To Collaborate, or Not?
Federated Learning Meets Control

James Anderson

Department of Electrical Engineering
Columbia University

MIT LIDS
Nov 27th, 2023

Acknowledgements

- **Han Wang**, Columbia University
- **Leonardo F. Toso**, Columbia University
- **Siddartha Marella**, ZF Group
- **NSF:** Grant 2144634 & Columbia Data Science Institute

The Problem

collaboration seems like a good idea

"Sometimes I think the collaborative process would work better without you."

...but is it always?

Outline

1) Train on data generated by "similar" systems seeded with a common model

2) Aggregate model and broadcast

3) Repeat

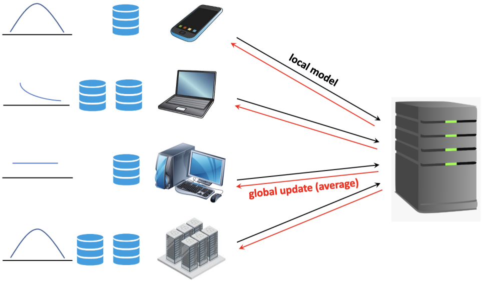
Q) How does heterogeneity affect sample complexity and performance?

1. system identification
2. clustering for personalization
3. extension to model-free optimal control

Federated Learning

a framework for distributed optimization that accounts for:

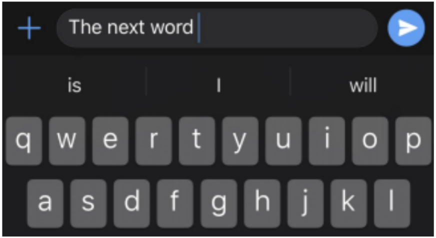
- device and data **heterogeneity**
- data **locality** (privacy)
- communication efficiency



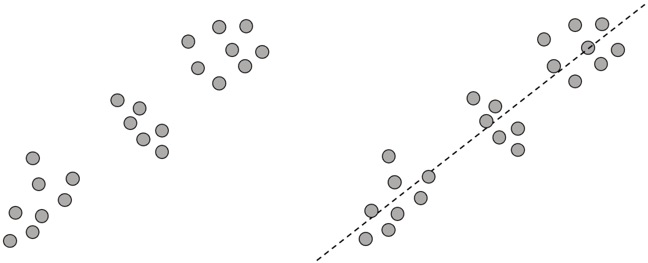
FEDERATED LEARNING FOR MOBILE KEYBOARD PREDICTION

Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays
Sean Augenstein, Hubert Eichner, Chloé Kiddon, Daniel Ramage

Google LLC,
Mountain View, CA, U.S.A.

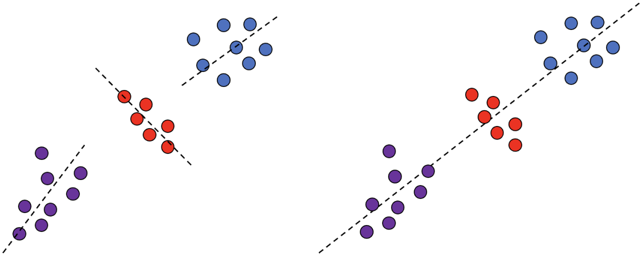


Centralized "Learning"



- all data in one place (or globally accessible)

Federated “Learning”



- data is **not** shared between clients, the **model** is shared and “averaged”

Data Privacy

Many high profile and large-scale data breaches have politicized data privacy



- EU's General Data Protection Regulation (GDPR) addresses the transfer of personal data outside the EU & EEA
- California Consumer Privacy Act (CCPA) intended to enhance privacy rights and consumer protection for residents
- many more countries have/will follow suit

Federated Learning

generic problem formulation:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad F(x) \triangleq \frac{1}{N} \sum_{i=1}^{N} f_i(x) + g(x)$$

assumptions:

- f_i non-convex, L-smooth
- g non-smooth, convex
- problem data is stored locally on each device and is **never** shared
- client-server computation model

Federated Learning

generic problem formulation:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad F(x) \triangleq \frac{1}{N} \sum_{i=1}^{N} f_i(x) + g(x)$$

we do not assume:

- bounded gradients: $\|\nabla f_i(x)\|^2 \leq G^2$ for all agents
- bounded heterogeneity: $\|\nabla f_i(x) - \nabla f_j(x)\| \leq \delta$ for all x

Federated Learning

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{N} \sum_{i=1}^{N} f_i(x) + g(x)$$

no shortage of federated algorithms:

- FedAvg, FedSplit, FedProx, FedDR, SCAFFOLD, FedPD, FedDyn,...
- our contribution: **FedADMM**
  - converges with **partial participation** and **approximate** local solutions
  - no bounded gradients
  - no bounded heterogeneity

FedADMM

rewrite the problem as

$$\begin{array}{ll} \underset{x,\bar{x}}{\text{minimize}} & \frac{1}{N}\sum_{i=1}^{N} f_i(x_i) + g(\bar{x}) \\ \text{s.t.} & \curvearrowleft = \bar{x} \end{array}$$

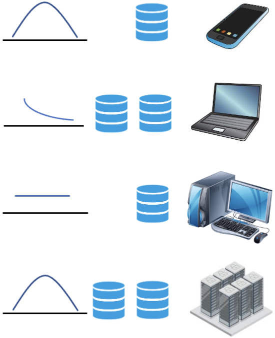
where

- x: concatenation of local variables $[x_1^T, x_2^T, \dots, x_N^T]$
- $\bar{x}$: global consensus variable

each agent has an augmented Lagrangian:

$$\mathcal{L}_i(x_i, \bar{x}, z_i) := f_i(x_i) + g(\bar{x}^k) + \langle z_i^k, x_i - \bar{x}^k \rangle + \frac{\eta}{2}\left\|x_i - \bar{x}^k\right\|^2$$

Client-side



▷ Client side

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

  receive $\bar{x}^k$ from the server.

  $x_i^{k+1} \approx \arg\min_{x_i} \mathcal{L}_i\left(x_i, \bar{x}^k, z_i^k\right)$

  $z_i^{k+1} = z_i^k + \eta\left(x_i^{k+1} - \bar{x}^k\right)$ ◊Dual updates

  $\hat{x}_i^{k+1} = x_i^{k+1} + \frac{1}{\eta} z_i^{k+1}$

  send $\Delta\hat{x}_i^k = \hat{x}_i^{k+1} - \hat{x}_i^k$ back to the server

end for

Client-side

approximation

clients do not have to minimize $\mathcal{L}_i$ precisely:

$$\left\| x_i^{k+1} - \arg\min_{x_i} \mathcal{L}_i(x_i, \bar{x}^k, z_i^k) \right\| \leq \epsilon_{i,k+1}$$

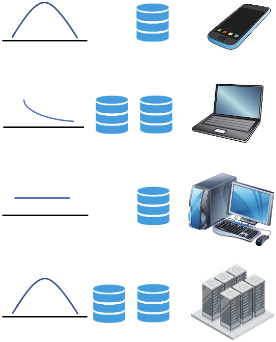
partial participation

at iteration k only a subset of clients $\mathcal{S}_k$ need to send local updates

mixing

each client is seeded with averaging vector $\bar{x}$

Server-side



▷ Server side

aggregation $\tilde{x}^{k+1} = \tilde{x}^k + \frac{1}{n} \sum_{i \in \mathcal{S}_k} \Delta \hat{x}_i^k$

update $\bar{x}^{k+1} = \mathrm{prox}_{g/\eta}\left(\tilde{x}^{k+1}\right)$

- FedADMM performs K server-side iterations

Analysis

convergence (informal, $g \equiv 0$):

$$\frac{1}{K+1}\sum_{k=0}^{K}\mathbb{E}\left[\left\|\nabla f(\bar{x}^k)\right\|^2\right] \leq \underbrace{\frac{c_1[F(x^0)-F^\star]}{K+1}}_{(1)} + \underbrace{\frac{1}{N(K+1)}l(\epsilon_{i,k},\epsilon_{i,k+1})}_{(2)}$$

where

$$l(\epsilon_{i,k},\epsilon_{i,k+1}) := \sum_{k=0}^{K}\sum_{i=1}^{n}(c_2\epsilon_{i,k}^2 + c_3\epsilon_{i,k+1}^2)$$

.

- (1) initial optimality gap
- (2) cost of working with approximate solutions and bennifit of N clients
- impact of partial participation reflected in the constants

Analysis

convergence (informal):

if the sum of the inaccuracies is bounded by $D > 0$, then FedADMM requires

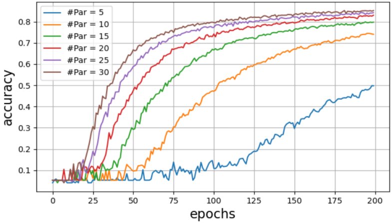
$$K = \left\lfloor \frac{c_1[F(x_0) - F^\star] + (c_2 + c_3)D}{\epsilon^2} \right\rfloor \equiv O(\epsilon^{-2})$$

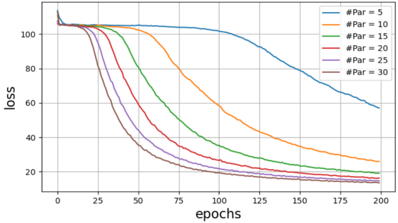
to achieve an ϵ-suboptimal stationary point.

- analysis can be extended to include g

Numerical Experiments

- FEMNIST Dataset: 62 classes, 1-10, A-Z, a-z, multiple writers, 30 clients
- 2 convolutional layers, 2 fully connected layers, 62 output neurons
- stochastic gradient descent, 300 iterations per client





FedADMM: A Federated Primal-Dual Algorithm Allowing Partial Participation

Wang, Marella, Anderson

Proc. IEEE CDC, 2022

System Identification

data

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ & & \vdots & & \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix} \longrightarrow$$

dynamic model

$$\frac{d}{dt}x(t) = f(x(t), u(t)) \longrightarrow$$

control

$$\frac{d}{dt}x(t) = f(x(t), u(t))$$

$$u(t) = \phi(x(t))$$

model-based control

- adaptive control: update model (and controller) online
- offline control: learn model once

Centralized Setting

- **ground truth** system

$$x_{t+1} = A_\star x_t + B_\star u_t + w_t, \quad t = 0, 1, 2, \ldots, T-1$$

- generates data

$$\{x_{l,t}, u_{l,t}\}_{t=0}^{T-1}, \quad l = 1, \ldots, N$$

- rewrite the system as

$$x_{t+1} = \Theta z_t + w_t, \quad \Theta \triangleq [A_\star \ B_\star], \quad z_t \triangleq \begin{bmatrix} x_t \\ u_t \end{bmatrix}$$

Data

- rollout l generates data

$$X_l = \begin{bmatrix} x_{l,T-1} & \dots & x_{l,1} \end{bmatrix} \in \mathbb{R}^{n \times T}$$
$$Z_l = \begin{bmatrix} x_{l,T-1} & \dots & x_{l,1} \end{bmatrix} \in \mathbb{R}^{(n+p) \times T}$$
$$W_l = \begin{bmatrix} w_{l,T-1} & \dots & w_{l,1} \end{bmatrix} \in \mathbb{R}^{n \times T}$$

- concatenating data from all rollouts

$$X = \begin{bmatrix} X_1 & \dots & X_N \end{bmatrix} \in \mathbb{R}^{n \times TN}, \quad Z = \begin{bmatrix} Z_1 & \dots & Z_N \end{bmatrix} \in \mathbb{R}^{(n+p) \times TN}$$

- relationship described by

$$X = \Theta Z + W$$

- **least-squares estimator**

$$\hat{\Theta} \triangleq \begin{bmatrix} A & B \end{bmatrix} = \arg\min_{\Theta \in \mathbb{R}^{n \times (n+p)}} \|X - \Theta Z\|_F^2.$$

Error Analysis

- optimal $\hat{\Theta}$ satisfies [Dean et al.]:

$$\max\{\|A_\star - A\|, \|B_\star - B\|\} \leq \underbrace{\frac{16\sigma_w}{\sqrt{\lambda_{\mathsf{min}}(\Sigma_T)}} \left(\frac{(n+2p)\log(36/\delta)}{N} \right)^{\frac{1}{2}}}_{\mathcal{O}(N^{-\frac{1}{2}})}$$

where Σ_T is the covariance of the state at time T

Note:

- $x_T = G_T u + F_T w$
- $\lambda_{\min}(\sigma_u^2 G_T G_T^T + \sigma_w^2 F_T F_T^T)$ quantifies how difficult to system is to control
- result only uses data at time T from each rollout

Federated System ID

- **ground truth** systems, $i = 1, \ldots, M$

$$x_{t+1}^{(i)} = A_\star^{(i)} x_t^{(i)} + B_\star^{(i)} u_t^{(i)} + w_t^{(i)}, \quad t = 0, 1, 2, \ldots, T-1$$

  where

$$x_0^{(i)} \sim \mathcal{N}(0, \sigma_{i,x}^2 I), \quad u_t^{(i)} \sim \mathcal{N}(0, \sigma_{i,u}^2 I), \quad w_0^{(i)} \sim \mathcal{N}(0, \sigma_{i,w}^2 I)$$

- system i generates

$$\{x_{l,t}^{(i)}, u_{l,t}^{(i)}\}_{t=0}^{T-1}, \quad l = 1, \ldots, N_i$$

- **system heterogeneity**

$$\max_{i,j} \|A_\star^{(i)} - A_\star^{(j)}\| \le \epsilon, \quad \text{and} \quad \max_{i,j} \|B_\star^{(i)} - B_\star^{(j)}\| \le \epsilon, \quad \text{for all } i, j$$

Federated System Identification

Objective:

Learn a common model $\bar{\Theta} = [\bar{A} \;\; \bar{B}]$ that performs well on all $\Theta^{(i)} = [A_\star^{(i)} \;\; B_\star^{(i)}]$

Challenges

- Data cannot be shared
- Systems are different

Formally, we will solve the following problem in a federated manner:

$$\bar{\Theta} \triangleq \left[\begin{array}{cc} \bar{A} & \bar{B} \end{array} \right] = \frac{1}{M} \sum_{i=1}^{M} \operatorname*{argmin}_{\Theta} \|X^{(i)} - \Theta Z^{(i)}\|_F^2$$

Aside: Quantifying System Heterogeneity

recall our definition:

$$\max_{i,j} \|A_\star^{(i)} - A_\star^{(j)}\| \le \epsilon, \quad \text{and} \quad \max_{i,j} \|B_\star^{(i)} - B_\star^{(j)}\| \le \epsilon, \quad \text{for all } i, j$$

are these systems really similar?

$$x_{t+1}^{(1)} = 0.99 x_t^{(1)} + 0.1 u_t^{(1)} \quad \text{and} \quad x_{t+1}^{(2)} = 1.01 x_t^{(2)} + 0.01 u_t^{(2)}$$

possible fixes:

- system norms
- ν-gap

`FedSysID`: Meta Algorithm

Algorithm 1 `FedSysID`

Initialize the server with $\bar{\Theta}_0$ and step size α ;
Initialize each client $i \in [M]$ with $\Theta_{0,0}^{(i)} = \bar{\Theta}_0$;
For each round $r = 0, 1, \ldots, R-1$ **do**
    uniformly sample $S_r \subseteq \{1, 2, \cdots, M\}$
    $\triangleright$ Client side:
    For each client $i \in S_r$ **in parallel do**
        $\Theta_{r+1}^{(i)} = \texttt{ClientUpdate}(i, \bar{\Theta}_r, K_i)$
        send $\Theta_{r+1}^{(i)}$ back to the server
    end for
    $\triangleright$ Server side:
    update $\bar{\Theta}_{r+1} = \frac{1}{M} \sum_{i=1}^{M} \Theta_{r+1}^{(i)}$ and send $\bar{\Theta}_{r+1}$ to each client
end for
Return $\bar{\Theta}_R$

Price of Heterogeneity

with high probability, the least squares estimator produces $\bar{\Theta}$ such that:

$\max\{\|\bar{A} - A_\star^{(i)}\|, \|\bar{B} - B_\star^{(i)}\|\} \leq$

$$\frac{1}{\sqrt{\sum_{i=1}^{M} N_i}} \times \underbrace{\frac{c\sqrt{\sum_{i=1}^{M} \sigma_{i,w}^2 \left(\sum_{t=0}^{T-1} \|(\Sigma_t^{(i)})^{\frac{1}{2}}\|\right)^2}}{\lambda}}_{C_1} + \epsilon \times \underbrace{\frac{9\sum_{j\neq i}\sqrt{(\sum_{t=0}^{T-1}\|\Sigma_t^{(i)}\|)^2}}{\lambda}}_{C_2}$$

where $\lambda = \min_i \lambda_{\min}(\sum_{t=0}^{T-1} \Sigma_t^{(i)})$.

- C_1: error constant
- C_2: heterogeneity constant

with high probability, the least squares estimator produces $\bar{\Theta}$ such that:

$$\max\{\|\bar{A} - A_\star^{(i)}\|, \|\bar{B} - B_\star^{(i)}\|\} \leq \frac{1}{\sqrt{\sum_{i=1}^{M} N_i}} \times \frac{\text{signal}}{\text{noise}} + \mathcal{O}(\text{heterogeneity})$$

Takeaways

- collaboration allows clients to improve their performance from $\mathcal{O}(\frac{1}{\sqrt{N}_i})$ to $\mathcal{O}\left(\frac{1}{\sum_{i=1}^{M}\sqrt{N}_i}\right)$
- despite not sharing the data, clients performance improves as if they did

`ClientUpdate`: Per-Round Analysis

$$\Delta_R \triangleq \max\left\{\mathbb{E}\|\bar{A}_R - A^{(i)}\|, \mathbb{E}\|\bar{B}_R - B^{(i)}\|\right\}$$

for all $R \geq 1$, the output of `FedSysID` $\bar{\Theta}$ satisfies:

- `FedAvg` [McMahan et al.]

$$\Delta_R \leq \mathcal{O}\left(\frac{1}{KR} + \frac{C_1}{\sqrt{\sum_{i=1}^{M} N_i}} + \epsilon C_2\right)$$

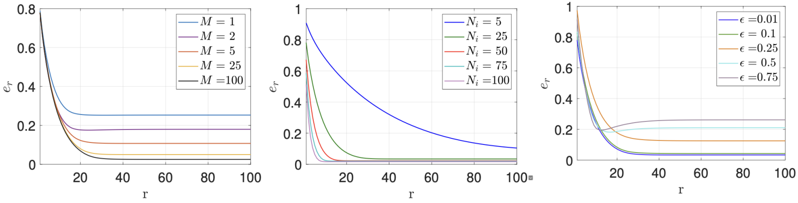
- `FedLin` [Mitra et al.]

$$\Delta_R \leq \mathcal{O}\left(e^{-\beta R} + \frac{C_1}{\sqrt{\sum_{i=1}^{M} N_i}} + \epsilon C_2\right)$$

Numerical Experiments

- nominal system (A_0, B_0)
- perturbed system $A^{(i)} = A_0 + \gamma_1^{(i)} V$, $B^{(i)} = B_0 + \gamma_2^{(i)} U$

$$A_0 = \begin{bmatrix} 0.6 & 0.5 & 0.4 \\ 0 & 0.4 & 0.3 \\ 0 & 0 & 0.3 \end{bmatrix}, \quad V = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad B_0 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \\ 0.5 & 0.5 \end{bmatrix}, \quad U = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}.$$



FedSysID: A Federated Approach to Sample-Efficient System ID
Wang, Toso, Anderson
Proc. L4DC, 2023

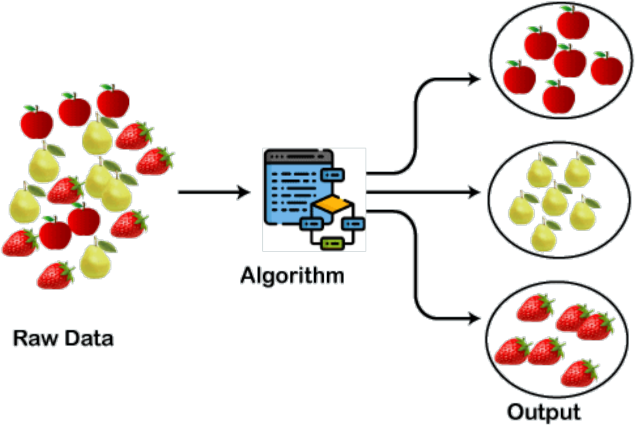
Clustering

- **Motivation:**
  is a common estimation for all the participants a good idea in heterogeneous settings?

  can we get **personalized models** when the systems participating in the collaboration are significantly different

- **Pros:**
  cluster systems with similar dynamics – run system identification on the clusters separately – reduces heterogeneity, therefore should speed up convergence.

- **Cons:**
  incorrect classification slows things down


Raw Data
Algorithm
Output

Clustering setup

- M systems generating data

$$x_{t+1}^{(i)} = A^{(i)} x_t^{(i)} + B^{(i)} u_t^{(i)} + w_t^{(i)}, \quad t = 0, \ldots, T-1$$

- M data sets of the form

$$\{x_{l,t}^{(i)}, u_{l,t}^{(i)}\}_{t=0}^{T-1}, \quad l = 1, \ldots, N_i$$

- each data set generated by one of $K \ll M$ system types

- define "clusters" $\mathcal{C}_1, \ldots, \mathcal{C}_K$, where

$$\mathcal{C}_j \triangleq (A_j, B_j) \quad \text{such that } A_j = A^{(i)}, B_j = B^{(i)} \text{ for some } i \in [M]$$

  and define $\Theta_j \triangleq \begin{bmatrix} A & B \end{bmatrix}$

Clustering for System Identification

Objective:

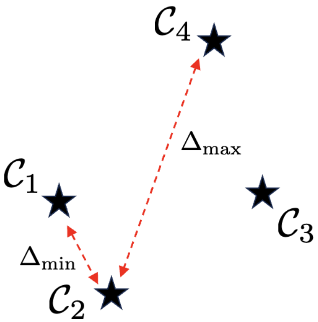
Given M data sets from K system types (clusters):

1. partition the data sets into clusters
2. learn a common model within the cluster

Challenges

- Data is unlabelled
- Misclassification may hinder progress

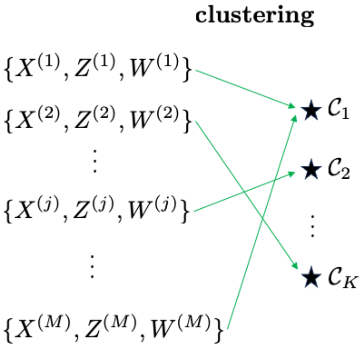
Assumptions



- $\|\widehat{\Theta}_j^{(0)} - \Theta_j\| \leq (\frac{1}{2} - \alpha^{(0)})\Delta_{\min}$ where $\alpha^{(0)} \in (0, \frac{1}{2})$
- $N_i n$ and $\Delta_{\min}$ sufficiently large

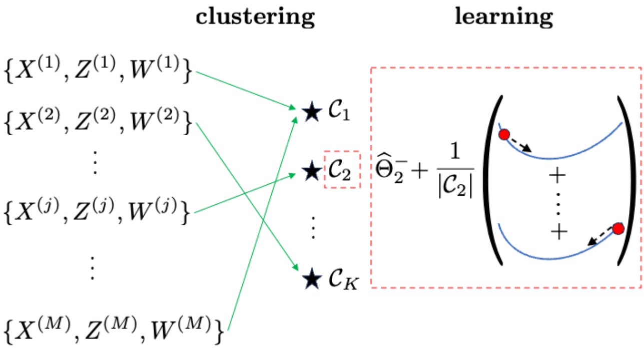
$$\Delta_{\min} \gtrsim 1 + \Delta_{\max} \sum_{i \in [M]} \sum_{t=1}^{T} \exp(-f(N_i, n, \alpha^{(0)}, \|\Sigma_t^{(i)}\|, \rho^{(i)}))$$

Phase 1: Clustering



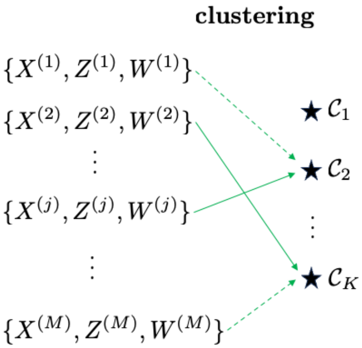
for each $i \in [M]$, $\quad \hat{j} = \underset{j \in [K]}{\operatorname{argmin}} \; \|X^{(i)} - \widehat{\Theta}_j^{(r)} Z^{(i)}\|_F^2$

Phase 2: Learning



$$\widehat{\Theta}_j^{(r+1)} \leftarrow \underbrace{\widehat{\Theta}_j^{(r)} + \frac{2\eta_j}{\sum_{i\in[M]} e_{i,j}} \sum_{i\in[M]} \underbrace{(X^{(i)} - \widehat{\Theta}_j^{(r)} Z^{(i)})(Z^{(i)})^\top}_{\text{localupdate}}}_{\text{centralized averaging}}$$

Phase 1: Clustering



and repeat...

Algorithm

Algorithm 1 `Clustered System Identification`

Initialization: number of clusters K, step-size η_j, and model initialization $\widehat{\Theta}_j^{(0)}\ \forall j \in [K]$,
for each iteration $r = 0, 1, \ldots, R-1$ **do**
    The systems receive the models $\{\widehat{\Theta}_1^{(r)}, \ldots, \widehat{\Theta}_K^{(r)}\}, \forall j \in [K]$,
    Cluster estimation (CE):
        for each system $i \in [M]$
            $\hat{j} = \text{argmin}_{j \in [K]} \|X^{(i)} - \widehat{\Theta}_j^{(r)} Z^{(i)}\|_F^2$,
            define $e_i = \{e_{i,j}\}_{j=1}^{K}$ with $e_{i,j} = \mathbb{1}\{j = \widehat{j}\}$,
        end for
    Model estimation (ME):
    $\widehat{\Theta}_j^{(r+1)} = \widehat{\Theta}_j^{(r)} + \frac{2\eta_j}{\sum_{i \in [M]} e_{i,j}} \sum_{i \in [M]} e_{i,j} (X^{(i)} - \widehat{\Theta}_j^{(r)} Z^{(i)}) Z^{(i),\top}$ for all $j \in [K]$
end for
Return $\widehat{\Theta}_j^{(R)}$ for all $j \in [K]$.

Theoretical Guarantees

Probability of misclassification:

$$\mathbb{P}\left\{\mathcal{M}_i^{j,j'}\right\} \leq c_1 \sum_{t=0}^{T-1} \exp\left(-c_2 N_i n_x \left(\frac{\alpha\rho^{(i)}\|\Sigma_t^{(i)}\|}{\rho^{(i)}\|\Sigma_t^{(i)}\| + \sqrt{n_x}}\right)^2\right)$$

$\mathcal{M}_i^{j,j'}$ is the event that system i is misclassified as belonging to cluster $\mathcal{C}_{j'}$

- reformulate so that misclassification happens with at most probability δ
- recovers the results of Ghosh et al. NeurIPS'20

Theoretical Guarantees

Convergence:

after R iterations, for every cluster $j \in [K]$, $||\widehat{\Theta}_j^{(R)} - \Theta_j||$ is upper bounded by

$$\underbrace{\frac{\tilde{c}_0}{\sqrt{\sum_{i \in \mathcal{C}_j} N_i}}}_{C_3} + \underbrace{\tilde{c}_1 \Delta_{\max} \sum_{i \in [M]} \sum_{t=0}^{T-1} \exp\left(-\tilde{c}_2 N_i n_x \left(\frac{\rho^{(i)} \|\Sigma_t^{(i)}\|}{\rho^{(i)} \|\Sigma_t^{(i)}\| + \sqrt{n_x}}\right)^2\right)}_{C_4}$$

- C_3: in-class sample complexity
- C_4 probability of misclassification
- bound holds w.h.p., + earlier assumptions
- bound is independent of $\alpha^{(0)}$

Comparison

- **Federated SysID:**

$$\max\{\|\bar{A} - A_\star^{(i)}\|, \|\bar{B} - B_\star^{(i)}\|\} \leq \frac{1}{\sqrt{\sum_{i=1}^{M} N_i}} \times \frac{\text{signal}}{\text{noise}} + \mathcal{O}(\text{heterogeneity})$$

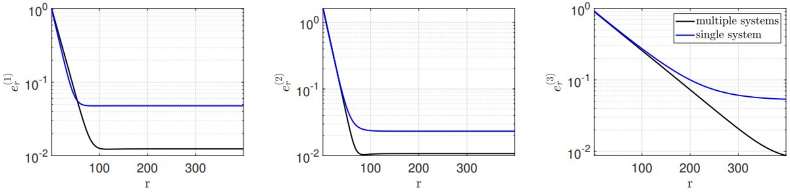
where $O(\text{heterogeneity})$ is **not** controlled by the number of trajectories N_i

- **Clustered SysID:**

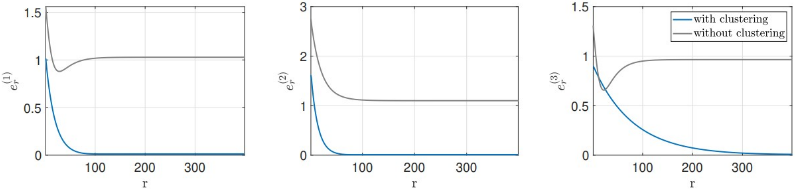
$$\max\{\|\hat{A}_j - A_j\|, \|\hat{B}_j - B_j\|\} \leq \frac{1}{\sqrt{\sum_{i \in \mathcal{C}_j} N_i}} \times \frac{\text{signal}}{\text{noise}} + \mathcal{O}(\exp(\text{misclass}))$$

with $\exp(\text{misclass})$ being controlled by N_i

- **Gain of collaboration:**

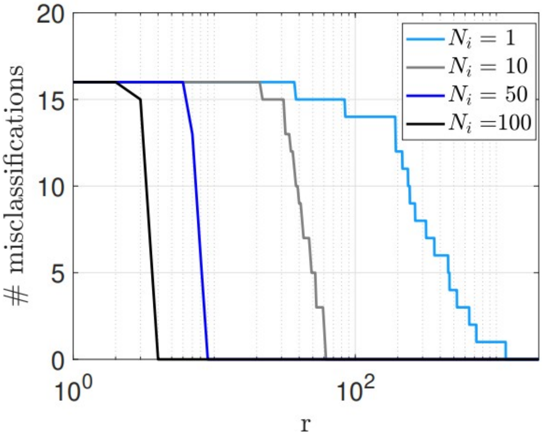


- **Gain of clustering:**



  – 50 systems, 3 clusters, $T = 50$

- **Number of misclassifications:**



Learning Personalized Models with Clustered System Identification

Toso, Wang, Anderson

To appear, Proc IEEE CDC'23: arXiv 2304.01395

Conclusions

- federated learning can be applied to system identification system identification
- characterized the cost/benefit of heterogeneity
- personalization can be achievd through clustering
- aimed for ideas – dodged the technical details
- related: federated LQR policy gradient and RL problems

james.anderson@columbia.edu www.columbia.edu/∼ja3451

Federated LQR

- Given M LTI systems

$$x_{t+1}^{(i)} = A^{(i)} x_t^{(i)} + B^{(i)} u_t^{(i)}, \quad x_0^{(i)} \sim \mathcal{D}, \quad i = 1, \dots, M$$

- construct a state feedback controller that solves

$$K^* = \underset{K}{\operatorname{argmin}} \left\{ C_{\mathsf{avg}}(K) \triangleq \frac{1}{M} \sum_{i=1}^{M} \mathbb{E} \left[\sum_{t=0}^{\infty} x_t^{(i)\top} Q x_t^{(i)} + u_t^{(i)\top} R u_t^{(i)} \right] \right\}$$

$$\text{s.t.} \quad u_t^{(i)} = -K x_t^{(i)}$$

$$\text{system dynamics}$$

- system heterogeneity

$$\max_{i,j} \|A^{(i)} - A^{(j)}\| \le \epsilon_1, \quad \text{and} \quad \max_{i,j} \|B^{(i)} - B^{(j)}\| \le \epsilon_2, \quad \text{for all } i, j$$

Single Agent Policy Gradient

- if (A, B) is **known** and $\mathbb{E}_{x_0 \sim \mathcal{D}}[x_0 x_0^T]$ is full rank, the iteration

$$K \leftarrow K - \eta \nabla C(K)$$

  finds the globally optimal controller

- closed-form expressions for the gradient: $\nabla C(K) = 2 E_K \Sigma_K$ where

$$E_K \triangleq \left(R + B^T P_K B\right) K - B^T P_K A, \quad \Sigma_K \triangleq \mathbb{E}_{x_0 \sim \mathcal{D}} \sum_{t=0}^{\infty} x_t x_t^T$$

  and P_K solves the Lyapunov equation

$$P_K = Q + K^T R K + (A - BK)^T P_K (A - BK)$$

Model-Free Case

- when (A, B, Q, R) not directly available, $\nabla C(K)$ can't be computed
- the controller only has simulation access to a model
- iterates are generated according to $K \leftarrow K - \eta \widehat{\nabla C(K)}$

Algorithm 2 Zeroth-order gradient estimation (ZO)

Input: K, number of trajectories n_s, trajectory length τ, smoothing radius r, dimension n_x and n_u, system index i.
for $s = 1, \ldots, n_s$ **do**
  Sample a policy $\widehat{K}_s = K + U_s$, with U_s drawn uniformly at random over matrices whose (Frobenius) norm is r.
  Simulate the i-th system for τ steps starting from $x_0 \sim \mathcal{D}$ using policy $\widehat{K}_s$. Let $\widehat{C}_s$ be the empirical estimate: $\widehat{C}_s = \sum_{t=1}^{\tau} c_t$, where $c_t := x_t^\top \left(Q + \widehat{K}_s^\top R \widehat{K}_s\right) x_t$ on this trajectory.
end for
Return the estimate: $\widehat{\nabla C(K)} = \frac{1}{n_s} \sum_{s=1}^{n_s} \frac{n_x n_u}{r^2} \widehat{C}_s U_s$.

FedLQR

Algorithm 1 Model-free Federated Policy Learning for the LQR (`FedLQR`)

Input: initial policy K_0, local step-size η_l and global step-size η_g.
Initialize the server with K_0 and η_g
for $n = 0, \ldots, N-1$ **do**
  for each system $i \in [M]$ **do**
    for $l = 0, \cdots, L-1$ **do**
      Agent i initializes $K_{n,0}^{(i)} = K_n$
      Agent i estimates $\widehat{\nabla C^{(i)}(K_{n,l}^{(i)})} = \mathtt{ZO}(K_{n,l}^{(i)}, i)$ and updates local policy as
$$K_{n,l+1}^{(i)} = K_{n,l}^{(i)} - \eta_l \widehat{\nabla C^{(i)}(K_{n,l}^{(i)})}$$
    end for
    send $\Delta_n^{(i)} = K_{n,L}^{(i)} - K_n$ back to the server
  end for
  Server computes and broadcasts global model $K_{n+1} = K_n + \frac{\eta_g}{M} \sum_{i=1}^{M} \Delta_n^{(i)}$
end for

Under the Hood

Details can be found in the paper:

- require a controller K_0 that stabilizes all systems
- sufficiently large smoothing radius of the gradient estimator
- have access to sufficient samples
- operate in a low heterogeneity regime

Algorithm Guarantees

- At every round n, K_n is stabilizing
- Every local controller $K_n^{(i)}$ is locally stabilzing
- After
$$N \geq \frac{c_{\mathsf{uni},4} \left\| \Sigma_{K_i^*} \right\|}{\eta \mu^2 \sigma_{\min}(R)} \log \left(\frac{2(C^{(i)}(K_0) - C^{(i)}(K_i^*))}{\epsilon'} \right)$$
rounds, `FedLQR` acheives
$$C^{(i)}(K_N) - C^{(i)}(K_i^*) \leq \epsilon' + c_{\mathsf{uni},2} \times \mathcal{B}(\epsilon_1, \epsilon_2), \quad \forall i \in [M]$$