A Federated Algorithm for Privacy-Preserving Empirical Risk Minimization

Zonghao Huang
ZONGHAO.HUANG@OKSTATE.EDU
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING
OKLAHOMA STATE UNIVERSITY

Editor: Zonghao Huang

1. Problem Statement

We consider a group of data owners $[m] = 1, 2, \ldots, m$, who collaboratively learn a model over their private datasets. We use $D_j = \{(x_i^{(j)}, y_i^{(j)}), i = 1, 2, \ldots, n_j\}$ to denote the dataset possessed by data owner j, where $x_i^{(j)}$ is the data feature vector and $y_i^{(j)}$ is the corresponding data label.

The goal of our problem is to learn a model over the aggregated dataset $\{D_j\}_{j\in[m]}$, which is formulated as a regularized empirical risk minimization:

$$\min_{\theta} J(\theta) = \min_{\theta} \frac{1}{m} \sum_{j=1}^{m} J_j(\theta) = \min_{\theta} \frac{1}{m} \sum_{j=1}^{m} \frac{1}{n_j} \sum_{i=1}^{n_j} \ell(\theta, x_i^{(j)}, y_i^{(j)}) + \lambda N(\theta). \tag{1}$$

2. Our Algorithm

Following the previous work (Jayaraman et al., 2018), our algorithm is based on the gradient-based iterative learning with calibrated noise under secure aggregation protocol. Instead of just computing the gradient, our algorithm allows the data owners to update their local model with calibrated noise, and aggregates the local models in every K iterations, which is similar to the federated learning framework proposed by (McMahan et al., 2016).

Our iterative approach is performed under a secure aggregation protocol. The secure aggregation protocol makes our approach work even if there are some dropping-off users during the communication phase, which is similar to that one proposed by (Bonawitz et al., 2017). Differently, my secure aggregation protocol requires only a small fraction of users to communicate with the server for aggregation. When the total number of users is large, the requirement of all nodes' participation in the aggregation will suffers from the serious "straggler's effect" (the server needs to wait for the slowest node). Thus, my secure aggregation protocol requires partial node participation. Such partial node participation requirement also improves the communication efficiency. The details of our algorithm are shown in Algorithm 1. The description of the secure aggregation protocol is introduced in Section 3.

Theorem 1 (Privacy Guarantee) *Given a estimator $\bar{\theta}^T$ obtained by Algorithm 1 involving a class of T gradient descent-based updates perturbed with noise $z \in N(0, \sigma^2 \mathrm{I})$. Assumed $\ell(\theta)$ is G-Lipschitz over $\theta \in C$. The estimator $\bar{\theta}^T$ is (ϵ, δ)-differentially private if we set:*

$$\sigma^2 = \frac{8G^2T\log(1/\delta)}{m^2 n_{(1)}^2 \epsilon^2}. \tag{2}$$

Proof We can prove the theorem by following the proof of Theorem 3.4 in (Jayaraman et al., 2018). ■

Algorithm 1 Differentially Private Federated Algorithm Under Secure Aggregation

Initialize $\{\theta_i^0\}$ and $\{\bar{\theta}_i^0\}$;
for $t = 1, 2, \ldots, T$ **do**
  for $j = 1, 2, \ldots, m$ **do**
    $\theta_j^t = \bar{\theta}_j^{t-1} - \eta_t(\nabla J_j(\bar{\theta}_j^{t-1}) + z)$, $z \sim N(0, \sigma^2 \mathrm{I})$;
  end for
  if $t \in \{nK, n = 1, 2, \ldots\}$ **then**
    Sample randomly E users without replacement (denoted by S_t);
    Perform $\theta = \frac{1}{E}\sum_{j \in S_t} \theta_j^t$ under a secure aggregation protocol;
    for $j \in S_t$ **do**
      $\bar{\theta}_j^t = \theta$;
    end for
  end if
end for

3. Secure Aggregation Protocol

Here we consider a secure aggregation protocol to reduce the privacy leakage in the iterative learning process, and is robust to dropping-off nodes during the communication, which is similar to that one proposed by (Bonawitz et al., 2017). As mentioned in the last section, to reduce the "straggler's effect" and improve the communication-efficiency, the server only communicates with a small fraction of users (E users) randomly during the aggregation phase. In our protocol, we define the following cryptographic primitives:

- Secret Sharing: (**SS.share**, **SS.reconstruct**).
- Key Agreement: (**KA.paramGen**, **KA.gen**, **KA.agree**).
- Authenticated Encryption: (**AE.enc**, **AE.dec**).
- Pseudorandom Generator: (**PRG**).
- Signature Scheme: (**SIG.gen**, **SIG.sign**, **SIG.ver**).

A detailed description of our secure summation protocol is shown as follows:

- **Setup**:
  - All the users and the server are given the security parameter k, the number of users m, the sampling number E, and a threshold Th, honestly generated $pp \leftarrow \mathbf{KA.gen}(k)$, parameters h and R such that $\mathbb{Z}_R^h$ is the space from which inputs are sampled, and a field $\mathbb{F}$ to be used for secret sharing. All users also have a private authenticated channel with the server.
  - All the users u receive their signing keys d_u^{SK} and the verification keys d_v^{PK} bound to other users pseudo-identity v (to ensure that the server samples the users randomly).

- **AdvertiseKeys**:
  User u:
  - Generate key pairs: $(c_u^{SK}, c_u^{PK}) \leftarrow \mathbf{KA.gen}(pp)$ and $(s_u^{SK}, s_u^{PK}) \leftarrow \mathbf{KA.gen}(pp)$, and generate $\sigma_u \leftarrow \mathbf{SIG.sign}(d_u^{SK}, c_u^{PK} \| s_u^{PK})$.
  - Send $(c_u^{PK} \| s_u^{PK} \| \sigma_u)$ to the server.

  Server:
  - Collect at least E messages from individual users from the previous round (denote with $\mathcal{U}_1$ this set of users). Otherwise, abort.
  - Sample E_0 ($E_0 \geq E$) users randomly denoted by $\mathcal{U}_2 \subseteq \mathcal{U}_1$ (The sampling aims to reduce the communication cost, E_0 should be larger than E to allow dropping-off users).
  - Broadcast $\{(v, c_v^{PK}, s_v^{PK}, \sigma_v)\}_{v \in \mathcal{U}_2}$ to all the users in the list $\mathcal{U}_2$.

- **ShareKeys**:
  User u:
  - Receive the list $\{(v, c_v^{PK}, s_v^{PK}, \sigma_v)\}_{v \in \mathcal{U}_2}$ broadcasted by the server. Assert that $|\mathcal{U}_2| \geq Th$, that all the public key pairs are different, and that $\forall v \in \mathcal{U}_2$, $\mathbf{SIG.ver}(d_v^{PK}, c_v^{PK} \| s_v^{PK}, \sigma_u) = 1$.
  - Sample a random element b_u (to be used as a seed for a **PRG**).
  - Generate Th-out-of-$|\mathcal{U}_2|$ shares of $s_u^{SK} : \{(v, s_{u,v}^{SK})\}_{v \in \mathcal{U}_2} \leftarrow \mathbf{SS.share}(s_u^{SK}, Th, \mathcal{U}_2)$.
  - Generate Th-out-of-$|\mathcal{U}_2|$ shares of $b_u : \{(v, b_{u,v})\}_{v \in \mathcal{U}_2} \leftarrow \mathbf{SS.share}(b_u, Th, \mathcal{U}_2)$.
  - For each other user $v \in \mathcal{U}_2 \setminus \{u\}$, compute $e_{u,v} \leftarrow \mathbf{AE.enc}(\mathbf{KA.agree}(c_u^{SK}, c_v^{PK}), u \| v \| \| s_{u,v}^{SK} \| b_{u,v})$.
  - If any of the above operations fails, abort.
  - Send all the ciphertexts $e_{u,v}$ to the server.
  - Store all messages received and values generated in this round.

  Server:
  - Collect lists of ciphertexts $e_{u,v}$ from at least E users (denote with $\mathcal{U}_3 \subseteq \mathcal{U}_2$ this set of users).

– Send to each user $u \in \mathcal{U}_3$ all the ciphertexts encrypted for it: $\{e_{u,v}\}_{v \in \mathcal{U}_3}$.

- **MaskedInputCollection**:
  User u:

  – Receive from the server the list of ciphertexts $\{e_{u,v}\}_{v \in \mathcal{U}_3}$. If the list is of size $< Th$, abort.

  – For each other user $v \in \mathcal{U}_3 \setminus \{u\}$, compute $s_{u,v} \leftarrow$ **KA.agree**(s_u^{SK}, s_v^{PK}) and expand this value using a **PRG** into a random vector $\boldsymbol{p}_{u,v} = \triangle_{u,v} \cdot$**PRG**$(s_{u,v})$, where $\triangle_{u,v} = 1$ when $u > v$, and $\triangle_{u,v} = -1$ when $u < v$. Additionally, define $\boldsymbol{p}_{u,u} = 0$.

  – Compute the user's own private mask vector $\boldsymbol{p}_u =$ **PRG**(b_u). Then, Compute the masked input vector $\boldsymbol{y}_u \leftarrow \boldsymbol{x}_u + \boldsymbol{p}_u + \sum_{v \in \mathcal{U}_3} \boldsymbol{p}_{u,v} (\text{mod } R)$.

  – If any of the above operations fails, abort. Otherwise, send $\boldsymbol{y}_u$ to the server.

  Server:

  – Collect $\boldsymbol{y}_u$ from at least E users (denote with $\mathcal{U}_4 \subseteq \mathcal{U}_3$ this set of users).

  – Sample E users randomly (denote with $\mathcal{U}_5 \subseteq \mathcal{U}_4$). Send to each user in $\mathcal{U}_4$ the list $\mathcal{U}_5$ ($\mathcal{U}_5$ is the list of users participating in aggregation).

- **ConsistencyCheck**:
  User u:

  – Receive from the server a list $\mathcal{U}_5 \subseteq \mathcal{U}_3$ consisting E users.If $\mathcal{U}_5$ is more than or less than E, abort.

  – Send to the server $\sigma_u^{'} \leftarrow$ **SIG.sign**$(d_u^{SK}, \mathcal{U}_5)$.

  Server:

  – Collect $\sigma_u^{'}$ from at least Th users (denote with $\mathcal{U}_6 \subseteq \mathcal{U}_4$). Send to each user in $\mathcal{U}_6$ the set $\{v, \sigma_v^{'}\}_{v \in \mathcal{U}_6}$.

- **Unmasking**:
  User u:

  – Receive from the server a list $\{v, \sigma_v^{'}\}_{v \in \mathcal{U}_4}$. Verify that $\mathcal{U}_6 \subseteq \mathcal{U}_4$, $|\mathcal{U}_4| \geq t$ and that **SIG.ver**$(d_u^{PK}, \mathcal{U}_5, \sigma_v^{'}) = 1$ for all $v \in \mathcal{U}_6$ (otherwise abort).

  – For each other user v in $\mathcal{U}_3 \setminus \{u\}$, decrypt the ciphertext $v^{'} \| u^{'} \| s_{u,v}^{SK} \| b_{u,v} \leftarrow$ **AE.dec**(**KA.agree**$(c_u^{SK}, c_v^{PK}), e_{v,u})$ received in the **MaskedInputCollection** round and assert that $u = u^{'} \wedge v = v^{'}$.

  – If any of the decryption operation fail, abort.

  – Send a list of shares to the server, which consists of $s_{v,u}^{SK}$ for users $v \in \mathcal{U}_3 \setminus \mathcal{U}_5$ and $b_{u,v}$ for users in $v \in \mathcal{U}_5$.

  Server (generating the output from E users):

– Collect responses from at least Th users (denote with $\mathcal{U}_7$ this set of users).
– For each user in $u \in \mathcal{U}_5 \setminus \mathcal{U}_3$, reconstruct $s_u^{SK} \leftarrow \textbf{SS.recon}(\{s_{u,v}^{SK}\}_{v\in\mathcal{U}_7}, Th)$ and use it (together with the public keys received in the **AdvertiseKeys** round) to recompute $\boldsymbol{p}_{v,u}$ for all $v \in \mathcal{U}_5$ using the PRG.
– For each user $u \in \mathcal{U}_5$, reconstruct $b_u \leftarrow \textbf{SS.recon}(\{b_{u,v}\}_{v\in\mathcal{U}_7}, t)$ and then recompute $\boldsymbol{p}_u$ using the PRG.
– Compute and output $\boldsymbol{O} = \sum_{u\in\mathcal{U}_5} \boldsymbol{x}_u$ as $\sum_{u\in\mathcal{U}_5} \boldsymbol{x}_u = \sum_{u\in\mathcal{U}_5} \boldsymbol{y}_u - \sum_{u\in\mathcal{U}_5} \boldsymbol{p}_u + \sum_{u\in\mathcal{U}_5, v\in\mathcal{U}_3\setminus\mathcal{U}_5} \boldsymbol{p}_{v,u}$.

4. Theoretical Result

In this section, we will give the utility analysis of our algorithm by the excess empirical risk. We use $\bar{\theta}^t$ to denote $\frac{1}{m}\sum_{j=1}^m \bar{\theta}_j^t$, and use θ^t to denote $\frac{1}{m}\sum_{j=1}^m \theta_j^t$. We also use $\mathbb{E}[\cdot]$ to denote the expectation over S_t and z.

Lemma 2 *Assumed that $J_j(\theta)$ is λ-strongly convex, and L-smooth over $\theta \in C$. If $\eta_t \le \frac{1}{4L}$ for all t, we have:*

$$\mathbb{E}[\|\bar{\theta}^{t+1}-\theta^*\|^2] \le (1-\eta^t\lambda)\mathbb{E}[\|\bar{\theta}^t-\theta^*\|^2]+6L\eta_t^2\Gamma+2\mathbb{E}[\frac{1}{m}\sum_{j=1}^m \|\theta^t-\theta_j^t\|^2]+\eta_t^2 d\sigma^2+\mathbb{E}[\|\theta^{t+1}-\bar{\theta}^{t+1}\|^2], \tag{3}$$

where $\Gamma = J^* - \frac{1}{m}\sum_{j=1}^m J_j^*$. J^* is the minimal value of $J(\theta)$ while J_j^* is the minimal value of $J_j(\theta)$. Γ quantifies the heterogeneity degree of the data distribution.

Proof Since we have:

$$\begin{aligned}\|\bar{\theta}^{t+1} - \theta^*\|^2 =& \|\bar{\theta}^{t+1} - \theta^{t+1} + \theta^{t+1} - \theta^*\|^2 \\ =& \|\bar{\theta}^{t+1} - \theta^{t+1}\|^2 + \|\theta^{t+1} - \theta^*\|^2 + 2\langle\bar{\theta}^{t+1} - \theta^{t+1}, \theta^{t+1} - \theta^*\rangle,\end{aligned} \tag{4}$$

by taking expectation, we have:

$$\mathbb{E}[\|\bar{\theta}^{t+1} - \theta^*\|^2] = \mathbb{E}[\|\bar{\theta}^{t+1} - \theta^{t+1}\|^2] + \mathbb{E}[\|\theta^{t+1} - \theta^*\|^2]. \tag{5}$$

If $t+1 \in \{nK\}$, we have: $\bar{\theta}^{t+1} = \theta^{t+1}$. Following the proof of Lemma 1 in (Li et al., 2019), we can prove that:

$$\begin{aligned}\mathbb{E}[\|\bar{\theta}^{t+1} - \theta^*\|^2] =& \mathbb{E}[\|\theta^{t+1} - \theta^*\|^2] \\ \le& (1 - \eta^t\lambda)\mathbb{E}[\|\bar{\theta}^t - \theta^*\|^2] + 6L\eta_t^2\Gamma + 2\mathbb{E}[\frac{1}{m}\sum_{j=1}^m \|\theta^t - \theta_j^t\|^2] + \eta_t^2 d\sigma^2.\end{aligned} \tag{6}$$

If $t+1 \notin \{nK\}$, by following the proof of Lemma 1 in (Li et al., 2019), we can prove that:

$$\begin{aligned}\mathbb{E}[\|\bar{\theta}^{t+1} - \theta^*\|^2] =& \mathbb{E}[\|\theta^{t+1} - \theta^*\|^2] + \mathbb{E}[\|\bar{\theta}^{t+1} - \theta^{t+1}\|^2] \\ \le& (1 - \eta^t\lambda)\mathbb{E}[\|\bar{\theta}^t - \theta^*\|^2] + 6L\eta_t^2\Gamma + 2\mathbb{E}[\frac{1}{m}\sum_{j=1}^m \|\theta^t - \theta_j^t\|^2] \\ &+ \eta_t^2 d\sigma^2 + \mathbb{E}[\|\theta^{t+1} - \bar{\theta}^{t+1}\|^2].\end{aligned} \tag{7}$$

■

Lemma 3 (Lemma 3 and Lemma 5 in (Li et al., 2019)) *Assumed that*

$$\mathbb{E}[\|\nabla J_j(\theta)\|^2] \leq H^2. \tag{8}$$

If η_t is non-increasing and $\eta_t \leq 2\eta_{t+K}$ for all t, we have:

$$\mathbb{E}[\frac{1}{m}\sum_{j=1}^{m}\|\theta^t - \theta_j^t\|^2] \leq 4\eta_t^2(K-1)^2H^2, \tag{9}$$

and

$$\mathbb{E}[\|\theta^{t+1} - \bar{\theta}^{t+1}\|^2] \leq \frac{m-E}{m-1}\frac{4}{E}\eta_t^2K^2H^2. \tag{10}$$

Theorem 4 (Utility Analysis) *Given the estimator θ^T obtained by Algorithm 1 where $J_j(\theta)$ is G-Lipschitz, λ-strongly convex, and L-smooth over $\theta \in C$. If we choose the learning rate $\eta_t = \frac{2}{\lambda(t+\gamma)}$ and $\gamma = \max\{\frac{8L}{\lambda} - 1, K\}$, and the gradients are perturbed with noise $z \in N(0, \sigma^2\mathrm{I})$ with σ^2 defined by Eq. (2), we have the following excess empirical risk:*

$$\mathbb{E}[J(\bar{\theta}^T)] - J(\theta^*) \leq \frac{2L}{\lambda(\gamma+T)}(\frac{B}{\lambda} + 2L\|\theta^0 - \theta^*\|^2), \tag{11}$$

where $B = 6L\Gamma + 8(K-1)^2G^2 + d\frac{8G^2T\log(1/\delta)}{m^2n_{(1)}^2\epsilon^2} + \frac{m-E}{m-1}\frac{4}{E}K^2G^2$.

Proof We follow (Li et al., 2019) to give the proof. Since $J_j(\theta)$ is G-Lipschitz, thus $\mathbb{E}[\|\nabla J_j(\theta)\|^2] \leq G^2$. Based on Lemma 2 and Lemma 3, we have:

$$\mathbb{E}[\|\bar{\theta}^{t+1} - \theta^*\|^2] \leq (1-\eta^t\lambda)\mathbb{E}[\|\bar{\theta}^t - \theta^*\|^2] + \eta_t^2B, \tag{12}$$

where $B = 6L\Gamma + 8(K-1)^2G^2 + d\frac{8G^2T\log(1/\delta)}{m^2n_{(1)}^2\epsilon^2}$.

By setting $\eta_t = \frac{2}{\lambda(t+\gamma)}$ where $\gamma = \max\{\frac{8L}{\lambda} - 1, K\}$ to enforce that $\eta_t \leq \frac{1}{4L}$ and $\eta_t \leq 2\eta_{t+E}$, we can prove that $\mathbb{E}[\|\theta^t - \theta^*\|^2] \leq \frac{v}{\lambda+t}$ where $v = \max\{\frac{4B}{\lambda^2}, (\gamma+1)\|\theta^0 - \theta^*\|^2\}$. By the smoothness of $J(\theta)$, we have:

$$\mathbb{E}[J(\theta^T)] - J(\theta^*) \leq \frac{L}{2}\|\theta^T - \theta^*\|^2 \leq \frac{Lv}{2(\lambda+T)} \leq \frac{2L}{\lambda(\gamma+T)}(\frac{B}{\lambda} + 2L\|\theta^0 - \theta^*\|^2). \tag{13}$$

■

5. Discussion

As shown in Theorem 4, a larger K (iteration numbers between two communications) and a smaller E (number of users for aggregation) will reduce the communication cost, but also brings utility loss ($\frac{16L(K-1)^2G^2}{\lambda^2(\gamma+T)}$ and $\frac{2L}{\lambda^2(\gamma+T)}\frac{m-E}{m-1}\frac{4}{E}K^2G^2$).

References

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191. ACM, 2017.

Bargav Jayaraman, Lingxiao Wang, David Evans, and Quanquan Gu. Distributed learning without distress: Privacy-preserving empirical risk minimization. In *Advances in Neural Information Processing Systems*, pages 6343–6354, 2018.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.