Tackling Heterogeneity in Federated Learning Systems

Othmane Marfoq[1,2,3]

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

Thursday 7th December, 2023

UNIVERSITÉ
CÔTE D'AZUR

Accenture Labs

Outline of talk

1. Motivation & Summary of Contributions
2. A Focus on Tackling Statistical Heterogeneity via Personalization
   - Why do we Need to Personalize?
   - An Impossibility Result
   - Personalized Federated Learning under a Mixture of Distributions
   - Personalized Federated Learning through Local Memorization
   - A Comparison between FedEM and kNN-Per
3. Other Main Contributions
4. Conclusion

Outline

1. Motivation & Summary of Contributions
2. A Focus on Tackling Statistical Heterogeneity via Personalization
   - Why do we Need to Personalize?
   - An Impossibility Result
   - Personalized Federated Learning under a Mixture of Distributions
   - Personalized Federated Learning through Local Memorization
   - A Comparison between FedEM and kNN-Per
3. Other Main Contributions
4. Conclusion

A Shift of Paradigm: From Centralized to Decentralized Data

- The standard in ML considers a centralized dataset processed in the cloud
- In practice, data is often decentralized:
  1. Sending the data may be too *costly*
  2. Data may be considered too *sensitive*

A Shift of Paradigm: From Centralized to Decentralized Data

- The standard in ML considers a centralized dataset processed in the cloud
- In practice, data is often decentralized:
  1. Sending the data may be too *costly*
  2. Data may be considered too *sensitive*

Solution: Federated Learning (FL) (McMahan et al. 2017)

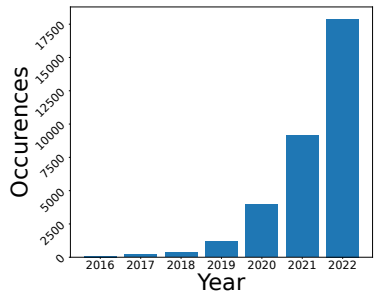


Figure: Occurrences of the key-word "federated learning" over time in academic papers.

A Baseline Algorithm: FedAvg (McMahan et al. 2017)

FedAvg (aka local SGD) aims at solving

$$\underset{\theta \in \mathbb{R}^d}{\text{minimize}}\, F(\theta) \triangleq \sum_{c=1}^{C} \frac{n_c}{n} F_c(\theta),$$

where

$$F_c(\theta) = \hat{\mathcal{L}}_c(h_\theta) = \frac{1}{n_c} \sum_{i=1}^{n_c} \ell(h_\theta(\mathbf{x}_{c,i}), y_{c,i})$$

Algorithm 1: FedAvg

server randomly initialize θ_1;
for $t = 1, \ldots, T$ **do**
  server selects a subset $\mathcal{C}_t \subset \mathcal{C}$ of clients ;
  server broadcast θ_t to the selected clients $\mathcal{C}_t$;
  for *each client* $k \in \mathcal{C}_t$ **in parallel do**
    $\theta_{t+1,c} \leftarrow$ `ClientUpdate`$(\theta_t, \mathcal{S}_c, E)$;
    client sends $\theta_{t+1,c}$ to the **server**;
  end
  $\theta_{t+1} \leftarrow \sum_{c \in \mathcal{C}_t} \frac{n_c}{n} \cdot \theta_{t+1,c}$;
end

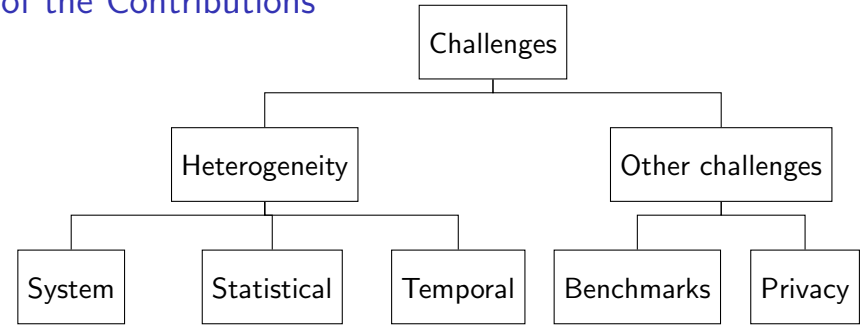
Challenges

Heterogeneity is a core and fundamental challenge in FL

System	**Statistical**	**Temporal**
• communication/computation	• imbalanced data	• data streams
• correlated clients' availability	• non-IID data	• concept shift

Other challenges: privacy, expensive communications, robustness, lack of benchmarks...

Summary of the Contributions

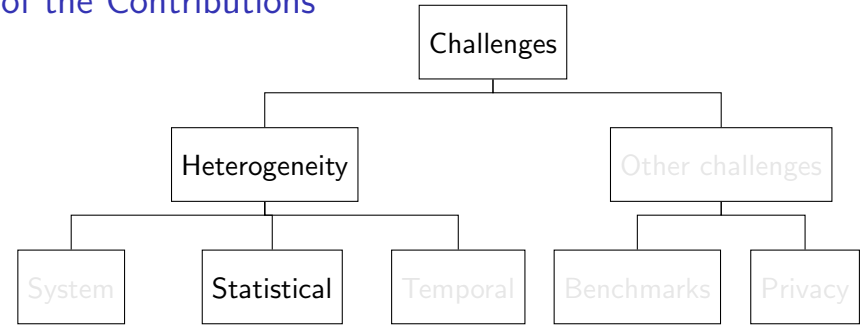


NeurIPS'20 **NeurIPS'21** **AISTATS'23** NeurIPS'22 PoPETS'24

ICML'22 **ICML'22** **Submitted**

Trans. on Netw.

Summary of the Contributions



NeurIPS'20 **NeurIPS'21** AISTATS'23 NeurIPS'22 PoPETS'24

ICML'22 **ICML'22** Submitted

Trans. on Netw.

Outline

1. Motivation & Summary of Contributions
2. A Focus on Tackling Statistical Heterogeneity via Personalization
   - Why do we Need to Personalize?
   - An Impossibility Result
   - Personalized Federated Learning under a Mixture of Distributions
   - Personalized Federated Learning through Local Memorization
   - A Comparison between FedEM and kNN-Per
3. Other Main Contributions
4. Conclusion

In this Presentation

Two personalization techniques to handle heterogeneity in FL. **The main focus is statistical heterogeneity**.

- Othmane Marfoq et al. (2021). "Federated Multi-Task Learning under a Mixture of Distributions". In: *NeurIPS'21*

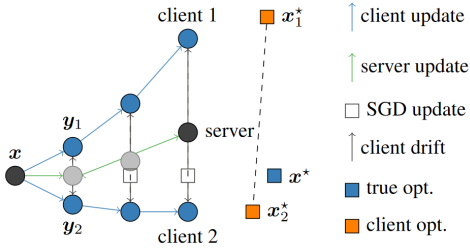
- Othmane Marfoq et al. (2022). "Personalized Federated Learning through Local Memorization". In: *ICML'22*

Outline

1. Motivation & Summary of Contributions
2. A Focus on Tackling Statistical Heterogeneity via Personalization
   - Why do we Need to Personalize?
   - An Impossibility Result
   - Personalized Federated Learning under a Mixture of Distributions
   - Personalized Federated Learning through Local Memorization
   - A Comparison between FedEM and kNN-Per
3. Other Main Contributions
4. Conclusion

Statistical Heterogeneity

Data is collected from clients with varying behaviors and preferences. A dual challenge:

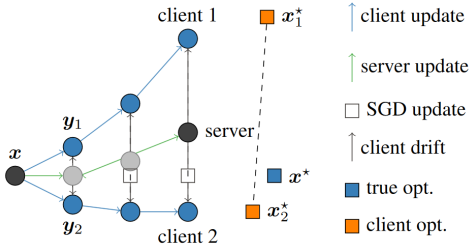


Slow convergence due to client drift

A common model for all clients is sub-optimal

Statistical Heterogeneity

Data is collected from clients with varying behaviors and preferences. A dual challenge:



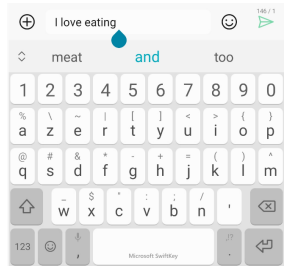
Slow convergence due to client drift

A common model for all clients is sub-optimal

Solutions: `SCAFFOLD`, `ProxSkip`

Statistical Heterogeneity

Data is collected from clients with varying behaviors and preferences. A dual challenge:



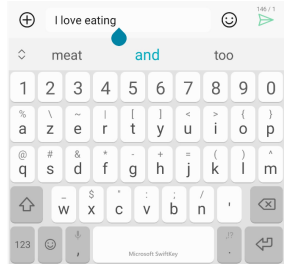
Slow convergence due to client drift

Solutions: `SCAFFOLD`, `ProxSkip`

A common model for all clients is sub-optimal

Statistical Heterogeneity

Data is collected from clients with varying behaviors and preferences. A dual challenge:



Slow convergence due to client drift

Solutions: `SCAFFOLD`, `ProxSkip`

A common model for all clients is sub-optimal

Personalized models are a necessity in many FL applications

Basic Notations

- We consider a (countable) set $\mathcal{C}$ of classification (or regression) tasks which represent the set of possible clients. Usually $|\mathcal{C}| = C$
- Data $\mathcal{S}_c = \{s_{c,i} \triangleq (\mathbf{x}_{c,i},\ y_{c,i})\}_{i=1}^{n_c}$ at client c is drawn according to a local probability distribution $\mathcal{P}_c$ over $\mathcal{X} \times \mathcal{Y}$. Let $n = \sum_{c=1}^{C} n_c$ be the total number of samples
- Client c wants to learn hypothesis $h_c^* \in \mathcal{H}$ minimizing its *local population risk*

$$h_c^* \in \arg\min_{h \in \mathcal{H}} \mathcal{L}_c(h) \triangleq \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{P}_c}\left[\ell\left(h\left(\mathbf{x}\right), y\right)\right]$$

- The empirical risk associated to client c is given by $\hat{\mathcal{L}}_c(h) = \sum_{i=1}^{n_c} \ell(h(\mathbf{x}_{c,i}), y_{c,i}) / n_c$
- Typically $\mathcal{H}$ is a set of parametric hypothesis (e.g., neural networks, linear models...): $\mathcal{H} = \{h_\theta, \theta \in \mathbb{R}^d\}$. We use $d_{\mathcal{H}}$ to denote its pseudo-dimension.

A Fundamental Question?

$\hat{h}_c \in \arg\min_{h\in\mathcal{H}} \hat{\mathcal{L}}_c(h)$; $\bar{h} \in \arg\min_{h\in\mathcal{H}} \sum_{c'} (n_{c'}/n) \cdot \hat{\mathcal{L}}_c(h)$; $\bar{\mathcal{P}} = \sum_{c'} (n_{c'}/n) \cdot \mathcal{P}_{c'}$

Proposition (Mansour et al. 2020)

Let $\delta \in (0,1)$. With probability at least $1-\delta$, the following holds:

Purely local model :
$$\mathcal{L}_c\left(\hat{h}_c\right) - \min_{h\in\mathcal{H}} \mathcal{L}_c(h) = \mathcal{O}\left(\sqrt{\frac{d_{\mathcal{H}} + \log 1/\delta}{n_c}}\right)$$

Global model :
$$\mathcal{L}_c\left(\bar{h}\right) - \min_{h\in\mathcal{H}} \mathcal{L}_c(h) = \mathcal{O}\left(\sqrt{\frac{d_{\mathcal{H}} + \log 1/\delta}{n}}\right) + \mathrm{disc}_{\mathcal{H}}\left(\mathcal{P}_c, \bar{\mathcal{P}}\right)$$

A Fundamental Question?

$\hat{h}_c \in \arg\min_{h\in\mathcal{H}} \hat{\mathcal{L}}_c(h)$; $\bar{h} \in \arg\min_{h\in\mathcal{H}} \sum_{c'} (n_{c'}/n) \cdot \hat{\mathcal{L}}_c(h)$; $\bar{\mathcal{P}} = \sum_{c'} (n_{c'}/n) \cdot \mathcal{P}_{c'}$

Proposition (Mansour et al. 2020)

Let $\delta \in (0,1)$. With probability at least $1-\delta$, the following holds:

Purely local model :
$$\mathcal{L}_c\left(\hat{h}_c\right) - \min_{h\in\mathcal{H}} \mathcal{L}_c(h) = \mathcal{O}\left(\sqrt{\frac{d_{\mathcal{H}} + \log 1/\delta}{n_c}}\right)$$

Global model :
$$\mathcal{L}_c\left(\bar{h}\right) - \min_{h\in\mathcal{H}} \mathcal{L}_c(h) = \mathcal{O}\left(\sqrt{\frac{d_{\mathcal{H}} + \log 1/\delta}{n}}\right) + \mathrm{disc}_{\mathcal{H}}\left(\mathcal{P}_c, \bar{\mathcal{P}}\right)$$

What is the optimal trade-off, and how can it be achieved?

Brief Overview of Related Work

- *Model agnostic meta-learning* (MAML) based personalized federated learning
- (Soft) Clustered FL: `CFL`, `IFCA`, `FedEM`, `FedSoft`
- Learning shared representation: `FedRep`, `FedPer`, `pFedGP`, `kNN-Per`
- Model interpolation: `APFL`, `MAPPER`, `kNN-Per`
- Federated MTL via task relationships: `MOCHA`, `pFedMe`, `L2SGD` and `FedU`
- Personalization as a stochastic optimization problem with biased gradients; gradient filtering algorithms: `ALL-FOR-ALL`, `ALL-FOR-ONE`

Outline

1. Motivation & Summary of Contributions
2. A Focus on Tackling Statistical Heterogeneity via Personalization
   - Why do we Need to Personalize?
   - An Impossibility Result
   - Personalized Federated Learning under a Mixture of Distributions
   - Personalized Federated Learning through Local Memorization
   - A Comparison between FedEM and kNN-Per
3. Other Main Contributions
4. Conclusion

An Impossibility Result

Proposition (Marfoq et al. 2021)

FL is impossible with no assumptions on local data distributions; some assumptions on local data distributions are needed for collaboration to be **provably beneficial**, *i.e., improve sample complexity.*

An Impossibility Result

Proposition (Marfoq et al. 2021)

FL is impossible with no assumptions on local data distributions; some assumptions on local data distributions are needed for collaboration to be **provably beneficial**, *i.e., improve sample complexity.*

Proof sketch by reduction to semi-supervised learning:

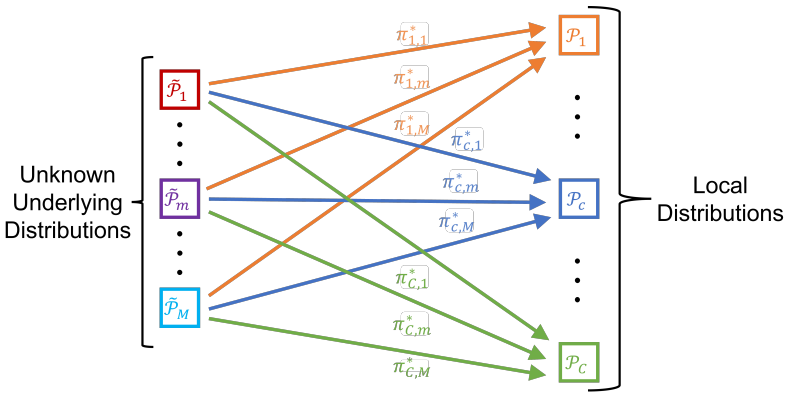
- Assume $\mathcal{P}_c(\mathbf{x})$ is identical across c$\in \mathcal{C}$, but $\mathcal{P}_c(y|\mathbf{x})$ can be arbitrarily different
- Collaborative learning with C clients is equivalent to C SSL problems
- With no assumptions on the data distribution, SSL does not improve sample complexity (Ben-David et al. 2008; Darnstädt et al. 2013; Göpfert et al. 2019)

Outline

1. Motivation & Summary of Contributions
2. A Focus on Tackling Statistical Heterogeneity via Personalization
   - Why do we Need to Personalize?
   - An Impossibility Result
   - Personalized Federated Learning under a Mixture of Distributions
   - Personalized Federated Learning through Local Memorization
   - A Comparison between FedEM and kNN-Per
3. Other Main Contributions
4. Conclusion

Main assumption

The data distribution of each client is a mixture of M underlying distribution



Main assumption

Assumption (Marfoq et al. 2021)

There exist M underlying (independent) distributions $\tilde{\mathcal{P}}_m$, $1 \leq m \leq M$, such that for $c \in \mathcal{C}$, $\mathcal{P}_c$ is mixture of the distributions $\{\tilde{\mathcal{P}}_m\}_{m=1}^M$ with weights $\pi_c^* = [\pi_{c1}^*, \ldots, \pi_{cm}^*] \in \Delta^{M-1}$, i.e.

$$z_c \sim \mathcal{M}(\pi_c^*), \quad ((\mathbf{x}_c, y_c) \,|\, z_c = m) \sim \tilde{\mathcal{P}}_m, \quad \forall c \in \mathcal{C},$$

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

Generalizing Existing Frameworks

Our assumptions generalizes previous personalized FL formulations

Example (Clustered Federated Learning)

The mixture assumption recovers this scenario considering $M = G$ and $\pi^*_{cg} = 1$ if task (client) c is in cluster g and $\pi^*_{cg} = 0$ otherwise.

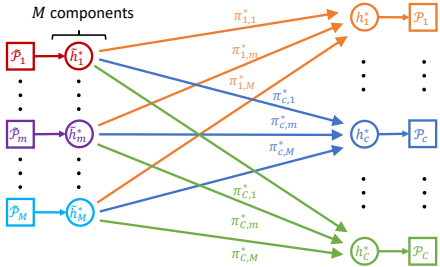
We also recover model interpolation (Deng et al. 2020; Mansour et al. 2020) and Fed-MTL with task relationships (Smith et al. 2017; Vanhaesebrouck et al. 2017) as special cases

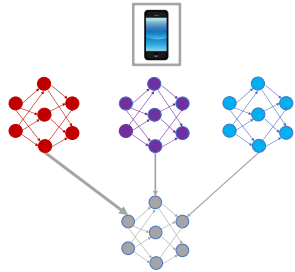
Learning within a Mixture Model

Proposition (Marfoq et al. 2021)

Let $\breve{\Theta}, \breve{\Pi} \in \arg\min_{\Theta,\Pi} \mathbb{E}_c \, \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{P}_c} [-\log \mathcal{P}_c(\mathbf{x}, y|\Theta, \pi_c)]$. *Then,*

$$h_c^* = \sum_{m=1}^{M} \breve{\pi}_{cm} h_{\breve{\theta}_m}, \quad \forall c \in \mathcal{C}$$





em

Learning within a mixture model

- Estimate the parameters $\breve{\Theta}$ and $\breve{\pi}_c,\ 1 \leq c \leq C$, minimizing:

$$f(\Theta, \Pi) \triangleq -\frac{\log p(\mathcal{S}_{1:C}|\Theta, \Pi)}{n} \triangleq -\frac{1}{n}\sum_{t=1}^{T}\sum_{i=1}^{n_t} \log p(s_{c,i}|\Theta, \pi_c)$$

- Use (1) to get the client predictor for the C clients present at training time

$$h_c^* = \sum_{m=1}^{M} \breve{\pi}_{cm} h_{\breve{\theta}_m}(\mathbf{x}), \quad \forall c \in \mathcal{C} \tag{1}$$

Learning within a mixture model

- Estimate the parameters $\breve{\Theta}$ and $\breve{\pi}_c,\ 1 \leq c \leq C$, minimizing:

$$f(\Theta, \Pi) \triangleq -\frac{\log p(\mathcal{S}_{1:C}|\Theta, \Pi)}{n} \triangleq -\frac{1}{n}\sum_{t=1}^{T}\sum_{i=1}^{n_t} \log p(s_{c,i}|\Theta, \pi_c)$$

- Use (1) to get the client predictor for the C clients present at training time

$$h_c^* = \sum_{m=1}^{M} \breve{\pi}_{cm} h_{\breve{\theta}_m}(\mathbf{x}), \quad \forall c \in \mathcal{C} \tag{1}$$

- Clients c' not participating at the training, learn $\pi_{c'}$ in a single forward-pass, then uses (1) to obtain the personalized model

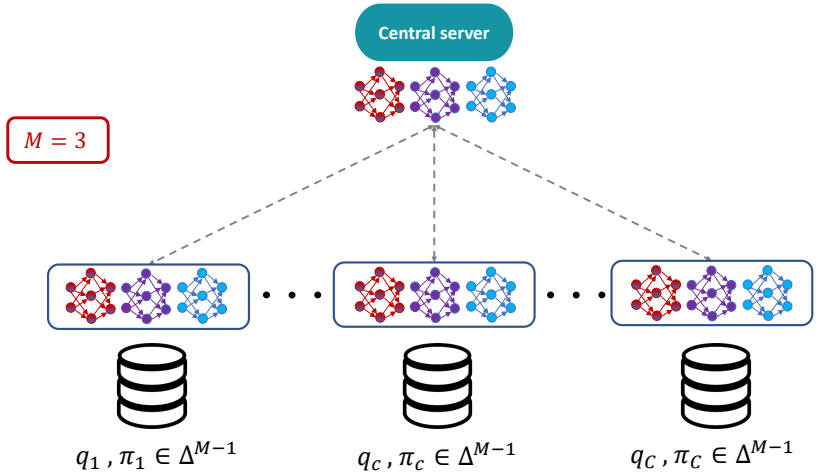
Federated Expectation Maximization

A natural approach to solve problem (23) is via the *Expectation-Maximization* (EM) algorithm

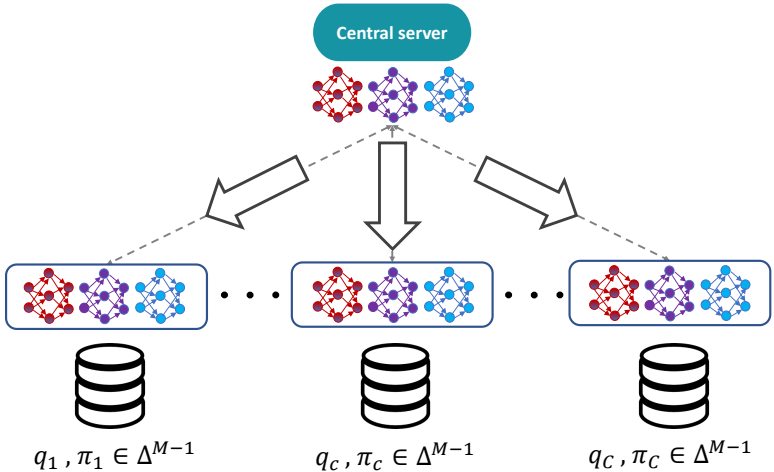
E-step:
$$q_c^{k+1}(z_{c,i} = m) \propto \pi_{cm}^k \cdot \exp\left(-\ell\big(h_{\theta_m^k}(\mathbf{x}_{c,i}), y_{c,i}\big)\right)$$

M-step:
$$\pi_{cm}^{k+1} = \frac{\sum_{i=1}^{n_c} q_c^{k+1}(z_{c,i} = m)}{n_c},$$
$$\theta_m^{k+1} \in \arg\min_{\theta \in \mathbb{R}^d} \sum_{c=1}^{C} \sum_{i=1}^{n_c} q_c^{k+1}(z_{c,i} = m) \cdot \ell\big(h_{\theta_m^k}(\mathbf{x}_{c,i}), y_{c,i}\big).$$

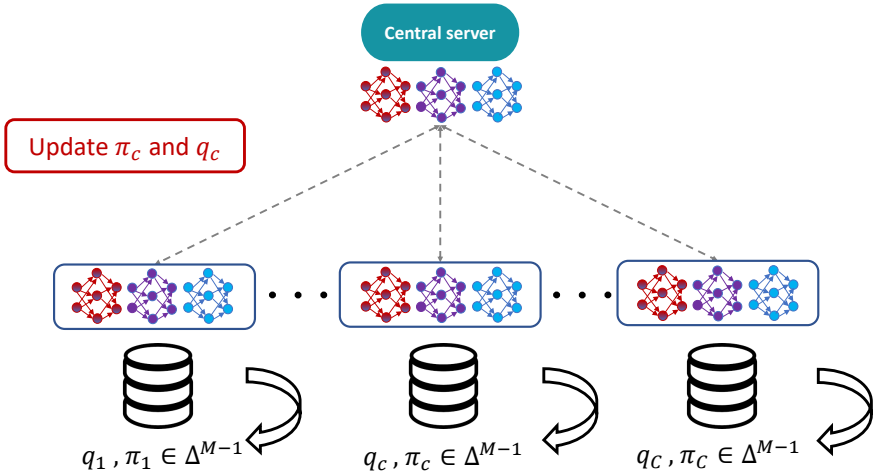
Federated Expectation-Maximization



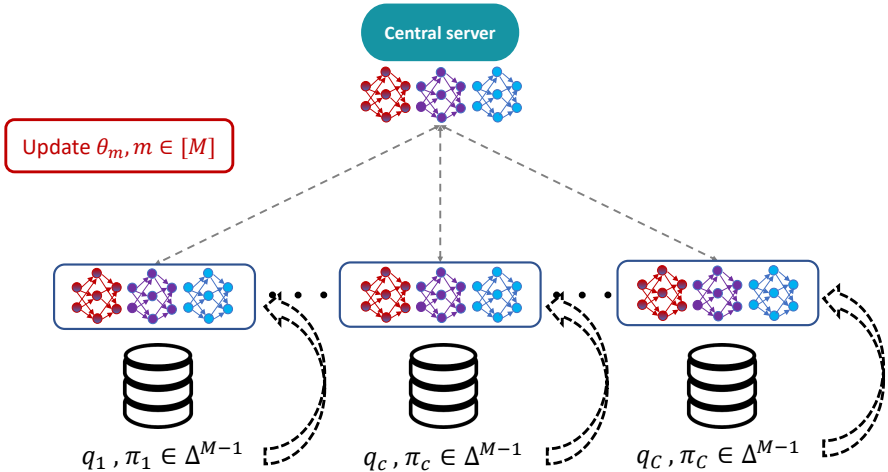
Federated Expectation-Maximization



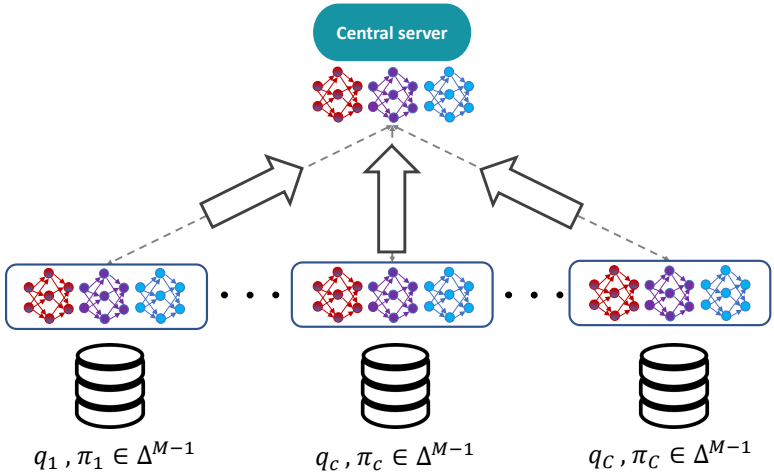
Federated Expectation-Maximization



Federated Expectation-Maximization



Federated Expectation-Maximization



Convergence Rate

Theorem (Marfoq et al. 2021)

Under mild assumptions, 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:*

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left\|\nabla_{\Theta} f\left(\Theta^k, \Pi^k\right)\right\|_F^2 \leq \mathcal{O}\left(\frac{1}{\sqrt{K}}\right),$$

$$\frac{1}{K}\sum_{k=1}^{K}\Delta_{\Pi} f(\Theta^k, \Pi^k) \leq \mathcal{O}\left(\frac{1}{K^{3/4}}\right),$$

where the expectation is over the random batches samples, and

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

Surrogate Federated Optimization

- `FedEM` can be seen as a particular instance of a more general framework that we call **federated surrogate optimization**, extending the centralized framework of (Mairal 2013)
- This framework minimizes an objective function $\sum_{c=1}^{C} \omega_c f_c(\mathbf{u}, \mathbf{v}_c)$
- Each client $c \in [C]$ can compute a partial first order surrogate of f_c
- Our framework can be used to analyze the convergence of other FL algorithms, such as `pFedMe` (Dinh et al. 2020)

Experiments: Average Accuracy / Fairness

Code available at `https://github.com/omarfoq/FedEM`

Dataset	Local	FedAvg	FedProx	FedAvg+	CFL	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: Test accuracy: average across clients / bottom decile.

How to Choose M

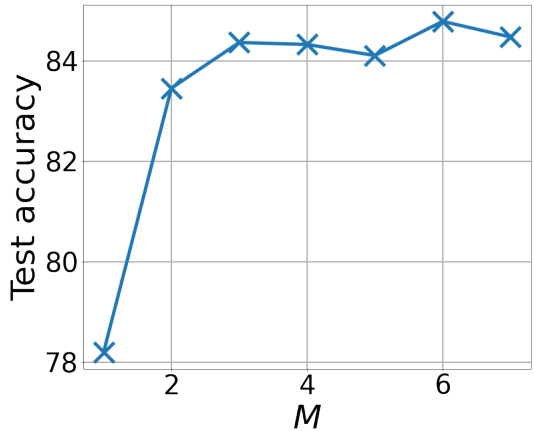


Figure: Effect of number of mixture components M on the test accuracy

A Limitation of FedEM

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+	CFL	pFedMe	FedEM (Ours)
FEMNIST	71.0	**78.6**	78.6	75.3	73.5	74.9	74.0
EMNIST	71.9	82.6	82.7	83.1	82.7	**83.3**	82.7
CIFAR10	70.2	78.2	78.0	82.3	78.6	81.7	**82.5**
CIFAR100	31.5	41.0	40.9	39.0	41.5	41.8	**42.0**
Shakespeare	32.0	**46.7**	45.7	40.0	46.6	41.2	43.8
Synthetic	65.7	68.2	68.2	68.9	69.1	69.2	**73.2**

Works Extending FedEM

- Personalization approaches expanding on the mixture paradigm: FedSoft, FedGMM, FedMN, and FedeRiCo

Works Extending FedEM

- Personalization approaches expanding on the mixture paradigm: FedSoft, FedGMM, FedMN, and FedeRiCo
- Beyond personalization, FedEM has found applications in addressing distributed concept shift (FedTEM, FEM-OMD) and characterizing internal evasion attacks (ARU)

Works Extending FedEM

- Personalization approaches expanding on the mixture paradigm: FedSoft, FedGMM, FedMN, and FedeRiCo
- Beyond personalization, FedEM has found applications in addressing distributed concept shift (FedTEM, FEM-OMD) and characterizing internal evasion attacks (ARU)
- FedEM's accompanying code provides an open source implementation of many (personalized) federated learning algorithms, e.g., FedeRiCo, ARU

Outline

1. Motivation & Summary of Contributions
2. A Focus on Tackling Statistical Heterogeneity via Personalization
   - Why do we Need to Personalize?
   - An Impossibility Result
   - Personalized Federated Learning under a Mixture of Distributions
   - Personalized Federated Learning through Local Memorization
   - A Comparison between FedEM and kNN-Per
3. Other Main Contributions
4. Conclusion

Local Memorization Mechanism: kNN-Per

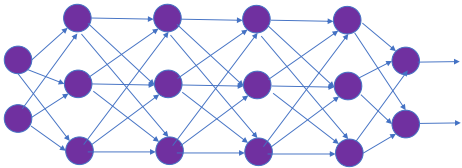
Suppose that each client $c \in [C]$ has access to a a global discriminative model $h_{\mathcal{S}} (\equiv \bar{h})$, such that

$$h_{\mathcal{S}} \in \underset{h \in \mathcal{H}}{\arg\min}\, \mathcal{L}_{\mathcal{S}}(h) \triangleq \sum_{c=1}^{C} \frac{n_c}{n} \cdot \sum_{i=1}^{n_c} \ell\left(h\left(\mathbf{x}_{c,i}\right), y_{c,i}\right).$$

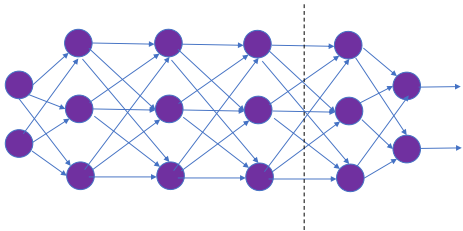
The global model $h_{\mathcal{S}}$ can be used to compute an embedding $\phi_{h_{\mathcal{S}}}(\mathbf{x}) \in \mathbb{R}^p$ for $\mathbf{x} \in \mathcal{X}$. Each client $c \in [C]$ builds a local datastore

$$(\mathcal{K}_c, \mathcal{V}_c) = \left\{\left(\phi_{h_{\mathcal{S}}}\left(\mathbf{x}_{c,i}\right), y_{c,i}\right), \forall \left(\mathbf{x}_{c,i}, y_{c,i}\right) \in \mathcal{S}_c\right\}.$$

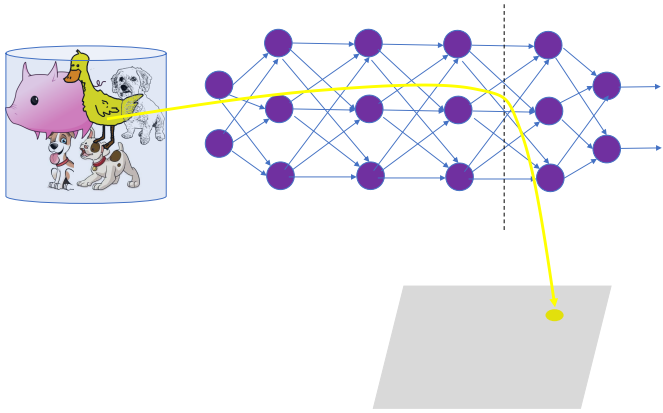
Local Memorization Mechanism: kNN-Per



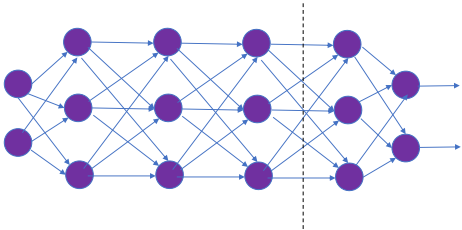
Local Memorization Mechanism: kNN-Per

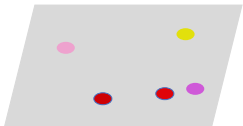


Local Memorization Mechanism: kNN-Per

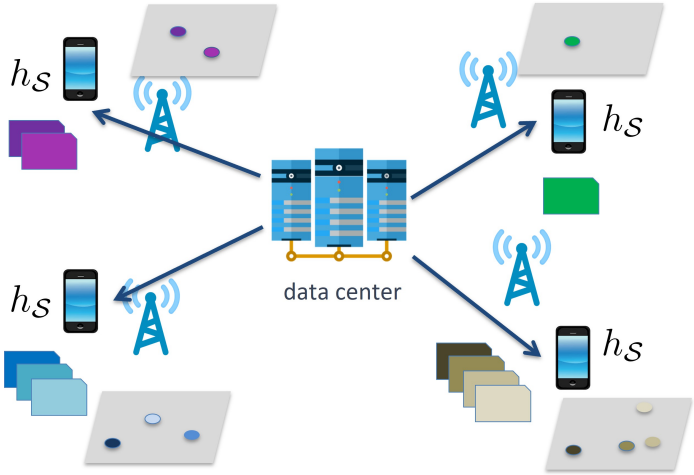


Local Memorization Mechanism: kNN-Per

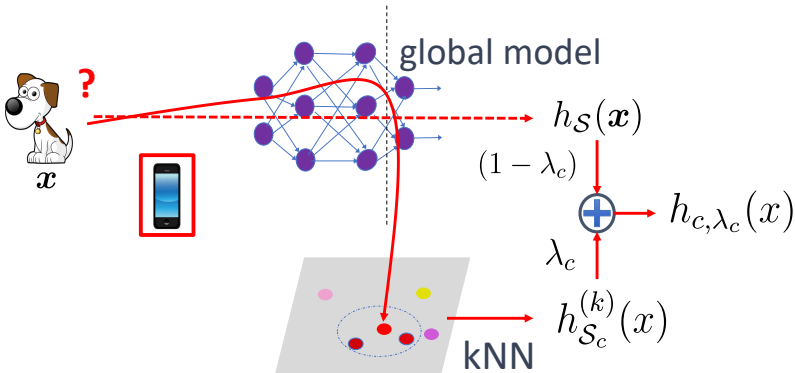




Local Memorization Mechanism: kNN-Per



Local Memorization Mechanism: kNN-Per



The Representation Assumption

Assumption (Marfoq et al. 2022)

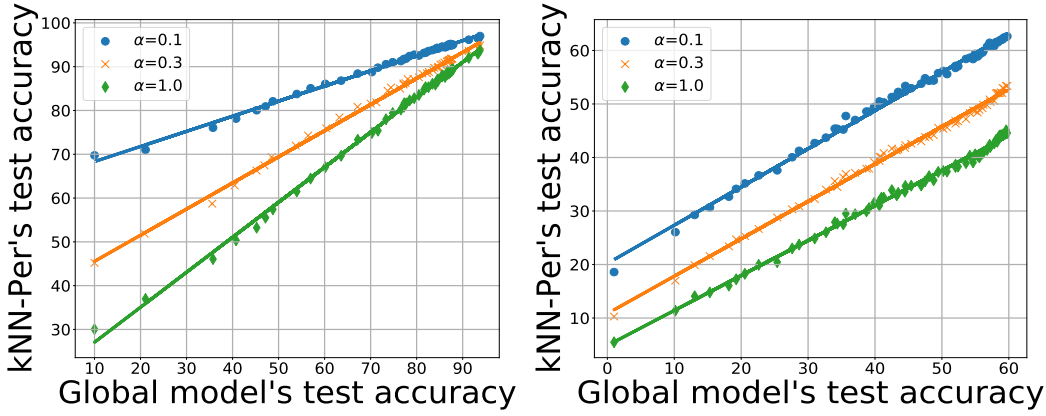
Let $\eta_c(\mathbf{x}) = \mathcal{P}_c(y = 1|\mathbf{x})$. *There exist constants* $\gamma_1, \gamma_2 > 0$, *such that for any dataset* $\mathcal{S}$ *drawn from* $\mathcal{X} \times \mathcal{Y}$ *and any data points* $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$, *we have*

$$\underbrace{\left|\eta_c(\mathbf{x}) - \eta_c(\mathbf{x}')\right|}_{\mathbf{x} \text{ \& } \mathbf{x}' \text{ have the same label}} \leq \underbrace{d\left(\phi_{h_S}(\mathbf{x}), \phi_{h_S}(\mathbf{x}')\right)}_{\text{representations' distance}} \times \left(\gamma_1 + \gamma_2 \underbrace{\left(\mathcal{L}_{\mathcal{P}_c}(h_S) - \mathcal{L}_{\mathcal{P}_c}(h_c^*)\right)}_{\text{global model's quality for client } c}\right),$$

where $h_c^* \in \arg\min_{h \in \mathcal{H}} \mathcal{L}_{D_c}(h)$.

If two samples have close representations, then their labels are likely to be the same. This is all the more so, the more suitable the global model is for the local distribution.

The Representation Assumption (Empirical Validation)



(a) CIFAR-10.

(b) CIFAR-100.

Figure: Effect of the global model quality on the test accuracy of `kNN-Per` with $\lambda = 1$.

Generalization Bound

Theorem (Marfoq et al. 2022)

In the non-agnostic case, i.e., $\mathcal{L}_{\mathcal{P}_c}(h_c^*)$, and under proper assumptions, consider $c \in [C]$ and $\lambda_c \in (0,1)$. Then,

$$\mathop{\mathbb{E}}_{\mathcal{S}\sim\otimes_{c=1}^{C}\mathcal{P}_c^{n_c}}\left[\mathcal{L}_{\mathcal{P}_c}\left(h_{c,\lambda_c}\right)\right] \leq (1-\lambda_c)\cdot \underbrace{\tilde{\mathcal{O}}\left(\mathrm{disc}_{\mathcal{H}}\left(\bar{\mathcal{P}},\mathcal{P}_c\right)+\left(\sqrt{\frac{d_{\mathcal{H}}}{n}}\right)\right)}_{\text{global model}} + \lambda_c \cdot \underbrace{\mathcal{O}\left(\frac{\sqrt{p}}{\sqrt[p+1]{n_c}}\right)}_{\text{purely local model}}$$

where $d_{\mathcal{H}}$ is the the VC dimension of the hypothesis class $\mathcal{H}$, $\bar{\mathcal{P}} = \sum_{c=1}^{C} \frac{n_c}{n}\cdot\mathcal{P}_c$ and $\mathrm{disc}_{\mathcal{H}}$ *is the label discrepancy associated to the hypothesis class $\mathcal{H}$.*

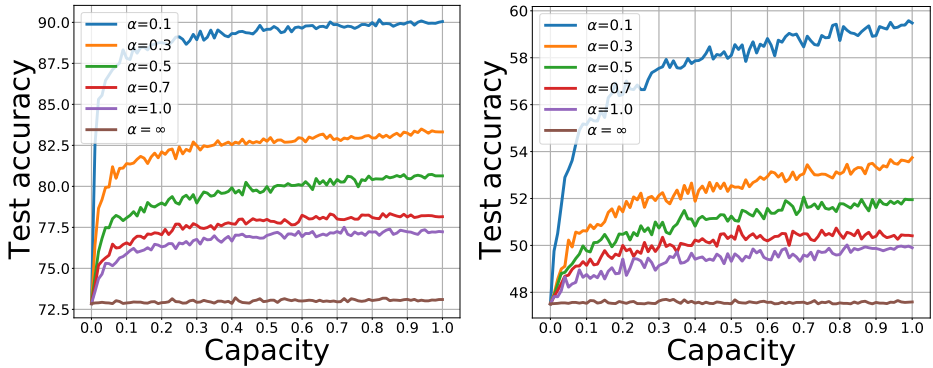
Experiments

Code available at `https://github.com/omarfoq/knn-per`

Dataset	Local	FedAvg	FedAvg+	CFL	Ditto	FedRep	APFL	kNN-Per
FEMNIST	71.0/57.5	83.4/68.9	84.3/69.4	83.7/69.4	84.3/71.3	85.3/72.7	84.1/69.4	**88.2**/**78.8**
CIFAR-10	57.6/41.1	72.8/59.6	75.2/62.3	73.3/61.5	80.0/66.5	77.7/65.2	78.9/68.1	**83.0**/**71.4**
CIFAR-100	31.5/19.8	47.4/36.0	51.4/41.1	47.2/36.2	52.0/41.4	53.2/41.7	51.7/41.1	**55.0**/**43.6**
Shakespeare	32.0/16.0	48.1/43.1	47.0/42.2	46.7/41.4	47.9/42.6	47.2/42.3	45.9/42.4	**51.4**/**45.4**

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

Effect of local dataset's size and data heterogeneity



(a) CIFAR-10 (b) CIFAR-100

Figure: Accuracy vs capacity (local datastore size). The capacity is normalized with respect to the initial size of the client's dataset partition. **Smaller values of α correspond to more heterogeneous data distributions across clients.**

Adding compression techniques

ProtoNN-like datastore compression

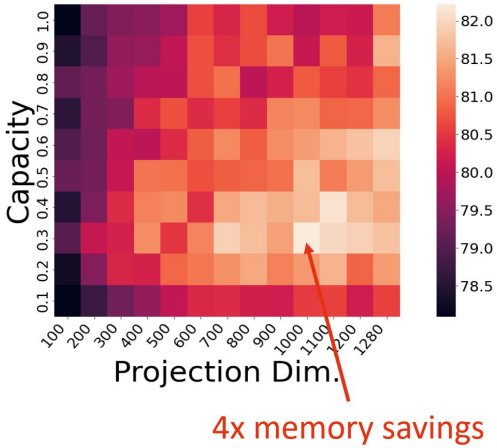
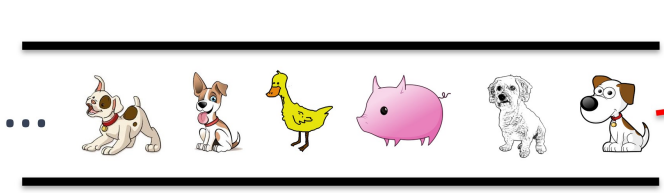
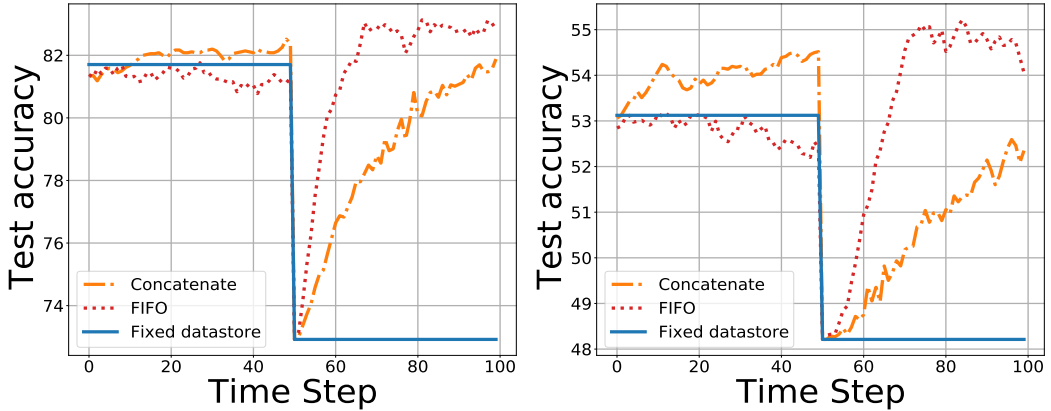


Figure: Test accuracy on CIFAR-10 dataset when the kNN mechanism is implemented through `ProtoNN` for different values of projection dimension and number of prototypes (expressed as a fraction of the local dataset).

Robustness to distribution shift



Robustness to distribution shift



(a) CIFAR-10.

(b) CIFAR-100.

Figure: Test accuracy when a distribution shift happens at time step $t_0 = 50$ for different datastore management strategies.

Outline

1. Motivation & Summary of Contributions
2. A Focus on Tackling Statistical Heterogeneity via Personalization
   - Why do we Need to Personalize?
   - An Impossibility Result
   - Personalized Federated Learning under a Mixture of Distributions
   - Personalized Federated Learning through Local Memorization
   - A Comparison between FedEM and kNN-Per
3. Other Main Contributions
4. Conclusion

Comparing FedEM and kNN-Per

FedEM / Mixture Assumption

- Flexible and generic assumption
- Quantifies data distribution similarity among clients
- Drawback: Incurs local computational and communication overhead due to maintaining and training multiple base models

kNN-Per / Representation Assumption

- Easily integrates as a lightweight module in standard FL
- Addresses statistical, system, and temporal heterogeneity
- Drawback: May not be suitable as a modeling tool in all scenarios due to its limited flexibility

Outline

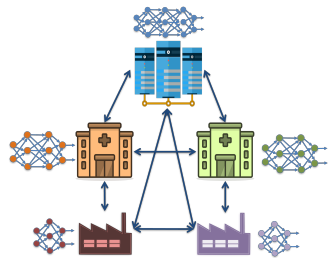
1. Motivation & Summary of Contributions
2. A Focus on Tackling Statistical Heterogeneity via Personalization
   - Why do we Need to Personalize?
   - An Impossibility Result
   - Personalized Federated Learning under a Mixture of Distributions
   - Personalized Federated Learning through Local Memorization
   - A Comparison between FedEM and kNN-Per
3. Other Main Contributions
4. Conclusion

System Heterogeneity

Clients exhibit variations in hardware specifications, network connectivity types, and power availability. Three distinct scenarios:

1. Cross-silo settings

- Different pairwise communication capabilities between data silo
- An orchestrator-centered communication topology is potentially inefficient
- Peer-to-peer communications are usually better



Contribution. How to design a communication setup that allows for the fastest convergence?

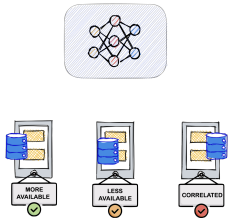
Othmane Marfoq et al. (2020). "Throughput-Optimal Topology Design for Cross-Silo Federated Learning". In: *NeurIPS'20*

System Heterogeneity

Clients exhibit variations in hardware specifications, network connectivity types, and power availability. Three distinct scenarios:

2. Cross-device settings

- System constraints affect client availability
- Heterogeneous and correlated client availability patterns
- Most works rely on simplistic assumptions such as independent availability patterns



Contribution. How to learn under Markovian clients' availability?

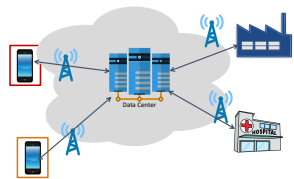
Angelo Rodio et al. (2023). "Federated Learning under Heterogeneous and Correlated Client Availability". In: *IEEE/ACM Transactions on Networking*

System Heterogeneity

Clients exhibit variations in hardware specifications, network connectivity types, and power availability. Three distinct scenarios:

3. Heterogeneous hardware environments
Why the same model architecture when clients have different capabilities?

Contribution. How to relieve the most powerful clients from the need to align their model to the weakest ones?

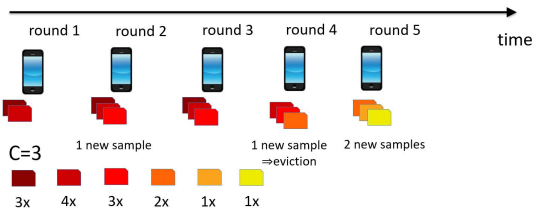


Othmane Marfoq et al. (2022). "Personalized Federated Learning through Local Memorization". In: *ICML'22*

Temporal Heterogeneity

FL in dynamic environments remains relatively unexplored

- Learning on static datasets is sub-optimal/impossible
  - new samples are ignored
  - clients have limited memory
- Distributed concept drift



Contributions. How to learn from distributed data streams in (1) the stationary regime, and in (2) the restricted adversarial settings?

- Othmane Marfoq et al. (2023). "Federated Learning for Data Streams". In: *AISTATS'23*
- Othmane Marfoq and Aryan Mokhtari (n.d.). "Online Federated Learning with Mixture Models".

Additional Contributions

Other challenges: privacy, expensive communications, robustness, free-riders, lack of standardized benchmarks...

Contributions:

- Comprehensive analysis of the role of reference data in empirical privacy defenses Caelin Kaplan et al. (2024). "A Cautionary Tale: On the Role of Reference Data in Empirical Privacy Defenses". In: *PoPETS'24*

- Open source cross-silo dataset suite for healthcare applications Jean Ogier du Terrail et al. (2022). "FLamby: Datasets and Benchmarks for Cross-Silo Federated Learning in Realistic Settings". In: NeurIPS'22 (Track on Datasets and Benchmarks)

Outline

1. Motivation & Summary of Contributions
2. A Focus on Tackling Statistical Heterogeneity via Personalization
   - Why do we Need to Personalize?
   - An Impossibility Result
   - Personalized Federated Learning under a Mixture of Distributions
   - Personalized Federated Learning through Local Memorization
   - A Comparison between FedEM and kNN-Per
3. Other Main Contributions
4. Conclusion

Future Research Directions

- Quantification of statistical heterogeneity
- Data-heterogeneity-aware topology design
- Privacy-preserving personalized federated learning
- Local cache update rules for federated learning
- Incentivizing client participation in federated learning

Conclusion

Thank you for your attention

References I

- Ben-David, Shai et al. (2008). "Does Unlabeled Data Provably Help? Worst-case Analysis of the Sample Complexity of Semi-Supervised Learning". In: *COLT*.
- Darnstädt, Malte et al. (2013). "Unlabeled Data Does Provably Help". In: *STACS*.
- Deng, Yuyang et al. (2020). "Adaptive Personalized Federated Learning". In: *arXiv preprint arXiv:2003.13461*.
- Dinh, Canh T et al. (2020). "Personalized Federated Learning with Moreau Envelopes". In: *arXiv preprint arXiv:2006.08848*.
- Göpfert, Christina et al. (2019). "When can unlabeled data improve the learning rate?" In: *Conference on Learning Theory*. PMLR, pp. 1500–1518.
- Kaplan, Caelin et al. (2024). "A Cautionary Tale: On the Role of Reference Data in Empirical Privacy Defenses". In: *PoPETS'24*.
- Mairal, Julien (2013). "Optimization with first-order surrogate functions". In: *International Conference on Machine Learning*, pp. 783–791.

References II

- Mansour, Yishay et al. (2020). "Three approaches for personalization with applications to federated learning". In: *arXiv preprint arXiv:2002.10619*.
- Marfoq, Othmane et al. (2020). "Throughput-Optimal Topology Design for Cross-Silo Federated Learning". In: *NeurIPS'20*.
- Marfoq, Othmane et al. (2021). "Federated Multi-Task Learning under a Mixture of Distributions". In: *NeurIPS'21*.
- Marfoq, Othmane et al. (2022). "Personalized Federated Learning through Local Memorization". In: *ICML'22*.
- — (2023). "Federated Learning for Data Streams". In: *AISTATS'23*.
- Marfoq, Othmane et al. (n.d.). "Online Federated Learning with Mixture Models".
- McMahan, Brendan et al. (2017). "Communication-efficient learning of deep networks from decentralized data". In: *Artificial Intelligence and Statistics*. PMLR, pp. 1273–1282.

References III

- Ogier du Terrail, Jean et al. (2022). "FLamby: Datasets and Benchmarks for Cross-Silo Federated Learning in Realistic Settings". In: NeurIPS'22 (Track on Datasets and Benchmarks).
- Rodio, Angelo et al. (2023). "Federated Learning under Heterogeneous and Correlated Client Availability". In: *IEEE/ACM Transactions on Networking*.
- Smith, Virginia et al. (2017). "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., pp. 4427–4437. ISBN: 9781510860964.
- Vanhaesebrouck, Paul et al. (2017). "Decentralized Collaborative Learning of Personalized Models over Networks". In: *AISTATS*.

Unseen Clients

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

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

Unseen Clients

E-step:

$$q_c^{k+1}(z_{c,i} = m) \propto \pi_{cm}^k \cdot \exp\left(-\ell(h_{\theta_m^k}(\mathbf{x}_{c,i}), y_{c,i})\right)$$

M-step:

$$\pi_{cm}^{k+1} = \frac{\sum_{i=1}^{n_c} q_c^{k+1}(z_{c,i} = m)}{n_c}$$

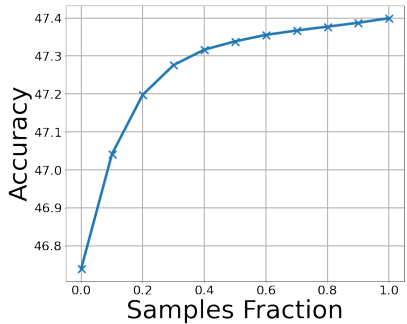


Figure: Effect of the number of samples on the average test accuracy across clients unseen at training.

Partial Participation

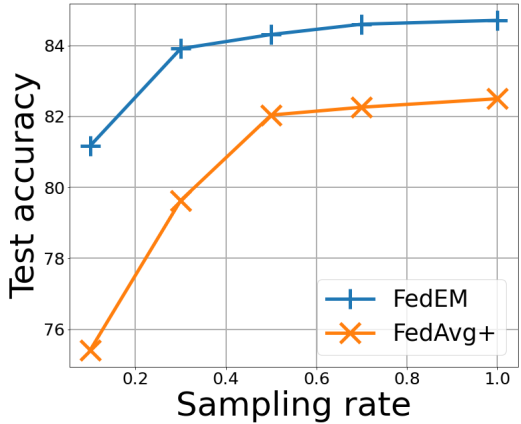


Figure: Effect of client sampling rate on the test accuracy for CIFAR10.

Local Memorization Mechanism: kNN-Per

At inference time, given input data $\mathbf{x} \in \mathcal{X}$, client $c \in [C]$

1. computes $h_{\mathcal{S}}(\mathbf{x})$ and the intermediate representation $\phi_{h_{\mathcal{S}}}(\mathbf{x})$.
2. queries $(\mathcal{K}_c, \mathcal{V}_c)$ with $\phi_{h_{\mathcal{S}}}(\mathbf{x})$ to retrieve its k-nearest neighbors $\mathcal{N}_c^{(k)}(\mathbf{x})$ according to a distance $d(\cdot, \cdot)$.
3. computes a local hypothesis $h_{\mathcal{S}_c}^{(k)}$, such that

$$\left[h_{\mathcal{S}_c}^{(k)}(\mathbf{x})\right]_y \propto \sum_{(\mathbf{x}', y') \in \mathcal{N}_c^{(k)}(\mathbf{x})} \mathbb{1}\{y = y'\} \times \exp\left\{-d\left(\phi_{h_{\mathcal{S}}}(\mathbf{x}), \phi_{h_{\mathcal{S}}}(\mathbf{x}')\right)\right\}.$$

4. outputs the final prediction

$$h_{c,\lambda}(\mathbf{x}) \triangleq \lambda \cdot h_{\mathcal{S}_c}^{(k)}(\mathbf{x}) + (1 - \lambda) \cdot h_{\mathcal{S}}(\mathbf{x}).$$