A review of distributed statistical inference

Yuan Gao, Weidong Liu, Hansheng Wang, Xiaozhou Wang, Yibo Yan & Riquan Zhang

Published online: 13 Sep 2021.

Submit your article to this journal ⏍

Article views: 1965

View related articles ⏍

View Crossmark data ⏍

Taylor & Francis
Taylor & Francis Group

A review of distributed statistical inference

Yuan Gao[a], Weidong Liu[b], Hansheng Wang[c], Xiaozhou Wang[a], Yibo Yan[a] and Riquan Zhang[a]

[a]School of Statistics and Key Laboratory of Advanced Theory and Application in Statistics and Data Science – MOE, East China Normal University, Shanghai, People's Republic of China; [b]School of Mathematical Sciences and Key Lab of Artical Intelligence – MOE, Shanghai Jiao Tong University, Shanghai, People's Republic of China; [c]Guanghua School of Management, Peking University, Beijing, People's Republic of China

ABSTRACT
The rapid emergence of massive datasets in various fields poses a serious challenge to traditional statistical methods. Meanwhile, it provides opportunities for researchers to develop novel algorithms. Inspired by the idea of divide-and-conquer, various distributed frameworks for statistical estimation and inference have been proposed. They were developed to deal with large-scale statistical optimization problems. This paper aims to provide a comprehensive review for related literature. It includes parametric models, nonparametric models, and other frequently used models. Their key ideas and theoretical properties are summarized. The trade-off between communication cost and estimate precision together with other concerns is discussed.

1. Introduction

With the rapid development of information technology, datasets of massive sizes become increasingly available. E-commerce companies like Amazon have to analyse billions of transaction data for personalized recommendation. Bioinformatics scientists need to locate relevant genes corresponding to some specific phenotype or disease from massive SNPs data. For Internet-related companies, large amounts of text, image, voice, and even video data are in urgent need of effective analysis. Due to the accelerated growth of data sizes, the computing power and memory of one single computer are no longer sufficient. Constraint on network bandwidth and other privacy or security considerations also make it difficult to process the whole data on one central machine. Accordingly, distributed computing systems become increasingly popular.

Similar to parallel computing executed on a single computer, distributed computing is closely related to the idea of divide-and-conquer. Simply speaking, for some statistical problems, we can divide a complicated large task into many small pieces so that they can be tackled simultaneously on multiple CPUs or machines. Their outcomes are then aggregated to obtain the final result. It is conceivable that this procedure can save the computing time substantially if the algorithm can be executed in a parallel way. The main difference between a traditional parallel computing system and a distributed computing system is the way they

access memory. For parallel computing, different processors can share the same memory. Consequently, they can exchange information with each other in a super-efficient way. While for distributed computing, distinct machines are physically separated. They are often connected by a network. Accordingly, each machine can only access its own memory directly. Therefore, the inter-machine communication cost in terms of time spending could be significant and thus should be prudently considered.

The rest of this article is organized as follows. Section 2 studies parametric models. Section 3 focuses on nonparametric methods. Section 4 expresses some other related methods. The article is concluded with a short discussion in Section 5.

2. Parametric models

Assume a total of N observations denoted as $Z_i = (X_i^\top, Y_i)^\top \in \mathbb{R}^{p+1}$ with $1 \le i \le N$. Here $X_i \in \mathbb{R}^p$ is the covariate vector and $Y_i \in \mathbb{R}$ is the corresponding scalar response. Define $\{\mathbb{P}_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta\}$ to be a family of statistical models parameterized by $\boldsymbol{\theta} \in \Theta \subset \mathbb{R}^p$. We further assume that Z_i's are independent and identically distributed with the distribution $\mathbb{P}_{\boldsymbol{\theta}^*}$, where $\boldsymbol{\theta}^* = (\theta_1^*, \ldots, \theta_p^*)^\top$ is the true parameter. Consider a distributed setting, where N sample units are allocated randomly and evenly to K local machines $\mathcal{M}_k$, $1 \le k \le K$, such that each machine has

---

n observations. Obviously, we should have $N = nK$. Write $\mathbb{S} = \{1, \ldots, N\}$ as the index set of whole sample. Then, let $\mathcal{S}_k$ denote the index set of local sample on $\mathcal{M}_k$ with $\mathcal{S}_{k_1} \cap \mathcal{S}_{k_2} = \emptyset$ for any $k_1 \neq k_2$. Other than the local machines, there also exists a central machine represented by $\mathcal{M}_{\text{center}}$. A standard architecture should have $\mathcal{M}_{\text{center}}$ to be connected with every $\mathcal{M}_k$.

Let $\mathcal{L} : \Theta \times \mathbb{R}^{p+1} \mapsto \mathbb{R}$ be the loss function. Assume that the true parameter $\boldsymbol{\theta}^*$ minimizes the population risk $\mathcal{L}^*(\boldsymbol{\theta}) = \mathbb{E}[\mathcal{L}(\boldsymbol{\theta}; Z)]$, where $\mathbb{E}$ stands for expectation with respect to $\mathbb{P}_{\boldsymbol{\theta}*}$. Define the local loss on the kth machine as $\mathcal{L}_k(\boldsymbol{\theta}) = n^{-1} \sum_{i \in \mathcal{S}_k} \mathcal{L}(\boldsymbol{\theta}; Z_i)$. Correspondingly, define the global loss function based on the whole sample as $\mathcal{L}(\boldsymbol{\theta}) = N^{-1} \sum_{i \in \mathbb{S}} \mathcal{L}(\boldsymbol{\theta}; Z_i) = K^{-1} \sum_{k=1}^{K} \mathcal{L}_k(\boldsymbol{\theta})$, whose minimizer is $\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta} \in \Theta} \mathcal{L}(\boldsymbol{\theta})$. In most cases, the whole sample estimator $\hat{\boldsymbol{\theta}}$ should be $\sqrt{N}$-consistent and asymptotically normal (Lehmann & Casella, 2006). If N is small enough so that the whole sample $\mathbb{S}$ can be read into the memory of one single computer, then $\hat{\boldsymbol{\theta}}$ can be easily computed. The entire computation can be executed in the memory of this computer. On the other hand, if N is too large so that the whole sample $\mathbb{S}$ cannot be placed on any single computer, then a distributed system must be used. In this case, the whole sample estimator $\hat{\boldsymbol{\theta}}$ is no longer computable (or at least very difficult to compute) in practice. Then, how to develop novel statistical methods for distributed systems becomes a problem of great importance.

2.1. One-shot approach

To solve the problems, various methods have been proposed. They can be roughly divided into two classes. The first class contains so-called one-shot methods. They are to be reviewed in this subsection. The other class contains various iterative methods. They are to be reviewed in the next subsection.

The basic idea of the one-shot approach is to calculate some relevant statistics on each local machine. Subsequently, these statistics are sent to a central machine, where they are assembled into the final estimator. The most popular and direct way of aggregation is simple average. Specifically, for each $1 \leq k \leq K$, machine $\mathcal{M}_k$ uses local sample $\mathcal{S}_k$ to compute the local empirical minimizer $\hat{\boldsymbol{\theta}}_k = \arg\min_{\boldsymbol{\theta} \in \Theta} \mathcal{L}_k(\boldsymbol{\theta})$. These local estimates (i.e., $\hat{\boldsymbol{\theta}}_k$'s) are then transferred to the centre machine $\mathcal{M}_{\text{center}}$, where they are averaged as $\bar{\boldsymbol{\theta}} = K^{-1} \sum_{k=1}^{K} \hat{\boldsymbol{\theta}}_k$. This leads to the final simple averaging estimator $\bar{\boldsymbol{\theta}}$ (see Figure 1(a)).

Obviously, the one-shot style of distributed framework is highly communication-efficient. Because it requires only one single round of communication between each $\mathcal{M}_k$ and $\mathcal{M}_{\text{center}}$. Hence, the communication cost is of the order $O(Kp)$, where p is the dimension of each estimate $\hat{\boldsymbol{\theta}}$. Theoretical properties of simple averaging estimator were also studied in the literature. For example, it was shown in Zhang et al. (2013, Corollary 2) that, under appropriate regularity conditions,

$$\mathbb{E}\left\| \bar{\boldsymbol{\theta}} - \boldsymbol{\theta}^* \right\|_2^2 \leq \frac{C_1}{N} + \frac{C_2}{n^2} + O\left(\frac{1}{Nn} + \frac{1}{n^3} \right), \quad (1)$$

where C_1, C_2 are some positive constants. If n is sufficiently large such that $n^{-2} = o(N^{-1})$, then the dominant term in (1) becomes C_1/N, and is of the order $O(N^{-1})$. It is the same as that of the whole sample estimator. This also implies that, in order to obtain the global convergence rate, we should not divide the whole sample into too many parts. A further improved theoretical result was obtained by Rosenblatt and Nadler (2016). They showed that the one-shot estimator is the first order equivalent to the whole sample estimator. However, the second-order error terms of $\bar{\boldsymbol{\theta}}$ can be non-negligible for nonlinear models. Similar observation was also obtained by Huang and Huo (2015). The work of Duchi et al. (2014) revealed that the minimal communication budget to attain the global estimation error for linear regression is $\mathcal{O}(Kp)$ bits up to a logarithmic factor under some
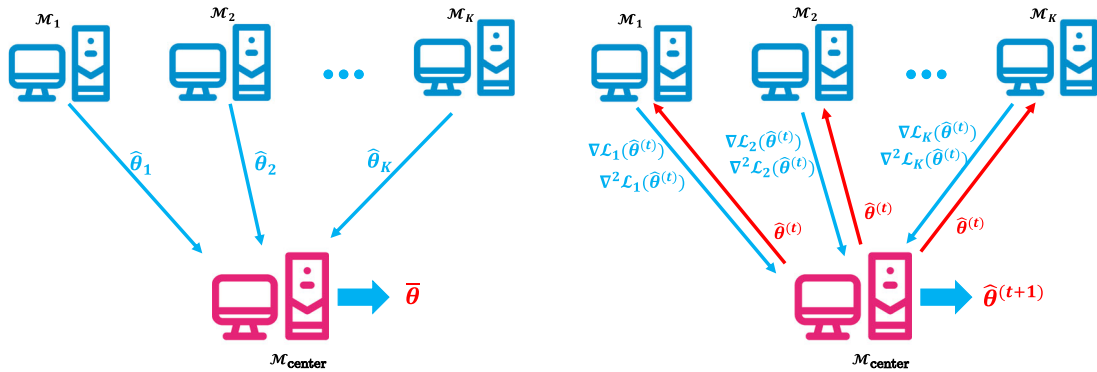
(a) Illustration of the one-shot approach.

(b) Illustration of the iterative approach.

Figure 1. Illustrations of the two different approaches. (a) one-shot approach and (b) iterative approach.

conditions. This result matches the simple averaging procedure and confirms the sharpness of the bound in (1). To further reduce the bias, a novel subsampling method was developed by Zhang et al. (2013). By this technique, the error bound is improved to be $O(N^{-1} + n^{-3})$, which relaxes the restriction on the number of machines.

Instead of the linear combination of local maximum likelihood estimates (MLEs) as simple average, Liu and Ihler (2014) proposed a KL-divergence based combination method. The final estimator is computed by

$$\hat{\boldsymbol{\theta}}_{\mathrm{KL}} = \arg\min_{\boldsymbol{\theta} \in \Theta} \sum_{k=1}^{K} \mathrm{KL}\left(p(x|\hat{\boldsymbol{\theta}}_k) \,\|\, p(x|\boldsymbol{\theta})\right),$$

where $p(x|\boldsymbol{\theta})$ is the probability density of $\mathbb{P}_{\boldsymbol{\theta}}$ with respect to some proper measure μ, and KL-divergence is defined by $\mathrm{KL}(p(x) \,\|\, q(x)) = \int_{\mathcal{X}} p(x) \, \log\{p(x)/q(x)\}\mathrm{d}\mu(x)$. It was shown that $\hat{\boldsymbol{\theta}}_{\mathrm{KL}}$ is exactly the global MLE $\hat{\boldsymbol{\theta}}$ if $\{\mathbb{P}_{\boldsymbol{\theta}} : \boldsymbol{\theta} \in \Theta\}$ is a full exponential family (defined in their paper). This sheds light on the inference about generalized linear models (GLMs) based on exponential likelihood.

In many cases, some local machines might suffer from data of poor quality. This could lead to abnormal local estimates, which further degrade the statistical efficiency of the final estimator. To fix the problem, Minsker (2019) devised a robust assembling method. It leads to an estimator as $\hat{\boldsymbol{\theta}}_{\mathrm{robust}} = \arg\min_{\boldsymbol{\theta} \in \Theta} \sum_{k=1}^{K} \rho(|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_k|)$, where $\rho(\cdot)$ is a robust loss function satisfying some conditions. For example, when $\rho(u) = u$ and $p = 1$ (univariate case), $\hat{\boldsymbol{\theta}}_{\mathrm{robust}}$ is the median of $\hat{\boldsymbol{\theta}}_k$'s. It should be more robust against outliers compared with the simple average. Under some regularity conditions, they showed that $\hat{\boldsymbol{\theta}}_{\mathrm{robust}}$ achieves the same convergence rate as the whole sample estimator provided $K \leq O(\sqrt{N})$.

2.2. Iterative approach

Although one-shot approach involves little communication cost, it suffers from several disadvantages. First, the local machines need to have sufficient amount of data (e.g., $n \gg \sqrt{N}$). Otherwise the aggregated estimator cannot reach the convergence rate as the global estimator. This prevents us from utilizing many machines to speed up the computation process (Jordan et al., 2019; Wang et al., 2017). Second, the simple averaging estimator is often poor in performance for nonlinear models (Huang & Huo, 2015; Jordan et al., 2019; Rosenblatt & Nadler, 2016). Last, when p is diverging with N, the situation could be even worse (Lee et al., 2017; Rosenblatt & Nadler, 2016). This suggests that carefully designed algorithms allowing a reasonable number of iterations should be useful for a distributed system.

Inspired by the one-step method in the M-estimator theory, Huang and Huo (2015) proposed an one-step refinement of the simple averaging estimator. Let us recall that $\bar{\boldsymbol{\theta}}$ is the one-shot averaging estimator. To further improve its statistical efficiency, it should be broadcast to each local machine. Next, local gradient $\nabla\mathcal{L}_k(\bar{\boldsymbol{\theta}})$ and local Hessian $\nabla^2\mathcal{L}_k(\bar{\boldsymbol{\theta}})$ can be computed on each $\mathcal{M}_k$. Then, they are reported to $\mathcal{M}_{\mathrm{center}}$ to form the whole sample gradient $\nabla\mathcal{L}(\bar{\boldsymbol{\theta}}) = K^{-1}\sum_{k=1}^{K} \nabla\mathcal{L}_k(\bar{\boldsymbol{\theta}})$ and Hessian $\nabla^2\mathcal{L}(\bar{\boldsymbol{\theta}}) = K^{-1}\sum_{k=1}^{K} \nabla^2\mathcal{L}_k(\bar{\boldsymbol{\theta}})$. Thus an one-step updated estimator can be constructed on $\mathcal{M}_{\mathrm{center}}$ as

$$\hat{\boldsymbol{\theta}}^{(1)} = \bar{\boldsymbol{\theta}} - [\nabla^2\mathcal{L}(\bar{\boldsymbol{\theta}})]^{-1}\nabla\mathcal{L}(\bar{\boldsymbol{\theta}}). \qquad (2)$$

Compared with one-shot estimator, $\hat{\boldsymbol{\theta}}^{(1)}$ involves one more round of communication cost. Nevertheless, the statistical efficiency of the resulting estimator could be well improved. In fact, Huang and Huo (2015) showed that

$$\mathbb{E}\|\hat{\boldsymbol{\theta}}^{(1)} - \boldsymbol{\theta}^*\|_2^2 \leq \frac{C_1}{N} + O\left(\frac{1}{n^4} + \frac{1}{N^2}\right),$$

where $C_1 > 0$ is some constant. Obviously, this is a lower upper bound of mean squared error than that in (1). To attain the global convergence rate, the local sample size needs to satisfy $n^{-4} = o(N^{-1})$, which is a much milder condition. Furthermore, they showed that $\hat{\boldsymbol{\theta}}^{(1)}$ also has the same asymptotic efficiency as the whole sample estimator $\hat{\boldsymbol{\theta}}$ under some regularity conditions.

A natural idea to further extend the one-step estimator is to allow the iteration (2) to be executed many times. Specifically, let $\hat{\boldsymbol{\theta}}^{(t)}$ be the estimator of the tth iteration. Then, we can use (2) by replacing $\bar{\boldsymbol{\theta}}$ with $\hat{\boldsymbol{\theta}}^{(t)}$ to generate the next step estimator $\hat{\boldsymbol{\theta}}^{(t+1)}$ (see Figure 1(b)). However, this requires a large number of Hessian matrices to be computed and transferred. If the parameter dimension p is relatively high, this will lead to a significant communication cost of the order $O(Kp^2)$. To fix the problem, Shamir et al. (2014) proposed an approximate Newton method, which conducts Newton-type iteration distributedly without transferring the Hessian matrices. Following this strategy, Jordan et al. (2019) developed an approximate likelihood approach. Their key idea is to update Hessian matrix on one single machine (e.g., $\mathcal{M}_{\mathrm{center}}$) only. Then, (2) can be revised to be

$$\hat{\boldsymbol{\theta}}^{(t+1)} = \hat{\boldsymbol{\theta}}^{(t)} - \left[\nabla^2\mathcal{L}_{\mathrm{center}}(\hat{\boldsymbol{\theta}}^{(t)})\right]^{-1}\nabla\mathcal{L}(\hat{\boldsymbol{\theta}}^{(t)}),$$

where $\nabla^2\mathcal{L}_{\mathrm{center}}$ is the Hessian matrix computed on the central machine. By doing so, the communication cost due to transmission of Hessian matrices can be saved.

Under some conditions, they showed that

$$\|\hat{\boldsymbol{\theta}}^{(t+1)} - \hat{\boldsymbol{\theta}}\|_2 \leq \frac{C_1}{\sqrt{n}} \|\hat{\boldsymbol{\theta}}^{(t)} - \hat{\boldsymbol{\theta}}\|_2, \quad \text{for } t \geq 0, \quad (3)$$

holds with high probability, where $C_1 > 0$ is some constant. By the linear convergence formula (3), we can see that it requires $[\log K / \log n]$ iterations to achieve the $\sqrt{N}$-consistency as the whole sample estimator $\hat{\boldsymbol{\theta}}$, provided $\hat{\boldsymbol{\theta}}^{(0)}$ is $\sqrt{n}$-consistent. Note that if $n = K = \sqrt{N}$, one iteration suffices to attain the optimal convergence rate. However, the satisfactory performance of this method relies on a good choice of the machine, on which the Hessian needs to be updated (Fan, Guo et al., 2019a). To fix the problems, Fan, Guo et al. (2019a) added an extra regularized term to the approximate likelihood used in Jordan et al. (2019). With this modification, the performance of the resulting estimator can be well improved. Theoretically, they showed a similar linear convergence rate under some more general conditions, which require no strict homogeneity of the local loss functions.

2.3. Shrinkage methods

We study the shrinkage methods for sparse estimation in this subsection. For a high-dimensional problem, especially when the dimension of $\boldsymbol{\theta}^*$ is larger than the sample size N, it is difficult to estimate $\boldsymbol{\theta}^*$ without any additional assumptions (Hastie et al., 2015). A popular constraint for tackling these problems is sparsity, which assumes only a subset of the entries in θ^* is non-zero. The index of non-zero entries is called the support of $\boldsymbol{\theta}^*$, that is

$$\text{supp}(\boldsymbol{\theta}^*) = \mathcal{A}^* = \left\{ 1 \leq j \leq p : \theta_j^* \neq 0 \right\}.$$

To induce a sparse solution, an additional regularization term of $\boldsymbol{\theta}$ is often introduced in the loss function. Specifically, we need to solve the shrinkage regression problem as $\min_{\boldsymbol{\theta} \in \Theta} \{ \mathcal{L}(\boldsymbol{\theta}) + \sum_{j=1}^{p} \rho_\lambda(|\theta_j|) \}$, where $\rho_\lambda(\cdot)$ is a penalty function with a regularization parameter $\lambda > 0$. Popular choices are LASSO (Tibshirani, 1996), SCAD (Fan & Li, 2001) and others discussed in Zhang and Zhang (2012). For simplicity, we consider the LASSO estimator in the framework of the linear regression problem. Specifically, the whole sample estimator is computed as

$$\hat{\boldsymbol{\theta}}_\lambda = \arg\min_{\boldsymbol{\theta} \in \Theta} \left\{ \frac{1}{N} \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_1 \right\},$$

where $\boldsymbol{Y} = (Y_1, \ldots, Y_N)^\top \in \mathbb{R}^N$ is the vector of response, $\boldsymbol{X} = (X_1, \ldots, X_N)^\top \in \mathbb{R}^{N \times p}$ is the design matrix, and $\|\boldsymbol{\theta}\|_1 = \sum_{j=1}^{p} |\theta_j|$ denotes the l_1-norm of $\boldsymbol{\theta}$. It is known that the LASSO procedure would produce biased estimators for the large coefficients. This is undesirable for the simple average procedures, since average

cannot eliminate the systematic bias. To reduce bias, Javanmard and Montanari (2014) proposed a debiasing technique for the lasso estimator, that is

$$\hat{\boldsymbol{\theta}}_\lambda^{(\text{d})} = \hat{\boldsymbol{\theta}}_\lambda + \frac{1}{N} \boldsymbol{M} \boldsymbol{X}^\top \left(\boldsymbol{Y} - \boldsymbol{X} \hat{\boldsymbol{\theta}}_\lambda \right), \quad (4)$$

where $\boldsymbol{M} \in \mathbb{R}^{p \times p}$ is an approximation to the inverse of $\hat{\Sigma} = \boldsymbol{X}^\top \boldsymbol{X} / N$. It appears that when $\hat{\Sigma}$ is invertible (e.g., when $N \gg p$), setting $\boldsymbol{M} = \hat{\Sigma}^{-1}$ gives $\hat{\boldsymbol{\theta}}_\lambda^{(\text{d})} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{Y}$, which is the ordinary least squares estimator and obviously unbiased. Hence, procedure (4) compensates for the bias incurred by ℓ_1 regularization in some sense.

By this debiasing technique, Lee et al. (2017) developed an one-shot type estimator for the LASSO problem. Specifically, let $\hat{\boldsymbol{\theta}}_{k,\lambda}^{(\text{d})}$ be the debiased LASSO estimator computed on $\mathcal{M}_k$. Then an averaging estimator can be constructed on $\mathcal{M}_{\text{center}}$ as $\bar{\boldsymbol{\theta}}_\lambda = K^{-1} \sum_{k=1}^{K} \hat{\boldsymbol{\theta}}_{k,\lambda}^{(\text{d})}$. Unfortunately, the sparsity level can be seriously degraded by averaging. For this reason, a hard threshold step often comes as a remedy. It was noticed that the debiasing step is computationally expensive. Hence an improved algorithm was also proposed to alleviate the computational cost of this step. Under certain conditions, they showed that the resulting estimator has the same convergence rate as the whole sample estimator. Battey et al. (2018) investigated the same problem with additional study on hypothesis testing. Furthermore, a refitted estimation procedure was used to preserve the global oracle property of the distributed estimator. An extension to high dimensional GLMs can also be found in Lee et al. (2017) and Battey et al. (2018). For this model, Chen and Xie (2014) implemented a majority voting method to aggregate the regularized local estimates. For the low dimensional sparse problem with smooth loss function (e.g., GLMs, Cox model), Zhu et al. (2019) developed a local quadratic approximation method with an adaptive-LASSO type penalty. They showed rigorously that the resulting estimator can be as good as the global oracle estimator.

Intuitively, above one-shot methods may need a stringent condition on the local sample size to meet the global convergence rate due to the limited communication. In fact, the simple averaging estimator requires $n \geq O(Ks^2 \log p)$ to match the oracle rate in the context of sparse linear model (Lee et al., 2017), where $s = |\mathcal{A}^*|$ is the number of non-zero entries of $\boldsymbol{\theta}^*$. For this problem, Wang et al. (2017) and Jordan et al. (2019) independently proposed a communication-efficient iterative algorithm, which constructs a regularized likelihood by using local Hessian matrix. As demonstrated by Wang et al. (2017), an one-step estimator $\hat{\boldsymbol{\theta}}^{(1)}$ suffices to achieve the global convergence rate if $n \geq O(Ks^2 \log p)$ (the condition used in Lee et al., 2017). Furthermore, if multi-round communication is allowed, $\hat{\boldsymbol{\theta}}^{(t+1)}$ (i.e., estimator of the $(t+1)$th

iteration) can match the estimator based on the whole sample as long as $n \geq O(s^2 \log p)$ and $t \geq O(\log K)$, under some certain conditions.

2.4. Non-smooth loss based models

The methods we described above typically require the loss function $\mathcal{L}$ to be sufficiently smooth, although a non-smooth regularization term is permitted (see e.g., Jordan et al., 2019; Wang et al., 2017; Zhang et al., 2013; Zhu et al., 2019). However, there are also some useful methods involving non-smooth loss functions, such as quantile regression and support vector machine. It is then of great interest to develop distributed methods for these methods.

We first focus on the quantile regression (QR) model. The QR model has a widespread use in statistics and econometrics, and performs more robustly against the outliers than the ordinary quadratic loss (Koenker, 2005). Specifically, a QR model assumes $Y_i = X_i^\top \boldsymbol{\theta}_\tau^* + \varepsilon_i$, $i \in \mathbb{S}$, where $X_i \in \mathbb{R}^p$ is the covariate vector, Y_i is the corresponding response, $\boldsymbol{\theta}_\tau^* \in \mathbb{R}^p$ is the true regression coefficient, and ε_i is the random noise satisfying $\mathbb{P}(\varepsilon_i \leq 0 | X_i) = \tau$, where $\tau \in (0,1)$ is a known quantile level. It is known that $\boldsymbol{\theta}_\tau^*$ is the minimizer of $\mathbb{E}[\rho_\tau(Y_i - X_i^\top \boldsymbol{\theta})]$. Here $\rho_\tau(u) = u(\tau - \mathbf{1}\{u \leq 0\}) = u(\mathbf{1}\{u > 0\} + \tau - 1)$ is the non-differentiable check-loss function, where $\mathbf{1}\{\cdot\}$ is the indicator function. When data size N is moderate, we can estimate $\boldsymbol{\theta}_\tau^*$ by $\hat{\boldsymbol{\theta}}_\tau = \min_{\boldsymbol{\theta} \in \Theta} N^{-1} \sum_{i \in \mathbb{S}} \rho_\tau(Y_i - X_i^\top \boldsymbol{\theta})$ on one single machine. However, when N is very large, a distributed system has to be used. Accordingly, distributed estimators have to be developed.

In this regard, Volgushev et al. (2019) studied the one-shot averaging type estimator. Specifically, a local estimator $\hat{\boldsymbol{\theta}}_{k,\tau}$ is first computed on each local machine $\mathcal{M}_k$. Then, the averaging estimator is assembled as $\bar{\boldsymbol{\theta}}_\tau = K^{-1} \sum_{k=1}^K \hat{\boldsymbol{\theta}}_{k,\tau}$ on the central machine $\mathcal{M}_{\text{center}}$. They further investigated the theoretical properties of the averaging estimator in detail. It was shown that the if the number of machines satisfies $K = o(\sqrt{N}/\log N)$, then $\bar{\boldsymbol{\theta}}_\tau$ should be as efficient as the whole sample estimator $\hat{\boldsymbol{\theta}}_\tau$ under some regularity conditions. Chen and Zhou (2020) proposed an estimating equation based one-shot approach for the QR problem. The asymptotic equivalence between the resulting estimator and the whole sample estimator was also established under $K = o(N^{1/4})$ and some other conditions. It can be seen that the performance of one-shot approaches relies more on the local sample size. In fact, Volgushev et al. (2019) showed that $K = o(\sqrt{N})$ is a necessary condition for the global efficiency of the simple averaging estimator $\bar{\boldsymbol{\theta}}_\tau$. To remove the constraint $K = o(\sqrt{N})$ on the number of machines, Chen et al. (2019) proposed an iterative approach. Their key idea is to approximate the check-loss function by a smooth alternative. More specifically, they approximated $\mathbf{1}\{u > 0\}$

by a smooth function $H(u/h)$, where $H(\cdot)$ is a smooth cumulative distribution function and $h > 0$ is the tuning parameter controlling the approximation goodness (see Figure 2(a)). With this modification, the algorithm can update the estimates by

$$\hat{\boldsymbol{\theta}}_\tau^{(t+1)} = \left[V(\hat{\boldsymbol{\theta}}_\tau^{(t)}) \right]^{-1} U(\hat{\boldsymbol{\theta}}_\tau^{(t)}), \qquad (5)$$

where $U(\boldsymbol{\theta}) = \sum_{k=1}^K U_k(\boldsymbol{\theta})$, $V(\boldsymbol{\theta}) = \sum_{k=1}^K V_k(\boldsymbol{\theta})$, and $U_k \in \mathbb{R}^p$, $V_k \in \mathbb{R}^{p \times p}$ depend only on the bandwidth h and local sample $\mathcal{S}_k$. It was shown that a constant number of rounds of iteration suffices to match the convergence rate of the whole sample estimator. Thus, the communication cost is roughly of the order $O(Kp^2)$. But this is not applicable when p is very large. Thus, for the high dimensional QR problem, Zhao et al. (2014) and Zhao et al. (2019) adopted an one-shot averaging method based on the debiased local estimates as that in (4). Accordingly, Chen et al. (2020) proposed a communication-efficient multi-round algorithm inspired by the approximate Newton method (Shamir et al., 2014). This iterative approach removes the restriction on the number of machines. A revised divide-and-conquer stochastic gradient descent method for QR and other models with diverging dimension can be found in Chen, Liu et al. (2021b).

We next consider the support vector machine (SVM), which is one of the most successful statistical learning methods (Vapnik, 2013). The classical SVM is aimed at the binary classification problem, i.e., the response variable $Y_i \in \{-1, 1\}$. Formally, a standard linear SVM solves the problem $\min_{\boldsymbol{\theta} \in \Theta} N^{-1} \sum_{i \in \mathbb{S}} (1 - Y_i X_i^\top \boldsymbol{\theta})_+ + \lambda \|\boldsymbol{\theta}\|_2^2$, where $(u)_+ = u\mathbf{1}(u > 0)$ is the hinge loss, and $\lambda > 0$ is the regularization parameter. By the same smooth technique used in Chen et al. (2019), i.e., replacing the hinge loss with a smooth alternative (see Figure 2(b)), Wang, Yang et al. (2019) proposed an iterative algorithm like (5). To reduce the communication cost incurred by transferring matrices, they further employed the approximate Newton method (Shamir et al., 2014). Theoretically, they showed the asymptotic normality of the estimator, which can be used to construct confidence interval. For the ultra-high dimensional SVM problem, Lian and Fan (2018) studied the one-shot averaging method with debiasing procedure similar to (4).

3. Nonparametric models

Different from parametric models, a nonparametric model typically involves infinite-dimensional parameters. In this section, we focus mainly on the nonparametric regression problems. Specifically, consider here a general regression model as $Y_i = f^*(X_i) + \varepsilon_i$, $i \in \mathbb{S}$, where $f^*(\cdot)$ is an unknown but sufficiently smooth function and ε_i is the random noise with zero mean. The aim of nonparametric regression is to estimate
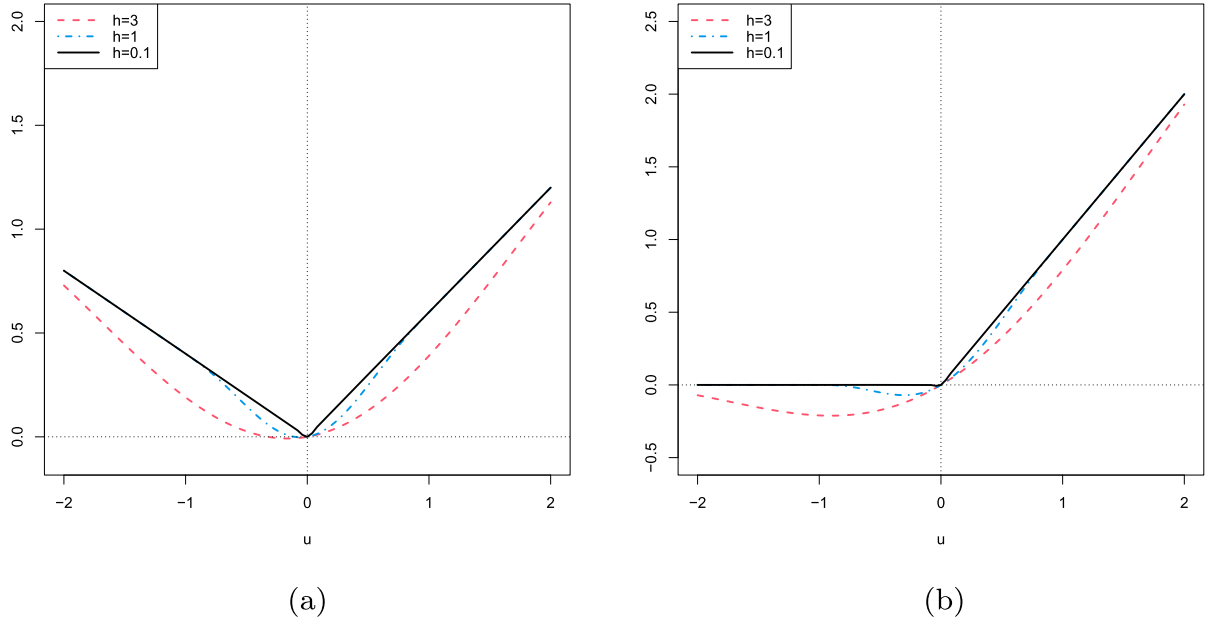
Figure 2. Approximation of two non-smooth loss functions. (a) QR loss with $\tau = 0.6$ and (b) hinge loss.

function $f^* \in \mathcal{F}$, where $\mathcal{F}$ is a given nonparametric class of functions.

3.1. Local smoothing

One way to estimate $f^*(\cdot)$ is to fit a locally constant model by kernel smoothing (Fan & Gijbels, 1996). More concretely, the whole sample estimator is given by

$$\hat{f}_h(x) = \sum_{i \in \mathbb{S}} W_{h,X_i}(x) Y_i,$$

where the $W_{h,X_i}(x) \geq 0$ is the local weight at $X = x$ satisfying $\sum_{i \in \mathbb{S}} W_{h,X_i}(x) = 1$. Specifically, for a Nadaraya–Watson kernel estimator, we should have $W_{h,X_i}(x) = K((X_i - x)/h)/\sum_{i' \in \mathbb{S}} K((X_{i'} - x)/h)$, where $K(\cdot)$ is a kernel function and $h > 0$ is the bandwidth. In the univariate case ($p = 1$), classical theory stated that the mean squared error of $\hat{f}_h(x)$ is of the order $O(h^4 + (Nh)^{-1})$ (Fan & Gijbels, 1996). Thus, the optimal rate $O(N^{-4/5})$ can be achieved by choosing bandwidth $h = O(N^{-1/5})$.

For a distributed kernel smoothing, an one-shot estimator can also be constructed. Let $\hat{f}_{k,h}(x)$ be the local estimator computed on $\mathcal{M}_k$. Then an averaging estimator can be obtained as $\bar{f}_h(x) = K^{-1} \sum_{k=1}^{K} \hat{f}_{k,h}(x)$. Chang, Lin, Wang (2017) studied the theoretical properties of $\bar{f}_h(x)$ in a specific function space $\mathcal{F}$. They established the same minimax convergence rate of $\bar{f}_h(x)$ as that of the whole sample estimator. However, they found that a strict restriction on the number of machines K is needed to achieve this optimal rate. To fix the problem, two solutions were provided. They are, respectively, a date-dependent bandwidth selection algorithm and an algorithm with a qualication step.

Nearest neighbours method can be regarded as another local smoothing method. Qiao et al. (2019) studied the Nearest neighbours classification in a distributed setting, where the optimal number of neighbours to achieve the optimal rate of convergence was derived. Li et al. (2013) discussed the problem of density estimation for scattered datasets. Kaplan (2019) focused on the choice of bandwidth for nonparametric smoothing techniques. All the works above in this subsection indicate that the bandwidth (or local smoothing parameter) used in the distributed setting should be adjusted according to the whole sample size N, other than the local sample size n.

3.2. RKHS methods

We next discuss another popular nonparametric regression method. This is reproducing kernel Hilbert space (RKHS) method. An RKHS $\mathcal{H}$ can be induced by a continuous, symmetric and positive semi-definite kernel function $\mathcal{K}(\cdot, \cdot) : \mathbb{R}^p \times \mathbb{R}^p \mapsto \mathbb{R}$. Two typical examples are: the polynomial kernel $\mathcal{K}(x_1, x_2) = (x_1^\top x_2 + 1)^d$ with an integer $d \geq 1$, and the radical kernel $\mathcal{K}(x_1, x_2) = \exp(-\gamma \|x_1 - x_2\|_2^2)$ with $\gamma > 0$. Refer to, for example, Berlinet and Thomas-Agnan (2011); Wahba (1990) for more details about RKHS. Then, our target is to find an $\hat{f} \in \mathcal{H}$ so that the following penalized empirical loss can be minimized. That leads to the whole sample estimator as

$$\hat{f}_\lambda = \arg\min_{f \in \mathcal{H}} \left\{ \frac{1}{N} \sum_{i \in \mathbb{S}} (Y_i - f(X_i))^2 + \lambda \|f\|_{\mathcal{H}}^2 \right\},$$

(6)

where $\| \cdot \|_{\mathcal{H}}$ is the norm associated with the RKHS $\mathcal{H}$ and $\lambda > 0$ is the regularization parameter. This problem is also known as kernel ridge regression (KRR). By the representer theorem for the RKHS (Wahba, 1990), any solution to the problem (6) must have the linear form as $\hat{f}_\lambda(x) = \sum_{i \in \mathbb{S}} \alpha_i \mathcal{K}(X_i, x)$, where $\alpha_i \in \mathbb{R}$ for each $i \in \mathbb{S}$. By this property, we can treat the KRR as a parametric problem with unknown parameter $\alpha = (\alpha_1, \ldots, \alpha_N)^\top \in \mathbb{R}^N$. The error bounds of the whole sample estimator $\hat{f}_\lambda$ has been well established in the existing literature (e.g., Steinwart et al., 2009; Zhang, 2005). However, a standard implementation of the KRR involves inverting a kernel matrix in $\mathbb{R}^{N \times N}$ (Saunders et al., 1998). Therefore, when N is extremely large, it is time consuming or even computationally infeasible to process the whole sample on a single machine. Thus, we should consider a distributed system.

In this regard, Zhang et al. (2015) studied the distributed KRR by taking the one-shot averaging approach. Specifically, each machine $\mathcal{M}_k$ computes local KRR estimate $\hat{f}_{k,\lambda}$ by (6) based on local sample $\mathcal{S}_k$. Then the central machine $\mathcal{M}_{\text{center}}$ averages them to obtain final estimator $\bar{f}_\lambda = K^{-1} \sum_{k=1}^{K} \hat{f}_{k,\lambda}$. Theoretically, they established the optimal convergence rate of mean squared error for $\bar{f}_\lambda$ with different types of kernel functions, under some regularity conditions. Lin et al. (2017) derived a similar optimal error bound under some relaxed conditions. Xu et al. (2016) extended the loss function in (6) to a further general form. Some related works on the distributed KRR problem by one-shot averaging approach can be found in Shang and Cheng (2017), Lin and Zhou (2018), Guo et al. (2019); Mücke and Blanchard (2018) and Wang (2019) and many others. It was noted that these one-shot approaches require the number of machines diverges in a relative slow speed to meet the global convergence rate. To fix the problem, Chang, Lin, Zhou (2017) proposed a semi-supervised learning framework by utilizing the additional unlabelled data (i.e., observations without response Y_i). Latest work of Lin et al. (2020) allowed communication between machines to improve the performance. In order to choose an optimal tuning parameter λ in (6), Xu et al. (2018) proposed a distributed generalized cross-validation method.

For semiparametric models, Zhao et al. (2016) considered a partially linear model with heterogeneous data in a distributed setting. Specifically, they assumed the following model

$$Y_i = X_i^\top \boldsymbol{\theta}_{(k)}^* + f^*(W_i) + \varepsilon_i, \quad i \in \mathcal{S}_k, \qquad (7)$$

where $W_i \in \mathbb{R}$ is an additional covariate, $f^*(\cdot)$ is the unknown function, and $\boldsymbol{\theta}_{(k)}^* \in \mathbb{R}^p$ is the true linear coefficient associated with the data on $\mathcal{M}_k$ for $1 \leq k \leq K$. In other words, the local data on different machines

are assumed to share the same nonparametric part, but are allowed to have different linear coefficients. To estimate the unknown function and coefficients, they extended the classical RKHS theory to cope with the partially linear function space. Under some regularity conditions, the resulting estimator of the nonparametric part is shown to be as efficient as the whole sample estimator, provided the number of machines does not grow too fast. The case with high dimensional linear part was also investigated. For example, under the homogeneity assumption (i.e., the linear coefficients $\boldsymbol{\theta}_{(k)}^*$'s are assumed to be identical to $\boldsymbol{\theta}^*$ across different machines), Lv and Lian (2017) adopted the one-shot averaging approach with debiasing technique analogous to (4) to estimate the linear coefficient. Lian et al. (2019) considered the same heterogeneous model as in (7), but the linear part is assumed in a high dimensional setting (i.e., $p > N$). For this model, they proposed a novel projection approach to estimate the common nonparametric part (not in an RKHS framework). Theoretically, the asymptotic normality of the one-shot averaging estimator for the nonparametric function was established under some certain conditions.

4. Other related works

4.1. Principal component analysis

Principal component analysis (PCA) is a common procedure to reduce the dimension of the data. It is widely used in the practical data analysis. Unlike the regression problems, PCA is an unsupervised method, which does not require a response variable Y. To conduct a PCA, a covariance matrix $\hat{\Sigma}$ needs to be constructed as $\hat{\Sigma} = N^{-1} \sum_{i \in \mathbb{S}} X_i X_i^\top$, where X_i's are assumed to be centralized already. Next, a standard singular value decomposition (SVD) is applied to $\hat{\Sigma}$. That leads to $\hat{\Sigma} = \hat{V} \hat{D} \hat{V}^\top$, where $\hat{D}$ is a diagonal matrix of eigenvalues and $\hat{V}$ is an orthogonal matrix of eigenvectors. Then, the columns of $\hat{V}$ are the principal component directions that we need.

In a distributed setting, simple average of the eigenvectors estimated locally cannot give a valid result. To solve the problem, Fan, Wang et al. (2019b) developed a divide-and-conquer algorithm for estimating eigenspaces. It involves only one single round of communication. This algorithm is quite easy to implement as well. We state it as follows (Fan, Wang et al., 2019b, Algorithm 1):

(1) For each $k = 1, \ldots, K$, machine $\mathcal{M}_k$ computes d leading eigenvectors of the local sample covariance matrix $\hat{\Sigma}_k = n^{-1} \sum_{i \in \mathcal{S}_k} X_i X_i^\top$, denoted by $\hat{v}_{1,k}, \ldots, \hat{v}_{d,k} \in \mathbb{R}^p$. Next, they are arranged by columns in $\hat{V}_k = (\hat{v}_{1,k}, \ldots, \hat{v}_{d,k}) \in \mathbb{R}^{p \times d}$, which is then sent to the central machine $\mathcal{M}_{\text{center}}$.

(2) The central machine $\mathcal{M}_{\text{center}}$ averages K local projection matrices to obtain $\tilde{\Sigma} = K^{-1} \sum_{k=1}^{K} \hat{V}_k \hat{V}_k^{\top}$. Then it computes d leading eigenvectors of $\tilde{\Sigma}$, denoted by $\tilde{v}_1, \ldots, \tilde{v}_d \in \mathbb{R}^p$. $\tilde{v}_1, \ldots, \tilde{v}_d$ are the estimators of the first d principal component directions that we need.

It is noticeable that the communication cost of above one-shot algorithm is of the order $O(Kdp)$. This can be considered to be communication-efficient since d is usually much smaller than p in practice. Fan, Wang et al. (2019b) showed that, under some appropriate conditions, the distributed estimator achieves the same convergence rate as the global estimator. The cases with heterogeneous local data were also investigated in their work. To further remove the restriction on the number of machines, Chen, Lee et al. (2021) proposed a communication-efficient multi-round algorithm based on the approximate Newton method (Shamir et al., 2014).

4.2. Feature screening

Massive datasets often involve ultrahigh dimensional data, for which feature screening is critically important (Fan & Lv, 2008). To fix the idea, consider a standard linear regression model as $Y_i = X_i^{\top} \boldsymbol{\theta}^* + \varepsilon_i$, $i \in \mathbb{S}$, where $X_i \in \mathbb{R}^p$ is the covariate vector, Y_i is the corresponding response, $\boldsymbol{\theta}^* \in \mathbb{R}^p$ is the true parameter, and ε_i is the random noise. To screen for the most promising features, the seminal method of sure independence screening (SIS) has been proposed by Fan and Lv (2008). Specifically, let $\mathcal{A}^* = \{1 \leq j \leq p : \theta_j^* \neq 0\}$ be the true sparse model. Let ω_j be the Pearson correlation between jth feature and response Y. Then, SIS screens features by a hard threshold procedure as $\hat{\mathcal{A}}_{\gamma} = \{1 \leq j \leq p : |\hat{\omega}_j| > \gamma\}$, where γ is a prespecified threshold and $\hat{\omega}_j$ is the whole sample estimator of ω_j. Under some specific conditions, Fan and Lv (2008) showed the sure screening property for SIS, that is,

$$\mathbb{P}(\mathcal{A}^* \subset \hat{\mathcal{A}}_{\gamma}) \to 1 \quad \text{as } N \to \infty.$$

However, the estimator $\hat{\omega}_j$ is usually biased for many correlation measures. This indicates that a direct one-shot averaging approach is unlikely to be the best practice for the distributed system. To fix the problem, Li et al. (2020) proposed a novel debiasing technique. They found that many correlation measures can be expressed as $\omega_j = g(v_1, \ldots, v_s)$, including Pearson correlation used above, Kendall τ rank correlation, SIRS correlation (Zhu et al., 2011), etc. Therefore, they used U-statistics to estimate the components $v_q, 1 \leq q \leq s$ on local machines. Then, these unbiased estimators of v_q's given by local machines are averaged on the central machine $\mathcal{M}_{\text{center}}$. Consequently, $\mathcal{M}_{\text{center}}$ can construct distributed estimator $\tilde{\omega}_j$ by the averaging estimators of

the components in the known function g. Finally, they showed the sure screening property of $\hat{\mathcal{A}}_{\gamma}$ based on the distributed estimators under some regularity conditions. When the feature dimension is much larger than the sample size (i.e., $p \gg N$), another distributed computing strategy is to partition the whole data by features, other than by samples. Refer to, for example, Song and Liang (2015); Yang et al. (2016) for more details.

4.3. Bootstrap

Bootstrap and related resampling techniques provide a general and easily implemented procedure for automatically statistical inference. However, these methods are usually computationally expensive. Especially when sample size N is very large, it would be even practically infeasible to conduct. To mitigate this computing issue, various alternative methods have been proposed, such as subsamping approach (Politis et al., 1999) and 'm-out-of-n' bootstrap (Bickel et al., 2012). Their key idea is to reduce the resample size. However, due to the difference between the size of whole sample and resample, an additional correction step is generally required to rescale the result. This makes these methods less automatic.

To solve this problem, Kleiner et al. (2014) proposed the bag of little bootstraps (BLB) method. It integrates the idea of subsampling and can be computed distributedly without a correction step. Suppose that N sample units have been randomly and evenly partitioned to K machines. Consider that we want to assess the accuracy of the point estimator for some parameter $\boldsymbol{\theta}$. Then we summarize their algorithm as follows.

(1) For each $1 \leq k \leq K$, machine $\mathcal{M}_k$ draws r samples of size N (instead of n) from $\mathcal{S}_k$ with replacement. Then it computes r estimates of $\boldsymbol{\theta}$ based on the r resamples drawn above, respectively. After that, each $\mathcal{M}_k$ computes some accuracy measure (e.g., variance, confidence region) by the r estimates above, denoted by $\hat{\xi}_k$. Finally, all of the local machines send $\hat{\xi}_k$'s to the central machine $\mathcal{M}_{\text{center}}$.
(2) The central machine $\mathcal{M}_{\text{center}}$ aggregates these $\hat{\xi}_k$'s by $\bar{\xi} = K^{-1} \sum_{k=1}^{K} \hat{\xi}_k$. And $\bar{\xi}$ is the final accuracy measure that we need.

It is remarkable that one does not need to process datasets of size N on local machines actually, although the nominal size of resample is N. This is because each machine contains at most n sample units. In fact, randomly generating some certain weight vectors of length n suffices to approximate the resampling process.

5. Future study

To conclude the article, we would like to discuss here a number of interesting topics for future study. First,

for datasets with massive sizes, a distributed system is definitely needed. Obviously, there could be no place to store the data. On the other hand, for datasets with sufficiently small sizes, traditional memory based statistical methods can be immediately used. Then, there leaves a big gap between the big and small datasets. Those middle-sized data are often of sizes much larger than the computer memory but smaller than the hard drive. Consequently, they can be comfortably placed on a personal computer, but can hardly be processed by memory as a whole. For those datasets, their sizes are not large enough to justify an expensive distributed system. They are also not small enough to be handled by traditional statistical methods. How to analyse datasets of this size seems to be a topic worth studying. Second, when the whole data are allocated to local machines randomly and evenly, the data on different machines are independent and identically distributed and balanced. Then, all of the methods discussed above can be safely implemented. However, when the data on different machines are collected from (for example) different regions, the homogeneity of the local data would normally be hard to satisfy. The situation could be even worse if the sample sizes allocated to different local machines are very different. How to cope with these heterogeneous and unbalanced local data is a problem of great importance (Wang et al., 2020). The idea of meta analysis may be applicable to these situations (Liu et al., 2015; Xu & Shao, 2020; Zhou & Song, 2017). Finally, in the era of big data, personal privacy is under unprecedented threat. How to protect users' private information during the learning process deserves urgent attention. In this regard, differential privacy (DP) provides a theoretical approach for privacy-preserving data analysis (Dwork, 2008). Some related works associated with distributed learning are Agarwal et al. (2018), Truex et al. (2019) and Wang, Ishii et al. (2019) and many others. Although it is a hot research area recently, there are still many open challenges. Thus, it is of great interest to study the privacy-preserving distributed statistical learning problems practically and theoretically.

Notes on contributors

Mr. Yuan Gao is a Ph.D. candidate in school of statistics at East China Normal University.

Dr. Weidong Liu is the Distinguished Professor in school of mathematical sciences at Shanghai Jiao Tong University.

Dr. Hansheng Wang is a professor in Guanghua School of Management at Peking University.

Dr. Xiaozhou Wang is an assistant professor in school of statistics at East China Normal University.

Mr. Yibo Yan is a Ph.D. candidate in school of statistics at East China Normal University.

Dr. Riquan Zhang is a professor in school of statistics at East China Normal University.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

References

Agarwal, N., Suresh, A. T., Yu, F., Kumar, S., & McMahan, H. B. (2018). cpsgd: Communication-efficient and differentially-private distributed sgd. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (pp. 7575–7586). Curran Associates Inc.

Battey, H., Fan, J., Liu, H., Lu, J., & Zhu, Z. (2018). Distributed testing and estimation under sparse high dimensional models. *Annals of Statistics*, *46*(3), 1352–1382. https://doi.org/10.1214/17-AOS1587

Berlinet, A., & Thomas-Agnan, C. (2011). *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Springer Science & Business Media.

Bickel, P. J., Götze, F., & van Zwet, W. R. (2012). Resampling fewer than *n* observations: Gains, losses, and remedies for losses. In *Selected Works of Willem van Zwet* (pp. 267–297). Springer.

Chang, X., Lin, S.-B., & Wang, Y. (2017). Divide and conquer local average regression. *Electronic Journal of Statistics*, *11*(1), 1326–1350. https://doi.org/10.1214/17-EJS1265

Chang, X., Lin, S.-B., & Zhou, D.-X. (2017). Distributed semi-supervised learning with kernel ridge regression. *The Journal of Machine Learning Research*, *18*(1), 1493–1514. https://jmlr.org/papers/volume18/16-601/16-601.pdf

Chen, L., & Zhou, Y. (2020). Quantile regression in big data: A divide and conquer based strategy. *Computational Statistics & Data Analysis*, *144*, 106892. https://doi.org/10.1016/j.csda.2019.106892

Chen, X., Lee, J. D., Li, H., & Yang, Y. (2021). Distributed estimation for principal component analysis: An enlarged eigenspace analysis. *Journal of the American Statistical Association*, 1–31. https://doi.org/10.1080/01621459.2021.1886937

Chen, X., Liu, W., Mao, X., & Yang, Z. (2020). Distributed high-dimensional regression under a quantile loss function. *Journal of Machine Learning Research*, *21*(182), 1–43. https://jmlr.org/papers/volume21/20-297/20-297.pdf

Chen, X., Liu, W., & Zhang, Y. (2019). Quantile regression under memory constraint. *The Annals of Statistics*, *47*(6), 3244–3273. https://doi.org/10.1214/18-AOS1777

Chen, X., Liu, W., & Zhang, Y. (2021b). First-order newton-type estimator for distributed estimation and inference.

Journal of the American Statistical Association, 1–40. https://doi.org/10.1080/01621459.2021.1891925

Chen, X., & Xie, M.-G. (2014). A split-and-conquer approach for analysis of extraordinarily large data. *Statistica Sinica*, 1655–1684. http://dx.doi.org/10.5705/ss.2013.088

Duchi, J. C., Jordan, M. I., Wainwright, M. J., & Zhang, Y. (2014). Optimality guarantees for distributed statistical estimation. *arXiv preprint arXiv:1405.0782*.

Dwork, C. (2008). Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation* (pp. 1–19). Springer.

Fan, J., & Gijbels, I. (1996). *Local polynomial modelling and its applications: Monographs on statistics and applied probability 66* (Vol. 66). CRC Press.

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

Fan, J., & Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456), 1348–1360. https://doi.org/10.1198/016214501753382273

Fan, J., & Lv, J. (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5), 849–911. https://doi.org/10.1111/rssb.2008.70.issue-5

Fan, J., Wang, D., Wang, K., & Zhu, Z. (2019b). Distributed estimation of principal eigenspaces. *The Annals of Statistics*, 47(6), 3009–3031. https://doi.org/10.1214/18-AOS1713

Guo, Z.-C., Lin, S.-B., & Shi, L. (2019). Distributed learning with multi-penalty regularization. *Applied and Computational Harmonic Analysis*, 46(3), 478–499. https://doi.org/10.1016/j.acha.2017.06.001

Hastie, T., Tibshirani, R., & Wainwright, M. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC press.

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

Javanmard, A., & Montanari, A. (2014). Confidence intervals and hypothesis testing for high-dimensional regression. *The Journal of Machine Learning Research*, 15(1), 2869–2909. https://jmlr.csail.mit.edu/papers/volume15/javanmard14a/javanmard14a.pdf

Jordan, M. I., Lee, J. D., & Yang, Y. (2019). Communication-efficient distributed statistical inference. *Journal of the American Statistical Association*, 114(526), 668–681. https://doi.org/10.1080/01621459.2018.1429274

Kaplan, D. M. (2019). Optimal smoothing in divide-and-conquer for big data. Technical report, working paper available at https://faculty.missouri.edu/kaplandm.

Kleiner, A., Talwalkar, A., Sarkar, P., & Jordan, M. I. (2014). A scalable bootstrap for massive data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(4), 795–816. https://doi.org/10.1111/rssb.2014.76.issue-4

Koenker (2005). *Quantile Regression (Econometric Society Monographs; No. 38)*. Cambridge University Press.

Lee, J. D., Liu, Q., Sun, Y., & Taylor, J. E. (2017). Communication-efficient sparse regression. *The Journal of Machine Learning Research*, 18(1), 115–144. https://jmlr.csail.mit.edu/papers/volume18/16-002/16-002.pdf

Lehmann, E. L., & Casella, G. (2006). *Theory of Point Estimation*. Springer Science & Business Media.

Li, R., Lin, D. K., & Li, B. (2013). Statistical inference in massive data sets. *Applied Stochastic Models in Business and Industry*, 29(5), 399–409. https://doi.org/10.1002/asmb.1927

Li, X., Li, R., Xia, Z., & Xu, C. (2020). Distributed feature screening via componentwise debiasing. *Journal of Machine Learning Research*, 21(24), 1–32. https://jmlr.csail.mit.edu/papers/volume21/19-537/19-537.pdf

Lian, H., & Fan, Z. (2018). Divide-and-conquer for debiased l1-norm support vector machine in ultra-high dimensions. *Journal of Machine Learning Research*, 18(1), 1–26. https://jmlr.org/papers/volume18/17-343/17-343.pdf

Lian, H., Zhao, K., & Lv, S. (2019). Projected spline estimation of the nonparametric function in high-dimensional partially linear models for massive data. *Annals of Statistics*, 47(5), 2922–2949. https://doi.org/10.1214/18-AOS1769

Lin, S.-B., Guo, X., & Zhou, D.-X. (2017). Distributed learning with regularized least squares. *The Journal of Machine Learning Research*, 18(1), 3202–3232. https://www.jmlr.org/papers/volume18/15-586/15-586.pdf

Lin, S.-B., Wang, D., & Zhou, D.-X. (2020). Distributed kernel ridge regression with communications. *arXiv preprint arXiv:2003.12210*.

Lin, S.-B., & Zhou, D.-X. (2018). Distributed kernel-based gradient descent algorithms. *Constructive Approximation*, 47(2), 249–276. https://doi.org/10.1007/s00365-017-9379-1

Liu, D., Liu, R. Y., & Xie, M. (2015). Multivariate meta-analysis of heterogeneous studies using only summary statistics: efficiency and robustness. *Journal of the American Statistical Association*, 110(509), 326–340. https://doi.org/10.1080/01621459.2014.899235

Liu, Q., & Ihler, A. T. (2014). Distributed estimation, information loss and exponential families. In *Advances in Neural Information Processing Systems* (pp. 1098–1106). MIT Press.

Lv, S., & Lian, H. (2017). Debiased distributed learning for sparse partial linear models in high dimensions. *arXiv preprint arXiv:1708.05487*.

Minsker, S. (2019). Distributed statistical estimation and rates of convergence in normal approximation. *Electronic Journal of Statistics*, 13(2), 5213–5252. https://doi.org/10.1214/19-EJS1647

Mücke, N., & Blanchard, G. (2018). Parallelizing spectrally regularized kernel algorithms. *The Journal of Machine Learning Research*, 19(1), 1069–1097. https://www.jmlr.org/papers/volume19/16-569/16-569.pdf

Politis, D. N., Romano, J. P., & Wolf, M. (1999). *Subsampling*. Springer Science & Business Media.

Qiao, X., Duan, J., & Cheng, G. (2019). Rates of convergence for large-scale nearest neighbor classification. In *Advances in Neural Information Processing Systems* (pp. 10768–10779). Curran Associates Inc.

Rosenblatt, J. D., & Nadler, B. (2016). On the optimality of averaging in distributed statistical learning. *Information and Inference: A Journal of the IMA*, 5(4), 379–404. https://doi.org/10.1093/imaiai/iaw013

Saunders, C., Gammerman, A., & Vovk, V. (1998). Ridge regression learning algorithm in dual variables. In *International Conference on Machine Learning* (pp. 515–521).

Shamir, O., Srebro, N., & Zhang, T. (2014). Communication-efficient distributed optimization using an approximate newton-type method. In *International Conference on Machine Learning* (pp. 1000–1008). JMLR.org.

Shang, Z., & Cheng, G. (2017). Computational limits of a distributed algorithm for smoothing spline. *Journal of Machine Learning Research*, 18(108), 1–37. https://jmlr.org/papers/volume18/16-289/16-289.pdf

Song, Q., & Liang, F. (2015). A split-and-merge bayesian variable selection approach for ultrahigh dimensional regression. *Journal of the Royal Statistical Society: Series*

B: Statistical Methodology, *77*(5), 947–972. https://doi.org/10.1111/rssb.12095.

Steinwart, I., Hush, D. R., & Scovel, C. (2009). Optimal rates for regularized least squares regression. In *COLT* (pp. 79–93). https://www.cs.mcgill.ca/∼colt2009/papers/038.pdf

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

Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R., & Zhou, Y. (2019). A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security* (pp. 1–11). Association for Computing Machinery.

Vapnik, V. (2013). *The Nature of Statistical Learning Theory*. Springer Science & Business Media.

Volgushev, S., Chao, S.-K., & Cheng, G. (2019). Distributed inference for quantile regression processes. *The Annals of Statistics*, *47*(3), 1634–1662. https://doi.org/10.1214/18-AOS1730

Wahba, G. (1990). *Spline Models for Observational Data* (Vol. 59). Siam.

Wang, F., Huang, D., Zhu, Y., & Wang, H. (2020). Efficient estimation for generalized linear models on a distributed system with nonrandomly distributed data. *arXiv preprint arXiv:2004.02414*.

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

Wang, S. (2019). A sharper generalization bound for divide-and-conquer ridge regression. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 33, pp. 5305–5312). AAAI Press.

Wang, X., Ishii, H., Du, L., Cheng, P., & Chen, J. (2019). Differential privacy-preserving distributed machine learning. In *2019 IEEE 58th Conference on Decision and Control (CDC)* (pp. 7339–7344). IEEE.

Wang, X., Yang, Z., Chen, X., & Liu, W. (2019). Distributed inference for linear support vector machine. *Journal of Machine Learning Research*, *20*(113), 1–41. https://www.jmlr.org/papers/volume20/18-801/18-801.pdf

Xu, C., Zhang, Y., Li, R., & Wu, X. (2016). On the feasibility of distributed kernel regression for big data. *IEEE Transactions on Knowledge and Data Engineering*, *28*(11), 3041–3052. https://doi.org/10.1109/TKDE.2016.2594060

Xu, G., Shang, Z., & Cheng, G. (2018). Optimal tuning for divide-and-conquer kernel ridge regression with massive data. In *Proceedings of Machine Learning Research*. PMLR.

Xu, M., & Shao, J. (2020). Meta-analysis of independent datasets using constrained generalised method of moments. *Statistical Theory and Related Fields*, *4*(1), 109–116. https://doi.org/10.1080/24754269.2019.1630545

Yang, J., Mahoney, M. W., Saunders, M. A., & Sun, Y. (2016). Feature-distributed sparse regression: A screen-and-clean approach. In *NIPS* (pp. 2712–2720). Curran Associates, Inc.

Zhang, C.-H., & Zhang, T. (2012). A general theory of concave regularization for high-dimensional sparse estimation problems. *Statistical Science*, *27*(4), 576–593. https://doi.org/10.1214/12-STS399

Zhang, T. (2005). Learning bounds for kernel regression using effective data dimensionality. *Neural Computation*, *17*(9), 2077–2098. https://doi.org/10.1162/0899766054323008

Zhang, Y., Duchi, J., & Wainwright, M. (2015). Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *The Journal of Machine Learning Research*, *16*(1), 3299–3340. https://jmlr.org/papers/volume16/zhang15d/zhang15d.pdf

Zhang, Y., Duchi, J. C., & Wainwright, M. J. (2013). Communication-efficient algorithms for statistical optimization. *The Journal of Machine Learning Research*, *14*(1), 3321–3363. https://www.jmlr.org/papers/volume14/zhang13b/zhang13b.pdf

Zhao, T., Cheng, G., & Liu, H. (2016). A partially linear framework for massive heterogeneous data. *Annals of Statistics*, *44*(4), 1400. https://doi.org/10.1214/15-AOS1410

Zhao, T., Kolar, M., & Liu, H. (2014). A general framework for robust testing and confidence regions in high-dimensional quantile regression. *arXiv preprint arXiv:1412.8724*.

Zhao, W., Zhang, F., & Lian, H. (2019). Debiasing and distributed estimation for high-dimensional quantile regression. *IEEE Transactions on Neural Networks and Learning Systems*, *31*(7), 2569–2577. https://doi.org/10.1109/TNNLS.2019.2933467

Zhou, L., & Song, P. X.-K. (2017). Scalable and efficient statistical inference with estimating functions in the mapreduce paradigm for big data. *arXiv preprint arXiv:1709.04389*.

Zhu, L.-P., Li, L., Li, R., & Zhu, L.-X. (2011). Model-free feature screening for ultrahigh-dimensional data. *Journal of the American Statistical Association*, *106*(496), 1464–1475. https://doi.org/10.1198/jasa.2011.tm10563

Zhu, X., Li, F., & Wang, H. (2019). Least squares approximation for a distributed system. *arXiv preprint arXiv:1908.04904*.