ANNUAL REVIEWS

Annual Review of Statistics and Its Application

Distributed Computing and Inference for Big Data

Ling Zhou, Ziyang Gong, and Pengcheng Xiang

Center of Statistical Research and School of Statistics, Southwestern University of Finance and Economics, Chengdu, China; email: zhouling@swufe.edu.cn

Annu. Rev. Stat. Appl. 2024. 11:533–51

First published as a Review in Advance on November 17, 2023

The *Annual Review of Statistics and Its Application* is online at statistics.annualreviews.org

https://doi.org/10.1146/annurev-statistics-040522-021241

Keywords

communication efficiency, distributed learning, federated learning, heterogeneity, statistical equivalence

Abstract

Data are distributed across different sites due to computing facility limitations or data privacy considerations. Conventional centralized methods—those in which all datasets are stored and processed in a central computing facility—are not applicable in practice. Therefore, it has become necessary to develop distributed learning approaches that have good inference or predictive accuracy while remaining free of individual data or obeying policies and regulations to protect privacy. In this article, we introduce the basic idea of distributed learning and conduct a selected review on various distributed learning methods, which are categorized by their statistical accuracy, computational efficiency, heterogeneity, and privacy. This categorization can help evaluate newly proposed methods from different aspects. Moreover, we provide up-to-date descriptions of the existing theoretical results that cover statistical equivalency and computational efficiency under different statistical learning frameworks. Finally, we provide existing software implementations and benchmark datasets, and we discuss future research opportunities.

1. INTRODUCTION

The distributed inference method integrates results derived from different analyses conducted on datasets from multiple study sites. If a supercomputer with infinite power and no data sharing barriers were available, there would be no need to develop new methods for processing big data, and existing methodologies and their associated software could be directly applied. Unfortunately, current computers do not have such capacity, nor do they allow operational convenience in terms of merging multiple datasets stored at different sites; thus, the distributed learning (DL) paradigm has emerged as a state-of-the-art computational solution to make distributed data computations feasible. Over the past 10 years, many studies have developed distributed computing and inference methods to address the various problems presented by distributed data. **Figure 1** shows the rapid development of DL in recent years and five highly cited articles each in computer science and statistics. In this article, we provide a selected review of the existing distributed computing and inference methods. First, we describe the basic idea of DL, which focuses on the divide-and-conquer strategy, and its early development. Second, we discuss the current state of the DL literature, as categorized by four essential aspects: statistical accuracy, computational efficiency, heterogeneity, and privacy. Third, we summarize the current theoretical advances that are related to the equivalence of distributed estimators and centralized estimators with all data stored together. Fourth, we briefly review the existing software implementations, open source platforms, and benchmark datasets. Finally, we recommend directions for future research to make DL more practical and powerful.

2. Distributed Learning

In response to the rapidly growing demands for big data analytics and computational tools, parallel computing and distributed data storage have become the leading innovations for solving big data

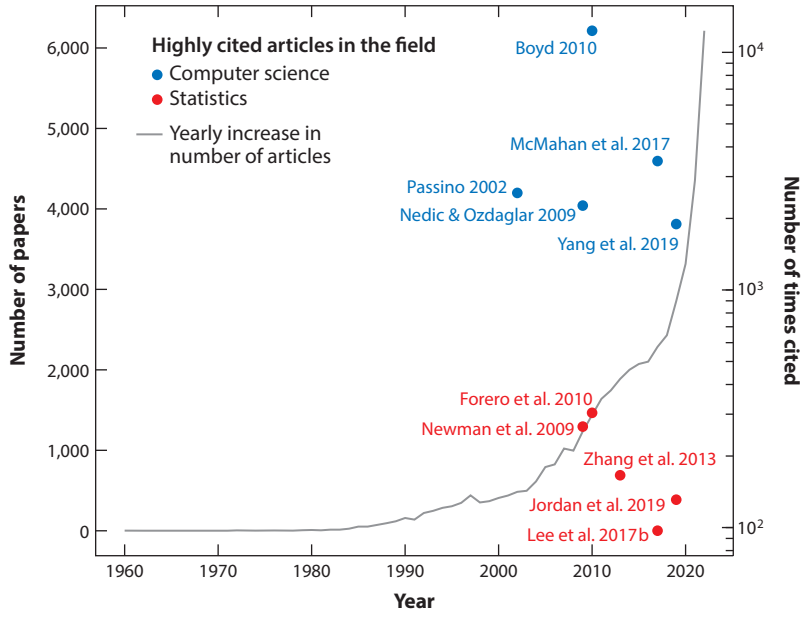

Figure 1

Frequency of published articles concerning distributed learning in recent years. All data were retrieved from **https://www.scopus.com/**.

problems. For example, data from different hospitals are isolated and become data islands. Because each data island is limited in terms of size and the number of variables, appropriately and collaboratively integrating all data islands allows practitioners to use data analysis to answer a scientific hypothesis of interest and/or obtain good predictive accuracy for a specific task. Multicore and cloud computing platforms, including the popular open source Hadoop (White 2015) platform, are now the standard software technologies that are extensively used in academia and industry (Taylor 2010, Hammoud et al. 2012, Jiao et al. 2012, Dittrich et al. 2013, Fernando et al. 2013). This new distributed file system necessitates the development of a general statistical method that allows for analyzing massive data through parallel and scalable operations in a distributed software framework.

DL can be traced back to Menabrea (1843). Its earlier development can be attributed to parallel computing, which was executed on a single computer and was built upon the divide-and-conquer strategy (Horowitz & Zorat 1983). For most statistical problems, we can divide a large complex task into many small pieces so that they can be approached simultaneously on multiple central processing units (CPUs) or machines. The outcomes are then aggregated to obtain the final results. For parallel computing purposes, different processors can share the same memory. Therefore, they can exchange information, including individual data, with each other in any efficient way (Gao et al. 2022). As the data size increases, the data must be stored at multiple locations. Built upon the divide-and-conquer strategy, the MapReduce paradigm refactors data processing into two primitives: a map function, which is written by the user to process distributed local data batches and generate intermediate results, and a reduce function, which is also written by the user to combine all intermediate results and then generate summary outputs (Dean & Ghemawat 2008).

Figure 2 shows a schematic outline of the MapReduce workflow, which splits data and performs computing tasks through parallel computation (Zhou & Song 2017). The salient features of MapReduce include the scalability and independence of data storage; the former enables the

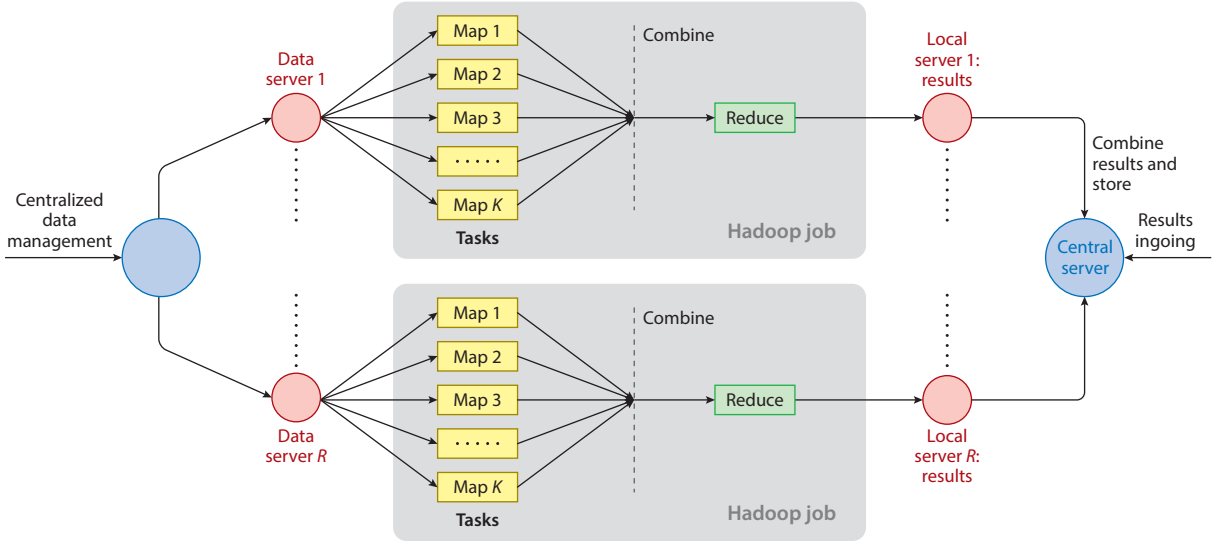

Figure 2

Schematic flow chart of the multiserver distributed data management and processing strategy according to the MapReduce paradigm as the heart of the Hadoop platform. The data partition scheme consists of R disjoint subsets that are distributed stored. Based on the data subset, a processor simultaneously processes K tasks in parallel. Adapted with permission from Zhou & Song (2017).

automatic parallelization and allocation of large-scale computations, and the latter allows data to be processed without requiring them to be loaded into a common data server. This process reduces the computational costs of loading input data into a centralized data server prior to conducting an analysis. Although MapReduce and its variants have performed well when processing large-scale data-intensive applications on high-performance clusters, most of these systems are restricted by acyclic data flows, which are not suitable for general statistical analyses because iterative numerical operations are involved. This issue arises because the operation of an iterative algorithm, such as Newton–Raphson, requires the repeated reloading of data from multiple data disks into a common server, incurring a large performance penalty. To solve the abovementioned problem, existing work can be categorized into two directions: (*a*) one-shot learning and (*b*) communication-iterative learning.

2.1. One-Shot Learning

Many real-world distributed data networks are not fully automated (Toh et al. 2017). Research sites often use their own analysts to manage their local data management and analysis tasks, which may not be automated by application programming interfaces; thus, full automation is challenging, and the number of allowable communication rounds is typically small. This type of data exists predominantly in distributed health care platforms, insurance provider networks, and collaborative research. One-shot learning requires just one round of communication between the local machines and the central server, which markedly reduces the required communication effort and broadens the application use cases to a wide variety of the abovementioned nonautomated systems. Due to the one-time communication process, one-shot methods often require a large sample size for each local site. Methods belonging to this category include the following.

2.1.1. Simple averaging. The most popular and direct aggregation method, simple averaging, averages the estimators obtained on each local machine. For each $1 \le k \le K$, the local estimates (i.e., the $\hat{\boldsymbol{\theta}}_k$s) are transferred to the central machine to obtain the final average estimator:

$$\hat{\boldsymbol{\theta}}_{\text{ave}} = \frac{1}{K}\sum_{k=1}^{K}\hat{\boldsymbol{\theta}}_k.$$

Under appropriate regularity conditions, $\hat{\boldsymbol{\theta}}_{\text{ave}}$ is the same as that of the entire sample estimator in terms of its convergence rate, i.e., it has first-order equivalence (Zhang et al. 2013, Rosenblatt & Nadler 2016, Battey et al. 2018, Chen et al. 2022).

2.1.2. Meta estimators. A simple average is easy to calculate but loses efficiency by ignoring the variance of each local estimator. To achieve improved efficiency, meta estimators $\hat{\boldsymbol{\theta}}_{\text{meta}}$ are defined as the inverse variance-weighted average of $\hat{\boldsymbol{\theta}}_k$:

$$\hat{\boldsymbol{\theta}}_{\text{meta}} = \left(\sum_{k=1}^{K}\text{Var}^{-1}(\hat{\boldsymbol{\theta}}_k)\right)^{-1}\left(\sum_{k=1}^{K}\text{Var}^{-1}(\hat{\boldsymbol{\theta}}_k)\hat{\boldsymbol{\theta}}_k\right).$$

These estimators were developed by Borenstein et al. (2021), Lin & Zeng (2010), and Liu et al. (2015). Liu et al. (2015) showed that their meta estimator was asymptotically as efficient as the maximum likelihood estimator (MLE) derived from using the entire input dataset once. With random-effects models, Zeng & Lin (2015) reported a similar finding; thus, their meta estimator was at least as efficient as that obtained from all data. Lin & Xi (2011) proposed an aggregated equation estimator.

2.1.3. Confidence distribution estimators. Meta-type estimators are objective-free methods. Later, using a confidence distribution (CD) (Xie & Singh 2013), the CD estimator was developed to offer a more general sample-dependent distribution-based estimator. Its biggest contribution is that it provides a flexible objective for aggregating local information by multiplying confidence densities. In particular, we denote $h_k(\boldsymbol{\theta}; \mathcal{D}_k)$ as a confidence density function derived from the kth study with dataset $\mathcal{D}_k$:

$$h_k(\boldsymbol{\theta}; \mathcal{D}_k) = \frac{1}{(2\pi)^{p/2}|\hat{\Sigma}_k|^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_k)^\top \hat{\Sigma}_k^{-1}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_k)\right\},$$

where $\hat{\boldsymbol{\theta}}_k$ is the estimator based on dataset $\mathcal{D}_k$ and $\hat{\Sigma}_k$ is its variance estimator. Then, a CD estimator is constructed by maximizing the following combined CDs:

$$\hat{\boldsymbol{\theta}}_{\mathrm{CD}} = \arg\max_{\boldsymbol{\theta}} h^{(c)}(\boldsymbol{\theta}; \mathcal{D}_1, \ldots, \mathcal{D}_k) := \arg\max_{\boldsymbol{\theta}} \prod_{k=1}^{K} h_k(\boldsymbol{\theta}; \mathcal{D}_k).$$

Related research includes the work of Singh et al. (2005) for univariate CDs, the study conducted by Liu et al. (2015) for multivariate common parameters, the work of Tang et al. (2020) for high-dimensional debiased estimators, and the research of Shen et al. (2020) for individualized fusion learning, among others. When each local sample size is large, the normal density assumption for the local CD is appropriately attributed to the central limit theorem. Clearly, CD-based estimators include meta estimators and average estimators as special cases. Due to the linear property of the Gaussian distribution, CD estimators can be treated as linear combinations of local estimators.

2.1.4. Nonlinear combined estimator. Liu & Ihler (2014) proposed a Kullback–Leibler (KL) divergence–based combination method, i.e.,

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

where $p(\mathcal{D}_k \mid \boldsymbol{\theta})$ is the density probability, and the KL divergence is defined by $\mathrm{KL}(p(x)\|q(x)) = \int p(x)\log\left(p(x)/q(x)\right)\mathrm{d}\mu(x)$. Liu & Ihler (2014) showed that $\hat{\boldsymbol{\theta}}_{\mathrm{KL}}$ is exactly the global MLE $\hat{\boldsymbol{\theta}}$ if p is a full exponential family.

To mitigate the impact of local sites with potentially poor quality, Minsker (2019) proposed a robust assembling method as follows:

$$\hat{\boldsymbol{\theta}}_{\mathrm{robust}} = \arg\min_{\boldsymbol{\theta}} \sum_{k=1}^{K} \rho(|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_k|),$$

where $\rho(\cdot)$ is a robust loss function.

2.2. Communication-Iterative Learning

Although one-shot approaches have the lowest communication costs, they suffer from several disadvantages. First, compared with a centralized estimator, which is estimated based on all individual data, a one-shot estimator incurs an estimation accuracy loss. To obtain the same convergence rate with a one-shot estimator and a centralized estimator, each local site should have sufficient data relative to the number of sites (Wang et al. 2017, Jordan et al. 2019). A higher-order equivalency between a one-shot estimator and a centralized estimator requires a larger sample size for each local site if a debiased procedure is not conducted. Second, a one-shot estimator requires a stronger homogeneity assumption across the local sites; for example, in regression problems, the covariates should share a common distribution. If some covariates are highly unbalanced in several local

sites, then one-shot estimators may suffer from large variations across the local sites or even fail to provide any results (Zhou et al. 2022). Therefore, objective-dependent algorithms allowing a reasonable number of iterations may lead to better estimation results in terms of stronger or higher-order equivalence and weaker conditions regarding the number of sites, the sample size needed for each local site, homogeneity, etc.

In particular, we consider an objective $\phi(\boldsymbol{\theta}; \mathcal{D}_1, \ldots, \mathcal{D}_K) = \frac{1}{K}\sum_{k=1}^{K} \phi_k(\boldsymbol{\theta}; \mathcal{D}_k)$ with K sites. We then denote $\dot{\phi}$ and $\ddot{\phi}$ as the first- and second-order derivatives of ϕ with respect to $\boldsymbol{\theta}$, respectively. Under the smoothness condition of ϕ, a centralized estimator can be obtained via the following iterative algorithms:

$$\hat{\boldsymbol{\theta}}_{\text{cen}} = \hat{\boldsymbol{\theta}}^{\infty}, \quad \hat{\boldsymbol{\theta}}^{(r)} = \hat{\boldsymbol{\theta}}^{(r-1)} - \eta \dot{\phi}(\boldsymbol{\theta}^{(r-1)}), \; r = 1, 2, \ldots,$$

where η is the step size tuning parameter. With η prespecified, the algorithm belongs to the class of gradient decent methods, which enjoy a linear convergence rate in general. When $\eta := (\ddot{\phi}(\boldsymbol{\theta}^{(r-1)}))^{-1}$, the algorithm is the Newton–Raphson algorithm, which has a quadratic convergence rate.

With a distributed framework, gradient-based methods require the gradient vector $\dot{\phi}$ to be transferred during each round of communication, and Newton–Raphson methods must transfer an additional Hessian matrix $\ddot{\phi}$ during each round. With unlimited time, both gradient descent methods and Newton–Raphson methods generate the same distributed estimators as a centralized estimator (i.e., $\hat{\boldsymbol{\theta}}_{\text{cen}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{K}\sum_{k=1}^{K} \phi_k(\boldsymbol{\theta}; \mathcal{D}_k)$). When computational complexity or time is considered, compared with gradient-based methods, the Newton–Raphson algorithm requires fewer communication rounds at the cost of transferring both gradients $\dot{\phi}$ and Hessians $\ddot{\phi}$. The balance between the number of communication rounds and computational complexity must therefore be considered. When the dimensionality of $\boldsymbol{\theta}$ is large, particularly in deep learning fields, transferring a Hessian matrix or calculating a Hessian matrix is difficult or impossible. For the maximum likelihood method, we can use the Bartlett identity to replace the Hessian matrix with the outer product of the gradient, which requires only a gradient transfer operation. However, for general loss functions, transferring an approximate Newton-type method is one direction to reduce communication costs and achieve communication efficiency. Methods of this type include the following.

2.2.1. Distributed approximate Newton-type method. Shamir et al. (2014) proposed a distributed approximate Newton-type method (DANE), where the Hessian matrices are not transferred. In particular, for an objective $\phi(\boldsymbol{\theta}) = \frac{1}{K}\sum_{k=1}^{K} \phi_k(\boldsymbol{\theta})$ with K local sites, they denoted $\nabla\phi$ and $\nabla^2\phi$ as the gradient and Hessian, respectively. They then updated $\boldsymbol{\theta}$ as follows:

$$\boldsymbol{\theta}_k^{(r)} = \arg\min_{\boldsymbol{\theta}} \left\{ \phi(\boldsymbol{\theta}^{(r-1)}) + \langle \nabla\phi(\boldsymbol{\theta}^{(r-1)}), \boldsymbol{\theta} - \boldsymbol{\theta}^{(r-1)} \rangle + \frac{1}{\eta} D_k(\boldsymbol{\theta}; \boldsymbol{\theta}^{(r-1)}) \right\},$$

where the first two terms $\phi(\boldsymbol{\theta}^{(r-1)}) + \langle \nabla\phi(\boldsymbol{\theta}^{(r-1)}), \boldsymbol{\theta}^{(r-1)} \rangle$ are linear approximations of the overall objective $\phi(\boldsymbol{\theta})$ concerning the current iteration $\boldsymbol{\theta}^{(r-1)}$ and do not depend on site k, and

$$D_i(\boldsymbol{\theta}; \boldsymbol{\theta}^{(r-1)}) = \phi_i(\boldsymbol{\theta}) - \phi_i(\boldsymbol{\theta}^{(r-1)}) - \langle \nabla\phi_i(\boldsymbol{\theta}^{(r-1)}), \boldsymbol{\theta} - \boldsymbol{\theta}^{(r-1)} \rangle + \frac{\mu}{2} \|\boldsymbol{\theta} - \boldsymbol{\theta}^{(r-1)}\|_2^2.$$

The DANE estimator at iteration r is $\hat{\boldsymbol{\theta}}_{\text{DANE}}^{(r)} = \frac{1}{K}\sum_{k=1}^{K} \hat{\boldsymbol{\theta}}_k^{(r)}$. After a simple calculation,

$$\boldsymbol{\theta}_k^{(r)} \approx \boldsymbol{\theta}^{(r-1)} - \eta \left(\nabla^2\phi_i(\boldsymbol{\theta}^{(r-1)}) + \mu I \right)^{-1} \nabla\phi(\boldsymbol{\theta}^{(r-1)}),$$

$$\boldsymbol{\theta}^{(r)} \approx \boldsymbol{\theta}^{(r-1)} - \eta \left(\frac{1}{K}\sum_{k=1}^{K} \left(\nabla^2\phi_i(\boldsymbol{\theta}^{(r-1)}) + \mu I \right)^{-1} \right) \nabla\phi(\boldsymbol{\theta}^{(r-1)}).$$

This result contrasts that of the true Newton-type update, which replaces $\left(\frac{1}{K}\sum_{k=1}^{K}(\nabla^2\phi_i(\boldsymbol{\theta}^{(r-1)}) + \mu I)^{-1}\right)$ with $\left(\frac{1}{K}\sum_{k=1}^{K}\nabla^2\phi_k(\boldsymbol{\theta}^{(r-1)})\right)^{-1}$. This difference is ignorable when $\nabla\phi_k(\boldsymbol{\theta}), k = 1, \ldots, K$ are similar. Thus, $\hat{\boldsymbol{\theta}}_{\text{DANE}}$ approximates the true Newton update without communicating Hessians. Zhang & Lin (2015) also avoided the direct transfer of Hessians from local sites to the center machine via an inexact damped Newton method.

2.2.2. Communication-efficient surrogate likelihood. Jordan et al. (2019) developed a communication-efficient surrogate likelihood (CSL) framework to solve distributed statistical inference problems, where the key idea is to update the Hessian matrix on a single site only. Thus,

$$\hat{\boldsymbol{\theta}}_{\text{CSL}}^{(r)} = \hat{\boldsymbol{\theta}}^{(r-1)} - \left(\nabla^2\ell_1(\hat{\boldsymbol{\theta}}^{(r-1)})\right)^{-1}\nabla\ell(\hat{\boldsymbol{\theta}}^{(r-1)}),$$

where ℓ is the log-likelihood and $\nabla\ell(\boldsymbol{\theta}) = \frac{1}{K}\sum_{k=1}^{K}\nabla\ell_k(\boldsymbol{\theta})$. The Hessian matrix is thus calculated on a single site, and transmission costs can be saved. However, the satisfactory performance of CSL relies on a properly selected local site, and Fan et al. (2021) added an additional regularized term to prevent sensitivity to the choice of a single machine.

2.2.3. Alternating direction method of multipliers–based method. Zhou et al. (2022) developed an alternating direction method of multipliers (ADMM)-based algorithm to distinguish between local parameters and global parameters and avoid transferring the Hessian matrix. Thus, we have

$$\hat{\boldsymbol{\theta}}_{\text{ADMM}} := \hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\theta}_k,\boldsymbol{\alpha}}\left\{\frac{1}{K}\sum_{k=1}^{K}\phi_k(\boldsymbol{\theta}_k; \mathcal{D}_k), \boldsymbol{\theta}_k \equiv \boldsymbol{\alpha}\right\}.$$

Utilizing the ADMM algorithm, we update the local parameters $\boldsymbol{\theta}_k$ at each local site by minimizing $\frac{1}{K}\phi_k(\boldsymbol{\theta}_k; \mathcal{D}_k) + \rho(\boldsymbol{\alpha}^{(r-1)} - \boldsymbol{\theta}_k - \boldsymbol{\lambda}_k^{(r-1)})^2$, where ρ is a tuning parameter. At a central machine, we update the global parameter $\boldsymbol{\alpha}^{(r)} = \frac{1}{K}\sum_{k=1}^{K}(\boldsymbol{\theta}_k^{(r-1)} + \boldsymbol{\lambda}_k^{(r-1)})$. Two sets of local vectors $\boldsymbol{\theta}_k$ and $\boldsymbol{\lambda}_k$ are then transferred to the central machine, and the global vector $\boldsymbol{\alpha}$ is transferred to each local site. Other related studies include those conducted by Boyd (2010) and Zhou & Li (2021).

When the given regression model has high dimensionality, regularization-based methods are commonly used. Further advancement in this area necessitates the formulation of meticulously designed transferring statistics. In a related study, the application of one-shot learning within a sparsity structure was demonstrated (Chen & Xie 2014). Battey et al. (2018) explored the utilization of a high-dimensional generalized linear model. Song et al. (2015) employed a linear model featuring feature splitting techniques. Lee et al. (2017b) adopted a broader perspective by utilizing a general smooth convex loss and debiased lasso techniques. Lv & Lian (2017) introduced a novel approach through a partial linear model incorporating a debiased reproducing kernel Hilbert space method. Lian & Fan (2018) innovatively addressed the challenge by developing a hinge loss with debiased l_1-support vector machine techniques. Wang et al. (2017) explored communication-iterative learning in the context of a sparsity structure. Tuning parameters, including the bandwidth in local polynomial methods and the number of splines in spline-based methods, usually determine the convergence rates of nonparametric methods and thus must be carefully chosen under a distributed framework. Related studies include those of Xu et al. (2016), Shang & Cheng (2017), Szabó et al. (2019), Banerjee et al. (2019), and Cai & Wei (2022), among others.

3. PROPERTIES

With the recent and rapid development of DL methods, the criteria for evaluating their performance can be summarized into the following types.

1. Statistical accuracy: Due to computing facility limitations or data use agreements, individual datasets from different data sites cannot be aggregated together into a single machine under distributed platforms. To aggregate information from different local data sites, several summary statistics are calculated and transferred. The equivalence of the resultant aggregated estimators to centralized estimators with all individual data combined is one of the most important criteria for measuring the performance of distributed methods. One-shot estimators (Shamir et al. 2014, Battey et al. 2018, Shi et al. 2018, Volgushev et al. 2019, Zhao et al. 2016, Fan et al. 2019) possess first-order equivalence to centralized estimators and typically require the number of sites to not be too high and the data size at each site to be large. To relax the conditions regarding the number of sites and the data size at each site, several subsampling methods (Kleiner et al. 2014) and debiased methods (Zhang et al. 2013, Lee et al. 2017b, Lian & Fan 2018) have been developed. Communication-iterative methods have also been developed to relax these conditions (Arjevani & Shamir 2015, Zhang & Lin 2015, Wang et al. 2017).
2. Computational efficiency: Under the distributed framework, communications implemented across different local data sites must integrate information. More communication rounds lead to greater communication time requirements. Conversely, fewer rounds of communication usually require the calculation of more complex statistics, implying increased information complexity being transferred and more restrictive conditions. For example, we denote p as the dimensionality of the parameters. Then, gradient-based distributed methods require more communications to converge but with only the gradient being transferred [i.e., the size of the transferred statistics is on the order of $O(p)$], while Newton-based methods require the Hessian and gradient to be transferred simultaneously [i.e., the size of the transferred statistics is on the order of $O(p^2)$] with fewer communications. Distributed methods with limited communications and restricted types of information transfer are becoming more important (Zhu & Jin 2020). To attain increased computational efficiency, we must balance the number of communication rounds and the communication complexity of such methods. Shamir et al. (2014) proposed DANE, which requires fewer communications than the Newton-based method by transferring gradients only. By measuring the dependence between the number of communication rounds, complexity and statistical accuracy, we can evaluate the performance of distributed methods from the computational aspect. As shown by Jordan et al. (2019), to ensure the same asymptotic distribution as that of a centralized estimator, at least $\lceil \log K / \log n \rceil$ iterations are needed, where K and n represent the number of sites and the data size at each local site, respectively. Other related studies include those by Wang et al. (2019), Shamir et al. (2014), Zhang & Lin (2015), and Fan et al. (2021), among others.
3. Heterogeneity: The success of DL methods lies in the integration of similar information. However, integrating all information, including both similar and heterogeneous information, generates biased estimates and unreliable inferences (Karimireddy et al. 2020, Yu et al. 2022). As the data distributions across local sites are typically not completely identical, the development of methods that are more personalized toward individual sites while borrowing strength from similar individuals has attracted growing interest (Smith et al. 2017). To appropriately address heterogeneity, the existing studies can be categorized into four directions:

- (*a*) Target (personalized) learning: Considering a target site, the goal of target learning is to achieve improved inference or prediction accuracy for the target site by appropriately borrowing information from other sites. Related studies include those conducted by Dinh et al. (2020) and Ghosh et al. (2022).
- (*b*) Federated learning with personalized awareness: The goal of this method is to achieve improved inference or prediction accuracy for the entire population while allowing heterogeneity to exist to some extent. Related studies include those by Li et al. (2020a) and Fallah et al. (2020a).
- (*c*) Fairness: In this category, researchers have tried to minimize the heterogeneity caused by specific variables, such as gender, race, and site differences. Related studies include those by Mohri et al. (2019) and Li et al. (2020b).
- (*d*) Robustness: The goal of this method is to obtain predictions that are robust to heterogeneity. Related studies include that of Reisizadeh et al. (2020).

4. Privacy: Many existing distributed methods require communication gradient or surrogate statistics only. Although free of individual data, individual information can still be recovered through several techniques, such as data poisoning attacks (Chen et al. 2017), model poisoning attacks (Bagdasaryan et al. 2020), Byzantine faults (Castro & Liskov 1999), and membership inference attacks (Shokri et al. 2017). Recently, several differential methods have been used to overcome privacy issues, including homomorphic encryption (Phong et al. 2018), secure multiparty computation (Bonawitz et al. 2017), and differential privacy (Geyer et al. 2017).

We summarize some important existing works in **Table 1** according to their statistical accuracy, their local data size conditions, their required communication rounds (time), the size of their transferred statistics (space), their tolerance to heterogeneity, and their targets. Furthermore, **Table 2** summarizes the existing methods for handling different types of heterogeneity.

4. THEORETICAL PROPERTIES

The existing theoretical results for DL focus on the equivalence between distributed estimators and centralized estimators, the dependence between the number of communication rounds and the convergence rate, the impact of heterogeneity, and privacy protection strategies. We now describe these properties in detail.

4.1. Equivalence

The equivalence between distributed estimators and centralized estimators can be summarized into three classes, which are as follows: (*a*) In first-order equivalence, a distributed estimator is equal to a centralized estimator when their error bounds have the same rate:

$$\mathrm{E}\|\hat{\boldsymbol{\theta}}_{DL} - \boldsymbol{\theta}^*\|_2^2 = O\left(\frac{1}{N}\right), \quad \text{and} \quad \mathrm{E}\|\hat{\boldsymbol{\theta}}_{\mathrm{cen}} - \boldsymbol{\theta}^*\|_2^2 = O\left(\frac{1}{N}\right),$$

where N is the size of the whole sample (Liu & Ihler 2014, Rosenblatt & Nadler 2016, Fan et al. 2021, Chen et al. 2022). (*b*) In the case of same asymptotic efficiency, the asymptotic distribution equivalence between a distributed estimator and a centralized estimator facilitates statistical inference (Chen & Peng 2021, Chang et al. 2023). Thus,

$$\sqrt{N}(\hat{\boldsymbol{\theta}}_{DL} - \boldsymbol{\theta}^*) \to \mathcal{N}(0, \Sigma), \quad \text{and} \quad \sqrt{N}(\hat{\boldsymbol{\theta}}_{\mathrm{cen}} - \boldsymbol{\theta}^*) \to \mathcal{N}(0, \Sigma).$$

Additionally, when heterogeneity exists to some extent, a distributed estimator with proper weighting methods generates more efficient estimators than a centralized estimator that does not account

Table 1 Comparison among the existing works in the field of distributed learning

Studies	Conditions			Computational Efficiency		Statistical Accuracy
	Target	Size	Heterogeneity[a]	$\mathbb{C}(time)$	$\mathbb{C}(space)$	
Battey et al. (2018)	GLM	$K = \mathcal{O}(\sqrt{N/(s^2 \log p)})$	NA	$\mathcal{O}(1)$	$\mathcal{O}(p)$	1st equiv.
Chang et al. (2023)	NA	n can be finite	NA	$\mathcal{O}(1)$	$\mathcal{O}(p+M)$	SAE
Chen & Peng (2021)	Symmetric	$K = o(N^{1-1/(2C)})$	NA	$\mathcal{O}(1)$	$\mathcal{O}(p)$	SAE
Chen & Xie (2014)	GLM	$\log(Kp) = o(n)$	NA	$\mathcal{O}(1)$	$\mathcal{O}(p^2)$	1st equiv.
Fan et al. (2021)	S, SC	$n \geq Cp$	δ-BHD	$\mathcal{O}\big(\log(\frac{N}{p})/\log(\frac{1}{\eta})\big)$, $\eta = \kappa^2(\log N)p/n$	$\mathcal{O}(p)$	1st equiv.
Gupta et al. (2021)	S, SC	NA	NA	$\mathcal{O}\big(\sqrt{n}\sqrt{\frac{1}{\varepsilon}}\big)$	$\mathcal{O}(p)$	NA
Jordan et al. (2019)	S	NA	NA	$\mathcal{O}\big(\frac{\log N}{\log n}\big)$	$\mathcal{O}(p)$	1st equiv.
Karimireddy et al. (2020)	S, SC	NA	(G, B)-BGD	$\mathcal{O}\big(\frac{\sigma^2}{\mu KE\varepsilon} + \frac{\sqrt{LG}}{\mu\sqrt{\varepsilon}} + \frac{B^2 L}{\mu}\big)$	$\mathcal{O}(p)$	NA
Karimireddy et al. (2020)	S, SC	NA	δ-BHD	$\mathcal{O}\big(\frac{\sigma^2}{\mu KE\varepsilon} + \frac{L}{\mu}\big)$	$\mathcal{O}(p)$	NA
Lee et al. (2017a)	S, SC	NA	$(G, 0)$-BGD	$\mathcal{O}\big(\frac{LK}{\mu n}\log\frac{1}{\varepsilon}\big)$	$\mathcal{O}(p)$	NA
Lee et al. (2017b)	Smooth spline	$n \gtrsim Ks^2 \log p$	NA	$\mathcal{O}(1)$	$\mathcal{O}(p)$	1st equiv.
Li et al. (2020a)	S, SC	NA	$(0, B)$-BGD	$\mathcal{O}\big(\frac{B^2}{\mu}\big)$	$\mathcal{O}(p)$	NA
Lian & Fan (2018)	SVM	$K \leq \mathcal{O}((N/\log p)^{1/3})$	NA	$\mathcal{O}(1)$	$\mathcal{O}(p)$	1st equiv.
Liu & Ihler (2014)	GLM	NA	NA	$\mathcal{O}(1)$	$\mathcal{O}(p)$	1st equiv.
Lv & Lian (2017)	PLM	$K \leq \sqrt{N/\log p}$	NA	$\mathcal{O}(1)$	$\mathcal{O}(p)$	1st equiv.
Shamir et al. (2014)	QR	NA	NA	$\mathcal{O}\big(\frac{L^2K}{\mu^2 N}\log\frac{1}{\varepsilon}\big)$	$\mathcal{O}(p)$	NA
Shang & Cheng (2017)	S	$K = \mathcal{O}(N^{(4L-1)/(4L+1)})$	NA	$\mathcal{O}(1)$	NA	1st equiv.
Smith et al. (2017)	S + SC/NC	NA	NA	$\mathcal{O}\big(\frac{L+\mu}{\mu}\log\frac{1}{\varepsilon}\big)$	$\mathcal{O}(p)$	NA
Song et al. (2015)	LM	NA	NA	$\mathcal{O}(1)$	NA	SE
Tang et al. (2020)	GLM	$N \gg p$	NA	$\mathcal{O}(1)$	NA	1st equiv.
Wang et al. (2017)	S, SC + l_1	$n \gtrsim s^2 \log p$	NA	$\mathcal{O}(\log K)$	$\mathcal{O}(p)$	1st equiv.
Wang et al. (2018)	QR (l_2)	NA	NA	$\mathcal{O}\big(\frac{\log(\kappa/\varepsilon)}{\log(n/K)}\big)$	$\mathcal{O}(p)$	NA
Wang et al. (2019)	SVM	NA	NA	$\log_2\big(\frac{\log N - \log p}{\log K - \log N}\big)$	$\mathcal{O}(p)$	1st equiv.
Zhang & Lin (2015)	S, SC	NA	NA	$\mathcal{O}\big(\sqrt{1+2\mu}\log\big(\frac{1}{\varepsilon}\big)\big)$	$\mathcal{O}(p)$	1st equiv.
Zhou et al. (2022)	NA	$K = \mathcal{O}(n^{1/2-\delta})$	NA	NA	$\mathcal{O}(p)$	SE

[a](G, B)-BGD indicates that there exist constants $G \geq 0$ and $B \geq 1$ such that $\forall \boldsymbol{\theta}$, $\frac{1}{K}\sum_{k=1}^{K} \|\nabla\phi(\boldsymbol{\theta})\|^2 \leq G^2 + B^2\|\nabla\phi(\boldsymbol{\theta})\|^2$. δ-BHD indicates that there exist constants $\delta > 0$ such that $\forall \boldsymbol{\theta}$, $\|\nabla^2\phi(\boldsymbol{\theta}) - \nabla^2\phi(\boldsymbol{\theta})\| \leq \delta$.

Other notation is as follows: C, a positive constant; E, the number of local update steps; K, the number of sites; L, the smoothness of true function f_0; N, the data size of the entire sites; n, the data size at each local site; p, the dimensionality of the parameters; s, the sparsity of the parameters; κ, the condition number of true function f_0; μ, the convexity of true function f_0; σ^2, the variance bounds of the stochastic gradient of local loss function ϕ_k; ε, the required optimization accuracy.

Abbreviations: 1st equiv., first-order equivalence; GLM, generalized linear model; l_1 and l_2, penalty terms; LM, linear model; NA, not applicable/not available; NC, nonconvex; NS, nonsmooth; PLM, partial linear model; QR, quadratic; S, smooth; SAE, same asymptotic efficiency; SC, strongly convex; SE, strong equivalence; SVM, support vector machine.

Table 2 Comparison among the existing methods for handling different types of heterogeneity in distributed learning

Scenario	Objective[a]	Algorithm and study[b]	Method
HDL-XY	*a*	pFedMe (Dinh et al. 2020)	Regularization
HDL-XY	*a*	FL+HC (Briggs et al. 2020)	Clustering
		IFCA (Ghosh et al. 2022)	
		FedSEM (Long et al. 2023)	
HDL-XY	*b*	FedProx (Li et al. 2020a)	Regularization
		FedCL (Yao & Sun 2020)	
		MOON (Li et al. 2021)	
HDL-XY	*b*	ARUBA (Khodak et al. 2019)	Meta-learning
		Per-FedAvg (Fallah et al. 2020a,b)	
HDL-XY	*c*	Agnostic FL (Mohri et al. 2019)	Weighted loss
		q-FedAvg (Li et al. 2020b)	
HDL-XY	*d*	FedRobust (Reisizadeh et al. 2020)	Perturbation
HDL-Y	*a*	FedCurv (Shoham et al. 2019)	Regularization
		MOCHA (Smith et al. 2017)	
		VIRTUAL (Corinzia et al. 2021)	
		FedAMP (Huang et al. 2021)	
VDL	*a*	FTL (Liu et al. 2020)	Regularization

[a]Objectives, corresponding to those listed in Section 3, are as follows: (*a*) target (personalized) learning, (*b*) federated learning with personalized awareness, (*c*) fairness, and (*d*) robustness.
[b]Abbreviations in algorithm names: Agnostic FL, agnostic federated learning; ARUBA, average regret-upper-bound analysis; FedAMP, federated attentive message passing; FedCL, federated learning with continual local training; FedCurv, federated curvature; FedProx, federated learning with proximal term; FedRobust, federated learning framework robust to affine distribution shifts; FedSEM, federated stochastic expectation maximization; FL + HC, federated learning with hierarchical clustering; FTL, federated transfer learning; IFCA, iterative federated clustering algorithm; MOCHA, framework for federated multi-task learning; MOON, model-contrastive learning; Per-FedAvg, personalized federated averaging; pFedMe, personalized federated learning with Moreau envelopes; q-FedAvg, q-federated averaging; VIRTUAL, variational federated multi task learning.
Other abbreviations: DL, distributed learning; HDL-XY, horizontal DL with varied covariates and outcomes distributions; HDL-Y, horizontal DL with varying conditional outcome distributions given covariates; VDL, vertical DL.

for such heterogeneity (Zhou et al. 2022). Thus, $\mathrm{Var}(\hat{\boldsymbol{\theta}}_{\mathrm{cen}}) - \mathrm{Var}(\hat{\boldsymbol{\theta}}_{DL})$ is semipositive definite. (*c*) In the case of strong equivalence, if a distributed estimator almost surely equals a centralized estimator, $\mathrm{P}(\hat{\boldsymbol{\theta}}_{DL} = \hat{\boldsymbol{\theta}}_{\mathrm{cen}}) = 1$ (Song et al. 2015, Zhou et al. 2022). More related work is reported in **Table 1**.

4.2. Dependence

To relax the strict conditions regarding the number of sites and the data size at each local site while achieving the same convergence rate as that of centralized estimators, iterative methods have been developed. The type of statistics being transferred determines the number of communications. To achieve the same first-order equivalence as that of the centralized estimator, many studies have listed the number of iterations and the statistics that must be transferred. Huang & Huo (2019) proposed a one-step update method $\hat{\boldsymbol{\theta}}^{(1)}$ via the use of the gradient and Hessian matrix of the global empirical criterion function, which achieves a lower upper bound on the mean squared error than does the simple averaging-based one-shot estimator $\hat{\boldsymbol{\theta}}_{\mathrm{ave}}$:

$$\mathrm{E}[\|\hat{\boldsymbol{\theta}}^{(1)} - \boldsymbol{\theta}^*\|_2^2] \leq \frac{C_1}{N} + O(N^{-2}) + O(K^4 N^{-4}).$$

Jordan et al. (2019) presented a CSL framework and noted that with $N^{-1/2}$-consistent initial estimators, at most $\lceil \log K / \log N \rceil$ iterations are required to obtain the same asymptotic distribution as that of the centralized estimator. Fan et al. (2021) proposed the communication-efficient accurate statistical estimation (CEASE) algorithm and showed that the proposed algorithm achieves statistical efficiency in $\mathcal{O}\left(\log\left(\frac{N}{p}\right)/\log\left(\frac{1}{\eta}\right)\right)$ iterations. Other related works are also summarized in **Table 1**.

4.3. Impact of Heterogeneity

The bias–variance tradeoff is particularly crucial to the success of DL methods when various levels of heterogeneity are present. Incorrectly borrowing information from other sites with large heterogeneity leads to unreliable inferences and/or low prediction power, which is even worse than that attained when using local sites only (Yu et al. 2022). In addition, utilizing DL without accounting for such heterogeneity can cause poor algorithmic convergence (Karimireddy et al. 2020). Existing methods that correctly identify similar sites include clustering (Briggs et al. 2020, Ghosh et al. 2022, Long et al. 2023) and regularization methods (Li et al. 2020a, Dinh et al. 2020, Yao & Sun 2020). Situations with different covariates (i.e., vertical DL), the same covariates with different distributions (i.e., horizontal DL with varied covariate and outcome distributions), and the same marginal covariate distribution but with different conditional outcome distributions given covariates (i.e., horizontal DL with varying conditional distributions) are three types of heterogeneous scenarios that are commonly considered. **Table 2** summarizes these and other studies that consider different heterogeneity types, targets and methods.

4.4. Impact of Privacy Protection

With regard to privacy protection, more iterations are required for homomorphic encryption and secure multiparty computations, thereby reducing communication efficiency (Bonawitz et al. 2017, Phong et al. 2018). Conversely, differential privacy strategies may affect the equivalence of distributed methods to centralized methods, although the careful selection of differential strategies may markedly reduce the impact of this choice (Dwork et al. 2006, Dwork & Roth 2013, Dong et al. 2022).

5. IMPLEMENTATIONS

DL research has undergone rapid progress due to the availability of powerful distributed computing frameworks. In **Table 3**, we present some widely used traditional distributed computing frameworks, such as MapReduce (Dean & Ghemawat 2008), Spark (Zaharia et al. 2012), Flink, Storm (Toshniwal et al. 2014), and Samza (Noghabi et al. 2017), and describe some recently developed platforms for federated learning research, such as TensorFlow, PySyft (Ryffel et al. 2018), FedML (federated machine learning) (He et al. 2020), FATE (federated AI technology enabler), and PaddleFL (paddle federated learning) (Ma et al. 2019). We provide a concise comparison of these frameworks to facilitate reader comprehension. For further details, please refer to the works cited.

MapReduce, Flink, Spark, Storm, and Samza do not directly offer data privacy protection capabilities. To achieve this goal, these methods can be combined with other technologies or frameworks. For example, we can use Spark or MapReduce to conduct differential privacy analysis using the Privacy on Beam module of Apache Beam. We can also perform encrypted computations and secure multiparty computations on Samza or Apache Flink using the PrivacyGuard module. Additionally, specific solutions, such as StreamShield (Nehme et al. 2009), have been designed to preserve the privacy of data streams.

Table 3 Summary of diverse distributed and federated computing frameworks

Framework	Data type	Supported language	Privacy protection	Built-in datasets	Built-in algorithms
MapReduce	B	C++, Java, Groovy, Perl, Ruby	✗	✗	✗
Flink	B, S	Java, Scala	✗	✗	✓
Spark	B, S	Java, Python, R, Scala	✗	✗	✓
Storm	S	JavaScript, Perl, Python, Ruby	✗	✗	✗
TensorFlow	B, S	C++, Java, Python	DP, HE, SMC	✓	✓
Samza	S	Java, JVM, Scala	✗	✗	✗
PySyft	B	Python	DP, HE, SMC	✓	✓
FATE	B	Python, Java	DP, HE, SMC	✓	✓
PaddleFL	B	C++, Python	DP, HE, SMC	✓	✓
FedML	B	Java, Python, Swift	DP, HE, SMC	✓	✓

Frameworks are listed in chronological order of introduction. Abbreviations: B, batch; DP, differential privacy; FATE, federated AI technology enabler; FL, federated learning; HE, homomorphic encryption; ML, machine learning; S, stream; SMC, secure multiparty computation.

Benchmark datasets are used to evaluate and compare the performance of machine learning algorithms on specific tasks or problems. They provide a common platform for researchers to test their methods and measure their accuracy, speed, robustness, etc., thus markedly advancing machine learning research. However, high-quality public datasets are not common, particularly in the field of DL. Fortunately, with the development of related fields, researchers have constructed popular and recognized high-quality benchmark datasets; some commonly used datasets include Federated EMNIST (extended MNIST), Federated CIFAR-10, Shakespeare, and Federated Reddit. Most existing federated frameworks provide some benchmark datasets for researchers to use; for example, LEAF (Caldas et al. 2018) provides various datasets, including federated EMNIST, Shakespeare, CelebA, and Sentiment140. FLamby (federated learning ample benchmark of your cross-silo strategies) (du Terrail et al. 2022) includes some federated cross-silo healthcare datasets, such as Fed-Camelyon16 (Cancer Metastases in Lymph Nodes Challenge 2016), Fed-KITS2019 (Kidney Tumor Segmentation Challenge 2019), Fed-ISIC2019 (International Skin Imaging Collaboration 2019), and Fed-Heart-Disease. FedGraphNN (federated graph neural networks) (He et al. 2021) provides datasets for federated graph neural network research, such as social networks, citation networks, and knowledge graphs.

6. FUTURE DIRECTIONS

DL is attracting increasing attention, and although many successes have been achieved from both theoretical and numerical perspectives, numerous interesting topics still require further study.

6.1. Robustness

Most existing studies have focused on equivalence, statistical efficiency, and communication efficiency. As distributed healthcare platforms, insurance provider networks, and other quality-of-life–related distributed data platforms increase in number and size, robustness to heterogeneity and contaminated data has attracted increasing interest. Zhou & Song (2017) considered the robustness of their proposed distributed Rao-CD method against contaminated data or heterogeneously correlated structures. Their numerical results showed that when the sample size is moderate, general distributed methods can still generate reasonable inference results. However,

when the sample size increases and more outliers are introduced to the data, distributed methods that are not robust to outliers suffer severe estimation biases due to these outliers. Thus, developing novel methods to address robustness is urgent and necessary. Recently, various techniques have been explored to address robustness in DL and provide inspiration for future studies. Related work includes the use of adversarial training (Reisizadeh et al. 2020), which employs a minimax optimization method to improve robustness against distribution shifts across local data sites, and quantile regression (Chen et al. 2020), which uses a nonsmooth quantile loss function to handle heavy-tailed noise.

6.2. Fairness

Researchers are paying more attention to the issue of fairness in DL as fair contribution-reward mechanisms encourage greater client participation in the real world (Li et al. 2023). When faced with heterogeneous scenarios, a training process using the sample size averaging–based scheme results in unfair resource allocation, and the generalization performance achieved by the aggregated model for all clients exhibits inconsistency with their corresponding contributions to the model. For instance, a few clients may dominate the training process and skew the results with disproportionately large samples. To address these learning fairness problems, most recent research has focused on using the performance achieved by a model for individual clients throughout the training process to adjust the weight of the transmitted model or parameters, hence ensuring balanced performance on local clients. Some interesting work in this area includes that of Li et al. (2020b), Lyu et al. (2020), and Zhang et al. (2020), but these methods come at the price of additional communication costs and information leakage risks. Thus, further exploration is needed for fairness considerations in DL.

6.3. Network and Survival Data

Many studies have used cross-sectional data and/or time series data. Embraced by the rapid development of computing software and online communications, distributed data with network structures and survival data in distributed platforms have emerged. However, the existing DL methods cannot be directly extended due to the non-Euclidean property of the network structure and the nonseparability of the objectives for survival data. Novel DL methods are required to address network and survival data.

6.4. Postregularization Inference

Statistical inference is also important, particularly for human life–related data. To appropriately integrate distributed data information, correctly identifying similar data sites is crucial. Regularization-based techniques are some of the most popular methods for automatically identifying homogeneous data sites. However, properly addressing the randomness caused by regularization is challenging and markedly increases the complexity of statistical inference. A few studies have conducted postselection inference (van de Geer et al. 2014, Zhang & Zhang 2014), but research in this area remains lacking for general postregularization inference; this topic is particularly interesting for future research, especially with a distributed framework.

6.5. Causal Distributed Learning

Causal inference, which is the process of drawing a conclusion about a causal connection based on the conditions of the occurrence of an effect (Yao et al. 2021), is a critical research topic. Recently, Tan et al. (2022) provided an interpretable tree-based ensemble of conditional average treatment effect estimators, which joined heterogeneous models across different sites. Determining causal

inference under a distributed framework is an appealing research direction due to the causal interpretability and large amount of available data.

DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

ACKNOWLEDGMENTS

The authors' research was partially supported by the National Key R&D Program of China (No. 2022YFA1003702) and National Natural Science Foundation of China (No. 11931014).

LITERATURE CITED

Arjevani Y, Shamir O. 2015. Communication complexity of distributed convex learning and optimization. In *NIPS'15: Proceedings of the 28th International Conference on Neural Information Processing Systems*, Vol. 1, ed. C Cortes, DD Lee, M Sugiyama, R Garnett, pp. 1756–64. Cambridge, MA: MIT Press

Bagdasaryan E, Veit A, Hua Y, Estrin D, Shmatikov V. 2020. How to backdoor federated learning. *Proc. Mach. Learn. Res.* 108:2938–48

Banerjee M, Durot C, Sen B. 2019. Divide and conquer in nonstandard problems and the super-efficiency phenomenon. *Ann. Stat.* 47(2):720–57

Battey H, Fan J, Liu H, Lu J, Zhu Z. 2018. Distributed testing and estimation under sparse high dimensional models. *Ann. Stat.* 46(3):1352–82

Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan HB, et al. 2017. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–91. New York: ACM

Borenstein M, Hedges LV, Higgins JP, Rothstein HR. 2021. *Introduction to Meta-Analysis*. New York: John Wiley & Sons

Boyd S. 2010. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends® Mach. Learn.* 3(1):1–122

Briggs C, Fan Z, Andras P. 2020. Federated learning with hierarchical clustering of local updates to improve training on non-IID data. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9. Piscataway, NJ: IEEE

Cai TT, Wei H. 2022. Distributed nonparametric function estimation: optimal rate of convergence and cost of adaptation. *Ann. Stat.* 50(2):698–725

Caldas S, Duddu SMK, Wu P, Li T, Konečnỳ J, et al. 2018. LEAF: a benchmark for federated settings. arXiv:1812.01097 [cs.LG]

Castro M, Liskov B. 1999. Practical Byzantine fault tolerance. In *3rd Symposium on Operating Systems Design and Implementation (OSDI 99)*, pp. 173–86. Berkeley, CA: USENIX

Chang C, Bu Z, Long Q. 2023. CEDAR: communication efficient distributed analysis for regressions. *Biometrics* 79(3):2357–69

Chen SX, Peng L. 2021. Distributed statistical inference for massive data. *Ann. Stat.* 49(5):2851–69

Chen X, Liu C, Li B, Lu K, Song D. 2017. Targeted backdoor attacks on deep learning systems using data poisoning. arXiv:1712.05526 [cs.CR]

Chen X, Liu W, Mao X, Yang Z. 2020. Distributed high-dimensional regression under a quantile loss function. *J. Mach. Learn. Res.* 21(182):1–43

Chen X, Liu W, Zhang Y. 2022. First-order Newton-type estimator for distributed estimation and inference. *J. Am. Stat. Assoc.* 117(540):1858–74

Chen X, Xie M. 2014. A split-and-conquer approach for analysis of extraordinarily large data. *Stat. Sin.* 24:1655–84

Corinzia L, Beuret A, Buhmann JM. 2021. Variational federated multi-task learning. arXiv:1906.06268 [cs.LG]

Dean J, Ghemawat S. 2008. MapReduce: simplified data processing on large clusters. *Commun. ACM* 51(1):107–13

Dinh CT, Tran NH, Nguyen TD. 2020. Personalized federated learning with Moreau envelopes. In *NIPS'20: Proceedings of the 34th International Conference on Neural Information Processing Systems*, ed. H Larochelle, M Ranzato, R Hadsell, MF Balcan, H Lin, pp. 21394–405. Red Hook, NY: Curran

Dittrich J, Richter S, Schuh S. 2013. Efficient OR Hadoop: Why not both? *Datenbank-Spektrum* 13(1):17–22

Dong J, Roth A, Su WJ. 2022. Gaussian differential privacy. *J. R. Stat. Soc. Ser. B* 84(1):3–37

du Terrail JO, Ayed SS, Cyffers E, Grimberg F, He C, et al. 2022. FLamby: datasets and benchmarks for cross-silo federated learning in realistic healthcare settings. arXiv:2210.04620 [cs.LG]

Dwork C, McSherry F, Nissim K, Smith A. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4–7, 2006*, pp. 265–84. New York: Springer

Dwork C, Roth A. 2013. The algorithmic foundations of differential privacy. *Found. Trends® Theor. Comput. Sci.* 9(3–4):211–407

Fallah A, Mokhtari A, Ozdaglar A. 2020a. On the convergence theory of gradient-based model-agnostic meta-learning algorithms. *Proc. Mach. Learn. Res.* 108:1082–92

Fallah A, Mokhtari A, Ozdaglar A. 2020b. Personalized federated learning with theoretical guarantees: a model-agnostic meta-learning approach. In *NIPS'20: Proceedings of the 34th International Conference on Neural Information Processing Systems*, ed. H Larochelle, M Ranzato, R Hadsell, MF Balcan, H Lin, pp. 3557–68. Red Hook, NY: Curran

Fan J, Guo Y, Wang K. 2021. Communication-efficient accurate statistical estimation. *J. Am. Stat. Assoc.* 118(543):1000–10

Fan J, Wang D, Wang K, Zhu Z. 2019. Distributed estimation of principal eigenspaces. *Ann. Stat.* 47(6):3009–31

Fernando N, Loke SW, Rahayu W. 2013. Mobile cloud computing: a survey. *Future Gener. Comput. Syst.* 29(1):84–106

Forero PA, Cano A, Giannakis GB. 2010. Consensus-based distributed support vector machines. *J. Mach. Learn. Res.* 11(55):1663–707

Gao Y, Liu W, Wang H, Wang X, Yan Y, Zhang R. 2022. A review of distributed statistical inference. *Stat. Theory Relat. Fields* 6(2):89–99

Geyer RC, Klein T, Nabi M. 2017. Differentially private federated learning: a client level perspective. arXiv:1712.07557 [cs.CR]

Ghosh A, Chung J, Yin D, Ramchandran K. 2022. An efficient framework for clustered federated learning. *IEEE Trans. Inf. Theory* 68(12):8076–91

Gupta V, Ghosh A, Derezinski M, Khanna R, Ramchandran K, Mahoney M. 2021. LocalNewton: reducing communication bottleneck for distributed learning. arXiv:2105.07320 [cs.DC]

Hammoud M, Rehman MS, Sakr MF. 2012. Center-of-gravity reduce task scheduling to lower MapReduce network traffic. In *2012 IEEE Fifth International Conference on Cloud Computing*, pp. 49–58. Piscataway, NJ: IEEE

He C, Balasubramanian K, Ceyani E, Yang C, Xie H, et al. 2021. FedGraphNN: a federated learning system and benchmark for graph neural networks. arXiv:2104.07145 [cs.LG]

He C, Li S, So J, Zeng X, Zhang M, et al. 2020. FedML: a research library and benchmark for federated machine learning. arXiv:2007.13518 [cs.LG]

Horowitz E, Zorat A. 1983. Divide-and-conquer for parallel processing. *IEEE Trans. Comput.* C-32(6):582–85

Huang C, Huo X. 2019. A distributed one-step estimator. *Math. Program.* 174:41–76

Huang Y, Chu L, Zhou Z, Wang L, Liu J, et al. 2021. Personalized cross-silo federated learning on non-IID data. *Proc. AAAI Conf. Artif. Intell.* 35(9):7865–73

Jiao S, He C, Dou Y, Tang H. 2012. Molecular dynamics simulation: implementation and optimization based on Hadoop. In *2012 8th International Conference on Natural Computation*, pp. 1203–7. Piscataway, NJ: IEEE

Jordan MI, Lee JD, Yang Y. 2019. Communication-efficient distributed statistical inference. *J. Am. Stat. Assoc.* 114(526):668–81

Karimireddy SP, Kale S, Mohri M, Reddi S, Stich S, Suresh AT. 2020. SCAFFOLD: stochastic controlled averaging for federated learning. *Proc. Mach. Learn. Res.* 119:5132–43

Khodak M, Balcan MFF, Talwalkar AS. 2019. Adaptive gradient-based meta-learning methods. In *33rd Conference on Neural Information Processing Systems (NeurIPS2019)*, ed. H Wallach, H Larochelle, A Beygelzimer, F d'Alché-Buc, E Fox, R Garnett, pp. 5917–28. Red Hook, NY: Curran

Kleiner A, Talwalkar A, Sarkar P, Jordan MI. 2014. A scalable bootstrap for massive data. *J. R. Stat. Soc. Ser. B* 76(4):795–816

Lee JD, Lin Q, Ma T, Yang T. 2017a. Distributed stochastic variance reduced gradient methods by sampling extra data with replacement. *J. Mach. Learn. Res.* 18(122):1–43

Lee JD, Liu Q, Sun Y, Taylor JE. 2017b. Communication-efficient sparse regression. *J. Mach. Learn. Res.* 18(1):115–44

Li Q, He B, Song D. 2021. Model-contrastive federated learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10708–17. Piscataway, NJ: IEEE

Li Q, Wen Z, Wu Z, Hu S, Wang N, et al. 2023. A survey on federated learning systems: vision, hype and reality for data privacy and protection. *IEEE Trans. Knowledge Data Eng.* 35:3347–66

Li T, Sahu AK, Zaheer M, Sanjabi M, Talwalkar A, Smith V. 2020a. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems 2 (MLSys 2020)*, ed. I Dhillon, D Papailiopoulos, V Sze, pp. 429–50. Indio, CA: Syst. Mach. Learn. Found.

Li T, Sanjabi M, Beirami A, Smith V. 2020b. Fair resource allocation in federated learning. In *Proceedings of the Eighth International Conference on Learning Representations*. N.p.: ICLR

Lian H, Fan Z. 2018. Divide-and-conquer for debiased l_1-norm support vector machine in ultra-high dimensions. *J. Mach. Learn. Res.* 18(182):1–26

Lin DY, Zeng D. 2010. On the relative efficiency of using summary statistics versus individual-level data in meta-analysis. *Biometrika* 97(2):321–32

Lin N, Xi R. 2011. Aggregated estimating equation estimation. *Stat. Interface* 4(1):73–83

Liu D, Liu RY, Xie M. 2015. Multivariate meta-analysis of heterogeneous studies using only summary statistics: efficiency and robustness. *J. Am. Stat. Assoc.* 110(509):326–40

Liu Q, Ihler AT. 2014. Distributed estimation, information loss and exponential families. In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, ed. Z Ghahramani, M Welling, C Cortes, N Lawrence, K Weinberger, pp. 1098–106. Red Hook, NY: Curran

Liu Y, Kang Y, Xing C, Chen T, Yang Q. 2020. A secure federated transfer learning framework. *IEEE Intell. Syst.* 35(4):70–82

Long G, Xie M, Shen T, Zhou T, Wang X, et al. 2023. Multi-center federated learning: clients clustering for better personalization. *World Wide Web* 26(1):481–500

Lv S, Lian H. 2017. Debiased distributed learning for sparse partial linear models in high dimensions. arXiv:1708.05487 [stat.ML]

Lyu L, Xu X, Wang Q. 2020. Collaborative fairness in federated learning. arXiv:2008.12161 [cs.LG]

Ma Y, Yu D, Wang H. 2019. PaddlePaddle: an open-source deep learning platform from industrial practice. *J. Front. Comput. Sci. Technol.* 13(1):11–23

McMahan B, Moore E, Ramage D, Hampson S, Aguera y Arcas B. 2017. Communication-efficient learning of deep networks from decentralized data. *Proc. Mach. Learn. Res.* 54:1273–82

Menabrea L. 1843. *Sketch of the Analytical Engine Invented by Charles Babbage.*, transl. A Lovelace. London: R. & J.E. Taylor

Minsker S. 2019. Distributed statistical estimation and rates of convergence in normal approximation. *Electron. J. Stat.* 13(2):5213–52

Mohri M, Sivek G, Suresh AT. 2019. Agnostic federated learning. *Proc. Mach. Learn. Res.* 97:4615–25

Nedic A, Ozdaglar A. 2009. Distributed subgradient methods for multi-agent optimization. *IEEE Trans. Autom. Control* 54(1):48–61

Nehme RV, Lim HS, Bertino E, Rundensteiner EA. 2009. StreamShield: a stream-centric approach towards security and privacy in data stream environments. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data, SIGMOD '09*, pp. 1027–30. New York: ACM

Newman D, Asuncion A, Smyth P, Welling M. 2009. Distributed algorithms for topic models. *J. Mach. Learn. Res.* 10(62):1801–28

Noghabi SA, Paramasivam K, Pan Y, Ramesh N, Bringhurst J, et al. 2017. Samza: stateful scalable stream processing at LinkedIn. In *Proceedings of the VLDB Endowment*, Vol. 10, ed. P Boncz, K Salem, pp. 1634–45. N.p.: VLDB Endow.

Passino K. 2002. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Syst. Mag.* 22(3):52–67

Phong LT, Aono Y, Hayashi T, Wang L, Moriai S. 2018. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Trans. Inform. Forensics Secur.* 13(5):1333–45

Reisizadeh A, Farnia F, Pedarsani R, Jadbabaie A. 2020. Robust federated learning: the case of affine distribution shifts. In *NIPS'20: Proceedings of the 34th International Conference on Neural Information Processing Systems*, ed. H Larochelle, M Ranzato, R Hadsell, M Balcan, H Lin, pp. 21554–65. Red Hook, NY: Curran

Rosenblatt JD, Nadler B. 2016. On the optimality of averaging in distributed statistical learning. *Inform. Inference J. IMA* 5(4):379–404

Ryffel T, Trask A, Dahl M, Wagner B, Mancuso J, et al. 2018. A generic framework for privacy preserving deep learning. arXiv:1811.04017 [cs.LG]

Shamir O, Srebro N, Zhang T. 2014. Communication-efficient distributed optimization using an approximate Newton-type method. *Proc. Mach. Learn. Res.* 32:1000–8

Shang Z, Cheng G. 2017. Computational limits of a distributed algorithm for smoothing spline. *J. Mach. Learn. Res.* 18(108):1–37

Shen J, Liu RY, Xie M. 2020. iFusion: individualized fusion learning. *J. Am. Stat. Assoc.* 115(531):1251–67

Shi C, Lu W, Song R. 2018. A massive data framework for M-estimators with cubic-rate. *J. Am. Stat. Assoc.* 113(524):1698–709

Shoham N, Avidor T, Keren A, Israel N, Benditkis D, et al. 2019. Overcoming forgetting in federated learning on non-IID data. arXiv:1910.07796 [cs.LG]

Shokri R, Stronati M, Song C, Shmatikov V. 2017. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18. Piscataway, NJ: IEEE

Singh K, Xie M, Strawderman WE. 2005. Combining information from independent sources through confidence distributions. *Ann. Stat.* 33:159–83

Smith V, Chiang CK, Sanjabi M, Talwalkar A. 2017. Federated multi-task learning. In *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, ed. U von Luxburg, I Guyon, S Bengio, H Wallach, R Fergus, pp. 4427–37. Red Hook, NY: Curran

Song Q, Liang F, et al. 2015. A split-and-merge Bayesian variable selection approach for ultrahigh dimensional regression. *J. R. Stat. Soc. Ser. B* 77(5):947–72

Szabó B, Van Zanten H, et al. 2019. An asymptotic analysis of distributed nonparametric methods. *J. Mach. Learn. Res.* 20(87):1–30

Tan X, Chang CCH, Zhou L, Tang L. 2022. A tree-based model averaging approach for personalized treatment effect estimation from heterogeneous data sources. *Proc. Mach. Learn. Res.* 162:21013–36

Tang L, Zhou L, Song PXK. 2020. Distributed simultaneous inference in generalized linear models via confidence distribution. *J. Multivariate Anal.* 176:104567

Taylor RC. 2010. An overview of the Hadoop/MapReduce/HBase framework and its current applications in bioinformatics. *BMC Bioinformatics* 11(S12):S1

Toh S, Rasmussen-Torvik LJ, Harmata EE, Pardee R, Saizan R, et al. 2017. The national Patient-Centered Clinical Research Network (PCORnet) bariatric study cohort: rationale, methods, and baseline characteristics. *JMIR Res. Protoc.* 6(12):e8323

Toshniwal A, Taneja S, Shukla A, Ramasamy K, Patel JM, et al. 2014. Storm@twitter. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, SIGMOD '14*, pp. 147–56. New York: ACM

van de Geer S, Bühlmann P, Ritov Y, Dezeure R. 2014. On asymptotically optimal confidence regions and tests for high-dimensional models. *Ann. Stat.* 42(3):1166–202

Volgushev S, Chao SK, Cheng G. 2019. Distributed inference for quantile regression processes. *Ann. Stat.* 47(3):1634–62

Wang J, Kolar M, Srebro N, Zhang T. 2017. Efficient distributed learning with sparsity. *Proc. Mach. Learn. Res.* 70:3636–45

Wang S, Roosta-Khorasani F, Xu P, Mahoney MW. 2018. GIANT: globally improved approximate Newton method for distributed optimization. In *NIPS'18: Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ed. S Bengio, HM Wallach, H Larochelle, K Grauman, N Cesa-Bianchi, pp. 2338–48. Red Hook, NY: Curran

Wang X, Yang Z, Chen X, Liu W. 2019. Distributed inference for linear support vector machine. *J. Mach. Learn. Res.* 20(113):1–41

White T. 2015. *Hadoop: The Definitive Guide*. Beijing: O'Reilly. 4th ed.

Xie M, Singh K. 2013. Confidence distribution, the frequentist distribution estimator of a parameter: a review. *Int. Stat. Rev.* 81(1):3–39

Xu C, Zhang Y, Li R, Wu X. 2016. On the feasibility of distributed kernel regression for big data. *IEEE Trans. Knowledge Data Eng.* 28(11):3041–52

Yang Q, Liu Y, Chen T, Tong Y. 2019. Federated machine learning: concept and applications. *ACM Trans. Intel. Syst. Technol.* 10(2):12

Yao L, Chu Z, Li S, Li Y, Gao J, Zhang A. 2021. A survey on causal inference. *ACM Trans. Knowledge Discov. Data* 15(5):74

Yao X, Sun L. 2020. Continual local training for better initialization of federated models. In *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 1736–40. Piscataway, NJ: IEEE

Yu T, Bagdasaryan E, Shmatikov V. 2022. Salvaging federated learning by local adaptation. arXiv:2002.04758 [cs.LG]

Zaharia M, Chowdhury M, Franklin MJ, Shenker S, Stoica I. 2012. Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, artic. 2. Berkeley, CA: USENIX

Zeng D, Lin D. 2015. On random-effects meta-analysis. *Biometrika* 102(2):281–94

Zhang CH, Zhang SS. 2014. Confidence intervals for low dimensional parameters in high dimensional linear models. *J. R. Stat. Soc. Ser. B* 76(1):217–42

Zhang M, Sapra K, Fidler S, Yeung S, Alvarez JM. 2020. Personalized federated learning with first order model optimization. arXiv:2012.08565 [cs.LG]

Zhang Y, Duchi JC, Wainwright MJ. 2013. Communication-efficient algorithms for statistical optimization. *J. Mach. Learn. Res.* 14(1):3321–63

Zhang Y, Lin X. 2015. DiSCO: distributed optimization for self-concordant empirical loss. *Proc. Mach. Learn. Res.* 37:362–70

Zhao T, Cheng G, Liu H. 2016. A partially linear framework for massive heterogeneous data. *Ann. Stat.* 44(4):1400–37

Zhou L, She X, Song PXK. 2022. Distributed empirical likelihood approach to integrating unbalanced datasets. *Stat. Sin.* 33:2209–31

Zhou L, Song PXK. 2017. Scalable and efficient statistical inference with estimating functions in the MapReduce paradigm for big data. arXiv:1709.04389 [stat.ME]

Zhou S, Li GY. 2021. Communication-efficient ADMM-based federated learning. arXiv:2110.15318 [cs.LG]

Zhu H, Jin Y. 2020. Multi-objective evolutionary federated learning. *IEEE Trans. Neural Netw. Learn. Syst.* 31(4):1310–22