The Internet of Federated Things (IoFT)

Raed Al Kontar
Assistant Professor, University of Michigan
Website: umich.edu/∼alkontar, Email: alkontar@umich.edu

Recent vision paper

Talk is partially based on the paper titled "The Internet of Federated Things (IoFT)" [9], led by the University of Michigan and written in collaboration with multiple universities and faculty with a wide variety of expertise.

Outline

1. The Internet of Things (IoT)

Outline

1. The Internet of Things (IoT)
2. The Internet of Federated Things (IoFT)
   - Defining properties
   - Application snapshot

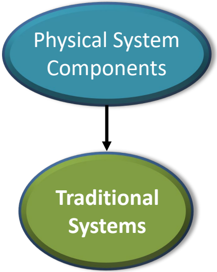
Outline

1. The Internet of Things (IoT)
2. The Internet of Federated Things (IoFT)
    - Defining properties
    - Application snapshot
3. Some Efforts on Federated Analytics
    - I. Personalized Predictive Analytics
    - II. Personalized Feature Extraction
    - III. Federated Bayes, Fairness & Others

* Disclaimer: A snapshot on opportunities within a field still in its infancy

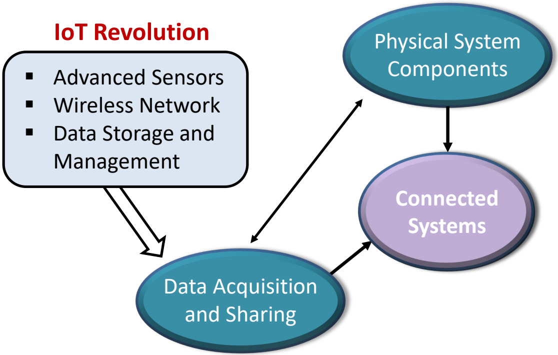
The Internet of Things (IoT)

- Smart and connected systems are transforming the competition and redefining the industry [17]



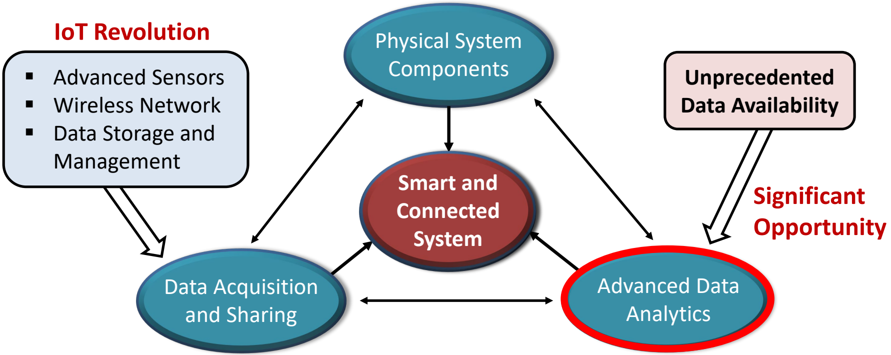
The Internet of Things (IoT)

- Smart and connected systems are transforming the competition and redefining the industry [17]



The Internet of Things (IoT)

- Smart and connected systems are transforming the competition and redefining the industry [17]



Basic feature

- **Connected**: Data from multiple similar units and from multiple components within the system are collected, often in real-time.
- **Smart**: Compare operations, share the information, and extract common knowledge to enable accurate prediction and control.
- **Old Notion**: Dates back to the time when artisans used to gather to share knowledge and perfect/standardize the quality of their crafted product.

Current IoT system

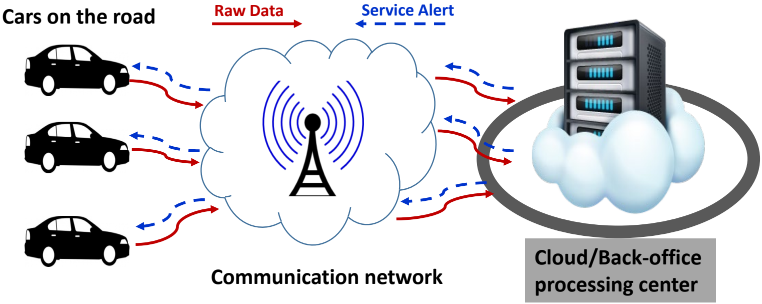


Figure 1: Example: Ford Sync or GM Onstar tele-service system

- Gigantic amounts of data are uploaded and stored in the cloud.
- Models (such as predictive maintenance, diagnostics, text prediction) are trained in these data centers.
- Models are then deployed to the edge devices.

Obvious drawbacks

- Is all the data utilized ?
- Communication burden
- Storage burden
- Deployment latency
- Energy cost of training large models
- Privacy * benefits large enterprises capable of building their own private cloud infrastructures at the expense of smaller entities.

What is changing in IoT?

- Computational power of edge devices is steadily increasing (as well as communication capabilities).

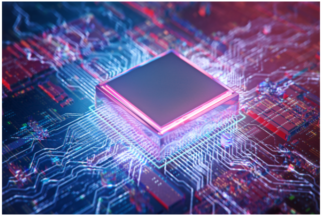


Figure 2: AI chips [4]

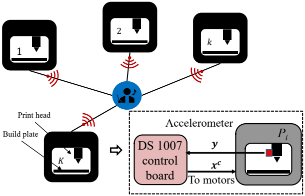


Figure 3: Smart 3D printers with Raspberry Pi's

What is changing in IoT?

- Computational power of edge devices is steadily increasing (as well as communication capabilities).

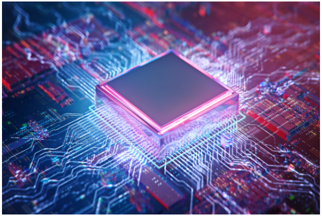


Figure 2: AI chips [4]

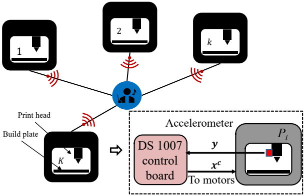


Figure 3: Smart 3D printers with Raspberry Pi's

- Tesla autopilot system has 150 million times more compute power than Apollo 11.

What is changing in IoT?

- Computational power of edge devices is steadily increasing (as well as communication capabilities).

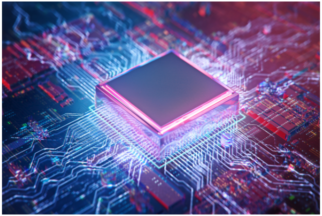


Figure 2: AI chips [4]

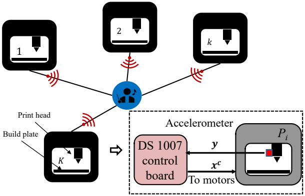


Figure 3: Smart 3D printers with Raspberry Pi's

- Tesla autopilot system has 150 million times more compute power than Apollo 11.
- Smart phones now have computational capabilities comparable to every day use laptops

The Internet of Federated Things (IoFT)

Federated Data Analytics (FDA): New data analytics paradigm within IoT

Exploits edge compute resources to process more of users' data where it's created.

The Internet of Federated Things (IoFT)

Federated Data Analytics (FDA): New data analytics paradigm within IoT

Exploits edge compute resources to process more of users' data where it's created.

Simple but powerful idea

With the availability of computing resources at the edge, IoT clients execute small computations locally and share the minimum information needed to learn a model

IoT moves from the "Cloud" to the "Crowd"

Our body filters external stimuli

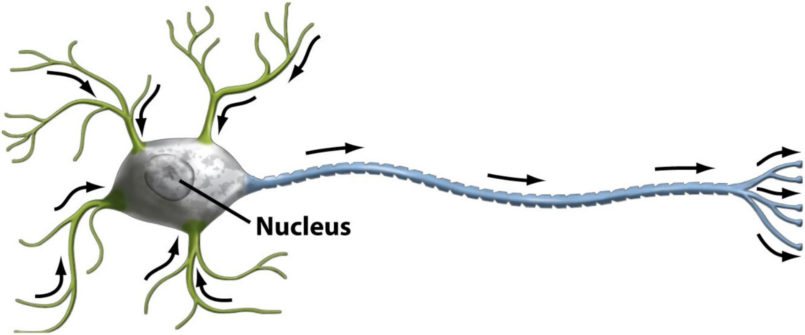
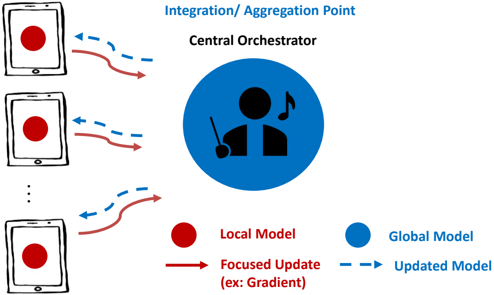


Figure 4: Information Flow via dendrites [1]

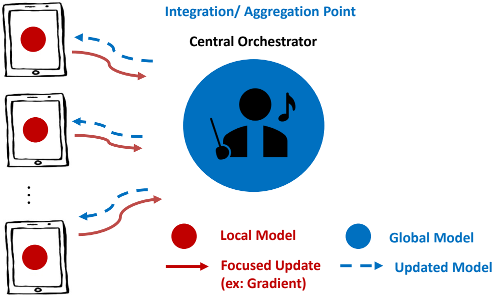
- Dendrites collect electric signals from different external stimuli
- Cell body integrates and condenses the signals
- Axons pass signals along
- **Our brain needs a healthy attention filter**

Simple example: exploiting edge compute resources



- How to learn the mean ($\bar{y}$) of a single feature (y) over all clients ?
- Exploiting compute capabilities, client i calculates $\bar{y}_i$
- Client i shares $\bar{y}_i$ instead of their entire feature vector ($\boldsymbol{y}_i$)
- $\bar{y}_i$ is a sufficient statistic to learn $\bar{y}$

An example of federated data analytics (FDA)



- IoFT devices perform local computations and report focused update to the orchestrator
- The orchestrator aggregates focused updates to update the global model
- This procedure is then iterated until a stopping criterion is met
- Global model goes through a quality testing on held-out devices

Why is IoFT Revolutionary ?

- **Privacy**: Users no longer have to share their valuable information, instead, it is kept local.
- **Computation and Energy**: No more fitting large models on the cloud. By exploiting edge compoute power, massive parallelization becomes a reality
- **Cost**:
  - Less information is transmitted to the cloud $\rightarrow$ less communication costs and efficient bandwidth ultilization.
  - Storage costs on cloud are minimal
- **Fast Alerts and Decisions**: Real-time decisions or service alerts can be achieved locally at the edge $\rightarrow$ no latency.

Why is IoFT revolutionary ?

- **Fast encryption** :Encryption of focused updates can be done readily and with better guarantees compared to encrypting entire datasets.
- **Resilience**: Edge devices are resilient to failures at the orchestrator level due to the existence of a local model.
- **Diversity and Fairness**: IoFT allows integrating information across uniquely diverse datasets, some of which have been restricted to be shared previously (ex: Medical institutes)
- **Minimal Infrastructure**: due to the increase compute power at the edge and AI chip penetration
- **Autonomy**: IoFT devices can be under independent control and opt-out of the collaborative training process at any time.

Difference from distributed learning

Distributed learning is often implemented to alleviate the huge computational burden via parallelization.

- Centralized systems where clients are compute nodes connected by large bandwidth.
- Follows a divide & conquer philosophy.

Difference from distributed learning

Distributed learning is often implemented to alleviate the huge computational burden via parallelization.

- Centralized systems where clients are compute nodes connected by large bandwidth.
- Follows a divide & conquer philosophy.

In IoFT, the data lives at the edge

- Data partitions are fixed and cannot be changed, shuffled, nor randomized.
- Devices have limited communication bandwidth with unstable or slow connection

Price to pay for privacy

Applications

Industry Interest

Industries have realized the disruptive potential of IoFT.
Google (Gboard, Android 13), Apple (QuickType keyboard), Microsoft (telemetry data), Facebook, some health care institutes, amongst many others.

Efforts are in their infancy phase

- Mostly tailored for mobile applications
- Methods focus on first order methods for deep learning
- A lot of development is needed for IoFT to become a norm in different industries

General purpose challenges

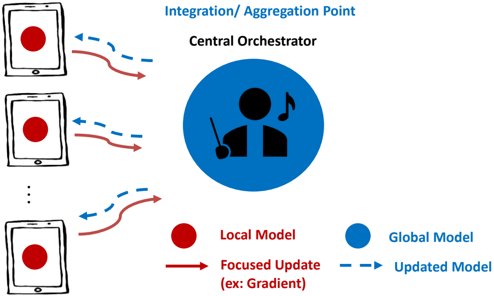
- Now let us discuss some challenges, opportunities & potential solutions

* Applications will dictate many challenges

FDA part I: Personalization via domain adaptation

Personalized Federated Learning via Domain Adaptation with Application to Distributed Manufacturing, *Technometrics*.

Global modeling



Global Modeling: One model to fit them all

Global modeling

- Assume N clients, the goal of FDA in IoFT is to collaboratively learn a global model $f_{\boldsymbol{w}}$ parametrized by $\boldsymbol{w}$

$$\min_{\boldsymbol{w}} F(\boldsymbol{w}) := \sum_{i=1}^{N} p_i F_i(\boldsymbol{w}), \tag{1}$$

where p_i is a weight (ex: $1/N$ or $n_i / \sum_{i=1}^{N} n_i$) and $F_i(\boldsymbol{w})$ is a risk function

$$F_i(\boldsymbol{w}) = \mathbb{E}_{(x_i, y_i) \sim \mathcal{D}_i} \left[\ell(f_{\boldsymbol{w}}(x_i), y_i) \right] \approx \frac{1}{n_i} \sum_{j=1}^{n_i} \left[\ell(f_{\boldsymbol{w}}(x_j), y_j) \right]$$

In IoFT, client i can only evaluate its own risk function $F_i(\boldsymbol{w})$ and orchestrator has no access to client datasets $D_i \sim \mathcal{D}_i$

Sample FDA framework

Sample FDA framework with weight sharing

1: **Input:** Client datasets $\{D_i\}_{i=1}^{N}$, T, number of local steps E, initialization for $\boldsymbol{w}$
2: **for** $t = 1, 2, \cdots T$ **do**
3: Orchestrator selects a subset of clients $\mathcal{S} \subseteq [N]$ and broadcasts global model $\boldsymbol{w}^t$

Sample FDA framework

Sample FDA framework with weight sharing

Input: Client datasets $\{D_i\}_{i=1}^N$, T, number of local steps E, initialization for $\boldsymbol{w}$

for $t = 1, 2, \cdots T$ **do**

  Orchestrator selects a subset of clients $\mathcal{S} \subseteq [N]$ and broadcasts global model $\boldsymbol{w}^t$

  for each $i \in \mathcal{S}$ **do**

    Client update: $\boldsymbol{w}_i^{t+1} = \mathit{client\_update}(\boldsymbol{w}^t, D_i, E)$

    Clients send updated parameters $\boldsymbol{w}_i^{t+1}$ to server.

  end for

Sample FDA framework

Sample FDA framework with weight sharing

Input: Client datasets $\{D_i\}_{i=1}^N$, T, number of local steps E, initialization for $\boldsymbol{w}$
for $t = 1, 2, \cdots T$ **do**
  Orchestrator selects a subset of clients $\mathcal{S} \subseteq [N]$ and broadcasts global model $\boldsymbol{w}^t$
  for each $i \in \mathcal{S}$ **do**
    Client update: $\boldsymbol{w}_i^{t+1} = \textit{client\_update}(\boldsymbol{w}^t, D_i, E)$
    Clients send updated parameters $\boldsymbol{w}_i^{t+1}$ to server.
  end for
  Orchestator update: $\boldsymbol{w}^{t+1} = \textit{server\_update}\left(\left\{\boldsymbol{w}_i^{t+1}\right\}_{i \in \mathcal{S}}\right)$
end for

Popular approach: `FedAvg` [16]

- **Client update**: running several E steps of stochastic gradient descents (SGD). More specifically, for $e = 0, \ldots, E-1$,

$$\boldsymbol{w}_i^{t+1,e+1} \leftarrow \boldsymbol{w}_i^{t+1,e} - \eta \nabla F_i(\boldsymbol{w}_i^{t+1,e}).$$

- **Orchestator update**: taking **average** of clients' model parameters:

$$\boldsymbol{w}^{t+1} = \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \boldsymbol{w}_i^{t+1,E}$$

where $|\mathcal{S}|$ denotes the cardinality of the set $\mathcal{S}$.

Heterogeneity: client drift

- Global and empirical risk different when data are *non-i.i.d*

$$F(\boldsymbol{w}^*) \neq \sum_{i=1}^{N} p_i F_i(\boldsymbol{w}_i^*)\,,$$

  where superscript * indicates some critical point. This phenomenon is known as "client-drift".
- Wide gap in a global model's performance across different devices when heterogeneity exists [6, 5, 20, 19, 7]
- Global model will be biased to devices with more data [13].

DistributedSGD vs Fedavg

- If one (i.e., $E = 1$) step ([15]), averaging weights and gradients is equivalent

$$\mathbb{E}_i \left[\boldsymbol{w}^t - \eta \nabla F_i(\boldsymbol{w}^t) \right] = \boldsymbol{w}^t - \eta \mathbb{E}_i \left[\nabla F_i(\boldsymbol{w}^t) \right].$$

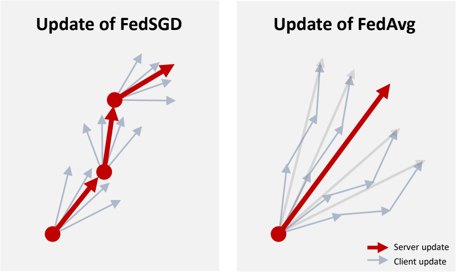
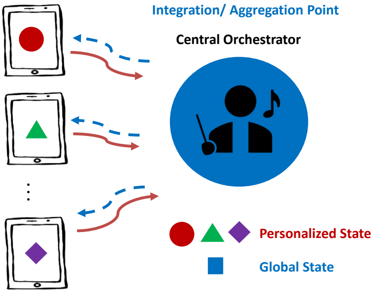


Figure 5: FedAvg vs DistrSGD

.

Personalized modeling



- IoFT devices often exhibit highly heterogeneous trends due to differences in operational, environmental, cultural, socio-economic and specification conditions [10, 11, 22]

Current approach

Learn $y_i = f_{\boldsymbol{\theta}_i}(x)$. A general objective for personalized FDA:

$$\min_{\boldsymbol{w},\boldsymbol{\theta}} F(\boldsymbol{w},\boldsymbol{\theta}) := \frac{1}{N}\sum_{i=1}^{N} F_i(\boldsymbol{w},\boldsymbol{\theta}_i)\,, \tag{2}$$

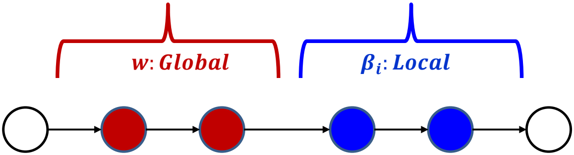
where $\boldsymbol{w}$ are shared global parameters while $\boldsymbol{\theta} = \{\boldsymbol{\theta}_i\}_{i=1}^{N}$ is a set of unique parameters for each client.

Approaches

- Weight sharing
- Regularization

Weight sharing

- The first set of literature solve (2) by using different layers of a neural network to represent $\boldsymbol{w}$ and $\boldsymbol{\theta}_i$ [21, 14].
- The underlying idea is that base layers process the input to learn a shared feature representation across clients, and top layers learn task-dependent weights based on the feature.

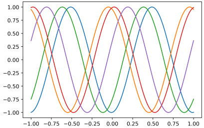


Regularization: train-then-personalize

- Learn global parameters $\boldsymbol{w}^*$ then regularize
- Proximal term
$$\min_{\boldsymbol{\theta}_i} \left(F_i(\boldsymbol{\theta}_i) + \frac{\mu}{2} \|\boldsymbol{\theta}_i - \boldsymbol{w}^*\|^2 \right)$$
- Similar to popular elastic weight consolidation model (EWC) [8]
$$\min_{\boldsymbol{\theta}_i} \left(F_i(\boldsymbol{\theta}_i) + \frac{\mu}{2} \sum_j \mathcal{F}\mathcal{I}_j \left\|\theta_j - w_j^*\right\|^2 \right),$$
where $\mathcal{F}_j$ are diagonal elements of the Fisher information
- Learning $\boldsymbol{w}^*$ and $\boldsymbol{\beta}_i$'s also done iteratively [12, 2]

Counter example

$$f_{u_i}(x) = \sin(2\pi(x + u_i)) \quad u_i \sim \mathcal{U}[0, 1]$$



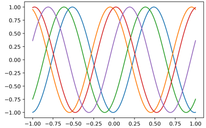
If we train a global model to minimize the population risk:

$$\min_{w} \mathbb{E}_i[||f_{\boldsymbol{w}} - f_{u_i}||_2^2]; \quad ||f||_2^2 = \int_0^1 f(x)^2 dx$$

- Then $f_{\boldsymbol{w}}$ should minimize: $\arg\min_{f_{\boldsymbol{w}}} \mathbb{E}_{u_i}\left[\int_0^1 f_{\boldsymbol{w}}(x)^2 dx\right]$
- The unique minimizer is $f_{\boldsymbol{w}_{zero}}(x) = 0$ for every x in $[0, 1]$

Counter example

$$f_{u_i}(x) = \sin(2\pi(x + u_i)) \quad u_i \sim \mathcal{U}[0, 1]$$



If we train a global model to minimize the population risk:

$$\min_{w} \mathbb{E}_i[||f_{\boldsymbol{w}} - f_{u_i}||_2^2]; \quad ||f||_2^2 = \int_0^1 f(x)^2 dx$$

- Then $f_{\boldsymbol{w}}$ should minimize: $\arg\min_{f_{\boldsymbol{w}}} \mathbb{E}_{u_i}\left[\int_0^1 f_{\boldsymbol{w}}(x)^2 dx\right]$
- The unique minimizer is $f_{\boldsymbol{w}_{zero}}(x) = 0$ for every x in $[0, 1]$
- Regularization augments problem

$$\int_0^1 \left(f_{\boldsymbol{\beta}_i}(x) - \sin\left(2\pi x + 2\pi u_i\right)\right)^2 dx + \lambda||\boldsymbol{\beta}_i - \boldsymbol{w}_{zero}||^2.$$

Structure of heterogeneity

Heterogeneity

$$\mathbb{P}^i_{x,y} = \mathbb{P}^i_x \times \mathbb{P}^i_{y|x}$$

- **Concept Shift** - current literature

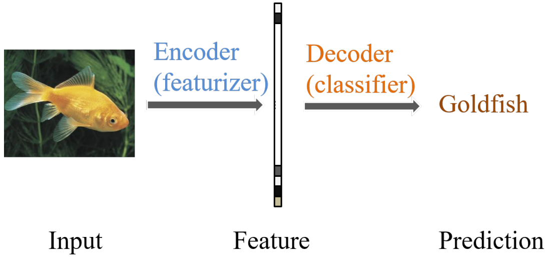
$$y_i = f_{\boldsymbol{\theta}_i}(x_i) \qquad x \sim \mathbb{P}_x$$

  Clients share the same f (a linear model, neural network) yet with different parameters $\boldsymbol{w}_i$

- **Covariate Shift**: $x_i \sim \mathbb{P}^i_x$

Domain Adaptation

Domain adaptation is a natural approach to handle the covariate shift.



$$f_{\boldsymbol{\theta}_i}(x) = g_{\gamma_i} \circ \Phi_{\boldsymbol{\beta}_i}(x) \tag{3}$$

- Encoder: $\Phi_{\beta_i} : \mathcal{X}_i \to \mathcal{H}$, output features in similar distributions $\to$ Handle covariate shift
- Decoder $g_{\gamma_i} : \mathcal{H} \to \mathcal{Y}$, classify features in similar distributions $\to$ Handle concept shift

Domain Adaptation: Bi-level Optimization

- How do we achieve (almost) domain-invariant features?
- Bi-level optimization:

$$\min_{\gamma_i} \tilde{F}_i(\boldsymbol{\gamma}_i, \boldsymbol{\beta}_i, \boldsymbol{w})$$

$$\text{s.t. } \{\boldsymbol{w}, \{\boldsymbol{\beta}_i\}\} \in \arg \min_{\widetilde{\boldsymbol{w}}, \{\widetilde{\boldsymbol{\beta}}_i\}} \sum_{i=1}^{N} p_i F_i(\widetilde{\boldsymbol{\beta}}_i, \widetilde{\boldsymbol{w}})$$

where F_i is the empirical loss on client i, and $\widetilde{F}_i$ is the regularized loss:

$$\tilde{F}_i = \frac{1}{n_i} \sum_{(x,y)\in D_i} \ell\left[y, g_{\boldsymbol{\gamma}_i}(\Phi_{\boldsymbol{\beta}_i}(x))\right] + \lambda_1 \left\|\boldsymbol{\gamma}_i - \boldsymbol{w}\right\|^2 .$$

- Inner level: train encoders with the help of a single decoder.
- Outer level: personalize decoders

Domain Adaptation: Train Encoders

- Use a single decoder function $g_{\boldsymbol{w}}$ to minimize:

$$\min_{\boldsymbol{w},\{\boldsymbol{\beta}_i\}} \sum_{i=1}^{N} p_i F_i(\boldsymbol{\beta}_i, \boldsymbol{w})$$

  where

$$F_i(\boldsymbol{\beta}_i, \boldsymbol{w}) = \frac{1}{n_i} \sum_{(x,y)\in D_i} \ell\left[y, g_{\boldsymbol{w}}(\Phi_{\boldsymbol{\beta}_i}(x))\right]$$

  is the empirical risk.
- $\Phi_{\boldsymbol{\beta}_i}(x_j)$'s learn common features from heterogeneous domains.
- $g_{\boldsymbol{w}}$ promotes learning of domain invariant features.

Domain Adaptation: Personalize Decoders

- Personalize decoders based on learned encoders.

$$\min_{\gamma_i} \tilde{F}_i(\boldsymbol{\gamma}_i, \boldsymbol{\beta}_i, \boldsymbol{w}) = \frac{1}{n_i} \sum_{(x,y)\in D_i} \ell \left[y, g_{\boldsymbol{\gamma}_i}(\Phi_{\boldsymbol{\beta}_i}(x))\right] + \lambda_1 \left\|\boldsymbol{\gamma}_i - \boldsymbol{w}\right\|^2.$$

- Use regularization since features admit similar distributions.
- $\boldsymbol{w}$ is a reference point to γ_i.
- γ_i learns the concept shifts.

Training

In each communication round:

- Client i:
  1. β_i^{t+1}, $\boldsymbol{w}_i^{t+1}$ updated using $\nabla_{\boldsymbol{w},\beta_i} F_i(\boldsymbol{w}_i^{t,q}, \beta_i^{t,q})$
  2. γ_i^{t+1} updated from $\nabla_{\gamma_i} \tilde{F}_i(\boldsymbol{w}_i^{t,q}, \beta_i^{t,q})$
- Server update:
  1. $\boldsymbol{w}^{t+1} = \sum_{i=1}^{N} p_i \boldsymbol{w}_i^{t+1}$

Convergence

If we E steps of local gradient descent for local updates, the algorithm converges under mild conditions.

Theorem (informal)

If all local objectives F_i are gradient Lipschitz continuous and the norm of the gradient of F_i and $\tilde{F}_i$ over $\boldsymbol{w}$, β_i's and γ_i's are all bounded:

$$\min_{t\in\{1,\cdots T\},q\in\{0,\cdots,E-1\}} \left[\left\| \sum_{k=1}^{N} p_i \nabla_{\boldsymbol{w}} F_i(\hat{\boldsymbol{w}}^{t,q}, \beta_i^{t,q}) \right\|^2 + \sum_{i=1}^{N} p_i \left\| \nabla_{\beta_i} F_i(\hat{\boldsymbol{w}}^{t,q}, \beta_i^{t,q}) \right\|^2 \right] \leq O\left(\frac{\log T}{\sqrt{T}} \right)$$

and

$$\min_{t\in\{1,\cdots T\},q\in\{0,\cdots,E-1\}} \left[\sum_{i=1}^{N} p_i \left\| \nabla_{\gamma_i} \tilde{F}_i(\hat{\boldsymbol{w}}^{t,q}, \beta_i^{t,q}, \gamma_i^{t,q}) \right\|^2 \right] \leq O\left(\frac{\sqrt{\log T}}{T^{\frac{1}{4}}} \right)$$

Testing: The Sine Counterexample

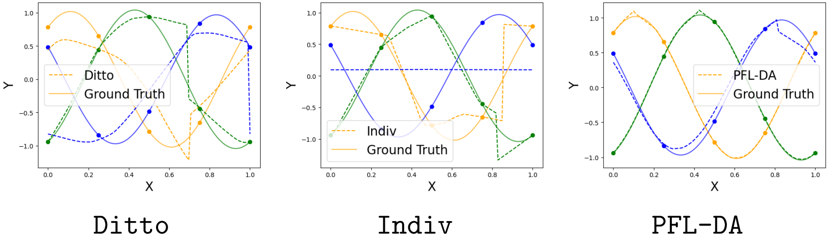


Figure 6: Regression of sine functions by three algorithms.

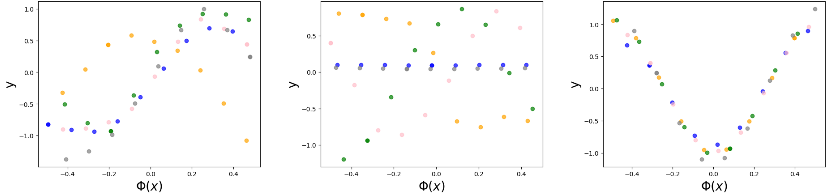


Figure 7: Learned decoder functions.

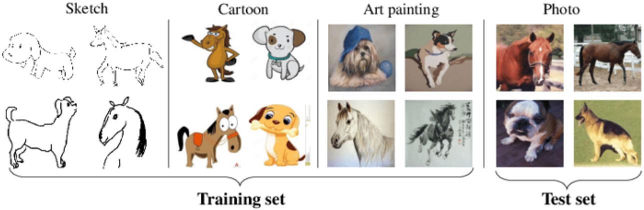
Insight

- $f_i(x) = \alpha_i \sin(2\pi(x+\theta_i))$, $\theta_i \sim \mathcal{U}[0,1]$ and $\alpha_i \sim \mathcal{N}(\mu_\alpha, \sigma_\alpha^\in)$
- Then with correct parametrizations, the optimal solution
  - $g_w(\phi) = \mu_a \sin(2\pi\phi)$. "center" of all decoder functions
  - $g_{\gamma_i}(\phi) = \gamma_i \sin(2\pi\phi)$

$$\gamma_i^\star = \frac{\alpha_i + 2\lambda_1\mu_\alpha}{1+2\lambda_1}$$

  weighted average of: α_i, the amplitude of the sine function on client i, and μ_α which is the average of all amplitudes.

Testing



Dataset	Fedavg	Indiv	TP	Ditto	Simple-DA	PFL-DA
CMNIST	68.8±0.2	75.6±0.2	54.5±0.1	71.8±0.2	75.6±0.2	**75.8**±0.1
RMNIST	93.8±0.4	**98.4**±0.1	93.9±0.2	93.9±0.2	**98.4**±0.1	**98.4**±0.1
FEMNIST	77.9±0.3	61.7 ±0.3	77.7±0.3	80.2±0.3	46.0±3	**80.8**±0.2
VLCS	82.8±0.3	82.5±0.3	82.7±0.3	82.4±0.3	82.6±0.1	**83.7**±0.1
PACS	84.4±0.8	93.9±0.4	85.0±1.9	92.7±0.2	94.4±0.5	**95.6**±0.1

Table 1: Average Test Accuracies

Testing: Distributed 3D Printers

- Data from 3D printers are collected by Raspberry Pis.
- Task: predict printhead vibrations at given printing speed.

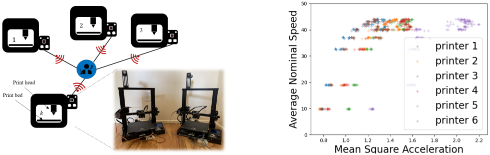


Figure 8: Left: an illustration of FL system consisting of 3D printers and Raspberry Pis. Right: the data collected.

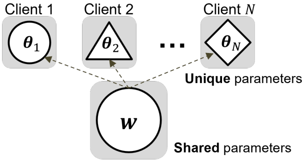
The proposed method `PFL-DA` has better predictive performance.

Table 2: Test loss with standard deviations.

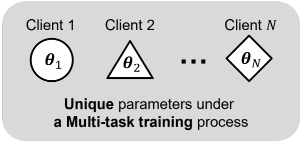
Model	`Ditto`	`indiv`	`PFL-DA`
Neural Network	0.48 ± 0.04	0.25 ± 0.02	$\mathbf{0.23} \pm 0.02$
sigmoid GLM	0.28 ± 0.02	0.268 ± 0.01	$\mathbf{0.23} \pm 0.01$
Gaussian GLM	0.26 ± 0.04	$\mathbf{0.25} \pm 0.01$	$\mathbf{0.25} \pm 0.01$

MTL has rich literature in centralized regimes [3, 25].

$$\min_{\boldsymbol{\theta},\Omega} \left\{ \sum_{i=1}^{N} p_i F_i(\theta_i) + \mathcal{R}(\boldsymbol{\theta}, \Omega) \right\}$$



Weight sharing (or regularization)



Multi-task learning-based

FDA part II: Personalized & Federated PCA

Personalized PCA: Decoupling Unique and Shared Features [18]

FDA part II: Personalized PCA

- An example of the application of Personalized PCA on video segmentation.

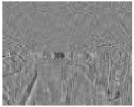
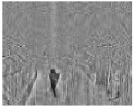
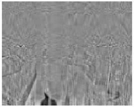
Image	1	2	3
Original			
Global PC			
Local PC			

Table 3: Video segmentation.

FDA part II: Personalized PCA

- Analyze data variance through principal component analysis.
- Global components $\boldsymbol{u}_q$'s: shared information
- Llobal components $\boldsymbol{v}_{(i),q}$'s: unique pattern
- Dataset i is modeled as:

$$\boldsymbol{y}_{(i)} \sim \underbrace{\sum_{q=1}^{r_1} \phi_{(i),q} \boldsymbol{u}_q}_{r_1 \text{ global components}} + \underbrace{\sum_{q=1}^{r_2} \varphi_{(i),q} \boldsymbol{v}_{(i),q}}_{r_2 \text{ local components}} + \underbrace{\boldsymbol{\epsilon}_{(i)}}_{noise}$$

  $\phi_{(i),q}$ and $\varphi_{(i),q}$ are data-dependent coefficients.
- **Decoupled** features: **global and local components are orthogonal**.

$$\left\langle \boldsymbol{u}_q, \boldsymbol{v}_{(i),q} \right\rangle = 0$$

Personalized PCA

- An example on personalized federated PCA.

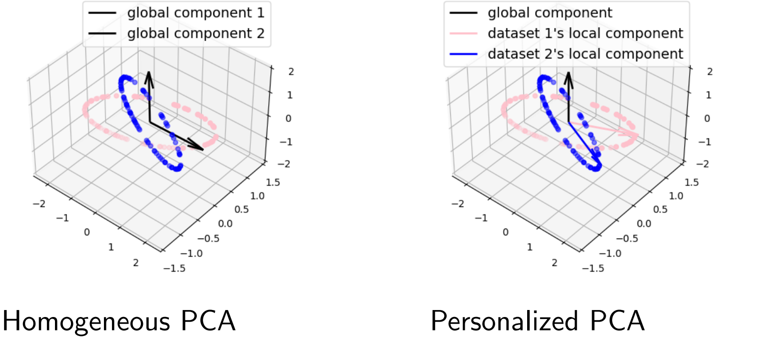


Figure 9: Comparison between Homogeneous PCA and personalized PCA.

- Personalized PCA learns better features.

Personalized PCA

- Task: recover features $\boldsymbol{u}_q$'s and $\boldsymbol{v}_{(i),q}$'s from (noisy) observations.
- In matrix form:

$$\begin{cases} \boldsymbol{U} = [\boldsymbol{u}_1, \cdots, \boldsymbol{u}_{r_1}] \\ \boldsymbol{V}_{(i)} = [\boldsymbol{v}_{(i),1}, \cdots, \boldsymbol{v}_{(i),r_2}] \\ \boldsymbol{Y}_{(i)} = [\boldsymbol{y}_{(i),1}, \cdots, \boldsymbol{y}_{(i),n_i}] \end{cases}$$

- Given N datasets, the objective is:

$$\max_{\boldsymbol{U}, \{\boldsymbol{V}_{(i)}\}_{i=1,\cdots,N}} \frac{1}{2}\mathrm{Tr}\left(\boldsymbol{U}^T \boldsymbol{S}_{(i)} \boldsymbol{U}\right) + \frac{1}{2}\mathrm{Tr}\left(\boldsymbol{V}_{(i)}^T \boldsymbol{S}_{(i)} \boldsymbol{V}_{(i)}\right) \quad (4)$$
$$\text{subject to } \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I},\ \boldsymbol{V}_{(i)}^T\boldsymbol{V}_{(i)} = \boldsymbol{I},\ \boldsymbol{V}_{(i)}^T\boldsymbol{U} = \boldsymbol{0},\ \forall i$$

- $\boldsymbol{S}_{(i)} = \boldsymbol{Y}_{(i)}\boldsymbol{Y}_{(i)}^T$ is the data covariance matrix on client i.
- Nonconvex constraints.
- Identifiable? Learnable?

Personalized PCA: Identifiability

- Given observed data, is it possible to recover the global end local subspaces?
- Eckart–Young theorem (PCA) - the solution to:

$$\arg\max_{\boldsymbol{U}} \frac{1}{2}\mathrm{Tr}\left(\boldsymbol{U}^T\boldsymbol{S}\boldsymbol{U}\right)$$
$$\text{subject to } \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I} \;\; rank(\boldsymbol{U}) = r$$

for PSD $\boldsymbol{S}$ is $\boldsymbol{U}^*$, then the column space of $\boldsymbol{U}^*$ spans the top-r eigenspace of $\boldsymbol{S}$.

Personalized PCA: Identifiability

- Given observed data, is it possible to recover the global end local subspaces?
- Eckart–Young theorem (PCA) - the solution to:

$$\arg\max_{\boldsymbol{U}} \frac{1}{2}\mathrm{Tr}\left(\boldsymbol{U}^T\boldsymbol{S}\boldsymbol{U}\right)$$
$$\text{subject to } \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I} \;\; rank(\boldsymbol{U}) = r$$

  for PSD $\boldsymbol{S}$ is $\boldsymbol{U}^*$, then the column space of $\boldsymbol{U}^*$ spans the top-r eigenspace of $\boldsymbol{S}$.
- Does **not** apply to personalized PCA.
- Simple reasoning: if all clients have the same global and local principal components, we cannot tell which are global and which are local.
- New conditions are required.

Personalized PCA: Identifiability

- $\mathbf{\Pi}_{(i)}$ the projections onto the subspace spanned by true local PCs: $\mathbf{\Pi}_{(i)} = \boldsymbol{V}_{(i),\text{true}}\boldsymbol{V}_{(i),\text{true}}^T$.
- $\mathbf{\Pi}_g$: true global projection
- **Identifiability assumption**: We assume there exists a positive constant $\theta \in (0, 1)$ such that:

$$\lambda_{\max}\left(\frac{1}{N}\sum_{i=1}^{N}\mathbf{\Pi}_{(i)}\right) \leq 1-\theta \tag{5}$$

- $0 \leq \lambda_{\max}\left(\frac{1}{N}\sum_{i=1}^{N}\mathbf{\Pi}_{(i)}\right) \leq \frac{1}{N}\sum_{i=1}^{N}\lambda_{\max}\left(\mathbf{\Pi}_{(i)}\right) = 1$

θ as a heterogeneity metric

θ measures subspace differences.

Personalized PCA: Identifiability

- $\boldsymbol{P_U} = \boldsymbol{U}\boldsymbol{U}^T$ is the projection to the column space of $\boldsymbol{U}$

Theorem (informal)

If the population covariance matrices satisfy the identifiability assumption, and have eigengaps larger than δ, and the noise is sub-Gaussian, then with probability at least 1-$\widetilde{\delta}$, we have:

$$\left\|\boldsymbol{P}_{\hat{\boldsymbol{U}}} - \boldsymbol{\Pi}_g\right\|_2^2 + \frac{1}{N}\sum_{i=1}^{N}\left\|\boldsymbol{P}_{\hat{\boldsymbol{V}}_{(i)}} - \boldsymbol{\Pi}_{(i)}\right\|_2^2 \leq O\left(\frac{d + \log\frac{2N}{\widetilde{\delta}}}{n\theta\delta^2}\right)$$

where $\hat{\boldsymbol{U}}$, and $\hat{\boldsymbol{V}}_{(i)}$'s are the optimal solution to the problem (4), d is data dimension, n is the number of samples per client, and N is the number of client.

- The global and local subspace can be recovered by solving a constrained optimization problem!

Personalized PCA: Stiefel Manifold

- Stiefel manifold: the manifold formed by all orthonormal matrices.

$$St(d, r) = \{\boldsymbol{U} \in \mathbb{R}^{d\times r} | \boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}\}$$

- **Clients**: Use Stiefel gradient descent to update $\boldsymbol{U}_{(i)}$ and $\boldsymbol{V}_{(i)}$. First projects the gradient to the tangent space

$$g_{(i),\tau} = \mathcal{P}_{\mathcal{T}_{\left[\boldsymbol{U}_\tau, \boldsymbol{V}_{(i),\tau}\right]}}\left(\boldsymbol{S}_{(i)}\left[\boldsymbol{U}_\tau, \boldsymbol{V}_{(i),\tau}\right]\right) \tag{6}$$

- **Server**: Aggregates $\boldsymbol{U}_{(i)}$'s. Ex: calculates the average

$$\boldsymbol{U} \leftarrow \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{U}_{(i)})$$

.

Challenge

- After server calculates the average, $\boldsymbol{U}^T\boldsymbol{V}_{(i)} \neq 0$, the variables become infeasible.
- Inspired by Gram-Schmit orthonormalization, we introduce a correction step for $\boldsymbol{V}_{(i)}$ at the local client:

$$\boldsymbol{V}_{(i)} \leftarrow \boldsymbol{V}_{(i)} - \boldsymbol{U}\boldsymbol{U}^T\boldsymbol{V}_{(i)}$$

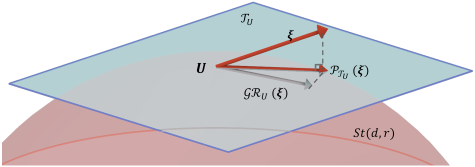
  BY projecting to column space of $\boldsymbol{U}$ and deflating $\boldsymbol{V}_{(i)}$, the resulting (deflated) matrix is orthogonal $\boldsymbol{U}$

- Still it does not lie on the Steifel Manifold (yet close, by theory!)

Generalized Retraction

- Generalized retraction:

$$\mathcal{GR}_{\boldsymbol{U}}(\cdot) : \mathbb{R}^{d\times r} \to St(d,r)$$



-
  1. preserves column space:
     $col(\mathcal{GR}_{\boldsymbol{U}}(\boldsymbol{\xi})) = col(\boldsymbol{U}+\boldsymbol{\xi}),\ \forall \boldsymbol{U} \in St(d,r),\ \forall \boldsymbol{\xi} \in \mathbb{R}^{d\times r}$
  2. is close to the projection to tangent space:
     $\|\mathcal{GR}_{\boldsymbol{U}}(\boldsymbol{\xi}) - (\boldsymbol{U} + \mathcal{P}_{\mathcal{T}_{\boldsymbol{U}}}(\boldsymbol{\xi}))\|_F \leq$
     $M_1 \|\mathcal{P}_{\mathcal{T}_{\boldsymbol{U}}}(\boldsymbol{\xi})\|_F^2 + M_2 \|\boldsymbol{\xi} - \mathcal{P}_{\mathcal{T}_{\boldsymbol{U}}}(\boldsymbol{\xi})\|_F,\ \forall \boldsymbol{U} \in St(d,r),\ \forall \boldsymbol{\xi} \in \mathbb{R}^{d\times r}$, for 2 constants $M_1, M_2 \geq 0$

- Polar projection is a generalized retraction:

$$\mathcal{GR}_{polar,\boldsymbol{U}}(\boldsymbol{\xi}) = (\boldsymbol{U}+\boldsymbol{\xi})\left(\boldsymbol{I} + \boldsymbol{\xi}^T\boldsymbol{U} + \boldsymbol{U}^T\boldsymbol{\xi} + \boldsymbol{\xi}^T\boldsymbol{\xi}\right)^{-\frac{1}{2}}$$

Generalized Retraction

- One key property of a generalized retraction is that it preserves the column space;

$$\boldsymbol{V}_{(i)} \leftarrow \mathcal{GR}_{\boldsymbol{V}_{(i)}}(\boldsymbol{V}_{(i)} - \boldsymbol{U}\boldsymbol{U}^T\boldsymbol{V}_{(i)})$$

- the retracted matrix is still orthogonal to $\boldsymbol{U}$.
- Also we do

$$\mathcal{GR}_{\boldsymbol{U}_\tau}\left(\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{U}_{(i),\tau+1}\right)$$

Personalized PCA: Algorithm

- In communication round τ:
  1. Client i receives $\boldsymbol{U}_\tau$ from server, then use it to correct $\boldsymbol{V}_{(i),\tau}$.
  2. Client i performs Stiefel gradient descent to obtain updates $\boldsymbol{U}_{(i),\tau+1}$ and $\boldsymbol{V}_{(i),\tau+1}$, then sends $\boldsymbol{U}_{(i),\tau+1}$ to server.
  3. Server averages received $\boldsymbol{U}_{(i),\tau+1}$'s, then retracts it to feasible set to obtain $\boldsymbol{U}_{\tau+1}$.
- Only global principal components are sent to the server!

Personalized PCA: Convergence

The algorithm has **convergence guarantee**

Theorem (informal)

If the stepsize η of Stiefel gradient descent is not too large, starting from general initializations, the updates $\{\boldsymbol{U}_\tau, \{\boldsymbol{V}_{(i),\tau}\}\}$ converge into (feasible) stationary points. Moreover, if the algorithm is initialized near the global optimum and the eigenvalues of sample covariance matrix are upper bounded by G_{max} and lower bounded by μ, the updates $\{\boldsymbol{U}_\tau, \{\boldsymbol{V}_{(i),\tau}\}\}$ converge into the true global and local subspace exponentially:

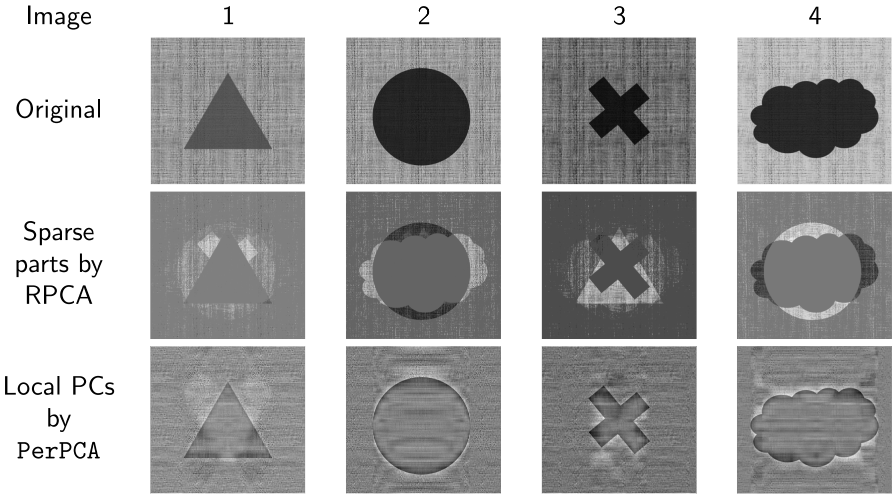
$$\|\boldsymbol{P}_{\boldsymbol{U}_\tau} - \boldsymbol{\Pi}_g\|_2^2 + \sum_{i=1}^{N} \left\|\boldsymbol{P}_{\boldsymbol{V}_{(i),\tau}} - \boldsymbol{\Pi}_{(i)}\right\|_2^2 \leq O\left(\left(1 - \frac{\eta\mu^2\omega(\theta)}{4NG_{max}}\right)^\tau\right)$$

where θ is the parameter in identifiability assumption and $\omega(\theta) = \frac{\theta^2 N}{(1+N)\left(1-\frac{\theta}{2}\right)+\sqrt{(1+N)^2\left(1-\frac{\theta}{2}\right)^2-\theta^2 N}}$

Personalized PCA: Convergence

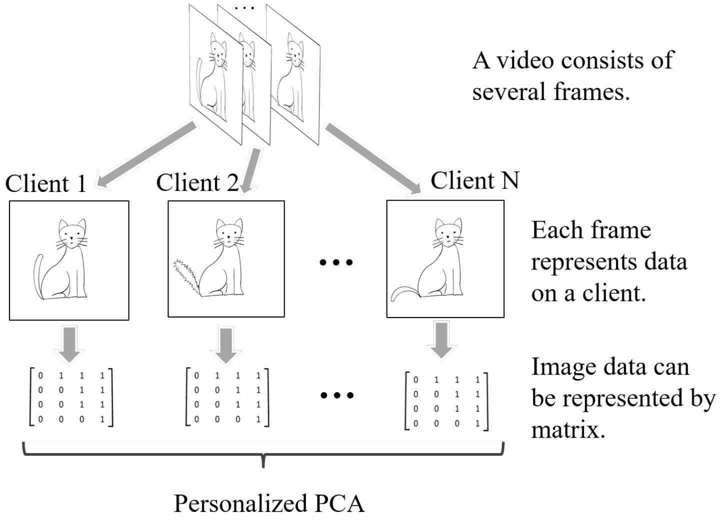
- We can recover the global and local subspaces.
- The algorithm converges faster when θ is larger!
- Comparison: In FDA, convergence is usually is slower for higher level of heterogeneity. Not the case here.

Comparison with Robust PCA



Personalized PCA: Video Segmentation

Personalized PCA can also solve the video segmentation task.



- Video segmentation. Global principal components capture stationary background. Local principal components capture moving parts.

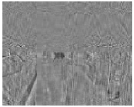
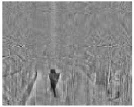
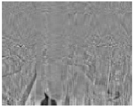
Image	1	2	3
Original			
Global PC			
Local PC			

Table 4: Video segmentation.

Personalized PCA: Topic Modeling

- Topic modeling in presidential debate transcripts, local components represent the most debated topics at the specific year.

Table 5: U.S. presidential debate key words.

Year	Top local principal components words
1960	peace, Castro, Africa, Kennedy, now, world, ...
1976	billion, Carter, Governor, Africa, Ford, people, world, ...
1980	coal, oil, money, energy, Social, Security, Reagan, ...
1984	Union, tax, Soviet, arms, leadership, proposal, ...
1988	drug, young, strong, build, future, enforcement, good, ...
1992	Bill, school, children, care, health, taxes, reform,plan, control, ...
1996	Clinton, Security, Medicare, budget, tax, Dole, Bob, ...
2000	school, public, plan, children, money, Social, Security, health, tax, ...
2004	wrong, plan, cost, free, Saddam, troops, Iraq, war, health, tax, ...
2008	nuclear, oil, troops, Iraq, Afghanistan, Pakistan, health, Iran, energy, ...
2012	million, small, business, China, Medicare, Romney, jobs, tax, ...
2016	Russia, Trump, Hillary, companies, taxes, Mosul, Iran, deal, ...
2020	Harris, Pence, Trump, down, Joe, Biden, jobs, Donald, health, ...
Common words	Tax, country, States, make, world, money, people, cut, ...

FDA part III: Other Efforts

- GIFAIR-FL: An Approach for Group and Individual Fairness in Federated, by Xubo Yue (PhD student), Maher Nouihed and Raed Al Kontar
- Federated Bayesian Linear Regression using Hierarchical Models by Xubo Yue (PhD student), Ana Estrada Gomez and Raed Al Kontar.
- Federated Gaussian Process: Convergence, Automatic Personalization and Multi-fidelity Modeling by Xubo Yue (PhD student) and Raed Al Kontar.

FDA part III: Fairness modeling

- `GIFAIR-FL-Global` [24] penalizes the spread of losses among all groups, while minimizing the training error:

$$\min_{\boldsymbol{w}} H(\boldsymbol{w}) \triangleq \underbrace{\sum_{i=1}^{N} p_i F_i(\boldsymbol{w})}_{\text{average of training losses}} + \underbrace{\lambda \sum_{1 \le a < b \le d} |L_a(\boldsymbol{w}) - L_b(\boldsymbol{w})|}_{\text{spread of group losses}},$$

where λ is a positive scalar that balances fairness and goodness-of-fit, and **$L_a(\boldsymbol{w})$ is the averaged loss for group a (i.e., group loss)**:

$$L_a(\boldsymbol{w}) \triangleq \frac{1}{|\mathcal{A}_a|} \sum_{i \in \mathcal{A}_a} F_i(\boldsymbol{w}).$$

Here, $\mathcal{A}_a$ is the set of indices of devices who belong to group a, and $|\mathcal{A}|$ is the cardinality of the set $\mathcal{A}$.

More details

- Ends up as a client reweighting scheme

$$H(\boldsymbol{w}) = \sum_{i=1}^{N} p_i \left(1 + \frac{\lambda}{p_i \left|\mathcal{A}_{s_i}\right|} \underbrace{r_i(\boldsymbol{w})}_{ordering} \right) F_i(\boldsymbol{w}) := \sum_{i=1}^{N} p_i H_i(\boldsymbol{w})$$

$$r_i(\boldsymbol{w}) \triangleq \sum_{1 \le a \neq s_i \le d} \operatorname{sign}(L_{s_i}(\boldsymbol{w}) - L_a(\boldsymbol{w})).$$

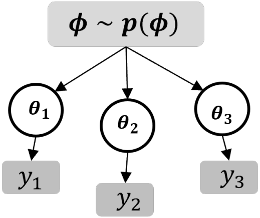
- One can view this approach as multiplying the original weight p_i by a factor $1 + \frac{\lambda}{p_i \left|\mathcal{A}_{s_i}\right|} r_i(\boldsymbol{w})$. The magnitude of this factor is determined by the ordering of losses.

FDA part III: FDA via hierarchical Bayes

- Can we learn a probabilistic model ?

$$y_i \sim \mathbb{P}^i_{y|x}\left(f_{\theta_i}(x_i)\right)$$

- Hierarchical Bayes is a natural way to **borrow strength** (and learn a good initialization)



More details

$$p(\phi, \{\boldsymbol{\theta}_i\}|\{\boldsymbol{y}_i\}) \propto p(\phi) \prod_{i=1}^{N} p(\boldsymbol{y}_i|\boldsymbol{\theta}_i, \phi) p(\boldsymbol{\theta}_i|\phi)$$

- Challenge: Joint local distribution (red) not available to the cloud

$$p(\phi|\{\boldsymbol{y}_i\}) \propto p(\phi) \prod_{i=1}^{N} \int p(\boldsymbol{y}_i, \boldsymbol{\theta}_i|\phi) d\boldsymbol{\theta}_i.$$

- The red part is not available in the central server. Therefore, one can approximate the red part by an approximation function $g_i(\phi)$. More specifically,

$$p(\phi|\{\boldsymbol{Y}_i\}_{i=1}^{N}) \approx p(\phi) \prod_{i=1}^{N} g_i(\phi) := q(\phi).$$

FDA part III: beyond ERM

- Functional Prior via Gaussain processes (GP)s

$$f \sim \mathcal{GP}(0, \mathcal{K}(\cdot, \cdot; \boldsymbol{\theta}_{\mathcal{K}})), \quad \epsilon \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2),$$

- Features correlations $\rightarrow$ biased gradients
- [23] proves that `Fedavg works as well`. Caveat, statistical error depends on batch size
- Automatic personalization as we jointly learn a functional prior

$$\begin{aligned} p(f_i^* | \boldsymbol{X}_i, \boldsymbol{y}_i, x^*) &= \int p(f_i^* | x^*, f_i) \frac{p(\boldsymbol{y}_i | \boldsymbol{X}_i, f_i) \overbrace{p(f_i)}^{\text{prior}}}{p(\boldsymbol{y}_i | \boldsymbol{X}_i)} df_i \\ &= \mathcal{N}(\mu_{i,pred}(x^*), \sigma^2_{i,pred}(x^*)), \end{aligned}$$

Applications dictate challenges

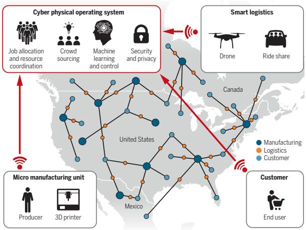
- I expect IoFT to infiltrate all industries that benefit from knowledge sharing, data analytics, and decision-making.
- **Only with a deep engineering understanding of the underlying system and domain, one formulates the right analytics**

IoFT website

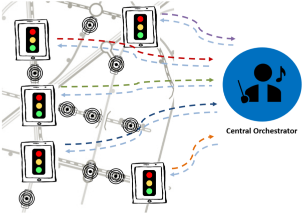
Website: `https://ioft-data.engin.umich.edu/`

Consider adding your data

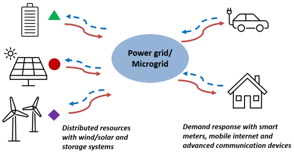
Applications



Distributed Manufacturing



Intersection Control



Energy Control

An application illustration

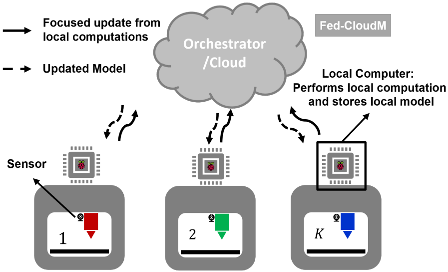


Figure 10: Cloud manufacturing powered by local computation

FE surrogate models also part of the nodes

Federated DoE

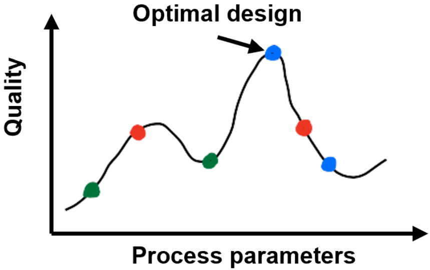


Figure 11: Collaborative optimal design

- Clients agree on next trial & error location via Blockchain consensus mechanisms

Papers & Youtube Channel

Youtube Channel: Link

- The Internet of Federated Things (IoFT). Link, Youtube.
- Federated Data Analytics: A Study on Linear Models. Link, Youtube.
- GIFAIR-FL: An Approach for Group and Individual Fairness in Federated Learning. Link, Youtube.
- Personalized PCA: Decoupling Shared and Unique Features. Link.
- Federated Gaussian Process: Convergence, Automatic Personalization and Multi-fidelity Modeling. Link.
- Personalized Federated Learning via Domain Adaptation with an Application to Distributed 3D Printing (paper attached).
- Fed-ensemble: Ensemble Models in Federated Learning for Improved Generalization and Uncertainty Quantification. Link.
- Federated Multi-output Gaussian Processes (coming soon).

Thank you

Questions

References

References I

[1] (2014). Information flow through neurons. https://jennysmoore.wordpress.com/2014/03/31/march-31-network-society-readings/. Accessed: 2020-07-18.

[2] Dinh, C. T., Tran, N. H., and Nguyen, T. D. (2020). Personalized federated learning with moreau envelopes. In *34th Conference on Neural Information Processing Systems*.

[3] Gonçalves, A. R., Von Zuben, F. J., Banerjee, A., et al. (2016). Multi-task sparse structure learning with gaussian copula models. *Journal of Machine Learning Research*.

[4] Gray, P. and Pancewicz, S. (2020). The ai chips race is on – what role will ip play? https://www.eenewseurope.com/news/ai-chips-race-what-role-will-ip-play. Accessed: 2020-07-18.

References II

[5] Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., and Ramage, D. (2018). Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*.

[6] Jiang, Y., Konečnỳ, J., Rush, K., and Kannan, S. (2019). Improving federated learning personalization via model agnostic meta learning. *arXiv preprint arXiv:1909.12488*.

[7] Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. (2019). Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*.

[8] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

References III

[9] Kontar, R., Shi, N., Yue, X., Chung, S., Byon, E., Chowdhury, M., Jin, J., Kontar, W., Masoud, N., Nouiehed, M., et al. (2021). The internet of federated things (ioft). *IEEE Access*, 9:156071–156113.

[10] Kontar, R., Zhou, S., Sankavaram, C., Du, X., and Zhang, Y. (2017). Nonparametric-condition-based remaining useful life prediction incorporating external factors. *IEEE Transactions on Reliability*, 67(1):41–52.

[11] Kontar, R., Zhou, S., Sankavaram, C., Du, X., and Zhang, Y. (2018). Nonparametric modeling and prognosis of condition monitoring signals using multivariate gaussian convolution processes. *Technometrics*, 60(4):484–496.

[12] Li, T., Hu, S., Beirami, A., and Smith, V. (2021). Ditto: Fair and robust federated learning through personalization. *arXiv preprint arXiv:2012.04221*.

References IV

[13] Li, T., Sanjabi, M., Beirami, A., and Smith, V. (2019). Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497*.

[14] Liang, P. P., Liu, T., Ziyin, L., Salakhutdinov, R., and Morency, L.-P. (2020). Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*.

[15] Mcdonald, R., Mohri, M., Silberman, N., Walker, D., and Mann, G. (2009). Efficient large-scale distributed training of conditional maximum entropy models. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C., and Culotta, A., editors, *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc.

[16] McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. JMLR.

References V

[17] Porter, M. E. and Heppelmann, J. E. (2014). How smart, connected products are transforming competition. *Harvard business review*, 92(11):64–88.

[18] Shi, N. and Kontar, R. A. (2022). Personalized pca: Decoupling shared and unique features. *arXiv preprint arXiv:2207.08041*.

[19] Smith, V., Chiang, C.-K., Sanjabi, M., and Talwalkar, A. (2017). Federated multi-task learning. *Conference on Neural Information Processing Systems*.

[20] Wang, K., Mathews, R., Kiddon, C., Eichner, H., Beaufays, F., and Ramage, D. (2019a). Federated evaluation of on-device personalization. *arXiv preprint arXiv:1910.10252*.

[21] Wang, K., Mathews, R., Kiddon, C., Eichner, H., Beaufays, F., and Ramage, D. (2019b). Federated learning with personalization layers. *arXiv preprint arXiv:1910.10252*.

References VI

[22] Yue, X. and Kontar, R. (2019). Variational inference of joint models using multivariate gaussian convolution processes. *arXiv preprint arXiv:1903.03867*.

[23] Yue, X. and Kontar, R. A. (2021). Federated gaussian process: Convergence, automatic personalization and multi-fidelity modeling. *arXiv preprint arXiv:2111.14008*.

[24] Yue, X., Nouiehed, M., and Kontar, R. A. (2021). Gifair-fl: An approach for group and individual fairness in federated learning. *arXiv preprint arXiv:2108.02741*.

[25] Zhang, Y. and Yeung, D.-Y. (2012). A convex formulation for learning task relationships in multi-task learning. *arXiv preprint arXiv:1203.3536*.