Simultaneous Inference for Massive Data: Distributed Bootstrap

Yang Yu [1] Shih-Kang Chao [2] Guang Cheng [1]

Abstract

In this paper, we propose a bootstrap method applied to massive data processed distributedly in a large number of machines. This new method is computationally efficient in that we bootstrap on the master machine without over-resampling, typically required by existing methods (Kleiner et al., 2014; Sengupta et al., 2016), while provably achieving optimal statistical efficiency with minimal communication. Our method does not require repeatedly re-fitting the model but only applies multiplier bootstrap in the master machine on the gradients received from the worker machines. Simulations validate our theory.

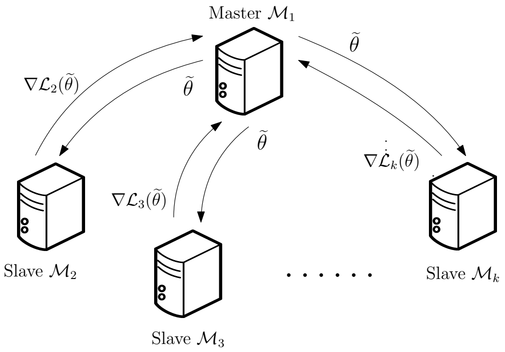

Figure 1.1. Master-worker architecture for storing and processing distributed data.

1. Introduction

1.1. Background

Modern massive data, with enormous sample size, are usually so hard to fit on a single machine. A master-slave architecture is often adopted using a cluster of nodes for data storage and processing; for example, Hadoop, as one of the most popular distributed framework, has facilitates distributed data processing; see Figure 1.1 for a diagram of the master-slave architecture (Singh & Kaur, 2014), where the master node has also a portion of the data. A shortcoming of this architecture is that inter-node communication (between master and worker nodes) is through the TCP/IP protocol, which can be over a thousand times slower than intra-node computation and always comes with significant overhead (Lan et al., 2018; Fan et al., 2019). For these reasons, statistical inference for modern distributed data is very challenging, and communication efficiency is a desirable feature when developing distributed learning algorithms.

However, classical statistical procedures, which typically require hundreds or even thousands passes over the entire data set, are very communication-inefficient or even impossible to perform, including popular methods such as bootstrap, Bayesian inference and many maximum likelihood estimation procedures. Over the last few years, many papers proposed computational procedures for estimation from the maximum likelihood criteria (Zhang et al., 2012; Li et al., 2013; Chen & Xie, 2014; Huang & Huo, 2015; Battey et al., 2015; Zhao et al., 2016; Fan et al., 2017; Lee et al., 2017; Wang & Zhang, 2017; Wang et al., 2017; Shi et al., 2018; Jordan et al., 2019; Volgushev et al., 2019; Banerjee et al., 2019; Fan et al., 2019).

As a popular method for approximating the sample distribution of an estimator, Bootstrap, without modifications, is inapplicable in the environment of distributed processing. It typically requires hundreds or thousands of resamples that is of the same size as the original data, which is impossible for large-scale data stored in different locations.

1.2. Our Contributions

In this paper, we first consider a naïve bootstrap method, named as `k-grad`, that uses local gradients from each machine, where k is the number of machines. To provide a higher accuracy, an improved version, named as `n+k-1-grad` bootstrap, is introduced. Both are (inter-node) communication and (intra-node) computation efficient for (generalized) linear models. Our methods can be easily extended to other statistical models. The statistical accuracy and efficiency are proved theoretically, and validated by

[1] Department of Statistics, Purdue University, USA [2] Department of Statistics, University of Missouri, USA. Correspondence to: Guang Cheng <chengg@purdue.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

simulations.

Our n+k-1-grad method overcomes many constraints faced by the existing methods:

- It preserves bootstrap validity, while relaxing the constraints on the number of machines.
- The computational cost of the bootstrap procedure is as small as it is conducted only on the master node.
- It performs statistical inference on a group of parameters simultaneously, rather than on only individual parameters.

1.3. Related Works

The bag of little bootstraps (BLB) (Kleiner et al., 2014) is one of the earliest methods that can be used in a distributed setting. However, to achieve the bootstrap validity, they require that the number of machines to be smaller than the sample size on local machine, while our methods relax such a requirement. In terms of intra-node computational cost, our methods are more efficient than BLB as expensive model re-fitting on each worker node is not required; see Table 1 for an empirical comparison on computational cost. The subsampled double bootstrap (SDB) approach (Sengupta et al., 2016) was proposed to improve upon BLB in terms of intra-node computational efficiency; however, it fails for both small and large number of machines, as witnessed in our simulation study, while our method can work under these regimes.

1.4. Outline

In Section 2, we formulate the problem of distributed simultaneous inference and present our bootstrap algorithms. We state our theoretical results of bootstrap validity in Section 3. Section 4 presents simulation results that corroborate our theoretical findings. Finally, Section 5 concludes the paper.

1.5. Notations

We denote the ℓ_p-norm ($p > 0$) of any vector $v = (v_1, \dots, v_n)$ by $\|v\|_p = (\sum_{i=1}^n |v_i|^p)^{1/p}$ ($\|v\|_\infty = \max_{1\le i\le n}|v_i|$). We denote the induced p-norm and the max-norm of any matrix $M \in \mathbb{R}^{m\times n}$ (with element M_{ij} at i-th row and j-th column) by $\|M\|_p = \sup_{x\in\mathbb{R}^n;\|x\|_p=1}\|Mx\|_p$ and $\|M\|_{\max} = \max_{1\le i\le m;1\le j\le n}|M_{i,j}|$. We write $a \lesssim b$ if $a = O(b)$, and $a \ll b$ if $a = o(b)$.

2. Methodology

We introduce the distributed framework and the problem setup in Section 2.1, and present the main methodology in Section 2.2. The application to statistical inferences is detailed in Sections 2.3.

We first discuss some preliminaries on the simultaneous inference. Suppose data $\{Z_i\}_{i=1}^N$ are i.i.d., and $\mathcal{L}(\theta; Z)$ is a twice-differentiable convex loss function of $\theta = (\theta_1, \dots, \theta_d) \in \mathbb{R}^d$, which depends on a random variable Z. Suppose that the parameter of interest θ^* is the minimizer of an expected loss:

$$\theta^* = \arg\min_{\theta\in\mathbb{R}^d} \mathcal{L}^*(\theta), \text{ where } \mathcal{L}^*(\theta) := \mathbb{E}_Z[\mathcal{L}(\theta; Z)].$$

2.1. Distributed Data Processing

Assuming N is so large that the data cannot be processed by a single machine, so an estimator of θ^* cannot be straightforwardly obtained by minimizing the empirical loss. Instead, we consider a distributed computation framework. Suppose the data are stored distributedly in k machines, where each machine has n data. Denote by $\{Z_{ij}\}_{i=1,\dots,n;j=1,\dots,k}$ the entire data, where Z_{ij} is i-th datum on the j-th machine $\mathcal{M}_j$, and $N = nk$. Without loss of generality, assume that the first machine $\mathcal{M}_1$ is the master node (see Figure 1.1). Define the local and global loss functions as

$$\text{global loss: } \mathcal{L}_N(\theta) = \frac{1}{k}\sum_{j=1}^k \mathcal{L}_j(\theta), \quad \text{where}$$
$$\text{local loss: } \mathcal{L}_j(\theta) = \frac{1}{n}\sum_{i=1}^n \mathcal{L}(\theta; Z_{ij}), \quad j = 1, \dots, k. \tag{2.1}$$

Recall that the communication between the master and worker nodes is costly in the parallel processing framework, e.g. Hadoop.

The goal in this paper is to obtain *simultaneous* confidence region for θ^*. Simultaneous inference has become a common problem in many areas of application, such as financial economics, signal processing, marketing analytics, biological sciences, and social science (Cai & Sun, 2017; Zhang & Cheng, 2017), where researchers want to investigate a group of variables at the same time, instead of a single variable at a time. Variable selection is usually done by simultaneous inference.

The empirical loss minimizer is defined as:

$$\widehat{\theta} = \arg\min_{\theta\in\mathbb{R}^d} \mathcal{L}_N(\theta). \tag{2.2}$$

Simultaneous confidence intervals can be found with confidence α, for large $\alpha \in (0,1)$, by finding the quantile

$$c(\alpha) := \inf\{t \in \mathbb{R} : P(\widehat{T} \le t) \ge \alpha\} \quad \text{where} \tag{2.3}$$
$$\widehat{T} := \big\|\sqrt{N}\big(\widehat{\theta} - \theta^*\big)\big\|_\infty. \tag{2.4}$$

The asymptotic distribution of $\widehat{\theta}$ has been derived (Eicker et al., 1963; Gourieroux & Monfort, 1981), and confidence intervals can be constructed by finding the quantiles of $\widehat{T}$ in (2.4).

While the procedure above has been well-developed if the data can be processed with a single machine, implementing $\widehat{\theta}$ in a distributed framework faces two challenges:

- $\widehat{\theta}$ usually cannot be easily obtained due to significant communication requirement, so statistical inference for θ^* has to be done via a surrogate estimator $\widetilde{\theta}$, which imitates the distribution of $\widehat{\theta}$ that is called the oracle estimator.
- Estimating $c(\alpha)$ is usually done via bootstrapping the distribution of (2.4) (DasGupta, 2008; Efron & Tibshirani, 1994). Unfortunately, implementing bootstrap is difficult in the distributed computational framework. The existing methods suffer from high computational cost due to resampling/model refitting in each worker nodes (Kleiner et al., 2014; Sengupta et al., 2016) or requiring a large number of machines (Sengupta et al., 2016).

To perform statistical inference in distributed computational framework, a surrogate estimator $\widetilde{\theta}$ satisfying $\|\widetilde{\theta}-\widehat{\theta}\|_\infty = o_p(N^{-1/2})$ (if d is fixed) will be obtained (see Section 2.3), and then we propose new distributed bootstrap algorithms to estimate the quantile $c(\alpha)$ of $\widehat{T}$ in (2.4).

2.2. Distributed Bootstrap Algorithms

We utilize the fact that under weak conditions, the centralized estimator $\widehat{\theta}$ has the following expansion:

$$\sqrt{N}(\widehat{\theta}-\theta^*) = \underbrace{-\nabla^2\mathcal{L}^*(\theta^*)^{-1}\frac{1}{\sqrt{N}}\sum_{i=1}^{n}\sum_{j=1}^{k}\nabla\mathcal{L}(\theta^*;Z_{ij})}_{:=A} + o_P(1). \tag{2.5}$$

It can be seen that the asymptotic distribution of $\sqrt{N}(\widehat{\theta}-\theta^*)$ is determined by that of A.

The multiplier bootstrap (Chernozhukov et al., 2013) can be applied to simulate the distribution of A. In particular, one repeatedly generates N i.i.d. $\mathcal{N}(0,1)$ multipliers $\{\epsilon_{ij}^{(b)}\}_{i=1,\dots,n;j=1,\dots,k}$ for each $b=1,\dots,B$, and then approximate $c(\alpha)$ by the percentile of $\{W^{*(b)}\}_{b=1}^{B}$, where

$$W^{*(b)} = \left\| -\nabla^2\mathcal{L}_N(\widehat{\theta})^{-1}\frac{1}{\sqrt{N}}\sum_{j=1}^{k}\sum_{i=1}^{n}\epsilon_{ij}^{(b)}(\widehat{\mathbf{g}}_{ij}-\widehat{\mathbf{g}})\right\|_\infty, \tag{2.6}$$

Algorithm 1 DistBoots(method, $\widetilde{\theta}, \{\mathbf{g}_j\}_{j=1}^k, \widetilde{\Theta}$): only need the master node $\mathcal{M}_1$

```
Input: master node $\mathcal{M}_1$ obtains local gradient $\mathbf{g}_j$, estimate $\widetilde{\Theta}$ of inverse population Hessian
$\bar{\mathbf{g}} \leftarrow k^{-1}\sum_{j=1}^k \mathbf{g}_j$
for $b = 1,\dots,B$ do
  if method='k-grad' then
    Draw $\epsilon_1^{(b)},\dots,\epsilon_k^{(b)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$
    Compute $W^{(b)}$ by (2.7)
  else if method='n+k-1-grad' then
    Draw $\epsilon_{11}^{(b)},\dots,\epsilon_{n1}^{(b)},\epsilon_2^{(b)},\dots,\epsilon_k^{(b)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$
    Compute $W^{(b)}$ by (2.8)
  end if
end for
Compute the percentile $c_W(\alpha)$ of $\{W^{(1)},\dots,W^{(B)}\}$ for $\alpha \in (0,1)$
Return $\widetilde{\theta}_l \pm N^{-1/2}c_W(\alpha)$, $l = 1,\dots,d$
```

with $\widehat{\mathbf{g}}_{ij} = \nabla\mathcal{L}(\widehat{\theta};Z_{ij})$, $\widehat{\mathbf{g}} = N^{-1}\sum_{j=1}^k\sum_{i=1}^n\widehat{\mathbf{g}}_{ij}$. However, computing $W^{*(b)}$ for each b requires one round of communication in the distributed computational framework, so the computational cost is formidable when, e.g. $B = 500$.

To adapt the above multiplier bootstrap for the distributed computational framework, we propose the k-grad bootstrap, which replaces (2.6) by

$$\overline{W}^{(b)} := \Bigg\| \underbrace{-\widetilde{\Theta}\frac{1}{\sqrt{k}}\sum_{j=1}^{k}\epsilon_j^{(b)}\sqrt{n}(\mathbf{g}_j-\bar{\mathbf{g}})}_{:=\overline{A}} \Bigg\|_\infty, \tag{2.7}$$

with $\epsilon_j^{(b)} \overset{iid}{\sim} \mathcal{N}(0,1)$, $\mathbf{g}_j = \nabla\mathcal{L}_j(\widetilde{\theta})$, $\bar{\mathbf{g}} = k^{-1}\sum_{j=1}^k\mathbf{g}_j$, and a communication-efficient surrogate estimator $\widetilde{\theta}$ to replace $\widehat{\theta}$ for communication efficiency, and an estimator $\widetilde{\Theta}$ of the inverse Hessian $\nabla^2\mathcal{L}^*(\widehat{\theta})^{-1}$. The key advantage of bootstrapping (2.7) over (2.6) is that, once the master has the gradients $\mathbf{g}_j$ from the worker nodes, the percentile of $\{\overline{W}^{(b)}\}_{b=1}^{B}$ can be computed in the master node only, without the need to communicate with worker nodes. See Algorithm 1 (method='k-grad') for a detailed description of the k-grad bootstrap.

A problem with the k-grad procedure is that it may perform rather poorly when k is small, e.g. $k = 2$ or 3, as can be seen from the simulation studies (Section 4). This is due to the failure of bootstrapping the variance with only 2 or 3 multipliers. This problem can be alleviated by using a unique multiplier to each datum in the master node $\mathcal{M}_1$;

that is,

$$\widetilde{W}^{(b)} := \Big\| \underbrace{-\widetilde{\Theta}\frac{1}{\sqrt{n+k-1}}\Big(\sum_{i=1}^{n} \epsilon_{i1}^{(b)}(\mathbf{g}_{i1}-\bar{\mathbf{g}}) + \sum_{j=2}^{k} \epsilon_{j}^{(b)}\sqrt{n}(\mathbf{g}_{j}-\bar{\mathbf{g}})\Big)}_{:=\widetilde{A}} \Big\|_{\infty}. \tag{2.8}$$

where $\epsilon_{i1}^{(b)}$ and $\epsilon_{j}^{(b)}$ are i.i.d. $\mathcal{N}(0,1)$ multipliers in i, j and b, and $\mathbf{g}_{i1} = \nabla\mathcal{L}(\widetilde{\theta}; Z_{i1})$ is based on a single datum Z_{i1} in the master. We call this method the `n+k-1-grad` bootstrap. Note that the percentile of $\{\widetilde{W}^{(b)}\}_{b=1}^{B}$ can still be computed using only $\mathcal{M}_1$, without needing to communicate with other machines. See Algorithm 1 (`method`=‘`n+k-1-grad`’) for details.

We remark that besides simultaneous inference, our methods can also be applied apply to pointwise confidence intervals and circular confidence region, by replacing $\|\cdot\|_\infty$ with $|(\cdot)_l|$ and $\|\cdot\|_2$, where we denote by $(\cdot)_l$ the l-th element of a vector.

2.3. CSL Estimator

To apply `k-grad` or `n+k-1-grad`, we use the communication-efficient surrogate likelihood [CSL, (Jordan et al., 2019)] algorithm with a quadratic approximation to compute the surrogate estimator $\widetilde{\theta}$ of $\widehat{\theta}$[1]. The CSL estimator $\widetilde{\theta}$ converges to $\widehat{\theta}$ even if $n \le k$ with sufficient rounds of communication, and when $n > k$, only one round of communication is required to achieve $\|\widetilde{\theta} - \widehat{\theta}\|_\infty = o_p(N^{-1/2})$ if d is fixed. We compute $\widetilde{\Theta}$ in (2.7) and (2.8) by inverting $\nabla^2\mathcal{L}_1(\widetilde{\theta})$ at $\mathcal{M}_1$. See Algorithm 2 for a detailed description of using `k-grad` or `n+k-1-grad` for constructing simultaneous confidence intervals with τ rounds of communication.

The number of iterations τ in Algorithm 2 steers the trade-off between accuracy and communication efficiency. A larger τ generally leads to a more accurate coverage of the simultaneous confidence interval; meanwhile, it induces a higher communication cost. We theoretically study the minimal τ that warrants the bootstrap accuracy in Section 3.

Remark 2.1. *Although in Algorithm 1 the same $\widetilde{\theta}$ is used for the center of the confidence interval and for evaluating the gradients $\mathbf{g}_{ij}$, allowing them to be different (such as in Algorithm 2) can save one round of communication. For example, we can use $\widetilde{\theta}^{(\tau)}$ for the center of the confidence interval, while the gradients are evaluated with $\widetilde{\theta}^{(\tau-1)}$ from the last iteration.*

Algorithm 2 `k-grad`/`n+k-1-grad` with CSL: τ rounds of communication, $\tau \ge 1$

```
Compute $\widetilde{\theta}^{(0)} = \arg\min_\theta \mathcal{L}_1(\theta)$ at $\mathcal{M}_1$
for $t = 1, \ldots, \tau$ do
  Transmit $\widetilde{\theta}^{(t-1)}$ to $\{\mathcal{M}_j\}_{j=2,\ldots,k}$
  Compute $\nabla\mathcal{L}_1(\widetilde{\theta}^{(t-1)})$ and $\nabla^2\mathcal{L}_1(\widetilde{\theta}^{(t-1)})^{-1}$ at $\mathcal{M}_1$
  for $j = 2, \ldots, k$ do
    Compute $\nabla\mathcal{L}_j(\widetilde{\theta}^{(t-1)})$ at $\mathcal{M}_j$
    Transmit $\nabla\mathcal{L}_j(\widetilde{\theta}^{(t-1)})$ to $\mathcal{M}_1$
  end for
  $\nabla\mathcal{L}_N(\widetilde{\theta}^{(t-1)}) \leftarrow k^{-1}\sum_{j=1}^{k}\nabla\mathcal{L}_j(\widetilde{\theta}^{(t-1)})$ at $\mathcal{M}_1$
  $\widetilde{\theta}^{(t)} \leftarrow \widetilde{\theta}^{(t-1)} - \nabla^2\mathcal{L}_1(\widetilde{\theta}^{(t-1)})^{-1}\nabla\mathcal{L}_N(\widetilde{\theta}^{(t-1)})$ at $\mathcal{M}_1$
end for
Run DistBoots('k-grad' or 'n+k-1-grad',
        $\widetilde{\theta} = \widetilde{\theta}^{(\tau)}, \{\mathbf{g}_j = \nabla\mathcal{L}_j(\widetilde{\theta}^{(\tau-1)})\}_{j=1}^{k}$,
        $\widetilde{\Theta} = \nabla^2\mathcal{L}_1(\widetilde{\theta}^{(\tau-1)})^{-1}$) at $\mathcal{M}_1$
```

Remark 2.2. *There exist other options than CSL for $\widetilde{\theta}$ such as the one-shot averaging estimator (Zhang et al., 2012), but an additional round of communication may be needed to compute the local gradients. More importantly, they may be inaccurate when $n < k$.*

3. Theoretical Results

Section 3.1 provides an overview of the theoretical results. Section 3.2 presents the theory in a linear model framework for `k-grad` and `n+k-1-grad`. Section 3.3 shows the results for the generalized linear models (GLMs).

3.1. An Overview

As discussed in Section 2.3, we would like τ to be large enough to ensure the accuracy of the simultaneous confidence interval, yet it may induce unmerited communication cost. Hence, we theoretically study the minimal number of iterations that is sufficient for Algorithm 2 to achieve the bootstrap validity; we denote it by $\tau_{\min}$. The results are illustrated in Figure 3.1. Panels in the top row illustrate the lower bound of τ for linear models given in Theorems 3.1 and 3.2 of Section 3.2, and those in the bottom row illustrating the results for the GLMs given in Theorems 3.6 and 3.7 of Section 3.3

As a general pattern of Figure 3.1, $\tau_{\min}$ is (logarithmically) increasing in k and decreasing in n for both `k-grad` and `n+k-1-grad` in (generalized) linear models; in addition, $\tau_{\min}$ is (logarithmically) increasing in d.

For the difference between `k-grad` and `n+k-1-grad`, we compare the left and right panel of Figure 3.1. With fixed (n, k, d), the $\tau_{\min}$ for `n+k-1-grad` is always no larger than that for `k-grad`, which indicates a greater effi-

[1] Particular, we adopt the $\widetilde{\theta}^{H}$ described in Section 3.1 of Jordan et al. (2019).

ciency of `n+k-1-grad`. As k is small, `k-grad` would not work, while `n+k-1-grad` can provably work. In addition, $\tau_{\min} = 1$ can work for certain instances of `n+k-1-grad` but never for `k-grad`.

For the comparison between the linear models (top panels) and GLMs (bottom panels), GLMs require larger n than linear models in order to ensure that our bootstrap procedures work.

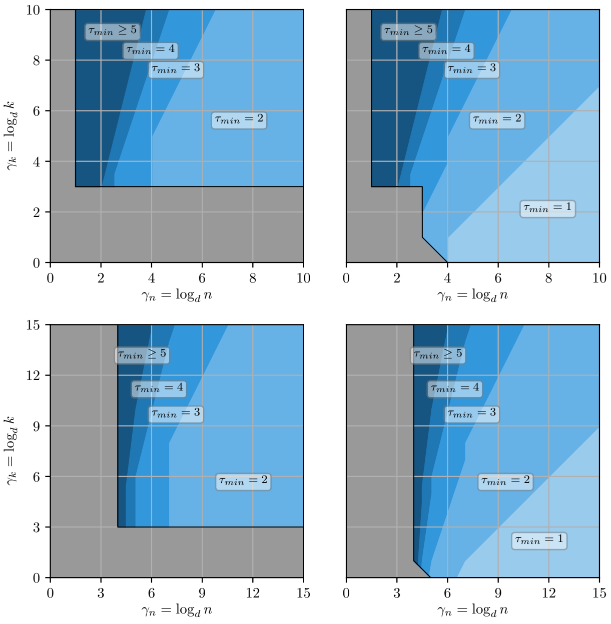

Figure 3.1. Illustration of Theorems 3.1 (**top left**: linear model, `k-grad`), 3.2 (**top right**: linear model, `n+k-1-grad`), 3.6 (**bottom left**: GLM, `k-grad`), and 3.7 (**bottom right**: GLM, `n+k-1-grad`). Gray area represents the region where the theorems do not validate the bootstrap procedures, and the other area is colored blue of varying lightness according to the lower bound of iteration τ.

3.2. Linear Model

For simplicity, we start with the linear model. Suppose that N i.i.d. observations come from a linear model, $y = x^\top\theta^* + e$, with unknown coefficient vector $\theta^* \in \mathbb{R}^d$, covariate random vector $x \in \mathbb{R}^d$, and noise $e \in \mathbb{R}$ independent of x with zero mean and variance of σ^2. We define $\Sigma = \mathbb{E}[xx^\top]$. We consider the least-squares loss $\mathcal{L}(\theta; z) = \mathcal{L}(\theta; x, y) = (y - x^\top\theta)^2/2$.

We impose the following assumptions on the linear model.

(A1) x is sub-Gaussian, that is,

$$\sup_{\|w\|_2 \leq 1} \mathbb{E}\big[\exp((w^\top x)^2/L^2)\big] = O(1),$$

for some absolute constant $L > 0$. Moreover, $1/\lambda_{\min}(\Sigma) \leq \mu$ for some absolute constant $\mu > 0$.

(A2) e is sub-Gaussian, that is,

$$\mathbb{E}\big[\exp(e^2/L'^2)\big] = O(1),$$

for some absolute constant $L' > 0$. Moreover, $\sigma > 0$ is an absolute constant.

Under the assumptions, we first investigate the theoretical property of Algorithm 2, where we apply `k-grad` along with the CSL estimator that takes advantage of multiple rounds of communication. We define

$$T := \big\|\sqrt{N}\big(\widetilde{\theta} - \theta^*\big)\big\|_\infty, \quad \text{and} \tag{3.1}$$
$$c_{\overline{W}}(\alpha) := \inf\{t \in \mathbb{R} : P_\epsilon(\overline{W} \leq t) \geq \alpha\},$$

where P_ϵ denotes the probability with respect to the randomness from all the multipliers, $\overline{W}$ has the same distribution as $\overline{W}^{(b)}$ in (2.7), and $\widetilde{\theta}$ and $\bar{\theta}$ are the τ-step and $\tau - 1$-step CSL estimators as specified in Algorithm 2. Recall the definition of $\widehat{T}$ in (2.4) with $\widehat{\theta}$ defined in (2.2). Now, we state a result for `k-grad` bootstrap procedure with the CSL estimator.

Theorem 3.1 (`k-grad`, linear model)**.** *Suppose (A1)-(A2) hold, and that we run Algorithm 2 with `k-grad` method in linear models. Assume $n = d^{\gamma_n}$ and $k = d^{\gamma_k}$ for some constants $\gamma_n, \gamma_k > 0$. If $\gamma_n > 1$, $\gamma_k > 3$, $\tau \geq \tau_{\min}$, where*

$$\tau_{\min} = 1 + \left\lfloor \max\left\{\frac{\gamma_k + 1}{\gamma_n - 1}, 1 + \frac{3}{\gamma_n - 1}\right\}\right\rfloor,$$

then we have

$$\sup_{\alpha \in (0,1)} |P(T \leq c_{\overline{W}}(\alpha)) - \alpha| = o(1). \tag{3.2}$$

In addition, (3.2) also holds if T is replaced by $\widehat{T}$.

Theorem 3.1 states that under certain conditions, simultaneous confidence intervals given by Algorithm 2 with `k-grad` method provide sufficient coverage. It also suggests that the bootstrap quantile approximates the quantile of the centralized estimator $\widehat{\theta}$, and therefore, the bootstrap procedure is also statistically efficient.

Next, we present a theorem that establishes the validity and the efficiency of `n+k-1-grad` bootstrap procedure in Algorithm 2. We define

$$c_{\widetilde{W}}(\alpha) := \inf\{t \in \mathbb{R} : P_\epsilon(\widetilde{W} \leq t) \geq \alpha\},$$

where $\widetilde{W}$ has the same distribution as $\widetilde{W}^{(b)}$ in (2.8).

Theorem 3.2 (`n+k-1-grad`, linear model)**.** *Suppose (A1)-(A2) hold, and that we run Algorithm 2 with `n+k-1-grad` method in linear models. Assume $n = d^{\gamma_n}$ and $k = d^{\gamma_k}$*

for some constants $\gamma_n, \gamma_k > 0$. If $\gamma_n > 1$, $\gamma_n \vee \gamma_k > 3$, $\gamma_n + \gamma_k > 4$, $\tau \geq \tau_{\min}$, where

$$\tau_{\min} = 1 + \left\lfloor \frac{(\gamma_k - 1) \vee (\gamma_n \wedge \gamma_k) \vee 1 + 2}{\gamma_n - 1} \right\rfloor,$$

then we have

$$\sup_{\alpha \in (0,1)} |P(T \leq c_{\widetilde{W}}(\alpha)) - \alpha| = o(1). \tag{3.3}$$

In addition, (3.3) also holds if T is replaced by $\widehat{T}$.

For a deeper look into the difference between `k-grad` and `n+k-1-grad`, we compare the difference between the covariance of the oracle score A (defined in (2.5)) and the conditional covariance of $\overline{A}$ (for `k-grad`, defined in (2.7)), and $\widetilde{A}$ (for `n+k-1-grad`, defined in (2.8)) conditioning on the data. These key quantities which determine how well the bootstrap procedure approximates the distribution of $\widehat{T}$. Conditioning on the data, we have the bounds

$$\begin{aligned} &\left|\!\left|\!\left|\mathrm{cov}_\epsilon(\overline{A}) - \mathrm{cov}(A)\right|\!\right|\!\right|_{\max} \leq d\|\widetilde{\theta}^{(\tau-1)} - \theta^*\|_1 \\ &\quad + nd\|\widetilde{\theta}^{(\tau-1)} - \theta^*\|_1^2 + O_P\left(\sqrt{\frac{d^2}{k}} + \sqrt{\frac{d}{n}}\right), \end{aligned} \tag{3.4}$$

$$\begin{aligned} &\left|\!\left|\!\left|\mathrm{cov}_\epsilon(\widetilde{A}) - \mathrm{cov}(A)\right|\!\right|\!\right|_{\max} \leq d\|\widetilde{\theta}^{(\tau-1)} - \theta^*\|_1 \\ &\quad + (n \wedge k)d\|\widetilde{\theta}^{(\tau-1)} - \theta^*\|_1^2 + O_P\left(\sqrt{\frac{d^2}{n+k}} + \sqrt{\frac{d}{n}}\right), \end{aligned} \tag{3.5}$$

up to factors that are logarithmic in d, n or k. Comparing the two preceding equations, we first see that overall, `n+k-1-grad` (3.5) has a smaller error than `k-grad` (3.4). In particular, `k-grad` requires both n and k to be large, while `n+k-1-grad` requires a large n but not a large k. In addition, a single round of communication could be enough for `n+k-1-grad`, but not for `k-grad`. To see it, if $\tau = 1$, $\|\widetilde{\theta}^{(0)} - \theta^*\|_1$ is of order $O_P(d/\sqrt{n})$, and the right-hand side of (3.4) will grow with d; by contrast, the error in (3.5) still shrinks to zero as long as $k \ll n$.

Remark 3.3. *Given fixed d, $\tau = \lceil \log k / \log n \rceil$ is enough for CSL to achieve the optimal estimation error rate (Jordan et al., 2019). Under same circumstance, bootstrap consistency is warranted at the expense of at most one additional communication round $\tau_{\min} = 1 + \lfloor \log k / \log n \rfloor$ (Theorem 3.2).*

Remark 3.4. *To apply BLB in the distributed setting, $k \lesssim n$ is required to achieve the higher order correctness of the bootstrap procedure (Kleiner et al., 2014). We conjecture that SDB requires $k \lesssim n$ as well, based on the observations from simulation study in Section 4.2. In contrast to BLB and SDB, `k-grad` (if $k \gg d^3$) and `n+k-1-grad` are both scalable to $k \gg n$, at the cost of a larger τ.*

Remark 3.5. *The non-asymptotic rate of $\sup_{\alpha \in (0,1)} |P(T \leq c_{\overline{W}}(\alpha)) - \alpha|$ may be proven to be polynomial in n and k, with a more delicate analysis. As an alternative, simultaneous inference can also be done with the the alternative extreme value distribution approach, but the convergence rate is at best logarithmic (Chernozhukov et al., 2013; Zhang & Cheng, 2017).*

3.3. Generalized Linear Model

In this section, we consider GLMs, which generate i.i.d. observations $(x, y) \in \mathbb{R}^d \times \mathbb{R}$. We assume that the loss function $\mathcal{L}$ is of the form $\mathcal{L}(\theta; z) = g(y, x^\top \theta)$ for $\theta, x \in \mathbb{R}^d$ and $y \in \mathbb{R}$ with $g : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, and $g(a, b)$ is three times differentiable with respect to b, and denote $\frac{\partial}{\partial b} g(a, b)$, $\left(\frac{\partial}{\partial b}\right)^2 g(a, b)$, $\left(\frac{\partial}{\partial b}\right)^3 g(a, b)$ by $g'(a, b)$, $g''(a, b)$, $g'''(a, b)$ respectively. We let θ^* be the unique minimizer of the expected loss $\mathcal{L}^*(\theta)$.

We impose the following assumptions on the GLM.

(B1) For some $\Delta > 0$, and $\Delta' > 0$ such that $|x^\top \theta^*| \leq \Delta'$,

$$\sup_{|b| \vee |b'| \leq \Delta + \Delta'} \sup_a \frac{|g''(a, b) - g''(a, b')|}{|b - b'|} \leq 1,$$

$$\max_{|b_0| \leq \Delta} \sup_a |g'(a, b_0)| = O(1), \quad \text{and}$$

$$\max_{|b| \leq \Delta + \Delta'} \sup_a |g''(a, b)| = O(1).$$

(B2) $\|x\|_\infty = O(1)$.

(B3) The smallest and largest eigenvalues of $\nabla^2 \mathcal{L}^*(\theta^*)$ and $\mathbb{E}\left[\nabla \mathcal{L}(\theta^*; Z) \nabla \mathcal{L}(\theta^*; Z)^\top\right]$ are bounded away from zero and infinity respectively.

(B4) For some constant $L > 0$,

$$\max_l \max_{q=1,2} \mathbb{E}[|\mathbf{h}_l^{2+q}|/L^q] + \mathbb{E}[\exp(|\mathbf{h}_l|/L)] = O(1), \quad \text{or}$$

$$\max_l \max_{q=1,2} \mathbb{E}[|\mathbf{h}_l^{2+q}|/L^q] + \mathbb{E}[(\max_l |\mathbf{h}_l|/L)^4] = O(1),$$

where $\mathbf{h} = \nabla^2 \mathcal{L}^*(\theta^*)^{-1} \nabla \mathcal{L}(\theta^*; Z)$ and $\mathbf{h}_l$ is the l-th coordinate.

Assumption (B1) imposes smoothness conditions on the loss function. For example, the logistic regression model has $g(a, b) = -ab + \log(1 + \exp(b))$. It is easy to see that $|g'(a, b)| \leq 2$, $|g''(a, b)| \leq 1$, $|g'''(a, b)| \leq 1$. Therefore, Assumption (B1) is met for the loss function of the logistic regression model. Assumption (B2) imposes boundedness condition on the input variables. Assumption (B3) is a standard assumption in the GLM literature. Assumption (B4) is required for proving the validity of multiplier bootstrap (Chernozhukov et al., 2013).

The following two theorems states the validity and the efficiency of `k-grad` and `n+k-1-grad` in the GLM. Recall the definitions of T, $\widehat{T}$, $\overline{W}$, and $\widetilde{W}$ in (3.1), (2.4), (2.7), and (2.8), respectively.

Theorem 3.6 (`k-grad`, GLM)**.** *Suppose (B1)-(B4) hold, and that we run Algorithm 2 with `k-grad` method in GLMs. Assume $n = d^{\gamma_n}$ and $k = d^{\gamma_k}$ for some constants $\gamma_n, \gamma_k > 0$. If $\gamma_n > 4$, $\gamma_k > 3$, $\tau \geq \tau_{\min}$, where*

$$\tau_{\min} = \tau_0 + \max\left\{ \left\lfloor \frac{\gamma_k - 2}{\gamma_n - 1} + \nu_0 \right\rfloor, 1 \right\},$$

$$\tau_0 = 1 + \left\lfloor \log_2 \frac{\gamma_n - 1}{\gamma_n - 4} \right\rfloor, \quad \nu_0 = 2 - \frac{2^{\tau_0}(\gamma_n - 4)}{\gamma_n - 1} \in (0, 1], \tag{3.6}$$

then we have (3.2). In addition, (3.2) also holds if T is replaced by $\widehat{T}$.

Theorem 3.7 (`n+k-1-grad`, GLM)**.** *Suppose (B1)-(B4) hold, and that we run Algorithm 2 with `n+k-1-grad` method in GLMs. Assume $n = d^{\gamma_n}$ and $k = d^{\gamma_k}$ for some constants $\gamma_n, \gamma_k > 0$. If $\gamma_n > 4$, $\gamma_n + \gamma_k > 5$, $\tau \geq \tau_{\min}$, where*

$$\tau_{\min} = \tau_0 + \left\lfloor \frac{(\gamma_k - 1) \vee (\gamma_n \wedge \gamma_k) - 1}{\gamma_n - 1} + \nu_0 \right\rfloor,$$

τ_0 and ν_0 defined as in (3.6), then we have (3.3). In addition, (3.3) also holds if T is replaced by $\widehat{T}$.

See Figure 3.1 for a comparison between the results of linear models and GLMs.

Remark 3.8. *In both Theorems 3.6 and 3.7, τ_0 is the communication rounds needed for the CSL estimator to go through the regions which are far from θ^*. As d grows, the time spent in these regions can increase. However, when n is large, e.g., $n \gg d^7$, the loss function is more well-behaved, and the time required reduces to $\tau_0 = 1$.*

4. Experiments

4.1. Accuracy and Efficiency

Fix the total sample size $N = 2^{16}$. Choose d from $\{2^1, 2^3, 2^5, 2^7\}$ and k from $\{2^0, 2^1, \ldots, 2^{11}\}$. θ^* is determined by drawing uniformly from $[-0.5, 0.5]^d$ and keep it fixed for all replications. We generate each covariate vector x independently from $\mathcal{N}(0, \Sigma)$ and specify two different covariance matrices: Toeplitz ($\Sigma_{l,l'} = 0.9^{|l-l'|}$) and equi-correlation ($\Sigma_{l,l'} = 0.8$ for all $l \neq l'$, $\Sigma_{l,l} = 1$ for all l), and the results for the latter are deferred to the appendix as they are similar to that under the Toeplitz design. For linear model, we generate e independently from $\mathcal{N}(0, 1)$, simulate the response from $y = x^\top \theta^* + e$; for GLM, we consider logistic regression and obtain each response from $y \sim \mathrm{Ber}(1/(1 + \exp[-x^\top \theta^*]))$. Under each choice of d and k, we run `k-grad` and `n+k-1-grad` with CSL on 1000 independent data sets, and compute the empirical coverage probability and the average width based on the results from these 1000 replications. At each replication, we draw $B = 500$ bootstrap samples, from which we calculate the 95% empirical quantile to further obtain the 95% simultaneous confidence interval (the level 95% is represented by a black solid line in all figures).

The average widths are compared against the oracle width. We compute the oracle width (represented by a black dashed line in all figures) for each model as follows. For a fixed N and d, we generate 500 independent data sets, and for each data set, we compute the centralized $\widehat{\theta}$. The oracle width is defined as two times the 95% empirical quantile of $\|\widehat{\theta} - \theta^*\|_\infty$.

The empirical coverage probabilities and the average widths of `k-grad` and `n+k-1-grad` are displayed in Figures 4.1 (linear regression with Toeplitz design) and 4.2 (logistic regression with Toeplitz design). Note that the sub-sample size n is determined by k as N is fixed, and therefore, a larger k indicates a smaller n.

When k is small, `k-grad` fails because k multipliers cannot provide enough perturbation to approximate the sampling distribution whereas `n+k-1-grad` has a good coverage (Theorems 3.2 and 3.7). When k gets too large (or n gets too small), the coverage of both algorithms starts to fall, due to both the deviation of the center (the estimator $\widetilde{\theta}^{(\tau)}$) from the centralized estimator $\widehat{\theta}$ and the deviation of the width from the oracle width [(3.4) and (3.5)]. We also see that the larger the dimension is, the harder it is for both algorithms to achieve 95% coverage, and the earlier both algorithm fail as k grows (or n decreases) [(3.4) and (3.5)]. However, increasing the number of communication rounds improves the coverage, and thus, the coverage of both algorithms, even when $k \geq n$. When k is too large (or n is too small; see, for example, Figure 4.1, `n+k-1-grad`, $d = 2^7$), the width could go further away from the oracle width as the number of communication rounds increases, as predicted by the increase of the right-hand sides of both (3.4) and (3.5) as n decreases.

The cases of $d = 2^3$ and 2^5 and the equi-correlation case are deferred to the appendix, as the patterns are similar to Figure 4.1 and Figure 4.2. Results on pointwise confidence intervals are also included in the appendix.

4.2. Comparisons to existing methods: BLB and SDB

Note that BLB (Kleiner et al., 2014) and SDB (Sengupta et al., 2016) do not give a confidence interval but just a bootstrap estimate of the percentile $c(\alpha)$ in (2.3). Here, we

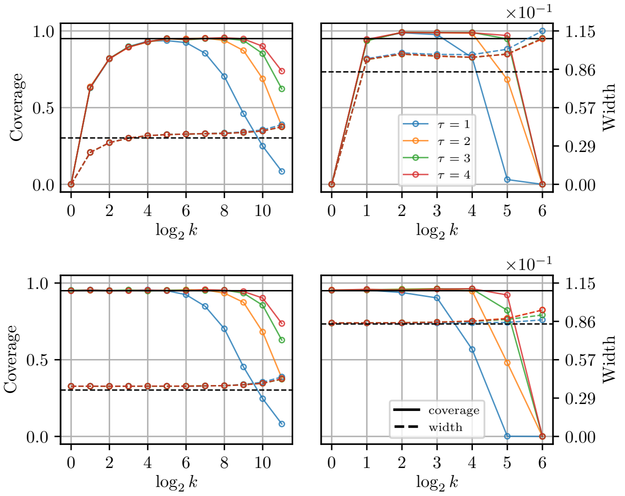

Figure 4.1. Empirical coverage probability (**left axis**) and average width (**right axis**) of simultaneous confidence intervals by `k-grad` (**top**) and `n+k-1-grad` (**bottom**) in a linear regression model with varying dimension (**left**: $d = 2^1$, **right**: $d = 2^7$). Black solid line represents nominal confidence level (95%) and black dashed line represents oracle width.

compare the width of `k-grad` and `n+k-1-grad` against BLB and SDB, using Toeplitz design and similar experimentals setting in Section 4.1. We use BLB and SDB to compute the width of a confidence interval and compare it against the oracle width, instead of constructing the entire confidence interval. The results are displayed in Figures 4.3.

SDB always has a significant deviation from the oracle width for small k and has the same behavior as BLB when k is large. The width of `n+k-1-grad` is closer to the oracle width than `k-grad`, as discussed in Section 4.1.

As `n+k-1-grad` and BLB appear to be the two best-performing methods, we compare the two into more details. For linear regression, `n+k-1-grad` performs as well as BLB, except in a few cases of large k. For logistic regression, the width of both `n+k-1-grad` and BLB deviate from the oracle width for large k, but `n+k-1-grad` mostly outperforms BLB, because n/k is too small for BLB, while `n+k-1-grad` improves as the number of communications τ increases.

4.3. Computational cost

Table 1 shows the computational cost of different bootstrap methods. The average run time (in seconds) is computed with 50 independent runs, and in each run a bootstrap method is carried out for linear regression model with Toeplitz design. We set $\tau = 1$ for `k-grad` and `n+k-1-grad`.

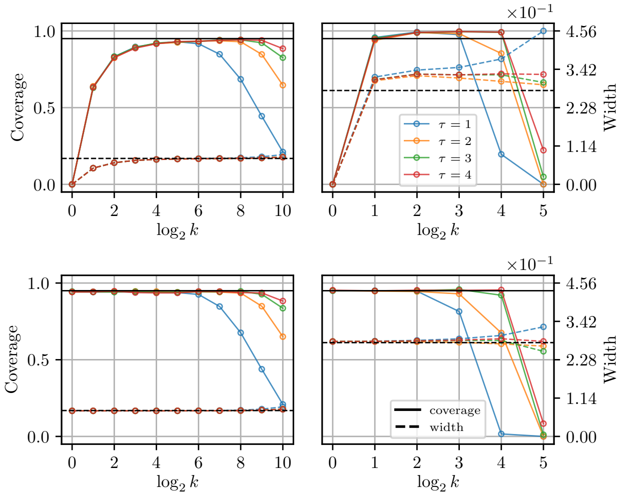

Figure 4.2. Empirical coverage probability (**left axis**) and average width (**right axis**) of simultaneous confidence intervals by `k-grad` (**top**) and `n+k-1-grad` (**bottom**) in a logistic regression model with varying dimension (**left**: $d = 2^1$, **right**: $d = 2^7$). Black solid line represents nominal confidence level (95%) and black dashed line represents oracle width.

Both BLB and SDB require each worker node to repeatedly resample and re-fit the model, so we expect they require more time. Particularly, Table 1 shows that BLB is much more computationally expensive than the others, and its computational time greatly increases as k and d grow. SDB has much lower computational time than BLB, but the computational time grows rapidly with the number of machines. On the other hand, computational time of `k-grad` and `n+k-1-grad` remains low as k grows, since the bootstrap is done only on the master node. We have even observed a decrease in the run time as k increases for `k-grad` and `n+k-1-grad`, which show that our methods can better take advantage of parallelism.

Table 1. Average run times (in seconds) of `k-grad`, `n+k-1-grad`, SDB, and BLB with different k and d (**top**: $d = 2^3$, **bottom**: $d = 2^7$).

Method	$k = 2^2$	$k = 2^6$	$k = 2^9$
`k-grad`	0.29	0.29	0.30
`n+k-1-grad`	0.85	0.45	0.45
SDB	0.08	0.30	5.39
BLB	22.66	35.12	159.88
Methods	$k = 2^2$	$k = 2^6$	$k = 2^9$
`k-grad`	0.82	0.51	0.50
`n+k-1-grad`	1.49	0.67	0.64
SDB	3.44	3.83	12.66
BLB	981.17	842.50	1950.91

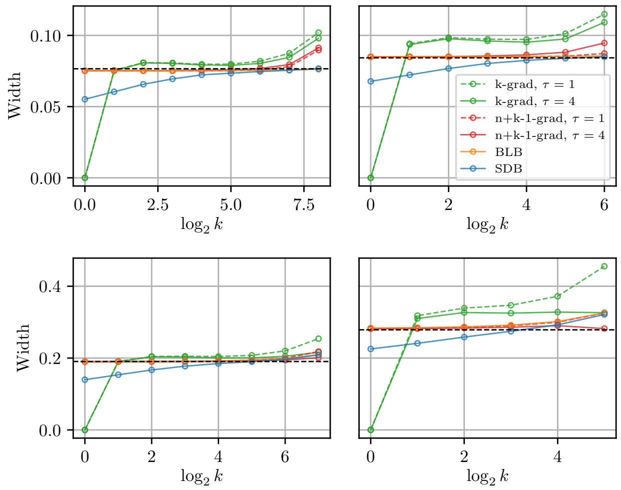

Figure 4.3. Comparison of `k-grad`, `n+k-1-grad`, BLB, and SDB in average width of simultaneous confidence intervals in linear regression (**top**) and logistic regression (**bottom**) with varying dimension (**left**: $d = 2^5$, **right**: $d = 2^7$). Black dashed line represents oracle width.

5. Discussions

We propose two communication-efficient and computation-efficient bootstrap methods, `k-grad` and `n+k-1-grad`, for simultaneous inference on distributed massive data. Our methods are robust to the number of machines. The accuracy and efficiency of the algorithms are theoretically proven and validated through simulations. Furthermore, our methods can potentially be extended for applications to high-dimensional (generalized) linear models and graphical models, which we discuss below.

5.1. Extension to High-Dimensional Models

We need to overcome the following two challenges for the high dimension extension ($d > n$). First, none of the existing high-dimensional distributed estimators enjoys a sample-average-like expression as in (2.5), when ℓ_1 regularization is used to induce sparsity. To meet this challenge, we can debias some distributed estimator (e.g., Wang et al. (2017)) by adapting approaches such as Van de Geer et al. (2014) to the distributed framework while maintaining communication efficiency. Second, the sample-average-like expression cannot be trivially approximated as done in the low-dimensional regime, as the local sample Hessian matrix is not invertible. We can apply approaches such as nodewise lasso (Van de Geer et al., 2014) on local sample to acquire approximate inverse Hessian matrices.

5.2. Applications in Graphical Models

Chang et al. (2018) and Yu et al. (2019) adopted Gaussian approximation and multiplier bootstrap to graphical models, based on sample-average-like approximations (see Equation (12) in Chang et al. (2018) and Equation (15) in Yu et al. (2019)). Therefore, we conjecture that our methods can be applied to the graphical models in a communication-efficient way under the distributed framework, by communicating vectors from the sample-average-like approximations, analogously to communicating gradients in our paper.

Acknowledgements

We thank the anonymous reviewers for the helpful comments. Chao would like to acknowledge the financial support from the Research Council of the University of Missouri. This work was completed while Cheng was a member of Institute for Advanced Study, Princeton in the fall of 2019. Cheng would like to acknowledge hospitality of IAS, and also financial support from NSF DMS-1712907, DMS-1811812, DMS-1821183, Office of Naval Research, (ONR N00014-18-2759) and Adobe Data Science Fund.

References

Banerjee, M., Durot, C., Sen, B., et al. Divide and conquer in nonstandard problems and the super-efficiency phenomenon. *The Annals of Statistics*, 47(2):720–757, 2019.

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

Cai, T. T. and Sun, W. Large-scale global and simultaneous inference: Estimation and testing in very high dimensions. *Annual Review of Economics*, 9:411–439, 2017.

Chang, J., Qiu, Y., Yao, Q., and Zou, T. Confidence regions for entries of a large precision matrix. *Journal of Econometrics*, 206(1):57–82, 2018.

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

Chernozhukov, V., Chetverikov, D., Kato, K., et al. Gaussian approximations and multiplier bootstrap for maxima of sums of high-dimensional random vectors. *The Annals of Statistics*, 41(6):2786–2819, 2013.

DasGupta, A. *Asymptotic theory of statistics and probability*. Springer Science & Business Media, 2008.

Efron, B. and Tibshirani, R. J. *An introduction to the bootstrap*. CRC press, 1994.

Eicker, F. et al. Asymptotic normality and consistency of the least squares estimators for families of linear regressions. *The Annals of Mathematical Statistics*, 34(2):447–456, 1963.

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

Fan, J., Guo, Y., and Wang, K. Communication-efficient accurate statistical estimation. *arXiv preprint arXiv:1906.04870*, 2019.

Gourieroux, C. and Monfort, A. Asymptotic properties of the maximum likelihood estimator in dichotomous logit models. *Journal of Econometrics*, 17(1):83–97, 1981.

Huang, C. and Huo, X. A distributed one-step estimator. *arXiv preprint arXiv:1511.01443*, 2015.

Jordan, M. I., Lee, J. D., and Yang, Y. Communication-efficient distributed statistical inference. *Journal of the American Statistical Association*, 114(526):668–681, 2019.

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

Lan, G., Lee, S., and Zhou, Y. Communication-efficient algorithms for decentralized and stochastic optimization. *Mathematical Programming*, pp. 1–48, 2018.

Lee, J. D., Liu, Q., Sun, Y., and Taylor, J. E. Communication-efficient sparse regression. *The Journal of Machine Learning Research*, 18(1):115–144, 2017.

Li, R., Lin, D. K., and Li, B. Statistical inference in massive data sets. *Applied Stochastic Models in Business and Industry*, 29(5):399–409, 2013.

Sengupta, S., Volgushev, S., and Shao, X. A subsampled double bootstrap for massive data. *Journal of the American Statistical Association*, 111(515):1222–1232, 2016.

Shi, C., Lu, W., and Song, R. A massive data framework for m-estimators with cubic-rate. *Journal of the American Statistical Association*, 113(524):1698–1709, 2018.

Singh, K. and Kaur, R. Hadoop: addressing challenges of big data. In *2014 IEEE International Advance Computing Conference (IACC)*, pp. 686–689. IEEE, 2014.

Tropp, J. A. et al. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

Van de Geer, S., Bühlmann, P., Ritov, Y., Dezeure, R., et al. On asymptotically optimal confidence regions and tests for high-dimensional models. *The Annals of Statistics*, 42(3):1166–1202, 2014.

Vershynin, R. How close is the sample covariance matrix to the actual covariance matrix? *Journal of Theoretical Probability*, 25(3):655–686, 2012.

Volgushev, S., Chao, S.-K., Cheng, G., et al. Distributed inference for quantile regression processes. *The Annals of Statistics*, 47(3):1634–1662, 2019.

Wang, J. and Zhang, T. Improved optimization of finite sums with minibatch stochastic variance reduced proximal iterations. *arXiv preprint arXiv:1706.07001*, 2017.

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

Yu, M., Gupta, V., and Kolar, M. Simultaneous inference for pairwise graphical models with generalized score matching. *arXiv preprint arXiv:1905.06261*, 2019.

Zhang, X. and Cheng, G. Simultaneous inference for high-dimensional linear models. *Journal of the American Statistical Association*, 112(518):757–768, 2017.

Zhang, Y., Wainwright, M. J., and Duchi, J. C. Communication-efficient algorithms for statistical optimization. In *Advances in Neural Information Processing Systems*, pp. 1502–1510, 2012.

Zhao, T., Cheng, G., and Liu, H. A partially linear framework for massive heterogeneous data. *Annals of statistics*, 44(4):1400, 2016.