Federated Multi-Task Learning under a Mixture of Distributions

Othmane MARFOQ [1,2,3]   Giovanni Neglia [1,2]   Aurélien Bellet [1]   Laetitia Kameni [3]   Richard Vidal [3]

[1]Inria   [2]Université Côte d'Azur   [3]Accenture Labs

Introduction

- A (countable) set $\mathcal{T}$ of classification (or regression) tasks which represent the set of possible clients.
- Data at client $t \in \mathcal{T}$ is drawn from a local distribution $\mathcal{D}_t$ over $\mathcal{X} \times \mathcal{Y}$.
- Client $t \in \mathcal{T}$ wants to learn a hypothesis h_t minimizing its true risk, i.e.,

$$\underset{h_t \in \mathcal{H}}{\text{minimize}}\ \mathcal{L}_{\mathcal{D}_t}(h_t) \triangleq \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_t}\left[l\left(h_t\left(\mathbf{x}\right), y\right)\right]. \quad (1)$$

- Client $t \in \mathcal{T}$ does not have access to the distribution $\mathcal{D}_t$. Instead, it has a access to n_t samples drawn i.i.d. from $\mathcal{D}_t$, denoted

$$\mathcal{S}_t = \{s_t^{(i)} \triangleq (\mathbf{x}_t^{(i)},\ y_t^{(i)})\}_{i=1}^{n_t}. \quad (2)$$

- Usually $n_t \ll n \triangleq \sum_{t\in\mathcal{T}} n_t$, thus collaboration among clients is needed in order to train better models.

An impossibility result

Some assumption on the local data distributions $\mathcal{D}_t,\ t \in \mathcal{T}$ are needed for federated learning to be possible:

- Federated learning with T clients is equivalent to T *semi-supervised learning* (SSL) problems, where the SSL problem associated with client t relies on labeled samples in $\mathcal{S}_t$ and unlabeled samples in

$$\mathcal{U}_t = \bigcup_{t' \in [T]\setminus\{t\}} \{\mathbf{x} : (\mathbf{x}, y) \in \mathcal{S}_{t'}\}.$$

- Even when the quantity of unlabeled data goes to infinity, the worst-sample complexity of SSL improves over supervised learning at most by a constant factor that only depends on the hypothesis class [1, 2, 3].

Main assumptions

Motivated by the above impossibility result, in this work we propose to consider that each local data distribution $\mathcal{D}_t$ is a mixture of M underlying distributions $\tilde{\mathcal{D}}_m,\ 1 \le m \le M$, as formalized below.

Assumption 1. There exist M underlying (independent) distributions $\tilde{\mathcal{D}}_m,\ 1 \le m \le M$, such that for $t \in \mathcal{T}$, $\mathcal{D}_t$ is mixture of the distributions $\{\tilde{\mathcal{D}}_m\}_{m=1}^M$ with weights $\pi_t^* = [\pi_{t1}^*, \dots, \pi_{tM}^*] \in \Delta^M$, i.e.

$$z_t \sim \mathcal{M}(\pi_t^*), \quad ((\mathbf{x}_t, y_t)\,|z_t = m) \sim \tilde{\mathcal{D}}_m, \quad \forall t \in \mathcal{T}, \quad (3)$$

where $\mathcal{M}(\pi)$ is a multinomial (categorical) distribution with parameters π.

Assumption 2. For all $m \in [M]$, we have $\tilde{\mathcal{D}}_m(\mathbf{x}) = \mathcal{D}(\mathbf{x})$.

Assumption 3. $\tilde{\mathcal{H}} = \{h_\theta\}_{\theta\in\mathbb{R}^d}$ is a set of hypotheses parameterized by $\theta \in \mathbb{R}^d$, whose convex hull is in $\mathcal{H}$. For each distribution $\tilde{\mathcal{D}}_m$ with $m \in [M]$, there exists a hypothesis $h_{\theta_m^*}$, such that

$$l\left(h_{\theta_m^*}(\mathbf{x}), y\right) = -\log p_m(y|\mathbf{x}) + c, \quad (4)$$

where $c \in \mathbb{R}$, is a normalization constant. $l(\cdot,\cdot)$ is then the log loss associated to $p_m(y|\mathbf{x})$.

Remark: *The generative model in Assumption 1 extends some popular mutli-task/personalized FL formulation in the literature, including* **CLustered FL** *[6],* **Personalization via model interpolation** *[5], and* **Federated MTL via task relationships** *[7]*

Learning under a Mixture Model

Proposition Let $l(\cdot,\cdot)$ be the mean squared error loss, the logistic loss or the cross-entropy loss, and $\breve{\Theta}$ and $\breve{\Pi}$ be a solution of the following optimization problem:

$$\underset{\Theta,\Pi}{\text{minimize}}\ \underset{t\sim D_{\mathcal{T}}}{\mathbb{E}}\ \underset{(\mathbf{x},y)\sim\mathcal{D}_t}{\mathbb{E}}\left[-\log \mathcal{D}_t(\mathbf{x}, y|\Theta, \pi_t)\right], \quad (5)$$

where $D_{\mathcal{T}}$ is any distribution with support $\mathcal{T}$. Under Assumptions 1, 2, and 3, the predictors

$$h_t^* = \sum_{m=1}^M \breve{\pi}_{tm} h_{\breve{\theta}_m}(\mathbf{x}), \quad \forall t \in \mathcal{T} \quad (6)$$

minimize $\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_t}\left[l(h_t(\mathbf{x}), y)\right]$ and thus solve Problem (1).

This Proposition suggests the following approach to solve Problem (1).

- **First**, estimate $\breve{\Theta}$ and $\breve{\pi}_t,\ 1 \le t \le T$, by minimizing

$$f(\Theta, \Pi) \triangleq -\frac{\log p(\mathcal{S}_{1:T}|\Theta, \Pi)}{n} \triangleq -\frac{1}{n}\sum_{t=1}^T\sum_{i=1}^{n_t} \log p(s_t^{(i)}|\Theta, \pi_t). \quad (7)$$

- **Second**, use Eq. (6) to get the client predictor for the T clients present at training time.

Federated Expectation-Maximization

- A natural approach to solve problem (7) is via the Expectation-Maximization algorithm (EM), which alternates between two steps.

E-step:
$$q_t^{k+1}(z_t^{(i)} = m) \propto \pi_{tm}^k \cdot \exp\left(-l(h_{\theta_m^k}(\mathbf{x}_t^{(i)}), y_t^{(i)})\right), \qquad t \in [T],\ m \in [M],\ i \in [n_t] \quad (8)$$

M-step:
$$\pi_{tm}^{k+1} = \frac{\sum_{i=1}^{n_t} q_t^{k+1}(z_t^{(i)} = m)}{n_t}, \qquad t \in [T],\ \ m \in [M] \quad (9)$$

$$\theta_m^{k+1} \in \underset{\theta\in\mathbb{R}^d}{\arg\min} \sum_{t=1}^T\sum_{i=1}^{n_t} q_t^{k+1}(z_t^{(i)} = m)\, l\big(h_\theta(\mathbf{x}_t^{(i)}), y_t^{(i)}\big), \qquad m \in [M] \quad (10)$$

- While the E-step (8) and the Π update (9) can be performed locally at each client, the Θ update (10) requires interaction with other clients.
- `FedEM` updates the local estimates of Θ through a solver which approximates the exact minimization in (10) using only the local dataset $\mathcal{S}_t$.

Algorithm 1 `FedEM`

Input: data $\mathcal{S}_{1:T}$; number of mixture distributions M; number of communication rounds K
for iterations $k = 1, \dots, K$ **do**
  server broadcasts θ_m^{k-1}, $1 \le m \le M$ to the T clients
  for tasks $t = 1, \dots, T$ in parallel over T clients **do**
    for component $m = 1, \dots, M$ **do**
      update $q_t^k(z_t^{(i)} = m)$ as in (8), $\forall i \in \{1, \dots, n_t\}$
      update π_{tm}^k as in (9)
      $\theta_{m,t}^k \leftarrow \texttt{LocalSolver}(m,\ \theta_m^{k-1},\ q_t^k,\ \mathcal{S}_t)$
    end for
  end for
  client t sends $\theta_{m,t}^k$, $1 \le m \le M$, to the server
  for component $m = 1, \dots, M$ **do**
    $\theta_m^k \leftarrow \sum_{t=1}^T \frac{n_t}{n} \times \theta_{m,t}^k$
  end for
end for

Theorem When clients use SGD as local solver with learning rate $\eta = \frac{a_0}{\sqrt{K}}$, after a large enough number of communication rounds K, `FedEM`'s iterates satisfy:

$$\frac{1}{K}\sum_{k=1}^K \mathbb{E}\left\|\nabla_\Theta f\left(\Theta^k, \Pi^k\right)\right\|_F^2 \le \mathcal{O}\left(\frac{1}{\sqrt{K}}\right), \qquad \frac{1}{K}\sum_{k=1}^K \Delta_\Pi f(\Theta^k, \Pi^k) \le \mathcal{O}\left(\frac{1}{K^{3/4}}\right), \quad (11)$$

where the expectation is over the random batches samples, and

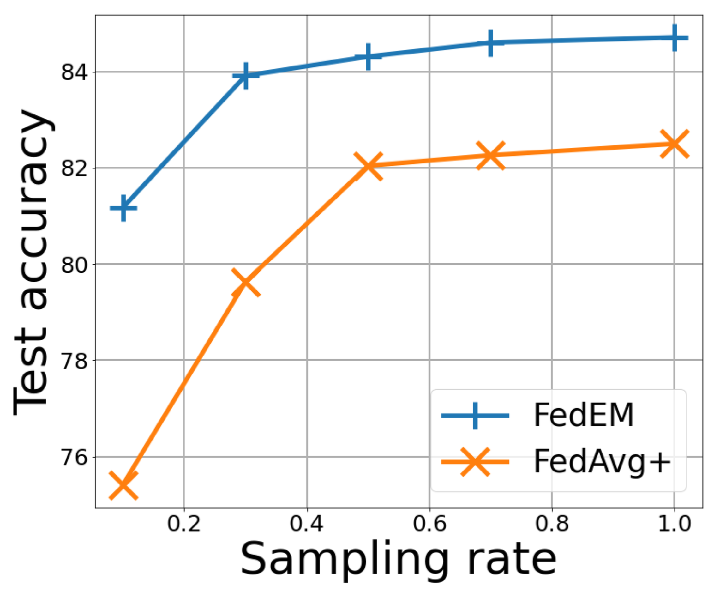
$$\Delta_\Pi f(\Theta^k, \Pi^k) \triangleq f\left(\Theta^k, \Pi^k\right) - f\left(\Theta^k, \Pi^{k+1}\right) \ge 0. \quad (12)$$

We also propose ***D-FedEM****, a fully decentralized version of our federated EM algorithm with similar convergence guarantees.*

Experiments

Dataset	Local	FedAvg	FedProx	FedAvg+	clustered FL	pFedMe	FedEM (Ours)
FEMNIST	71.0 / 57.5	78.6 / 63.9	78.9 / 64.0	75.3 / 53.0	73.5 / 55.1	74.9 / 57.6	**79.9 / 64.8**
EMNIST	71.9 / 64.3	82.6 / 75.0	83.0 / 75.4	83.1 / 75.8	82.7 / 75.0	83.3 / 76.4	**83.5 / 76.6**
CIFAR10	70.2 / 48.7	78.2 / 72.4	78.0 / 70.8	82.3 / 70.6	78.6 / 71.2	81.7 / 73.6	**84.3 / 78.1**
CIFAR100	31.5 / 19.9	40.9 / 33.2	41.0 / 33.2	39.0 / 28.3	41.5 / 34.1	41.8 / 32.5	**44.1 / 35.0**
Shakespeare	32.0 / 16.6	**46.7** / 42.8	45.7 / 41.9	40.0 / 25.5	46.6 / 42.7	41.2 / 36.8	**46.7 / 43.0**
Synthetic	65.7 / 58.4	68.2 / 58.9	68.2 / 59.0	68.9 / 60.2	69.1 / 59.0	69.2 / 61.2	**74.7 / 66.7**

Table 1:Test accuracy: average across clients / bottom decile.



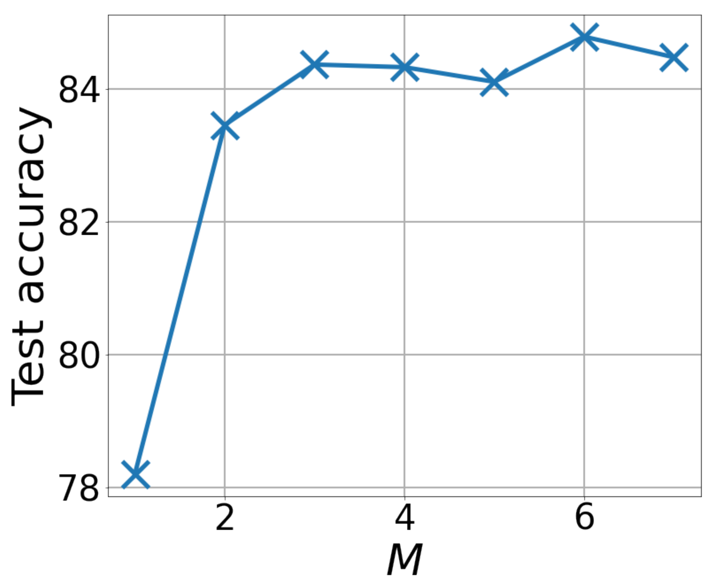


Figure 1:Effect of client sampling rate (left) and number of mixture components M (right) on test accuracy for CIFAR10 [4].

References

[1] Shai Ben-David, Tyler Lu, and D. Pál. ``Does Unlabeled Data Provably Help? Worst-case Analysis of the Sample Complexity of Semi-Supervised Learning''. In: *COLT*. 2008.

[2] Malte Darnstädt, H. U. Simon, and Balázs Szörényi. ``Unlabeled Data Does Provably Help''. In: *STACS*. 2013.

[3] Christina Göpfert et al. ``When can unlabeled data improve the learning rate?'' In: *Conference on Learning Theory*. PMLR. 2019, pp. 1500–1518.

[4] Alex Krizhevsky. *Learning multiple layers of features from tiny images*. Tech. rep. 2009.

[5] Yishay Mansour et al. ``Three approaches for personalization with applications to federated learning''. In: *arXiv preprint arXiv:2002.10619* (2020).

[6] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. ``Clustered Federated Learning: Model-Agnostic Distributed Multitask Optimization Under Privacy Constraints''. In: *IEEE Transactions on Neural Networks and Learning Systems* (2020).

[7] Virginia Smith et al. ``Federated Multi-Task Learning''. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 4427–4437. ISBN: 9781510860964.