Heterogeneous Federated Learning on a Graph

Huiyuan Wang, Xuyang Zhao and Wei Lin*

School of Mathematical Sciences, Peking University

huiyuan.wang@pku.edu.cn, xuyangzhao@pku.edu.cn and weilin@math.pku.edu.cn

Abstract

Federated learning, where algorithms are trained across multiple decentralized devices without sharing local data, is increasingly popular in distributed machine learning practice. Typically, a graph structure G exists behind local devices for communication. In this work, we consider parameter estimation in federated learning with heterogeneity in communication and data distribution, coupled with a limited computational capacity of local devices. We encode the distribution heterogeneity by parametrizing distributions on local devices with a set of distinct p-dimensional vectors. We then propose to jointly estimate parameters of all devices under the M-estimation framework with the fused Lasso regularization, encouraging an equal estimate of parameters on connected devices in G. We provide a general statistical guarantee for our estimator, which can be further calibrated to obtain convergence rates for various specific problem setups. Surprisingly, our estimator attains the optimal rate under certain graph fidelity condition on G, as if we could aggregate all samples sharing the same distribution. If the graph fidelity condition is not met, we propose an edge selection procedure via multiple testing to ensure the optimality. To ease the burden of local computation, a decentralized stochastic version of ADMM is provided, with convergence rate $O(T^{-1}\text{Log}\,T)$ where T denotes the number of iterations. We highlight that our algorithm transmits only parameters along edges of G at each iteration, without requiring a central machine, which preserves privacy. To address communication heterogeneity, we further extend it to the case where devices are randomly inaccessible during the training process, with a similar algorithmic convergence guarantee. The computational and statistical efficiency of our method is evidenced by simulation experiments and the 2020 US presidential election data set.

1. Problem setup

(i) *Heterogeneous data distribution.* The target parameter $\boldsymbol{\theta}_u^*$ of the local device u is defined as the unique minimizer

$$\boldsymbol{\theta}_u^* = \arg\min_{\boldsymbol{\theta}\in\Xi} M_u(\boldsymbol{\theta}) \equiv \mathbb{E}\{m_u(\mathbf{z};\boldsymbol{\theta})\}, \qquad \forall u \in V,$$

where the expectation is taken with respect to the distribution of local samples. To model distribution heterogeneity, we introduce the *characteristic graph* G_0 such that device i and device j are connected in G_0 *if and only if* $\boldsymbol{\theta}_i^* = \boldsymbol{\theta}_j^*$.

(ii) *Heterogeneous communication among local devices.* To avoid the expensive cost of communicating between local devices and the central server, we consider *decentralized federated learning*; specifically, we assume that a *communication graph* G is given a priori, along whose edges local devices can transmit information with negligible costs. Here, G is required to be similar but not necessarily equal to G_0; see Figure 1 for an illustration. Moreover, we allow *local devices to be off-line* randomly during the communication process.

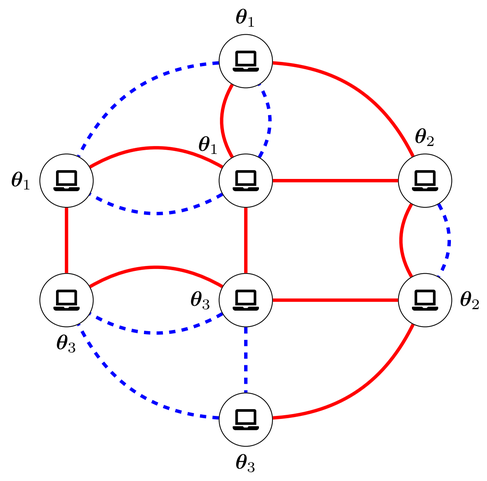

Figure 1: *Decentralized Federated Learning network illustrating communication (red solid lines) and distribution heterogeneity (blue dashed lines) among local devices with varying target parameters ($\boldsymbol{\theta}_1$–$\boldsymbol{\theta}_3$).*

(iii) *Limited computing power of local devices.* We consider the case where local devices have such a weak computing power that it will cost prohibitive time to derive local estimators. As such, our goal is to develop a *real-time* algorithm so that local devices are only required to perform simple iterations.

2. Methodology

To exploit the similarity between G and G_0, we propose the network fusion penalized estimator

$$\widehat{\boldsymbol{\Theta}} = (\widehat{\boldsymbol{\theta}}_u : u \in V) = \arg\min_{\boldsymbol{\Theta}} F(\boldsymbol{\Theta}) = \frac{1}{|V|}\sum_{u\in V} \widehat{M}_u(\boldsymbol{\theta}_u) + \lambda R(\boldsymbol{D\Theta}), \tag{1}$$

where $\boldsymbol{\Theta} = (\boldsymbol{\theta}_u : u \in V) \in \mathbb{R}^{|V|\times p}$, $\boldsymbol{D\Theta} = (\boldsymbol{\theta}_i - \boldsymbol{\theta}_j : (i,j) \in E) \in \mathbb{R}^{|E|\times p}$, λ is a tuning parameter, and $R(\boldsymbol{D\Theta}) = \sum_{(i,j)\in E}\phi(\boldsymbol{\theta}_i - \boldsymbol{\theta}_j)$ with $\phi(\cdot)$ being a norm defined on $\mathbb{R}^p$. The fused lasso regularization $R(\boldsymbol{D\Theta})$ encourages an *equal* estimate for devices connected in G. When G deviates significantly from G_0, such an estimator may be erroneous. We then propose an edge selection procedure.

(i) *Edge selection procedure.* Since $(i,j) \in E_0 = E(G_0)$ is equivalent to $\boldsymbol{\theta}_i^* = \boldsymbol{\theta}_j^*$, we consider the simultaneous testing of the following null hypotheses $H_{0,(i,j)}\colon \boldsymbol{\theta}_i^* = \boldsymbol{\theta}_j^*$ versus $H_{1,(i,j)}\colon \boldsymbol{\theta}_i^* \neq \boldsymbol{\theta}_j^*, (i,j) \in E$. We construct a test statistic via $\widehat{\mathsf{W}}_{(i,j)} = \left\{\left(\widehat{\boldsymbol{\theta}}_i^{\text{loc}} - \widehat{\boldsymbol{\theta}}_j^{\text{loc}}\right)^T\left(\widehat{\boldsymbol{\Omega}}_i + \widehat{\boldsymbol{\Omega}}_j\right)^{-1}\left(\widehat{\boldsymbol{\theta}}_i^{\text{loc}} - \widehat{\boldsymbol{\theta}}_j^{\text{loc}}\right)\right\}^{1/2}$, where $\widehat{\boldsymbol{\Omega}}_i = \{n_i\widehat{\mathbf{H}}_i(\widehat{\boldsymbol{\theta}}_i^{\text{loc}})\}^{-1}$ denotes the asymptotic variance of $\widehat{\boldsymbol{\theta}}_i^{\text{loc}}$. Adopting Bonferroni correction, we select $E \cap E_0$ by

$$\widehat{E} = \left\{(i,j) \in E : |\widehat{\mathsf{W}}_{(i,j)}|^2 \le \chi_p^2(\alpha/|E|)\right\}, \tag{2}$$

where $\chi_p^2(\alpha)$ is the upper α-quantile of the χ_p^2 distribution. The edge selection procedure can eliminate wrong edges in $E \setminus E_0$, and thus ensure the global estimator in (1) achieve the *optimal* performance.

(ii) *Decentralized real-time optimization procedure.* To solve the optimization problem (1), we consider the augmented Lagrangian defined as

$$\begin{aligned} L(\boldsymbol{\Theta},\mathbf{B},\mathbf{A}) = &\frac{1}{|V|}\sum_{i\in V}\widehat{M}_i(\boldsymbol{\theta}_i) + \lambda\sum_{(i,j)\in E}\phi(\boldsymbol{\beta}_{ij} - \boldsymbol{\beta}_{ji}) \\ &- \sum_{(i,j)\in E}\left\{\boldsymbol{\alpha}_{ij}^T(\boldsymbol{\theta}_i - \boldsymbol{\beta}_{ij}) + \boldsymbol{\alpha}_{ji}^T(\boldsymbol{\theta}_j - \boldsymbol{\beta}_{ji})\right\} + \frac{\rho}{2}\sum_{(i,j)\in E}\left\{\|\boldsymbol{\theta}_i - \boldsymbol{\beta}_{ij}\|_2^2 + \|\boldsymbol{\theta}_j - \boldsymbol{\beta}_{ji}\|_2^2\right\}, \end{aligned} \tag{3}$$

where $\mathbf{B} = (\boldsymbol{\beta}_{ij}, \boldsymbol{\beta}_{ji} : (i,j) \in E)$ and $\mathbf{A} = (\boldsymbol{\alpha}_{ij}, \boldsymbol{\alpha}_{ji} : (i,j) \in E)$. To accommodate the *weak computing power of local devices*, we adopt one-step stochastic gradient update in the t-th iteration:

$$\boldsymbol{\theta}_i(t+1) = \boldsymbol{\theta}_i(t) - \eta(t)\left\{\widetilde{\mathbf{g}}_i(t) + \rho\sum_{j\in N_i}\left(\boldsymbol{\theta}_i(t) - \boldsymbol{\beta}_{ij}(t) - \rho^{-1}\boldsymbol{\alpha}_{ij}(t)\right)\right\}, \tag{4}$$

where $\widetilde{\mathbf{g}}_i(t) = |\mathcal{B}_i(t)|^{-1}\sum_{b\in\mathcal{B}_i(t)}\boldsymbol{\psi}_i(\mathbf{z}_b^{(i)};\boldsymbol{\theta}_i(t))$, $\eta(t)$ denotes the learning rate, $\mathcal{B}_i(t)$ denotes the mini-batch randomly sampled from $\{\mathbf{z}_k^{(i)}\}_{k=1}^{n_i}$ on device i in the t-th iteration, and $\widetilde{\mathbf{g}}_i(t)$ is an unbiased estimator of $\nabla_{\boldsymbol{\theta}}\widehat{M}_i(\boldsymbol{\theta})$ evaluated at $\boldsymbol{\theta}_i(t)$. Except for local samples, the update equation (4) only requires $\boldsymbol{\beta}_{ij}(t)$ and $\boldsymbol{\alpha}_{ij}(t)$ which can be transmitted from device j. Thus, (4) *can be executed in parallel for all devices.* With $\boldsymbol{\theta}_i(t+1), i \in V$ at hand, we then update $\boldsymbol{\beta}_{ij}(t)$ and $\boldsymbol{\beta}_{ji}(t)$ by

$$\begin{aligned} \begin{pmatrix}\boldsymbol{\beta}_{ij}(t+1)\\ \boldsymbol{\beta}_{ji}(t+1)\end{pmatrix} = \arg\min_{\boldsymbol{\beta}_{ij},\boldsymbol{\beta}_{ji}}\Big\{&\lambda\phi(\boldsymbol{\beta}_{ij} - \boldsymbol{\beta}_{ji}) \\ &+\frac{\rho}{2}\left(\|\boldsymbol{\theta}_i(t+1) - \boldsymbol{\beta}_{ij} - \rho^{-1}\boldsymbol{\alpha}_{ij}(t)\|_2^2 + \|\boldsymbol{\theta}_j(t+1) - \boldsymbol{\beta}_{ji} - \rho^{-1}\boldsymbol{\alpha}_{ji}(t)\|_2^2\right)\Big\}. \end{aligned} \tag{5}$$

The update equation (5) can be implemented on either device i or device j, as long as $(\boldsymbol{\theta}_j(t+1), \boldsymbol{\beta}_{ji}(t), \boldsymbol{\alpha}_{ji}(t))$ or $(\boldsymbol{\theta}_i(t+1), \boldsymbol{\beta}_{ij}(t), \boldsymbol{\alpha}_{ij}(t))$ is transmitted to the corresponding device. *For specific choice of $\phi(\cdot)$, for example, $\phi(\cdot) = \|\cdot\|_1$ and $\phi(\cdot) = \|\cdot\|_2$, we can obtain an explicit update equation from* (5). Finally, we update $\boldsymbol{\alpha}_{ij}(t)$ and $\boldsymbol{\alpha}_{ji}(t)$ by

$$\begin{pmatrix}\boldsymbol{\alpha}_{ij}(t+1)\\ \boldsymbol{\alpha}_{ji}(t+1)\end{pmatrix} = \begin{pmatrix}\boldsymbol{\alpha}_{ij}(t)\\ \boldsymbol{\alpha}_{ji}(t)\end{pmatrix} - \rho\begin{pmatrix}\boldsymbol{\theta}_i(t+1) - \boldsymbol{\beta}_{ij}(t+1)\\ \boldsymbol{\theta}_j(t+1) - \boldsymbol{\beta}_{ji}(t+1)\end{pmatrix}. \tag{6}$$

Notice that update equation (6) also only requires parameter communication among connected devices. Both (5) and (6) can be performed in parallel across edges.

We refer to (4) *as node optimization step, and* (5) *and* (6) *as edge communication step.*

3. Assumptions and theoretic guarantees

We maintain the following regularity assumptions.

(A1) *Identifiability.* $\mathbb{E}\{m_u(\mathbf{z};\boldsymbol{\theta})\}$ is convex and twice differentiable with respect to $\boldsymbol{\theta}$, the Hessian matrix $\mathbf{H}_u(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}}^2\mathbb{E}\{m_u(\mathbf{z};\boldsymbol{\theta})\}$ is Lipschitz continuous at $\boldsymbol{\theta}_u^*$.

(A2) *Bounded conditional number.* The conditional number of $\widehat{\mathbf{H}}_u(\boldsymbol{\theta})$ is bounded by κ.

(A3) *Compatibility factor.* For $S = E \setminus E_0 \neq \emptyset$, $\kappa_S(\mathbf{D}) \equiv \inf_{\boldsymbol{\Theta}}\frac{\sqrt{|S|}\|\boldsymbol{\Theta}\|_F}{R\{(\mathbf{D\Theta})_S\}} \ge \kappa_0 > 0$, where $R\{(\mathbf{D\Theta})_S\} = \sum_{(u,v)\in S}\phi(\boldsymbol{\theta}_u - \boldsymbol{\theta}_v)$.

Based on Assumptions (A1)–(A3), we obtain the following *non-asymptotic bounds* regarding both statistical and algorithmic errors.

(i) *Deterministic estimation bound.* For noises from *any* distribution, the penalized M-estimator $\widehat{\boldsymbol{\Theta}}$ satisfies

$$\frac{1}{|V|}\|\widehat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}^*\|_F^2 \le 2\kappa^2\left(\rho^2 + \frac{4|S|}{\kappa_0}\lambda^2\right),$$

where $\rho = \frac{1}{\sqrt{|V|}}\|\Pi_{\text{Ker}(\mathbf{D})}\widehat{\boldsymbol{\Psi}}(\boldsymbol{\Theta}^*)\|_F$, and $\lambda = \frac{1}{\sqrt{|V|}}R^*\{(\mathbf{D}^+)^T\widehat{\boldsymbol{\Psi}}(\boldsymbol{\Theta}^*)\}$, $\widehat{\boldsymbol{\Psi}}$ is the gradient of the empirical risk function and $R^*(\cdot)$ denotes the Fréchet dual of $R(\cdot)$.

(ii) *Probabilistic estimation bound.* If the score function $\boldsymbol{\psi}_u(\mathbf{z}_i^{(u)};\boldsymbol{\theta}_u^*) = \nabla_{\boldsymbol{\theta}}m_u(\mathbf{z}_i^{(u)};\boldsymbol{\theta}_u^*)$ is sub-Gaussian with parameter σ^2, then we obtain

$$\frac{1}{|V|}\|\widehat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}^*\|_F^2 = O_p\left\{\frac{\sigma^2}{\kappa_0}\left(\frac{pK(G)}{n|V|} + \frac{p|E\setminus E_0|}{n|V|}\right)\right\},$$

where $K(G)$ is the number of connected components of G. We remark that the optimal estimation error scales as $O_p\left(\frac{pK(E_0)}{n|V|}\right)$.

(iii) *Edge selection ensures the optimality of the network fusion estimator.* Under mild conditions, the output of the edge selection procedure maximizes the *graph fidelity* $\mathsf{GF}_{G_0}(G) \equiv \frac{K(E_0)}{K(E)+|E\setminus E_0|}$.

(iv) *Algorithmic convergence.* Under mild conditions, by choosing $\eta(t) = \kappa/t$, we have

$$\frac{1}{|V|}\mathbb{E}\left(\|\overline{\boldsymbol{\Theta}} - \widehat{\boldsymbol{\Theta}}\|_F^2\Big|\,\mathbf{z}_k^{(u)}, 1 \le k \le n_u, u \in V\right) \le \frac{2\kappa^2 C_\psi\text{Log}\,T}{T},$$

for sufficiently large T such that $\kappa C_\psi|V|\text{Log}\,T \ge C|E|$ for some constant C, where the expectation is taken with respect to the choice of mini-batches $\{\mathcal{B}_u(t)\colon u \in V, 1 \le t \le T\}$.

4. Selected simulation

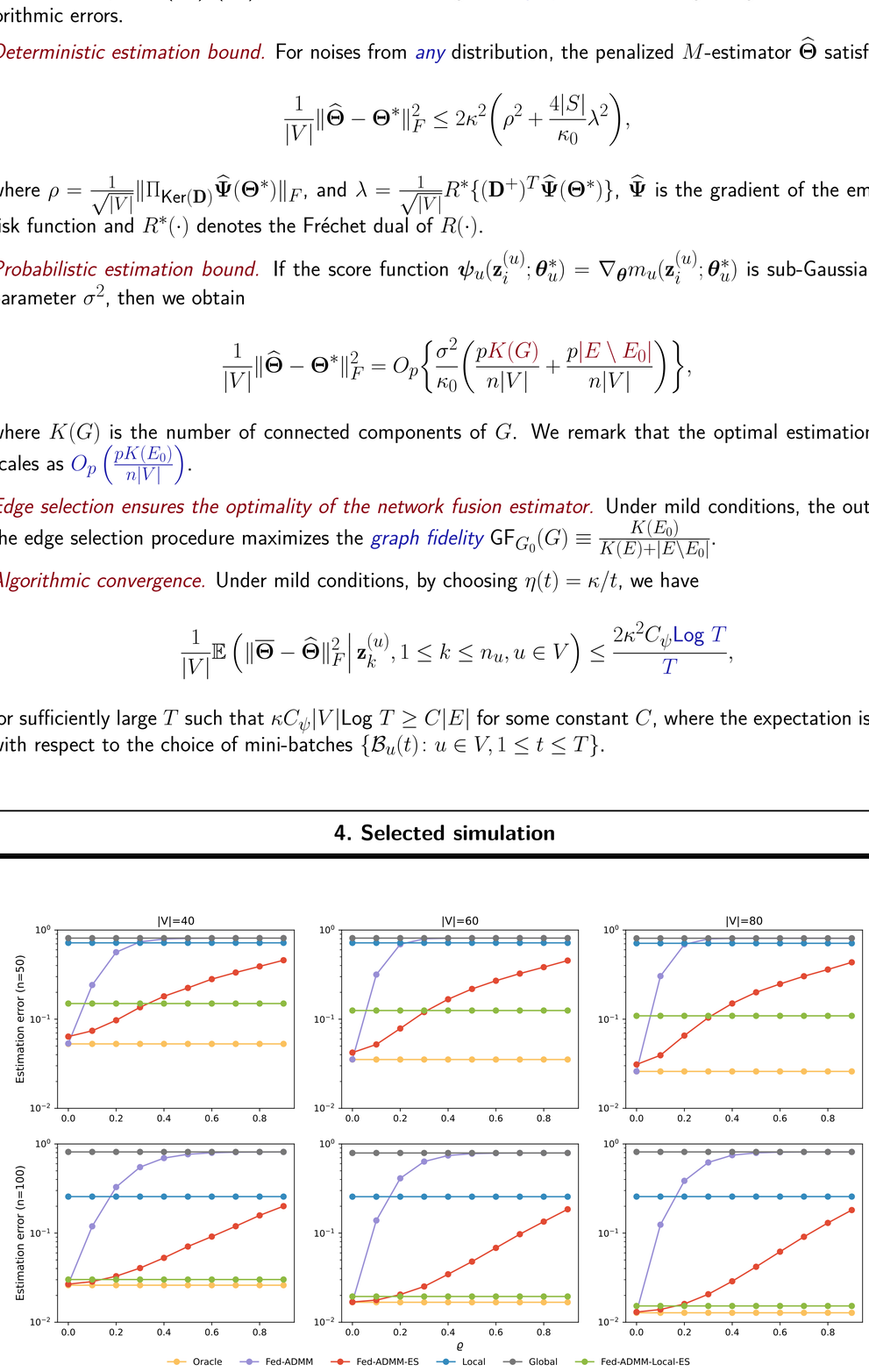

Figure 2: *Sensitivity to graph corruption. G is obtained by randomly flipping the status of edge in G_0 with probability ϱ. Fed-ADMM-ES is obtained by performing the edge selection procedure on G and Fed-ADMM-Local-ES is obtained by performing the edge selection procedure on fully connected graph.*