Federated Learning for Non-IID Data: From Theory to Algorithm

Bojian Wei[1,2][0000−0001−5882−2523], Jian Li*[1][0000−0003−4977−1802], Yong Liu[3][0000−0002−6739−621X], and Weiping Wang[1][0000−0002−8618−4992]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{weibojian,lijian9026,wangweiping}@iie.ac.cn
[2] University of Chinese Academy of Sciences, Beijing, China
[3] Renmin University of China, Beijing, China
liuyonggsai@ruc.edu.cn

Abstract. Federated learning suffers from terrible generalization performance because the model fails to utilize global information over all clients when data is non-IID (not independently or identically distributed) partitioning. Meanwhile, the theoretical studies in this field are still insufficient. In this paper, we present an *excess risk* bound for federated learning on non-IID data, which measures the error between the model of federated learning and the optimal centralized model. Specifically, we present a novel error decomposition strategy, which decomposes the *excess risk* into three terms: *agnostic error*, *federated error*, and *approximation error*. By estimating the error terms, we find that Rademacher complexity and discrepancy distance are the keys to affecting the learning performance. Motivated by the theoretical findings, we propose `FedAvgR` to improve the performance via additional regularizers to lower the *excess risk*. Experimental results demonstrate the effectiveness of our algorithm and coincide with our theory.

Keywords: Federated Learning · Non-IID · Excess Risk Bound.

1 Introduction

Federated learning (FL) [25] is a new machine learning paradigm where a large number of clients collaboratively train a model under the coordination of a central server. Different from centralized learning (CL), in FL setting, the raw data of each client is stored locally, other clients and the central server have no access to it. Instead, the global model is updated by alternately performing local training and server aggregating. At present, FL still faces many problems [14], one severe problem in FL is that training data is usually non-IID among clients, and this leads to the decline of the model's effectiveness compared to CL.

Some studies [33, 31, 28] try to solve this problem by designing new optimization algorithms. `FedAvg` [25] is an efficient algorithm based on iterative model averaging, but it might be less accurate when dealing with non-IID data. `FedProx`

* Corresponding author

[15] adds a proximal term to local objectives to constrain the gap between local models and the global model, but the convergence is slower. `SCAFFOLD` [9] controls variates to reduce the variance among local updates, while it increases communication costs. Using local momentum [38] instead of local SGD empirically improves the accuracy in heterogeneous settings, but such methods also require additional communication. FL with server momentum [35] performs better than many existing methods including `SCAFFOLD` without increasing communication costs, but such methods do not consider the specific non-IID setting. `FedNova` [34] was proposed to tackle objective inconsistency problem and it could be combined with some acceleration techniques [17], while it has not taken the distribution discrepancy into account. Another effective way is to apply clustering [30, 5] to FL, where clients are divided into several groups based on their similarities, but the metric of clustering and the number of clusters needs to be determined in advance.

On the contrary, there are only a few generalization analysis [21, 13] for FL under non-IID setting. Many works have analyzed federated optimization from the aspect of homogeneity [2, 32] or heterogeneity [36, 18], where some works focus on the convergence of federated stochastic algorithms [7] and have made progress in relaxing the assumptions [18]. Most theoretical works paid more attention to the optimization problem with convergence analysis [16, 9] on non-IID data, some of which showed that the heterogeneity of data slows down the convergence. From the perspective of generalization, agnostic federated learning [26] provided a new point on FL, but the target is to optimize the worst case in the hypothesis, which often performs not well in practice, and it only focused on the generalization error of FL. Thus, there is still a lack of generalization analysis between FL and CL under the traditional framework, which may help to further improve the performance of FL under non-IID setting.

In this paper, we analyze the *excess risk* of FL on non-IID data, which measures the gap between FL and the optimal CL, and we give the corresponding *excess risk* bound. With proper error decompositions, the *excess risk* can be divided into *agnostic error*, *federated error*, and *approximation error*, then we further construct ingenious error decompositions to derive the upper bound of these errors by means of Rademacher complexity [27, 1] and discrepancy distance [24, 3, 40]. Based on the theoretical analysis, we devise an effective algorithm, where we introduce three regularizers to ensure the performance of FL on non-IID data. Experimental results on the synthetic dataset and real-world datasets show that our proposed algorithm outperforms the previous methods and validates our theory.

The contributions of our work are summarized as follows:

- Theoretically, we give the *excess risk* bound between FL on non-IID data and CL for the first time and find out the factors that affect the accuracy decline. We give a reasonable explanation for the bound by decomposing the *excess risk* into three terms: *agnostic error*, *federated error* and *approximation error*, where each term has a detailed analysis with complete proof.

– Algorithmically, we propose a novel algorithm `FedAvgR` (Federated Averaging with Regularization) to improve the performance of FL on non-IID data, which is regularized by Rademacher complexity and discrepancy distance. Furthermore, we design a learning framework for a linear classifier with nonlinear feature mapping, where all the parameters will be updated automatically through back-propagation.

2 Preliminaries and Notations

There are some general notations used in this paper. Assume that there are K clients in a FL setting, where data on the k-th client is drawn i.i.d. from distribution ρ_k, data on different clients may not have the same distribution ($\rho_i \neq \rho_j$), and all clients participate in each round (cross-silo FL). The global distribution is assumed to be a mixture distribution of local distributions on all K clients: $\rho = \sum_{k=1}^{K} p_k \rho_k$, where p_k is the mixture weight ($\sum_{k=1}^{K} p_k = 1$). Actually, the mixture weight p_k is unknown, so an estimated weight $\widehat{p}_k$ will be applied in practice, which brings us the estimated global distribution $\widetilde{\rho} = \sum_{k=1}^{K} \widehat{p}_k \rho_k$.

In this paper, we focus on the multi-classification task. We denote the hypothesis space $\mathcal{H} = \{\boldsymbol{x} \to f(\boldsymbol{x})\}$ consisting of labeling functions $f : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X} \subseteq \mathbb{R}^d$ represents the input space and $\mathcal{Y} \subseteq \mathbb{R}^C$ represents the label space, training samples $(\boldsymbol{x}^k, y^k)$ on the k-th client with size of n_k are i.i.d. drawn from $\rho_k(\boldsymbol{x}, y)$. The labeling function f is formed as $f(\boldsymbol{x}) = \boldsymbol{W}^T \phi(\boldsymbol{x})$, where $\boldsymbol{W} \in \mathbb{R}^{D \times C}$, $\phi(\boldsymbol{x}) \in \mathbb{R}^D$ and $\phi(\cdot)$ is the feature mapping with learnable parameters $\boldsymbol{\varphi}$.

Let $\ell(f(\boldsymbol{x}), y)$ be the loss function, which is assumed to be upper bounded by M ($M > 0$), and $\mathcal{L} = \{\ell(f(\boldsymbol{x}), y) | f \in \mathcal{H}\}$ be the family of loss functions on the hypothesis $\mathcal{H}$, the expected loss of FL on ρ can be described as

$$\mathcal{E}_\rho(f) = \sum_{k=1}^{K} p_k \mathcal{E}_{\rho_k}(f) = \sum_{k=1}^{K} p_k \int_{\mathcal{X} \times \mathcal{Y}} \ell(f(\boldsymbol{x}), y) d\rho_k(\boldsymbol{x}, y),$$

and the corresponding empirical loss is

$$\widehat{\mathcal{E}}_\rho(f) = \sum_{k=1}^{K} p_k \widehat{\mathcal{E}}_{\rho_k}(f) = \sum_{k=1}^{K} p_k \frac{1}{n_k} \sum_{i=1}^{n_k} \ell(f(\boldsymbol{x}_i^k), y_i^k).$$

The empirical learner of FL on the estimated distribution $\widetilde{\rho}$ is denoted by $\widetilde{f}_{fl} = \arg\min_{f \in \mathcal{H}} \sum_{k=1}^{K} \widehat{p}_k \widehat{\mathcal{E}}_{\rho_k}(f)$, and we define the expected (optimal) learner in $\mathcal{H}$ as $f^* = \arg\min_{f \in \mathcal{H}} \mathcal{E}_\rho(f)$, which minimizes the expected loss on ρ.

The performance of a learning model is usually measured by the *excess risk*: $\mathcal{E}_\rho(\widetilde{f}_{fl}) - \mathcal{E}_\rho(f^*)$. Unlike the generalization error, *excess risk* represents the gap between an empirical model and the optimal model, which has not been considered recently in FL. In the following, we consider bounding this *excess risk*.

3 Generalization Analysis

In this section, we will derive the *excess risk* bound between FL and CL.

To this end, we decompose the *excess risk* into *agnostic error* A_1, *federated error* A_2, and *approximation error* A_3:

$$\mathcal{E}_\rho(\widetilde{f}_{fl})-\mathcal{E}_\rho(f^*) \leq \underbrace{\mathcal{E}_\rho(\widetilde{f}_{fl}) - \mathcal{E}_\rho(\widehat{f}_{fl})}_{A_1:=} + \underbrace{\mathcal{E}_\rho(\widehat{f}_{fl}) - \mathcal{E}_\rho(\widehat{f}_{cl})}_{A_2:=} + \underbrace{\mathcal{E}_\rho(\widehat{f}_{cl}) - \mathcal{E}_\rho(f^*)}_{A_3:=}, \quad (1)$$

where $\widehat{f}_{fl} = \arg\min_{f\in\mathcal{H}} \sum_{k=1}^K p_k \widehat{\mathcal{E}}_{\rho_k}(f)$ denotes the empirical learner on the unknown real distribution ρ and $\widehat{f}_{cl} = \arg\min_{f\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^n \ell(f(\boldsymbol{x}_i), y_i)$ denotes the empirical learner of CL.

As mentioned above, $\widehat{p}_k \neq p_k$ results in the difference between $\widetilde{f}_{fl}$ and $\widehat{f}_{fl}$, which is caused by the agnostic nature of the mixture weight. And, in CL setting, model is trained directly on the samples $\{(\boldsymbol{x}_n, y_n), ..., (\boldsymbol{x}_n, y_n)\}$ i.i.d. drawn from $\rho(\boldsymbol{x}, y)$ with size of n ($n = \sum_{k=1}^K n_k$).

In (1), A_1 represents the difference of expected loss for FL between the estimated distribution and the real distribution, A_2 represents the difference of expected loss between FL and CL, and A_3 represents the approximation error of CL to the optimal solution.

3.1 Bounds of Three Error Terms

To measure the performance gap of a model on different distributed data, we introduce the discrepancy distance [24] as follows:

$$disc_L(Q_1, Q_2) = \sup_{f\in\mathcal{H}} \left|\mathcal{E}_{Q_1}(f) - \mathcal{E}_{Q_2}(f)\right|, \quad (2)$$

where Q_1 and Q_2 are two different distributions.

Using Rademacher complexity and discrepancy distance, we bound A_1, A_2, and A_3 as follows.

Theorem 1 (Agnostic Error Bound). *Assume that $\ell(f(\boldsymbol{x}), y)$ is λ-Lipschitz equipped with the 2-norm, that is $|\ell(f(\boldsymbol{x}), y) - \ell(f(\boldsymbol{x}'), y')| \leq \lambda\|f(\boldsymbol{x}) - f(\boldsymbol{x}')\|_2$, $B = \sup_{f=\boldsymbol{W}^T\phi(\boldsymbol{x})\in\mathcal{H}} \|\boldsymbol{W}\|_*$, where $\|\cdot\|_*$ denotes the trace norm. With probability at least $1-\delta$ ($\delta > 0$), we have:*

$$A_1 \leq 2disc_L(\widetilde{\rho}, \rho) + 4\sqrt{2}\lambda B \sum_{k=1}^K \frac{\widehat{p}_k}{n_k}\sqrt{C}\|\phi(\boldsymbol{X}^k)\|_F + 6M\sqrt{\frac{\mathcal{S}(\widehat{\boldsymbol{p}}||\bar{\boldsymbol{n}})\log(2/\delta)}{2n}},$$

where $\|\phi(\boldsymbol{X}^k)\|_F = \sqrt{\sum_{i=1}^{n_k}\left\langle\phi(\boldsymbol{x}_i^k), \phi(\boldsymbol{x}_i^k)\right\rangle}$, $\mathcal{S}(\widehat{\boldsymbol{p}}||\bar{\boldsymbol{n}}) = \chi^2(\widehat{\boldsymbol{p}}||\bar{\boldsymbol{n}}) + 1$, χ^2 denotes the chi-squared divergence, $\widehat{\boldsymbol{p}} = [\widehat{p}_1, ..., \widehat{p}_K]$, and $\bar{\boldsymbol{n}} = \frac{1}{n}[n_1, ..., n_K]$.

Proof. We first decompose A_1 into the following parts:

$$A_1 = \mathcal{E}_\rho(\widetilde{f}_{fl}) - \mathcal{E}_{\widetilde{\rho}}(\widetilde{f}_{fl}) + \underbrace{\mathcal{E}_{\widetilde{\rho}}(\widetilde{f}_{fl}) - \mathcal{E}_{\widetilde{\rho}}(\widehat{f}_{fl})}_{A_1':=} + \mathcal{E}_{\widetilde{\rho}}(\widehat{f}_{fl}) - \mathcal{E}_\rho(\widehat{f}_{fl}), \tag{3}$$

According to 2, we know that $\mathcal{E}_\rho(\widetilde{f}_{fl}) - \mathcal{E}_{\widetilde{\rho}}(\widetilde{f}_{fl}) \leq disc_L(\rho, \widetilde{\rho})$ and $\mathcal{E}_{\widetilde{\rho}}(\widehat{f}_{fl}) - \mathcal{E}_\rho(\widehat{f}_{fl}) \leq disc_L(\widetilde{\rho}, \rho)$. Then, We further decompose A_1' as:

$$A_1' = \underbrace{\mathcal{E}_{\widetilde{\rho}}(\widetilde{f}_{fl}) - \widehat{\mathcal{E}}_{\widetilde{\rho}}(\widetilde{f}_{fl})}_{A_{11}:=} + \underbrace{\widehat{\mathcal{E}}_{\widetilde{\rho}}(\widetilde{f}_{fl}) - \widehat{\mathcal{E}}_{\widetilde{\rho}}(\widehat{f}_{fl})}_{A_{12}:=} + \underbrace{\widehat{\mathcal{E}}_{\widetilde{\rho}}(\widehat{f}_{fl}) - \mathcal{E}_{\widetilde{\rho}}(\widehat{f}_{fl})}_{A_{13}:=}.$$

where A_{11} and A_{13} represent the generalization errors of $\widetilde{f}_{fl}$ and $\widehat{f}_{fl}$, respectively, which can be bounded by weighted Rademacher complexity.

Definition 1 (Weighted Rademacher Complexity). *Let $\mathcal{H}$ be a hypothesis space of f defined over $\mathcal{X}$, $\mathcal{L}$ be the family of loss functions associated to $\mathcal{H}$, $\boldsymbol{n} = [n_1, ..., n_K]$ be the vector of sample sizes and $\boldsymbol{p} = [p_1, ..., p_K]$ be the mixture weight vector, the empirical weighted Rademacher complexity of $\mathcal{L}$ is*

$$\widehat{\mathcal{R}}(\mathcal{L}, \boldsymbol{p}) = \mathbb{E}_\epsilon \left[\sup_{f \in \mathcal{H}} \sum_{k=1}^{K} \frac{p_k}{n_k} \sum_{i=1}^{n_k} \epsilon_i^k l(f(\boldsymbol{x}_i^k), y_i^k) \right],$$

and the empirical weighted Rademacher complexity of $\mathcal{H}$ is

$$\widehat{\mathcal{R}}(\mathcal{H}, \boldsymbol{p}) = \mathbb{E}_\epsilon \left[\sup_{f \in \mathcal{H}} \sum_{k=1}^{K} \frac{p_k}{n_k} \sum_{i=1}^{n_k} \sum_{c=1}^{C} \epsilon_{ic}^k f_c(\boldsymbol{x}_i^k) \right],$$

where $f_c(\boldsymbol{x}_i^k)$ is the c-th value of $f(\boldsymbol{x}_i^k)$ corresponding to the C classes, ϵ_i^ks and ϵ_{ic}^ks are independent Rademacher variables, which are uniformly sampled from $\{-1, +1\}$, respectively.

For any sample $S = \{S_1, ...S_n\}$ drawn from ρ, define $\Phi(S)$ by $\Phi(S) = \sup_{f \in \mathcal{H}}(\mathcal{E}_{\widetilde{\rho}}(f) - \widehat{\mathcal{E}}_{\widetilde{\rho}}(f))$. According to[26], we have

$$\Phi(S) \leq 2\widehat{\mathcal{R}}(\mathcal{L}, \widehat{\boldsymbol{p}}) + 3M\sqrt{\frac{\chi^2(\widehat{\boldsymbol{p}}||\bar{\boldsymbol{n}}) + 1}{2n} \log \frac{2}{\delta}}. \tag{4}$$

According to [8], it holds that $\widehat{\mathcal{R}}(\mathcal{L}, \widehat{\boldsymbol{p}}) \leq \sqrt{2}\lambda\widehat{\mathcal{R}}(\mathcal{H}, \widehat{\boldsymbol{p}})$ under the Lipschitz assumption. Applying Hölder's inequality, we have:

$$\begin{aligned}
\widehat{\mathcal{R}}(\mathcal{H}, \widehat{\boldsymbol{p}}) &= \mathbb{E}_\epsilon \left[\sup_{f \in \mathcal{H}} \sum_{k=1}^{K} \frac{\widehat{p}_k}{n_k} \langle \boldsymbol{W}_k, \boldsymbol{\Phi}_k \rangle \right] \leq \mathbb{E}_\epsilon \left[\sum_{k=1}^{K} \frac{\widehat{p}_k}{n_k} \sup_{f \in \mathcal{H}} \langle \boldsymbol{W}_k, \boldsymbol{\Phi}_k \rangle \right] \\
&\leq \mathbb{E}_\epsilon \left[\sum_{k=1}^{K} \frac{\widehat{p}_k}{n_k} \sup_{f \in \mathcal{H}} \|\boldsymbol{W}_k\|_* \|\boldsymbol{\Phi}_k\|_F \right] \leq \mathbb{E}_\epsilon \left[\sum_{k=1}^{K} \frac{\widehat{p}_k}{n_k} B \|\boldsymbol{\Phi}_k\|_F \right] \\
&= B \sum_{k=1}^{K} \frac{\widehat{p}_k}{n_k} \mathbb{E}_\epsilon \left[\|\boldsymbol{\Phi}_k\|_F \right] \leq B \sum_{k=1}^{K} \frac{\widehat{p}_k}{n_k} \sqrt{\mathbb{E}_\epsilon [\|\boldsymbol{\Phi}_k\|_F^2]}.
\end{aligned} \tag{5}$$

where $\boldsymbol{W}_k, \boldsymbol{\Phi}_k = [\sum_{i=1}^{n_k} \epsilon_{i1}^k \phi(\boldsymbol{x}_i^k), ..., \sum_{i=1}^{n_k} \epsilon_{iC}^k \phi(\boldsymbol{x}_i^k)] \in \mathbb{R}^{D\times C}$ and $\langle \boldsymbol{W}_k, \boldsymbol{\Phi}_k \rangle = \mathrm{Tr}(\boldsymbol{W}_k^T \boldsymbol{\Phi}_k)$. $\mathbb{E}_\epsilon[\|\boldsymbol{\Phi}_k\|_F^2]$ can be further bounded as follows [10]:

$$\begin{aligned}\mathbb{E}_\epsilon[\|\boldsymbol{\Phi}_k\|_F^2] &\leq \mathbb{E}_\epsilon\left[\sum_{c=1}^{C}\left\|\sum_{i=1}^{n_k}\epsilon_{ic}^k\phi(\boldsymbol{x}_i^k)\right\|_2^2\right] \leq \sum_{c=1}^{C}\mathbb{E}_\epsilon\left[\left\|\sum_{i=1}^{n_k}\epsilon_{ic}^k\phi(\boldsymbol{x}_i^k)\right\|_2^2\right] \\ &\leq \sum_{c=1}^{C}\mathbb{E}_\epsilon\left[\sum_{i,j=1}^{n_k}\epsilon_{ic}^k\epsilon_{jc}^k\langle\phi(\boldsymbol{x}_i^k),\phi(\boldsymbol{x}_j^k)\rangle\right] = C\|\phi(\boldsymbol{X}^k)\|_F^2.\end{aligned} \tag{6}$$

Based on the definition of learners, $\widetilde{f}_{fl}$ minimizes the empirical loss on $(\boldsymbol{x}, y) \sim \widetilde{\rho}$, while $\widehat{f}_{fl}$ minimizes the empirical risk on $(\boldsymbol{x}, y) \sim \rho$, so it is obvious that $\widehat{\mathcal{E}}_{\widetilde{\rho}}(\widetilde{f}_{fl}) \leq \widehat{\mathcal{E}}_{\rho}(\widehat{f}_{fl})$. Therefore, the proof of Theorem 1 is completed.

A_1 (*agnostic error*) is mainly caused by the gap between the estimated distribution $\widetilde{\rho}$ and the real distribution ρ, because the underlying mixture weight p_k is unknown. $\mathcal{S}(\widehat{\boldsymbol{p}}||\bar{\boldsymbol{n}})$ represents the distance between $\widehat{p}_k$ and the uniform mixture weight $\frac{n_k}{n}$, which gives a guidance on the choice of $\widehat{p}_k$.

Theorem 2 (Federated Error Bound). *Under the same assumptions as Theorem 1, with probability at least* $1-\delta(\delta>0)$*, we have:*

$$A_2 \leq \sum_{k=1}^{K} p_k \left(disc_L(\rho_k, \rho) + \frac{4\sqrt{2}\lambda B}{n_k}\sqrt{C}\|\phi(\boldsymbol{X}^k)\|_F \right) + \sum_{k=1}^{K} p_k \left(6M\sqrt{\frac{\log(2/\delta)}{2n_k}} \right).$$

Proof. Note that $A_2 = \sum_{k=1}^{K} p_k [\underbrace{\mathcal{E}_{\rho_k}(\widehat{f}_{fl}) - \mathcal{E}_{\rho}(\widehat{f}_{cl})}_{A_2' :=}]$, we decompose A_2' as:

$$\underbrace{\mathcal{E}_{\rho_k}(\widehat{f}_{fl}) - \widehat{\mathcal{E}}_{\rho_k}(\widehat{f}_{fl})}_{A_{21}} + \underbrace{\widehat{\mathcal{E}}_{\rho_k}(\widehat{f}_{fl}) - \widehat{\mathcal{E}}_{\rho_k}(\widehat{f}_{cl})}_{A_{22}} + \underbrace{\widehat{\mathcal{E}}_{\rho_k}(\widehat{f}_{cl}) - \mathcal{E}_{\rho_k}(\widehat{f}_{cl})}_{A_{23}} + \underbrace{\mathcal{E}_{\rho_k}(\widehat{f}_{cl}) - \mathcal{E}_{\rho}(\widehat{f}_{cl})}_{A_{24}}.$$

Substituting A_{22} into the equation of A_2, due to the definition of $\widehat{f}_{fl}$, we have $\sum_{k=1}^{K} p_k[\widehat{\mathcal{E}}_{\rho_k}(\widehat{f}_{fl}) - \widehat{\mathcal{E}}_{\rho_k}(\widehat{f}_{cl})] \leq 0$. Similar to Theorem 1, the rest parts of A_2' can be bounded by Rademacher complexity [27] and discrepancy distance.

Therefore, the proof is completed by bounding the four parts.

A_2 (*federated error*) is mainly caused by the FL setting. Samples on different clients are drawn from different distributions, which results in the discrepancy between ρ_k and ρ, where the CL model is directly trained on ρ.

Theorem 3 (Approximation Error Bound). *Under the same assumptions as Theorem 1, with probability* $1-\delta(\delta>0)$*, we have:*

$$A_3 \leq \frac{4\sqrt{2}\lambda B}{n}\sqrt{C}\|\phi(\boldsymbol{X})\|_F + 3M\sqrt{\frac{\log(2/\delta)}{2n}},$$

where $\|\phi(\boldsymbol{X})\|_F = \sqrt{\sum_{i=1}^{n}\langle\phi(\boldsymbol{x}_i),\phi(\boldsymbol{x}_i)\rangle}$.

A_3 (*approximation error*) is a classic excess risk bound in CL setting, which represents the gap between an empirical learner and the optimal learner in $\mathcal{H}$.

Remark 1 (Proof Novelty). 1) The *excess risk* bound for FL on non-IID data can not be derived directly. To bridge $\widetilde{f}_{fl}$ to f^*, we decompose the *excess risk* into three error terms. 2) There is no available tool can be applied to bound A_1 and A_2 directly. Thus, we propose a two-stage error decomposition for A_1 and a novel decomposition for A_2 (See proofs for details).

3.2 Excess Risk Bound

The *excess risk* bound is obtained by combining the above bounds together.

Theorem 4 (Excess Risk Bound). *Under the same assumptions as Theorem 1, With probability at least $1-\delta$ ($\delta > 0$), the excess risk bound of federated learning on non-IID data holds as follows:*

$$\mathcal{E}_\rho(\widehat{f}_{fl}) - \mathcal{E}_\rho(f^*) \leq \mathcal{O}\left(G_1 + G_2 + G_3\right), \tag{7}$$

where $G_1 = disc_L(\widetilde{\rho}, \rho) + \sum_{k=1}^{K} \frac{\widehat{p}_k B\sqrt{C}}{n_k}\|\phi(\boldsymbol{X}^k)\|_F + \sqrt{\frac{\mathcal{S}(\widehat{\boldsymbol{p}}||\bar{\boldsymbol{n}})}{n}}$, $G_2 = \frac{B\sqrt{C}}{n}\|\phi(\boldsymbol{X})\|_F + \sqrt{\frac{1}{n}}$ *and* $G_3 = \sum_{k=1}^{K} p_k[disc_L(\rho_k, \rho) + \frac{B\sqrt{C}}{n_k}\|\phi(\boldsymbol{X}^k)\|_F + \sqrt{\frac{1}{n_k}}]$.

According to Theorem 4, to lower the *excess risk*, we need to reduce $disc_L(\rho_k, \rho)$, constrain $\|\boldsymbol{W}\|_*$ and $\|\phi(\boldsymbol{X}^k)\|_F$, and at the same time reduce $disc_L(\widetilde{\rho}, \rho)$.

In non-IID condition, samples on different clients are drawn from different distributions, so the gap between ρ_k and ρ certainly exists. Furthermore, p_k is unknown, how can we reduce $disc_L(\widetilde{\rho}, \rho)$? Actually, if we reduce $disc_L(\rho_k, \rho)$, the differences among local distributions will become smaller, that is, the degree of non-IID will be reduced. At this time, ρ_k is approximate to ρ, so $\widehat{p}_k$ has a small effect on $disc_L(\widetilde{\rho}, \rho)$, especially, when $\rho_k = \rho$, whatever value we choose for $\widehat{p}_k$, it's not going to make big difference to the global distribution. Therefore, we are able to lower the *excess risk* by reducing $disc_L(\rho_k, \rho)$, $\|\boldsymbol{W}\|_*$ and $\|\phi(\boldsymbol{X}^k)\|_F$.

On the other hand, when $\phi(\cdot)$ is upper bounded by κ^2 and $\widehat{p}_k$ is equal to p_k, if we can reduce $disc_L(\rho_k, \rho)$ to 0, then (7) will be $\mathcal{O}[(\kappa B\sqrt{C}+1)\sum_{k=1}^{K}\widehat{p}_k\sqrt{\frac{1}{n_k}}]$. In this case, if the number of samples is equal ($n_k = n/K$, $\forall k = 1, ..., K$), we have $\mathcal{O}(\kappa B\sqrt{KC/n})$, which is the convergence rate for the counterpart of distributed learning. Moreover, if we have only one client, we have $\mathcal{O}(\kappa B\sqrt{C/n})$, which is the convergence rate for the counterpart of centralized learning. Thus, our theory gives a more general framework that can be applied to FL as well as distributed learning [37] and CL, with the latter two being a special case of the former.

Remark 2 (Novelty). Few of the existing theoretical studies of FL are concerned with the excess risk. [26] analyzed federated learning under the agnostic framework, which aims to improve the performance under the worst condition, and

this may not get the optimal solution. Also, they just give the generalization bound of agnostic federated learning. In this paper, we analyze the excess risk between federated learning model on non-IID data and the optimal centralized model under a more general framework and derive the excess risk bound, which may provide a new path for theoretical analysis of federated learning.

4 Algorithm

Motivated by the *excess risk* bound, we propose `FedAvgR` (Federated Averaging with Regularization) to improve the performance of FL on non-IID data.

Algorithm 1 `FedAvgR`. K clients are indexed by k, $\mathcal{B}$ is the local mini-batch size, E is the number of local epochs, η is the learning rate, $\boldsymbol{F}$ represents the objective function.

Server-Aggregate

initialize $\boldsymbol{W}_0$ and $\boldsymbol{\varphi}_0$
for $k = 1, ..., K$ **do**
  $\widehat{\rho}_k^{\phi} \leftarrow$ estimate the distribution of $\phi(\boldsymbol{x})$
  upload the parameters of $\widehat{\rho}_k^{\phi}$ to the server
end for
get the global distribution $\widehat{\rho}^{\phi} = \sum_{k=1}^{K} \widehat{p}_k \widehat{\rho}_k^{\phi}$
for each round $t = 1, 2, ...$ **do**
  for each client k **do**
    $\boldsymbol{W}_{t+1}^k, \boldsymbol{\varphi}_{t+1}^k, \widehat{\rho}_k^{\phi} \leftarrow$ Client-Update$(k, \boldsymbol{W}_t, \boldsymbol{\varphi}_t, \widehat{\rho}^{\phi})$
  end for
  update the global distribution $\widehat{\rho}^{\phi}$
  $\boldsymbol{W}_{t+1} \leftarrow \sum_{k=1}^{K} \widehat{p}_k \boldsymbol{W}_{t+1}^k, \boldsymbol{\varphi}_{t+1} \leftarrow \sum_{k=1}^{K} \widehat{p}_k \boldsymbol{\varphi}_{t+1}^k$
end for

Client-Update$(k, \boldsymbol{W}_t, \boldsymbol{\varphi}_t, \widehat{\rho}^{\phi})$

draw samples $Z_{\widehat{\rho}^{\phi}}$ from $\widehat{\rho}^{\phi}$
for epoch$= 1, ..., E$ **do**
  for $(\boldsymbol{x}, y) \in \mathcal{B}$ **do**
    calculate MMD$[\widehat{\rho}_k^{\phi}, \widehat{\rho}^{\phi}]$ by $(\boldsymbol{x}, y)$ and $Z_{\widehat{\rho}^{\phi}}$
    $\boldsymbol{F} = \frac{1}{\mathcal{B}} \sum_{(\boldsymbol{x},y)\in\mathcal{B}} \ell(f(\boldsymbol{x}), y) + \alpha\|\boldsymbol{W}\|_* + \beta\|\phi(\boldsymbol{X})\|_F + \gamma \text{MMD}[\widehat{\rho}_k^{\phi}, \widehat{\rho}^{\phi}]$
    $\boldsymbol{W}_{t+1}^k \leftarrow \boldsymbol{W}_t - \eta \nabla_{\boldsymbol{W}_t} \boldsymbol{F}, \boldsymbol{\varphi}_{t+1}^k \leftarrow \boldsymbol{\varphi}_t - \eta \nabla_{\boldsymbol{\varphi}_t} \boldsymbol{F}$
  end for
  $\widehat{\rho}_k^{\phi} \leftarrow$ estimate the distribution of $\phi(\boldsymbol{x})$
end for

4.1 Regularization

Based on Theorem 4, we can constrain $\|\boldsymbol{W}\|_*$, $\|\phi(\boldsymbol{X}^k)\|_F$, and $disc_L(\rho_k, \rho)$ by adding them to the objective function as regularizers [11, 12].

Unlike $\|\boldsymbol{W}\|_*$ and $\|\phi(\boldsymbol{X}^k)\|_F$, the discrepancy distance $disc_L(\rho_k, \rho)$ is not an explicit variable-dependent term, so we need to find an approach to quantify it.

Another problem is that the local distribution ρ_k won't change during training, so we shall reduce the discrepancy after feature mapping. In other words, we can reduce $disc_L(\rho_k^\phi, \rho^\phi)$ instead of $disc_L(\rho_k, \rho)$, where ρ_k^ϕ and ρ^ϕ are respectively the local feature distribution on client k and global feature distribution.

We choose MMD (Maximum Mean Discrepancy) [4] to measure the distance between different distributions Q_1 and Q_2, which is formed as $\mathrm{MMD}[Q_1, Q_2] = \sup_{f\in\mathcal{H}}(\mathbb{E}_{Q_1}[f(\boldsymbol{x})] - \mathbb{E}_{Q_2}[f(\boldsymbol{x})])$. Assume that $\mathcal{H}$ is a complete inner product space of f, then $\mathcal{H}$ can be termed a reproducing kernel Hilbert space when the continuous linear point evaluation mapping $f \to f(\boldsymbol{x})$ exists for all $\boldsymbol{x} \in \mathcal{X}$. Thus, we can use inner product to represent $f(\boldsymbol{x})$: $f(\boldsymbol{x}) = \langle f, \phi(\boldsymbol{x})\rangle_{\mathcal{H}}$, so it holds that $\mathrm{MMD}[Q_1, Q_2] = \|\mathbb{E}_{Q_1}[\phi(\boldsymbol{x})], \mathbb{E}_{Q_2}[\phi(\boldsymbol{x}')]\|_{\mathcal{H}}$, and the related expansion is:

$$\frac{1}{m^2}\sum_{i,j=1}^{m}\langle\phi(\boldsymbol{x}_i),\phi(\boldsymbol{x}_j)\rangle_{\mathcal{H}} + \frac{1}{n^2}\sum_{i,j=1}^{n}\langle\phi(\boldsymbol{x}'_i),\phi(\boldsymbol{x}'_j)\rangle_{\mathcal{H}} - \frac{2}{mn}\sum_{i,j=1}^{m,n}\langle\phi(\boldsymbol{x}_i),\phi(\boldsymbol{x}'_j)\rangle_{\mathcal{H}},$$

where m and n denotes the number of samples on Q_1 and Q_2, respectively.

Taking $\mathrm{MMD}[\rho_k^\phi, \rho^\phi]$ as a regularizer with $\|\boldsymbol{W}\|_*$ and $\|\phi(\boldsymbol{X}^k)\|_F$, the objective function on the k-th client is

$$\min_{\boldsymbol{W},\boldsymbol{\varphi}} \frac{1}{n_k}\sum_{i=1}^{n_k}\ell(f(\boldsymbol{x}_i^k), y_i^k) + \alpha\|\boldsymbol{W}\|_* + \beta\|\phi(\boldsymbol{X}^k)\|_F + \gamma\mathrm{MMD}[\rho_k^\phi, \rho^\phi].$$

4.2 Learning Framework

The procedure of `FedAvgR` is listed in Algorithm 1. First, the server sends the initial parameters to all clients, then we estimate the empirical local distribution $\widehat{\rho}_k^\phi$ and upload them to the server to get the empirical global distribution $\widehat{\rho}^\phi$. Next, at each communication round, each client updates the model parameters and reestimates $\widehat{\rho}_k^\phi$ locally. Then, the server aggregates local updates and renews $\widehat{\rho}^\phi$ based on $\widehat{\rho}_k^\phi$, which will be sent to all clients again [23].

In order to calculate $\mathrm{MMD}[\rho_k^\phi, \rho^\phi]$ in client-update, we first drawn samples $Z_{\widehat{\rho}^\phi}$ from $\widehat{\rho}^\phi$, and then calculate $\mathrm{MMD}[\rho_k^\phi, \rho^\phi]$ by $\phi(\boldsymbol{x}^k)$ and $Z_{\widehat{\rho}^\phi}$. In server-aggregate, we choose $\widehat{p}_k = n_k/n$ to aggregate the updates, so that $\mathcal{S}(\widehat{\boldsymbol{p}}||\bar{\boldsymbol{n}})$ can be minimized. Particularly, when $disc_L(\rho_k^\phi, \rho^\phi)$ is close to 0, the learning problem degenerates into the distributed learning, where n_k/n is widely used.

We design a learning framework (Figure 1) for linear classifier to update all the parameters automatically through back-propagation, where $\boldsymbol{W}^T\phi(\boldsymbol{x})$ can be treated as a fully-connected neural network with one hidden layer and we only need to initialize the parameters. Besides, we apply D feature mappings with different parameters to reduce variance. Moreover, this framework is generalizable, where $\phi(\cdot)$ can be replaced by neural network, kernel method [22], etc.

5 Experiment

In this section, we will introduce our experimental setup and conduct extensive experiments to demonstrate our theory and show the effectiveness of `FedAvgR`.

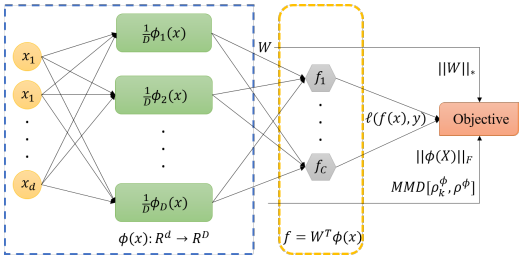


Fig. 1. Architecture of local learning framework

5.1 Experimental Setup

We evaluate our algorithm and make further analysis on some real-world datasets and the synthetic dataset. All the experiments are trained on a Linux_x86_64 server (CPU: Intel(R) Xeon(R) Silver 4214 (RAM: 196 GB) / GPU: NVIDIA GeForce RTX-2080ti).

The synthetic dataset in our experiment is generated related to the method in [16], where the number of samples n_k on client k follows a power law. We choose three binary-classification datasets (a1a, svmguide1 and splice) and six multi-classification datasets (vehicle, dna, pendigits, satimage, usps and MNIST) from LIBSVM [6]. We apply the partitioning method related to [25] to all these datasets to get non-IID data. We sort each dataset by the label and divide it into N/N_s shards of size N_s, where N is the total number of samples, then we assign each client 2 shards. The detailed information for the real-world datasets [6] is listed in Table 1, where the training sets and the test sets are officially splited except vehicle.

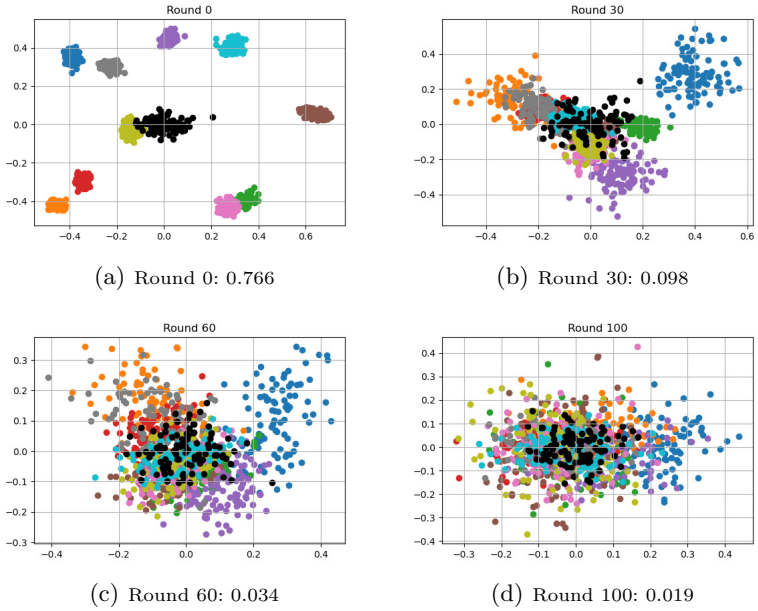
Table 1. Information of Different Datasets

Dataset	Class	Training Size	Testing Size	Features
a1a	2	1605	30956	123
svmguide1	2	3089	4000	4
splice	2	1000	2175	60
vehicle	4	500	446	18
dna	3	2000	1186	180
pendigits	10	7494	3498	16
satimage	6	4435	2000	36
usps	10	7291	2007	256
MNIST	10	60000	10000	28×28

In the following experiments, we use random Fourier feature to do the feature mapping. According to [29], random feature mapping can be formed as $\sqrt{2}\cos(\boldsymbol{\omega}^T\boldsymbol{x} + b)$, where $\boldsymbol{\omega}$ is sampled from $\mathcal{N}(0, \sigma^2)$, σ is related to the corresponding Gaussian kernel, and b is uniformly sampled from $[0, 2\pi]$.

5.2 Analysis of FedAvgR

In this part, we will discuss the effects of different components on the performance of our algorithm. We set the feature dimension as 100, the minimum number of local samples as 100, and the number of clients as 10.



(a) Round 0: 0.766 (b) Round 30: 0.098

(c) Round 60: 0.034 (d) Round 100: 0.019

Fig. 2. Distributions Changes via Training: Black points are sampled from $\widehat{\rho}^{\phi}$, and others are sampled from $\widehat{\rho}_k^{\phi}$s, the corresponding discrepancy distance is labeled at the bottom of each figure.

Impact of MMD$[\boldsymbol{\rho_k^{\phi}}, \boldsymbol{\rho^{\phi}}]$ MMD$[\rho_k^{\phi}, \rho^{\phi}]$ is used to match local distributions to the global distribution, which is the key component to solve the non-IID problem. We run 100 rounds on the synthetic dataset with $(u, v) = (1, 1)$ and sample 100 points from each $\widehat{\rho}_k^{\phi}$ and $\widehat{\rho}^{\phi}$. To show its impact, we visualize the distributions changes via training process in Figure 2, where all the points are transformed to 2D by PCA (Principal Component Analysis) and $disc_L(\rho_k^{\phi}, \rho^{\phi})$ labeled in Figure 2 is calculated by the distance among the centroids of each group of points. (a) shows the distributions after initializing by random feature, we find that there exists a certain distance between $\widehat{\rho}_k^{\phi}$ and $\widehat{\rho}^{\phi}$. (b) shows the distributions after 30 rounds training, and (c) shows the result after 60 rounds training, it is apparent that $disc_L(\widehat{\rho}_k^{\phi}, \widehat{\rho}^{\phi})$ is getting smaller. (d) shows the distributions after 100 rounds, where the local distributions of all clients converge toward the global distribution, which can reduce the negative impact of non-IID, and this also demonstrate the effectiveness of FedAvgR.

Impacts of Different Regularizers We conduct an experiment to analyze the three regularizers. We run 250 rounds on the synthetic dataset with $(u, v) = (0.5, 0.5)$ and some real-world datasets with non-iid partitioning. As shown in Table 2, `FedAvgR` mostly performs the best, and `FedAvgR` without regularization (equal to `FedAvg`) performs the worst. The performances are close when `FedAvgR` only contains $\|\boldsymbol{W}\|_*$ or $\|\phi(\boldsymbol{X}^k)\|_F$, because both of them are designed to limit the Rademacher complexity. The performance of `FedAvgR` only with MMD$[\rho_k^\phi, \rho^\phi]$ is only second to `FedAvgR` with all three regularizers on most datasets, which exactly demonstrates our theory that when the gap between ρ_k and ρ becomes smaller, the performance of the model will be improved.

Table 2. Test Accuracy of `FedAvgR` with Different Regualrizers

Dataset	No Regularizer	$\|\boldsymbol{W}\|_*$	$\|\phi(\boldsymbol{X}^k)\|_F$	MMD	All Regularizers
svmguide1	89.05	89.20	89.45	89.61	**90.70**
vehicle	77.12	77.17	77.17	77.46	**78.32**
dna	95.33	95.52	95.36	95.45	**95.70**
pendigits	95.70	95.71	95.74	**95.94**	95.90
usps	94.57	94.72	**94.82**	94.80	**94.82**
synthetic	95.82	96.07	96.06	96.12	**96.23**

5.3 Comparison with Other Methods

In this part, we compare `FedAvgR` with `OneShot` [39], `FedAvg` [25], `FedProx` [15] and `FL+HC` [5] on several LIBSVM datasets. The regularization parameters of `FedAvgR` are selected in $\alpha \in \{10^{-8}, 10^{-7}, ..., 10^{-4}\}$, $\beta \in \{10^{-6}, 10^{-5}, ..., 10^{-2}\}$, and $\gamma \in \{10^{-4}, 10^{-3}, ..., 10^{-1}\}$ through 3-folds cross-validation [20, 19], the regularization parameters of `FedProx` are selected in $\{10^{-4}, 10^{-3}, ..., 10^{-1}\}$, and the number of clusters is set as 2 in `FL+HC`. The top-1 accuracy is used to evaluate the performance, and the communication round is set as 300 with 10 epochs on each client per round. We implement all the methods based on Pytorch and use Momentum as optimizer with 10 instances in a mini-batch for training. We run all the methods on each dataset 10 times with different random seeds, and we apply t-test to estimate the statistical significance.

Instead of partitioning the test samples to each client, we test all the algorithms with the entire test set of each dataset, because our target is to learn a global model that has the best generalized performance on the global distribution ρ. `OneShot` aggregates local models when local trainings converge, `FedAvg` iteratively averages local models by n_k/n, `FedProx` adds the last-round's global model to local training as regularization based on `FedAvg`, and `FL+HC` uses hierarchical clustering to divide clients into several clusters and applies `FedAvg` separately.

According to the results in Table 3, `FedAvgR` shows the best performances on all datasets, which means that the use of three regularizers brings notable improvement coincides with our theoretical analysis. `OneShot`, `FedAvg` and `FedProx`

do not consider or explicitly deal with the differences among local distributions, which limits the model's performance on non-IID data, while `FedAvgR` reduces the discrepancies between $\widehat{\rho}_k^{\phi}$s and $\widehat{\rho}^{\phi}$. `FL+HC` is a personalized method for scenarios where each client has its own test samples. In particular, when the number of clusters is 1, `FL+HC` is equal to `FedAvg`.

On most datasets, `FedAvgR` is significantly better than other methods with confidence at level 95%. However, on a1a and splice, the advantage of our algorithm is not significant. The reason is that the datasets are not balanced, where the number of training samples is far less than the number of test samples.

Table 3. Test Accuracy on Real-World Datasets. We run methods on each dataset 10 times, each with 300 rounds. We bold the numbers of the best method and underline the numbers of other methods which are not significantly worse than the best one.

Dataset	`OneShot`	`FedAvg`	`FedProx`	`FL+HC`	`FedAvgR`
a1a	76.86±0.30	84.29±0.06	84.27±0.06	81.63±0.94	**84.30±0.06**
svmguide1	71.50±4.21	90.95±0.86	91.19±0.84	85.66±4.48	**91.77±1.01**
splice	75.95±4.56	90.37±0.21	90.38±0.20	85.12±2.14	**90.40±0.26**
vehicle	52.31±4.36	78.61±1.08	78.58±1.06	62.24±8.12	**78.82±0.98**
dna	63.73±1.02	95.23±0.17	95.18±0.21	92.09±3.25	**95.59±0.23**
pendigits	46.70±2.32	94.87±0.58	94.85±0.59	86.81±4.58	**95.12±0.48**
satimage	73.07±2.39	88.83±0.41	88.46±0.31	76.72±2.96	**88.93±0.39**
usps	56.83±4.06	94.57±0.15	94.53±0.13	88.03±3.62	**94.80±0.19**
MNIST	68.80±2.06	97.26±0.09	97.24±0.07	85.13±2.23	**97.34±0.06**

6 Conclusion

In this paper, we give an *excess risk* bound for federated learning on non-IID data through Rademacher complexity and discrepancy distance, analyzing the error between it and the optimal centralized learning model. Based on our theory, we propose `FedAvgR` to improve the performance of federated learning in non-IID setting, where three regularizers are added to achieve a sharper bound. Experiments show that our algorithm outperforms the previous methods. As the first work to analyze the *excess risk* under a more general framework, our work will provide a reference for the future study of generalization properties in federated learning with non-IID data. Besides, the proof techniques in this paper are helpful to the research of error analysis related to the distributed framework.

Acknowledgement

This work was supported in part by Excellent Talents Program of Institute of Information Engineering, CAS, Special Research Assistant Project of CAS (No. E0YY231114), Beijing Outstanding Young Scientist Program (No.BJJWZYJH01 2019100020098) and National Natural Science Foundation of China (No.62076234).

References

1. Bartlett, P.L., Bousquet, O., Mendelson, S.: Localized rademacher complexities. In: COLT. vol. 2375, pp. 44–58 (2002)
2. Basu, D., Data, D., Karakus, C., Diggavi, S.N.: Qsparse-local-sgd: Distributed SGD with quantization, sparsification and local computations. In: NeurIPS. pp. 14668–14679 (2019)
3. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains. Mach. Learn. **79**(1-2), 151–175 (2010)
4. Borgwardt, K.M., Gretton, A., Rasch, M.J., Kriegel, H., Schölkopf, B., Smola, A.J.: Integrating structured biological data by kernel maximum mean discrepancy. In: Proceedings of the 14th International Conference on Intelligent Systems for Molecular Biology. pp. 49–57 (2006)
5. Briggs, C., Fan, Z., Andras, P.: Federated learning with hierarchical clustering of local updates to improve training on non-iid data. In: International Joint Conference on Neural Networks, IJCNN. pp. 1–9. IEEE (2020)
6. Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology pp. 2:27:1–27:27 (20)
7. Charles, Z., Konečný, J.: Convergence and accuracy trade-offs in federated learning and meta-learning. In: AISTATS. vol. 130, pp. 2575–2583 (2021)
8. Cortes, C., Kuznetsov, V., Mohri, M., Yang, S.: Structured prediction theory based on factor graph complexity. In: NIPS. pp. 2514–2522 (2016)
9. Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S.J., Stich, S.U., Suresh, A.T.: SCAFFOLD: stochastic controlled averaging for federated learning. In: ICML. vol. 119, pp. 5132–5143 (2020)
10. Li, J., Liu, Y., Wang, W.: Automated spectral kernel learning. In: AAAI. pp. 4618–4625 (2020)
11. Li, J., Liu, Y., Yin, R., Wang, W.: Approximate manifold regularization: Scalable algorithm and generalization analysis. In: IJCAI. pp. 2887–2893 (2019)
12. Li, J., Liu, Y., Yin, R., Wang, W.: Multi-class learning using unlabeled samples: Theory and algorithm. In: IJCAI. pp. 2880–2886 (2019)
13. Li, J., Liu, Y., Yin, R., Zhang, H., Ding, L., Wang, W.: Multi-class learning: From theory to algorithm. In: NeurIPS. pp. 1593–1602 (2018)
14. Li, T., Sahu, A.K., Talwalkar, A., Smith, V.: Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine **37**(3), 50–60 (2020)
15. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks. In: MLSys (2020)
16. Li, X., Huang, K., Yang, W., Wang, S., Zhang, Z.: On the convergence of fedavg on non-iid data. In: ICLR (2020)
17. Li, Z., Kovalev, D., Qian, X., Richtárik, P.: Acceleration for compressed gradient descent in distributed and federated optimization. In: ICML. vol. 119, pp. 5895–5904 (2020)
18. Lian, X., Zhang, C., Zhang, H., Hsieh, C., Zhang, W., Liu, J.: Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent. In: NeurIPS. pp. 5330–5340 (2017)
19. Liu, Y., Jiang, S., Liao, S.: Efficient approximation of cross-validation for kernel methods using bouligand influence function. In: ICML. vol. 32, pp. 324–332 (2014)
20. Liu, Y., Liao, S., Jiang, S., Ding, L., Lin, H., Wang, W.: Fast cross-validation for kernel-based algorithms. IEEE Trans. Pattern Anal. Mach. Intell. **42**(5), 1083–1096 (2020)

21. Liu, Y., Liao, S., Lin, H., Yue, Y., Wang, W.: Generalization analysis for ranking using integral operator. In: AAAI. pp. 2273–2279 (2017)
22. Liu, Y., Liao, S., Lin, H., Yue, Y., Wang, W.: Infinite kernel learning: Generalization bounds and algorithms. In: AAAI. pp. 2280–2286 (2017)
23. Liu, Y., Liu, J., Wang, S.: Effective distributed learning with random features: Improved bounds and algorithms. In: ICLR (2021)
24. Mansour, Y., Mohri, M., Rostamizadeh, A.: Domain adaptation: Learning bounds and algorithms. In: COLT (2009)
25. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: AISTATS. vol. 54, pp. 1273–1282 (2017)
26. Mohri, M., Sivek, G., Suresh, A.T.: Agnostic federated learning. In: ICML. vol. 97, pp. 4615–4625 (2019)
27. Mohri Mehryar, A.R., Talwalkar, A.: Foundations of Machine Learning. The MIT Press, second edn. (2018)
28. Pustozerova, A., Rauber, A., Mayer, R.: Training effective neural networks on structured data with federated learning. In: AINA. vol. 226, pp. 394–406 (2021)
29. Rahimi, A., Recht, B.: Random features for large-scale kernel machines. In: NIPS. pp. 1177–1184 (2007)
30. Sattler, F., Müller, K.R., Samek, W.: Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. IEEE Transactions on Neural Networks and Learning Systems (2020)
31. Smith, V., Chiang, C., Sanjabi, M., Talwalkar, A.S.: Federated multi-task learning. In: NIPS. pp. 4424–4434 (2017)
32. Stich, S.U.: Local SGD converges fast and communicates little. In: ICLR (2019)
33. Wang, H., Yurochkin, M., Sun, Y., Papailiopoulos, D.S., Khazaeni, Y.: Federated learning with matched averaging. In: ICLR (2020)
34. Wang, J., Liu, Q., Liang, H., Joshi, G., Poor, H.V.: Tackling the objective inconsistency problem in heterogeneous federated optimization. In: NeurIPS (2020)
35. Wang, J., Tantia, V., Ballas, N., Rabbat, M.G.: Slowmo: Improving communication-efficient distributed SGD with slow momentum. In: ICLR (2020)
36. Wang, S., Tuor, T., Salonidis, T., Leung, K.K., Makaya, C., He, T., Chan, K.: Adaptive federated learning in resource constrained edge computing systems. IEEE J. Sel. Areas Commun. **37**(6), 1205–1221 (2019)
37. Yin, R., Liu, Y., Lu, L., Wang, W., Meng, D.: Divide-and-conquer learning with nyström: Optimal rate and algorithm. In: AAAI. pp. 6696–6703 (2020)
38. Yu, H., Jin, R., Yang, S.: On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization. In: ICML. vol. 97, pp. 7184–7193 (2019)
39. Zhang, Y., Duchi, J.C., Wainwright, M.J.: Divide and conquer kernel ridge regression: a distributed algorithm with minimax optimal rates. J. Mach. Learn. Res. **16**, 3299–3340 (2015)
40. Zhang, Y., Liu, T., Long, M., Jordan, M.I.: Bridging theory and algorithm for domain adaptation. In: ICML. vol. 97, pp. 7404–7413 (2019)