0.05	0.06	0.06	0.05	0.05	0.05	0.06	0.05		
		DLSA	0.96	0.99	0.99	0.98	0.98	0.99	0.99	0.95		
		SDLSA	1.01	-	-	1.10	-	-	1.12	-	3.00	1.00
100	10	OS	0.80	0.78	0.80	0.79	0.78	0.81	0.77	0.78		
		CSL	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.03		
		DLSA	0.98	0.98	1.00	0.98	0.98	0.99	0.98	0.97		
		SDLSA	1.04	-	-	1.07	-	-	1.08	-	3.00	1.00

Table 6: Simulation results for Example 5 with 500 replications. The numerical performances are evaluated for different sample sizes N ($\times 10^3$) and numbers of workers K. For the OS, CSL, and DLSA method, the REE_j is reported. For the shrinkage DLSA (SDLSA) method, the REE_j^s is reported. Finally, the MS and CM are reported for the SDLSA method to evaluate the variable selection accuracy.

N	K	Est.	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	MS	CM
Setting 1: i.i.d Covariates												
10	5	OS	0.97	0.99	0.99	0.95	0.99	0.99	0.99	0.99		
		CSL	0.99	1.00	0.99	0.99	0.99	0.99	0.98	0.99		
		DLSA	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00		
		SDLSA	1.00	-	-	1.00	-	-	1.00	-	3.01	1.00
20	5	OS	0.99	1.00	1.00	0.98	1.00	1.00	0.99	1.00		
		CSL	0.99	1.00	0.99	0.99	1.00	1.00	0.99	0.99		
		DLSA	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00		
		SDLSA	1.00	-	-	1.00	-	-	1.00	-	3.01	1.00
100	10	OS	0.99	1.00	1.00	0.98	1.00	1.00	1.00	1.00		
		CSL	0.98	0.99	1.00	0.99	0.99	0.99	0.99	0.99		
		DLSA	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00		
		SDLSA	1.00	-	-	1.00	-	-	1.00	-	3.00	1.00
Setting 2: Heterogeneous Covariates												
10	5	OS	0.88	0.90	0.93	0.91	0.91	0.94	0.91	0.91		
		CSL	0.14	0.19	0.20	0.20	0.20	0.18	0.19	0.12		
		DLSA	0.95	0.95	0.98	0.98	0.96	0.98	0.97	0.92		
		SDLSA	0.90	-	-	0.98	-	-	0.93	-	3.01	1.00
20	5	OS	0.92	0.93	0.95	0.94	0.93	0.93	0.94	0.93		
		CSL	0.13	0.20	0.19	0.20	0.19	0.19	0.19	0.12		
		DLSA	0.94	0.96	0.98	0.99	0.97	0.99	0.96	0.93		
		SDLSA	0.92	-	-	1.03	-	-	1.00	-	3.01	1.00
100	10	OS	0.89	0.95	0.97	0.96	0.97	0.95	0.96	0.93		
		CSL	0.09	0.13	0.14	0.15	0.13	0.14	0.13	0.09		
		DLSA	0.93	0.97	0.97	0.98	0.98	0.97	0.98	0.93		
		SDLSA	0.88	-	-	0.99	-	-	1.02	-	3.00	1.00

Table 7: Variable description for the U.S. airline data. Numerical variables are standardized to have mean zero and variance one.

Variable	Description	Variable used in the model
Delayed	Whether the flight is delayed, 1 for Yes; 0 for No.	Used as the response variable
Year	Year between 1987 and 2008	Used as numerical variable
Month	Which month of the year	Converted to 11 dummies
DayofMonth	Which day of the month	Used as numerical variable
DayofWeek	Which day of the week	Converted to 6 dummies
DepTime	Actual departure time	Used as numerical variable
CRSDepTime	Scheduled departure time	Used as numerical variable
CRSArrTime	Scheduled arrival time	Used as numerical variable
ElapsedTime	Actual elapsed time	Used as numerical variable
Distance	Distance between the origin and destination in miles	Used as numerical variable
Carrier	Flight carrier code for 29 carriers	Top 7 carries converted to 7 dummies
Destination	Destination of the flight (total 348 categories)	Top 75 destination cities converted to 75 dummies
Origin	Departing origin (total 343 categories)	Top 75 origin cities converted to 75 dummies