

# Federated Multi-Task Learning under a Mixture of Distributions

Othmane Marfoq<sup>1, 2, 3</sup>, Giovanni Neglia<sup>1, 2</sup>, Aurélien Bellet<sup>1</sup>,  
Laetitia Kamani<sup>3</sup>, Richard Vidal<sup>3</sup>,

<sup>1</sup>Inria and <sup>2</sup>Université Côte d'Azur and <sup>3</sup>Accenture Labs



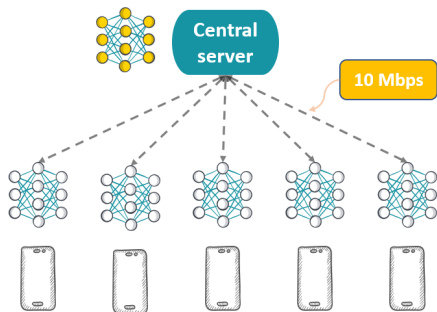
**Accenture** Labs

## Federated Learning

Federated learning (FL) “involves training statistical models over remote devices or siloed data centers, such as mobile phones or hospitals, while keeping data localized”. (Li et al. 2020)

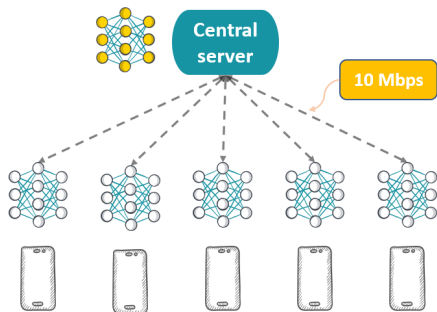
# Federated Learning

Federated learning (FL) “involves training statistical models over remote devices or siloed data centers, such as mobile phones or hospitals, while keeping data localized”. (Li et al. 2020)



# Federated Learning

Federated learning (FL) “involves training statistical models over remote devices or siloed data centers, such as mobile phones or hospitals, while keeping data localized”. (Li et al. 2020)



## Introduction

A (countable) set  $T$  of classification (or regression) tasks which represent the set of possible clients.

## Introduction

A (countable) set  $T$  of classification (or regression) tasks which represent the set of possible clients.

Data  $S_t = \{f_{S_t}^{(i)}, (\mathbf{x}_t^{(i)}; y_t^{(i)})\}_{i=1}^{n_t}$  at client  $t$  is drawn from a local distribution  $D_t$  over  $X \times Y$ .

## Introduction

A (countable) set  $T$  of classification (or regression) tasks which represent the set of possible clients.

Data  $S_t = \{(\mathbf{x}_t^{(i)}; y_t^{(i)})\}_{i=1}^{n_t}$  at client  $t$  is drawn from a local distribution  $D_t$  over  $X \times Y$ .

Client  $t$  wants to learn hypothesis  $h_t$

$$\underset{h_t \in H}{\text{minimize}} L_{D_t}(h_t), \quad \mathbb{E}_{(\mathbf{x}; y) \sim D_t} [l(h_t(\mathbf{x}); y)]:$$

## Introduction

A (countable) set  $T$  of classification (or regression) tasks which represent the set of possible clients.

Data  $S_t = \{S_t^{(i)}, (\mathbf{x}_t^{(i)}; y_t^{(i)})\}_{i=1}^{n_t}$  at client  $t$  is drawn from a local distribution  $D_t$  over  $X \times Y$ .

Client  $t$  wants to learn hypothesis  $h_t$

$$\underset{h_t \in H}{\text{minimize}} L_{D_t}(h_t), \quad \mathbb{E}_{(\mathbf{x}; y) \sim D_t} [l(h_t(\mathbf{x}); y)]:$$

Having personalized models for each client is a necessity in many FL applications.



## Related Work

*Model agnostic meta-learning* (MAML) based federated *multi-task learning* (MTL).

Clustered FL.

Model interpolation: APFL and MAPPER.

Federated MTL via task relationships: MOCHA, pFedMe, L2SGD and FedU.

## Related Work

*Model agnostic meta-learning* (MAML) based federated *multi-task learning* (MTL).

Clustered FL.

Model interpolation: APFL and MAPPER.

Federated MTL via task relationships: MOCHA, pFedMe, L2SGD and FedU.

**Limitation:** restrictive assumptions or complex algorithms.

## An impossibility result

Some assumption on the local data distributions  $D_t; t \in T$  are needed for federated learning to be beneficial:

## An impossibility result

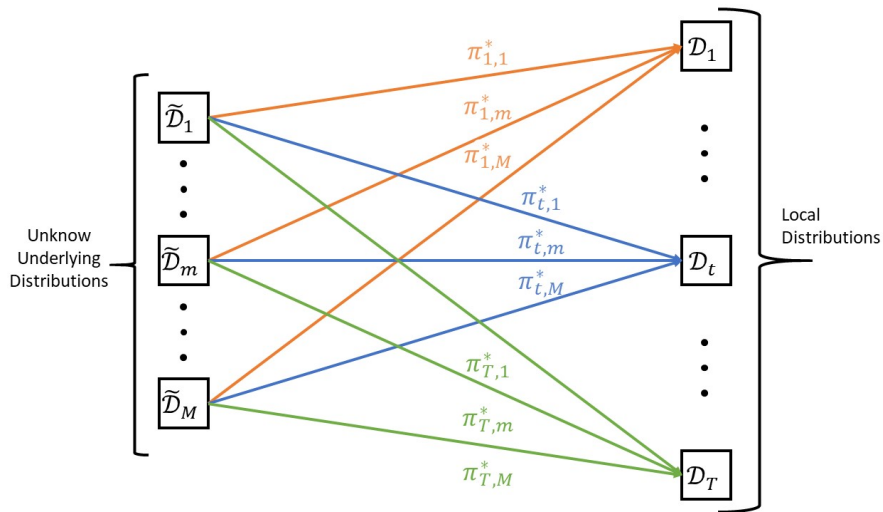
Some assumption on the local data distributions  $D_t; t \in T$  are needed for federated learning to be beneficial:

Federated learning with  $T$  clients is equivalent to  $T$  *semi-supervised learning* (SSL) problems.

With no assumptions on the data distribution, SSL is impossible.

(Ben-David et al. 2008; Darnstädt et al. 2013; Göpfert et al. 2019).

# Main assumption



# Main assumption

## Assumption

There exist  $M$  underlying (independent) distributions  $\mathcal{D}_m$ ;  $1 \leq m \leq M$ , such that for  $t \in T$ ,  $D_t$  is mixture of the distributions  $\mathcal{D}_m$   $g_{m=1}^M$  with weights  $\theta_t = [\theta_{t1}; \dots; \theta_{tm}] \in \mathbb{R}^M$ , i.e.

$$z_t \sim M(\theta_t); ((x_t; y_t) | z_t = m) \sim \mathcal{D}_m; \quad \forall t \in T; \quad (1)$$

where  $M(\cdot)$  is a multinomial (categorical) distribution with parameters

# Generalizing Existing Frameworks

The generative model in the mixture assumption extends/covers some popular multi-task/personalized FL formulations in the literature.

# Generalizing Existing Frameworks

The generative model in the mixture assumption extends/covers some popular multi-task/personalized FL formulations in the literature.

## Example (Clustered Federated Learning)

The mixture assumption recovers this scenario considering  $\mathcal{C}$  and  $t_c = 1$  if task (client)  $t$  is in cluster  $c$  and  $t_c = 0$  otherwise.



## Main Contributions

Flexible assumption for personalized FL (mixtures of components).

EM-like learning algorithms with convergence guarantees (both in client-server and fully-decentralized settings).

## Main Contributions

Flexible assumption for personalized FL (mixtures of components).

EM-like learning algorithms with convergence guarantees (both in client-server and fully-decentralized settings).

More general federated surrogate optimization framework.

## Main Contributions

Flexible assumption for personalized FL (mixtures of components).

EM-like learning algorithms with convergence guarantees (both in client-server and fully-decentralized settings).

More general federated surrogate optimization framework.

Higher accuracy and fairness than SOTA algorithms, even for clients not present at training time.

## Learning under a mixture model

### Proposition (informal)

$$h_t = \sum_{m=1}^M \alpha_{tm} h_m; \quad \forall t \in \{1, \dots, T\} \quad (2)$$

## Learning under a mixture model

Estimate the parameters and  $\theta_t; 1 \leq t \leq T$ , minimizing:

$$f(\theta; \mathcal{S}) = \frac{\log p(\mathcal{S}_{1:T} | \theta)}{n}, \quad \frac{1}{n} \sum_{t=1}^T \sum_{i=1}^{X^t} \log p(s_t^{(i)} | \theta_t); \quad (3)$$

## Learning under a mixture model

Estimate the parameters and  $\theta_t; 1 \leq t \leq T$ , minimizing:

$$f(\theta; \mathcal{X}, \mathcal{Y}) = \frac{1}{n} \sum_{t=1}^T \sum_{i=1}^{|\mathcal{X}^t|} \log p(s_t^{(i)} | j; \theta_t); \quad (3)$$

Use Eq. (4) to get the client predictor for the  $\theta_t$  clients present at training time.

$$h_t = \sum_{m=1}^{|\mathcal{X}^t|} \theta_{tm} h_m(x); \quad \forall t \in \{1, 2, \dots, T\} \quad (4)$$

## Expectation-Maximization

A natural approach to solve problem (3) is via the Expectation-Maximization(EM) algorithm

# Expectation-Maximization

A natural approach to solve problem (3) is via the Expectation-Maximization(EM) algorithm

E-step:  $q_t^{k+1}(z_t^{(i)} = m) / \prod_{tm}^k \exp \left( l(h_{\frac{k}{m}}(x_t^{(i)}); y_t^{(i)}) \right) :$

M-step:  $m_{tm}^{k+1} = \frac{\sum_{i=1}^{n_t} q_t^{k+1}(z_t^{(i)} = m)}{n_t};$

$m_m^{k+1} = 2 \arg \min_{2R^d} \sum_{t=1}^T \sum_{i=1}^{n_t} q_t^{k+1}(z_t^{(i)} = m) l(h(x_t^{(i)}); y_t^{(i)}) :$



# Federated Expectation-Maximization

# Federated Expectation-Maximization

# Federated Expectation-Maximization

# Federated Expectation Maximization

## Theorem

Under Assumptions 1-3 and some other mild assumptions, when clients use SGD as local solver with learning rate  $\eta = \frac{\eta_0}{K}$ , after a large enough number of communication rounds  $K$ , FedEM iterates satisfy:

$$\frac{1}{K} \sum_{k=1}^K \mathbb{E} \|r^k\|^2 \leq O\left(\frac{1}{K}\right);$$

$$\frac{1}{K} \sum_{k=1}^K f(\theta^k; \theta^k) \leq O\left(\frac{1}{K^{3/4}}\right);$$

where the expectation is over the random batches samples, and

$$f(\theta^k; \theta^k) = \mathbb{E} \sum_{i \in \mathcal{K}} \sum_{b \in \mathcal{B}_i} \ell(\theta^k; \mathcal{D}_i^b) - \mathbb{E} \sum_{i \in \mathcal{K}} \sum_{b \in \mathcal{B}_i} \ell(\theta^{k+1}; \mathcal{D}_i^b) \geq 0;$$

# Fully Decentralized Settings

## Theorem (Informal)

In the same setting of the previous theorem and under an additional mild assumption on the connectivity of the communication graph, FedEMs individual estimates  $(\hat{x}_t^k)_{1 \leq t \leq T}$  converge to a common value  $x^k$ . Moreover,  $\hat{x}^k$  and  $\hat{y}^k$  converge to a stationary point of  $f$ .

# Surrogate Federated Optimization

FedEM can be seen as a particular instance of a more general framework that we call federated surrogate optimization

# Surrogate Federated Optimization

FedEM can be seen as a particular instance of a more general framework that we call federated surrogate optimization

This framework minimizes an objective function  $\sum_{t=1}^T f_t(u; v_t)$

Each client  $i \in [T]$  can compute a partial first order surrogate of  $f_t$ .

# 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	30.2=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:** Test accuracy: average across clients / bottom decile.



# Experiments

**Figure:** Effect of client sampling rate on the test accuracy for CIFAR10.

## Experiments

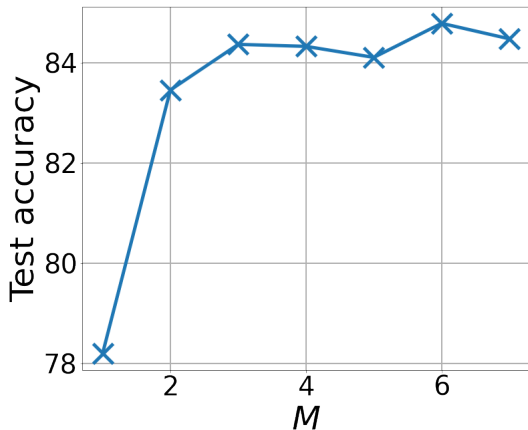


Figure: Effect of number of mixture components  $M$  on the test accuracy

## Experiments

Dataset	FedAvg	FedAvg+	FedEM
FEMNIST	78.3 (80.9)	74.2 (84.2)	<b>79.1</b> (81.5)
EMNIST	83.4 (82.7)	83.7 (92.9)	<b>84.0</b> (83.3)
CIFAR10	77.3 (77.5)	80.4 (80.5)	<b>85.9</b> (90.7)
CIFAR100	41.1 (42.1)	36.5 (55.3)	<b>47.5</b> (46.6)
Shakespeare	<b>46.7</b> (47.1)	40.2 (93.0)	<b>46.7</b> (46.6)
Synthetic	68.6 (70.0)	69.1 (72.1)	<b>73.0</b> (74.1)

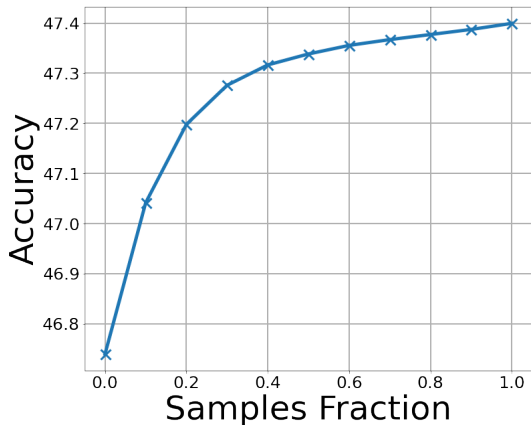
**Table:** Average test accuracy across **clients unseen at training** (train accuracy in parenthesis).

# Experiments

**Table:** Test accuracy comparison across different tasks. For each method, the best test accuracy is reported. For FedEM we run only  $\frac{K}{M}$  rounds, where  $K$  is the total number of rounds for other methods— $K = 80$  for Shakespeare and  $K = 200$  for all other datasets—and  $M = 3$  is the number of components used in FedEM.

Dataset	Local	FedAvg	FedProx	FedAvg+	Clustered FL	pFedMe	FedEM (Ours)
FEMNIST	71.0	<b>78.6</b>	78:6	75:3	73:5	74:9	74:0
EMNIST	71.9	82:6	82:7	83:1	82:7	<b>83.3</b>	82:7
CIFAR10	70:2	78:2	78:0	82:3	78:6	81:7	<b>82.5</b>
CIFAR100	31.5	41:0	40:9	39:0	41:5	41:8	<b>42.0</b>
Shakespeare	32:0	<b>46.7</b>	45:7	40:0	46:6	41:2	43:8
Synthetic	65:7	68:2	68:2	68:9	69:1	69:2	<b>73.2</b>

## Experiments



**Figure:** Effect of the number of samples on the average test accuracy across clients unseen at training.

## Conclusion

Thank you for your attention

**Project link:** <https://github.com/omarfoq/FedEM>

**Email:** othmane.marfoq@inria.fr

## References I

- Ben-David, Shai, Tyler Lu, and D. Pál (2008). “Does Unlabeled Data Provably Help? Worst-case Analysis of the Sample Complexity of Semi-Supervised Learning”. In: *COLT*.
- Darnstädt, Malte, H. U. Simon, and Balázs Szörényi (2013). “Unlabeled Data Does Provably Help”. In: *STACS*.
- Göpfert, Christina et al. (2019). “When can unlabeled data improve the learning rate?” In: *Conference on Learning Theory*. PMLR, pp. 1500–1518.
- Li, Tian et al. (2020). “Federated learning: Challenges, methods, and future directions”. In: *IEEE Signal Processing Magazine* 37.3, pp. 50–60.