Multi-Task Learning for Control with MAML-LQR

James Anderson

Department of Electrical Engineering
Columbia University

Johns Hopkins University: ECE Seminar

October 29, 2024

Acknowledgements

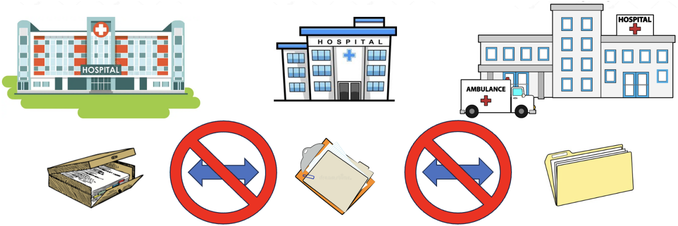



- **Han Wang**, Columbia University
- **Leonardo F. Toso**, Columbia University
- **Donglin Zhan**, Columbia University
- **Aritra Mitra**, NC State

"sometimes I think this collaboration would work better without you"

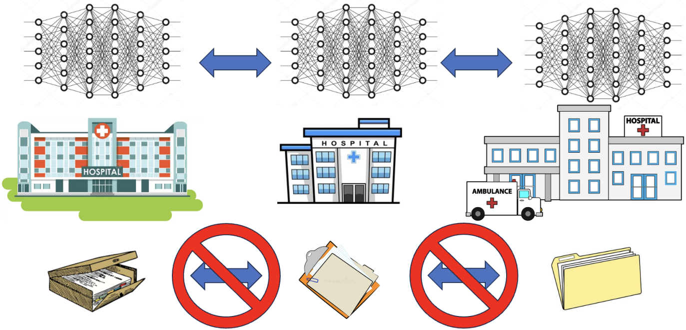
Motivation: Collaborative (Supervised) Learning

- data is collected from different sources, it **cannot** be shared
- goal is to build a model that captures **all** the data



Motivation: Collaborative (Supervised) Learning

- data is collected from different sources, it **cannot** be shared
- goal is to build a model that captures **all** the data



Task Adaptability



learn controller from training data

deploy learned controller

- sample tasks from a distribution
- learn a policy that does well on all of them
- quickly adapt policy to an unseen task

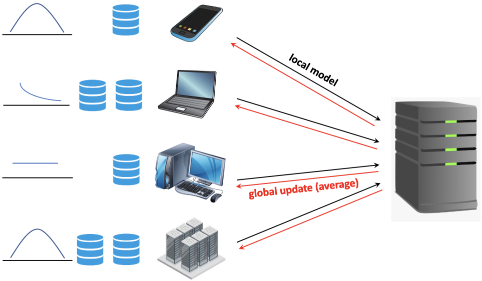
Outline

- Federated Learning
- Model-Free Learning for control
- The Federated LQR problem
- Meta-LQR

Federated Learning

a framework for distributed optimization that accounts for:

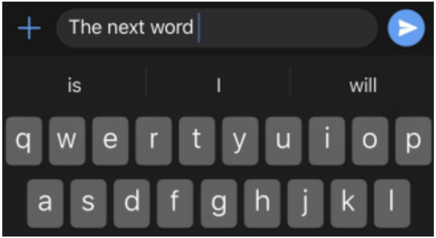
- device and data **heterogeneity**
- data **locality** (privacy)
- communication efficiency



FEDERATED LEARNING FOR MOBILE KEYBOARD PREDICTION

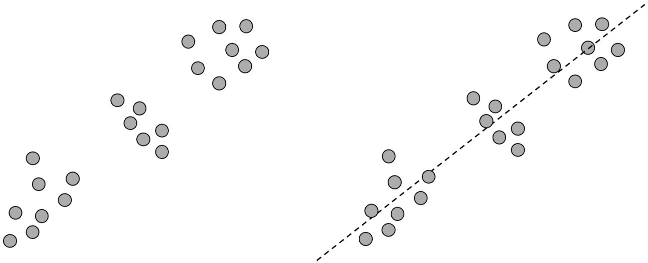
Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays
Sean Augenstein, Hubert Eichner, Chloé Kiddon, Daniel Ramage

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



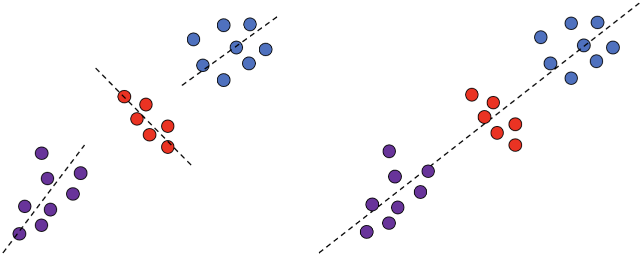
Federated Learning Problem Setup

Centralized "Learning"



- all data in one place (or globally accessible)

Federated “Learning”



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

Problem Formulation

consider the stochastic optimization problem

$$\underset{x}{\text{minimize}} \quad \mathbb{E}_{\zeta}\left[l(x,\zeta)\right] \qquad \texttt{// population risk}$$

where

- $l : \mathbb{R}^p \times \mathbb{R}^u$ is the expected loss function
- x is the model parameter vector
- $\zeta \sim \mathcal{P}$ with $\mathcal{P}$ unknown
- N clients each generate m samples denoted $\mathcal{D}^i = \{\zeta_1^i, \ldots, \zeta_m^i\}$ for $i \in [N]$

Problem Formulation

consider the stochastic optimization problem

$$\underset{x}{\text{minimize}} \quad \mathbb{E}_\zeta \left[l(x, \zeta) \right] \qquad \texttt{// population risk}$$

where

- $l : \mathbb{R}^p \times \mathbb{R}^u$ is the expected loss function
- x is the model parameter vector
- $\zeta \sim \mathcal{P}$ with $\mathcal{P}$ unknown
- N clients each generate m samples denoted $\mathcal{D}^i = \{\zeta_1^i, \ldots, \zeta_m^i\}$ for $i \in [N]$

to highlight the distributed nature of the problem, rewrite as

$$\underset{x}{\text{minimize}} \quad \underbrace{\frac{1}{N} \sum_{i=1}^{N} f_i(x),}_{\texttt{empirical risk}} \quad \text{where} \quad \underbrace{f_i(x) \triangleq \frac{1}{m} \sum_{\zeta \in \mathcal{D}^i} l(x, \zeta)}_{\texttt{client } i \texttt{ solves}}$$

FedAvg

a prototypical federated learning algorithm **[McMahan et al. 2016]**

Algorithm 1: Federated Averaging (`FedAvg`)

Input: global iterations K, local iterations τ, stepsize $\eta_{k,t}$

for $k = 0, 1, \ldots, K-1$ **do**
  // server operations
  randomly select subset of clients $\mathcal{S}_k$
  broadcast x_k to all clients in $\mathcal{S}_k$

  for *each client in* $\mathcal{S}_k$ **in parallel do**
    $x_{k,0}^{(i)} \leftarrow x_k$
    for $t = 0, 1, \ldots, \tau - 1$ **do**
      pick data point $\eta \in \mathcal{D}^i$ **and** compute $g_i(x) = \nabla l(x, \zeta)$
      $x_{k,t+1}^{(i)} \leftarrow x_{k,t}^{(i)} - \eta_{k,t} g_i(x_{k,t}^{(i)})$ // SGD iteration
    send $\Delta_{k,\tau}^{(i)} \leftarrow x_{k,\tau}^{(i)} - x_k$ to server // new − old

  aggregate the updates $x_{k+1} \leftarrow x_k + \frac{1}{n_c} \sum_{i \in \mathcal{S}_k} \Delta_{k,\tau}^{(i)}$ // global update

Linear Quadratic Control

System

consider the discrete-time dynamical system

$$x_{t+1} = Ax_t + Bu_t, \quad x_0 \sim \mathcal{D} \quad t = 0, 1, 2, \ldots \quad \text{(dynamics)}$$

with

- state $x_t \in \mathbb{R}^n$, input $u_t \in \mathbb{R}^m$
- initial condition $\mathbb{E}x_0 = 0$, and $\mathbb{E}x_0 x_0^T \succeq \mu I$

Linear Quadratic Control

System

consider the discrete-time dynamical system

$$x_{t+1} = Ax_t + Bu_t, \quad x_0 \sim \mathcal{D} \quad t = 0, 1, 2, \ldots \quad \text{(dynamics)}$$

with

- state $x_t \in \mathbb{R}^n$, input $u_t \in \mathbb{R}^m$
- initial condition $\mathbb{E}x_0 = 0$, and $\mathbb{E}x_0 x_0^T \succeq \mu I$

Objective

design a static linear control policy $u_t = -Kx_t$ such that:

$$K \in \mathcal{K} \triangleq \{K \mid \rho(A - BK) < 1\} \quad \text{(stability)} \qquad \texttt{// non-convex}$$

and the quadratic cost

$$C(K) \triangleq \mathbb{E}_{x_0 \sim \mathcal{D}} \left[\sum_{t=0}^{\infty} x_t^T \left(Q + K^\top RK \right) x_t \right] \quad \text{s.t. (dynamics)+(stability)}$$

is minimized

Model-Based Solution

LQR problem:

$$\underset{K}{\text{minimize}} \quad C(K)$$
$$\text{s.t.} \quad \text{(dynamics)} + \text{(stability)}$$

LQR solution:

- solve the DARE for P_K

$$P_K = Q + A^T P_K A - A^T P_K B(R + B^T P_K B)^{-1} B^T P_K A$$

- construct $K^\star$ from (A, B, P_K, R)

$$K^\star = -(R + B^T P_K B)^{-1} B^T P_K A$$

Model-Based Solution

LQR problem:

$$\underset{K}{\text{minimize}} \quad C(K)$$
$$\text{s.t.} \quad \text{(dynamics)} + \text{(stability)}$$

LQR solution:

- solve the DARE for P_K

$$P_K = Q + A^T P_K A - A^T P_K B(R + B^T P_K B)^{-1} B^T P_K A$$

- construct $K^\star$ from (A, B, P_K, R)

$$K^\star = -(R + B^T P_K B)^{-1} B^T P_K A$$

Q: How do we compute K without a model, i.e., (A, B, Q, R)?

Model-Free LQR

we **do not** have access to the model (A, B) or cost matrices (Q, R)

- Riccati approach won't work
- gradient descent to find K? $\checkmark$

Model-Free LQR

we **do not** have access to the model (A, B) or cost matrices (Q, R)

- Riccati approach won't work
- gradient descent to find K? ✓

Policy Iteration

initially assume **we do** have access to (A, B, Q, R) and we want to solve

$$\begin{aligned} \underset{K}{\text{minimize}} \quad & C(K) \\ \text{s.t.} \quad & \text{(dynamics)} + \text{(stability)} \end{aligned}$$

try to apply gradient descent:

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

[Fazel, Ke, Kakade, Meshahi, ICML, 2018]

LQR Reformulation

we can equivalently rewrite the quadratic cost function

$$C(K) \triangleq \mathbb{E}_{x_0 \sim \mathcal{D}} \left[\sum_{t=0}^{\infty} x^T \left(Q + K^\top R K \right) x_t \right] = \mathbb{E}_{x_0 \sim \mathcal{D}} \; x_0^T P_K x_0$$

where P_K solves the Lyapunov equation

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

Reformulated LQR problem:

$$\begin{aligned} \underset{K}{\text{minimize}} \quad & \mathbb{E}_{x_0 \sim \mathcal{D}} \; x_0^T P_K x_0 \\ \text{s.t.} \quad & \text{(dynamics)} + \text{(stability)} \end{aligned}$$

- for $n \geq 3$ there exist non-convex problem instances

LQR Gradients

$\mathbb{E}_{x_0 \sim \mathcal{D}}\ x_0^T P_K x_0$ formulation of $C(K)$ makes it easier to compute a gradient:

$$\nabla C(K) = 2(\underbrace{(R + B^T P_K B)K - B^T P_K A}_{E_K})\Sigma_K,$$

where Σ_K is the state-correlation matrix:

$$\text{choose } K, \quad \underbrace{x_{t+1} = (A - BK)x_t}_{\texttt{closed-loop dynamics}}, \quad \Sigma_K \triangleq \mathbb{E}_{x_0 \sim \mathcal{D}} \sum_{t=0}^{\infty} x_t x_t^T$$

- **not** useful as an "object" in the model-free setting
- for analysis...

LQR Landscape

Gradient Dominance: [Polyak–Łojasiewicz]

a function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be **gradient dominated** if the there exits a scalar $\mu > 0$ such that

$$f(x) - f(x^\star) \leq \mu \|\nabla f(x)\|^2.$$

- used in place of strong convexity to ensure linear convergence rate of gradient descent

LQR Landscape

Gradient Dominance: [Polyak–Łojasiewicz]

a function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be **gradient dominated** if the there exits a scalar $\mu > 0$ such that

$$f(x) - f(x^\star) \leq \mu \|\nabla f(x)\|^2.$$

- used in place of strong convexity to ensure linear convergence rate of gradient descent

LQR is Gradient Dominated! [Fazel et. al]

for any K such that $C(K) < \infty$, we have

$$C(K) - C(K^\star) \leq \underbrace{\frac{\|\Sigma_{K^\star}\|}{\sigma_{\min}(\Sigma_K)^2 \sigma_{\min}(R)}}_{\mu} \|\nabla C(K)\|_F^2$$

$\implies \quad \nabla C(K) = 0$ then K is optimal (or Σ_K not full-rank)

Model-Based Policy Gradient

LQR landscape is "approximately smooth", for $t = 1, \ldots, N$:

- **Gradient Descent:**

$$K_{t+1} \leftarrow K_t - \eta \nabla C(K_t)$$

produces a controller that satisfies

$$C(K_N) - C(K^\star) \leq \epsilon.$$

Model-Based Policy Gradient

LQR landscape is "approximately smooth", for $t = 1, \ldots, N$:

- **Gradient Descent:**
$$K_{t+1} \leftarrow K_t - \eta \nabla C(K_t)$$
produces a controller that satisfies
$$C(K_N) - C(K^\star) \leq \epsilon.$$

- **Natural Policy Gradient:**
$$K_{t+1} \leftarrow K_t - \eta \nabla C(K_t) \Sigma_{K_t}^{-1}$$

- **Gauss-Newton:**
$$K_{t+1} \leftarrow K_t \eta \nabla (R + B^T P_{K_t} B)^{-1} \nabla C(K_t) \Sigma_{K_t}^{-1}$$

Model-Based LQR

- **Gradient Descent:**

$$K_{t+1} \leftarrow K_t - \eta \nabla C(K_t)$$

- **Natural Policy Gradient:**

$$K_{t+1} \leftarrow K_t - \eta \nabla C(K_t) \Sigma_{K_t}^{-1}$$

- **Gauss-Newton:**

$$K_{t+1} \leftarrow K_t - \eta (R + B^T P_{K_t} B)^{-1} \nabla C(K_t) \Sigma_{K_t}^{-1}$$

methods require oracle access to: $\nabla C(K_t)$, $\Sigma_{K_t}^{-1}$, $(R + B^T P_{K_t} B)^{-1}$

Model-Free LQR

- we **do not** have access to (A, B, Q, R)
- have access to a **closed-loop** simulation that for a given K, produces

$$\{x_t, u_t\}_{t=0}^{l}$$

- use the simulation data to provide a gradient estimate and then run

$$K_{t+1} \leftarrow K_t - \eta \widehat{\nabla C(K_t)}$$

Model-Free LQR

- we **do not** have access to (A, B, Q, R)
- have access to a **closed-loop** simulation that for a given K, produces

$$\{x_t, u_t\}_{t=0}^{l}$$

- use the simulation data to provide a gradient estimate and then run

$$K_{t+1} \leftarrow K_t - \eta \widehat{\nabla C(K_t)}$$

One-Point Gradient Estimate: `ZeroOrder`(K, r, n_s, τ)

- draw n_s random matrices U_s, s.t. $\|U_s\|_F = r$, for $s = 1, \ldots, n_s$

$$\widehat{\nabla C(K)} = \frac{1}{n_s} \sum_{s=1}^{n_s} C(K + U_s; x_0) \frac{nm}{r^2}, \quad \texttt{// } C \texttt{ horizon length } \tau$$

is a biased estimate of $\nabla C(K)$

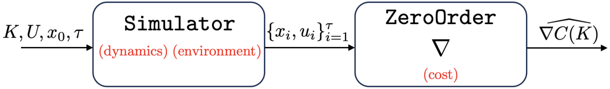
Model-Free LQR: Convergence

- we **do not** have access to (A, B, Q, R)
- have access to a closed-loop simulation that for a given K, produces

$$\{x_t, u_t\}_{t=0}^{l}$$

- use the simulation data to provide a gradient estimate and then run

$$K_{t+1} \leftarrow K_t - \eta \widehat{\nabla C(K_t)}$$



[Malik et al., JMLR, 2020], [Neshaei et al. arXiv, 2024],[Mohammadi et al. TAC, 2022]

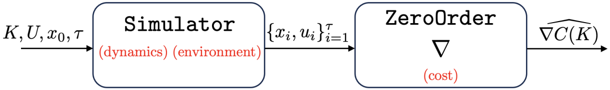
Model-Free LQR: Convergence

- we **do not** have access to (A, B, Q, R)
- have access to a closed-loop simulation that for a given K, produces

$$\{x_t, u_t\}_{t=0}^{l}$$

- use the simulation data to provide a gradient estimate and then run

$$K_{t+1} \leftarrow K_t - \eta \widehat{\nabla C(K_t)}$$



- for k sufficiently large, the one-point gradient estimates converges w.h.p.:

$$C(K_k) - C(K^\star) \leq \epsilon$$

- polynomial computational and sample complexity

[Malik et al., JMLR, 2020], [Neshaei et al. arXiv, 2024],[Mohammadi et al. TAC, 2022]

Federated LQR

for full details...

arXiv > math > arXiv:2308.11743

Mathematics > Optimization and Control

[Submitted on 22 Aug 2023]

Model-free Learning with Heterogeneous Dynamical Systems: A Federated LQR Approach

Han Wang, Leonardo F. Toso, Aritra Mitra, James Anderson

Problem formulation

- given $i = 1, \ldots, M$ (stabilzable) LTI systems

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

Problem formulation

- given $i = 1, \ldots, M$ (stabilzable) LTI systems

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

- construct a **common** state feedback controller, $u_t^{(i)} = K x_t^{(i)}$, that solves

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

s.t. $\{\text{(dynamics}_i\text{)}\}_{i=1}^{M} + \{\text{(stability}_i\text{)}\}_{i=1}^{M}$

Questions & Challenges

1. **Is this common policy stabilizing for all the systems? If so, under what conditions?**

Questions & Challenges

1. **Is this common policy stabilizing for all the systems? If so, under what conditions?**

2. **How far is the learned common policy from each agent's locally optimal policy?**

Questions & Challenges

1. **Is this common policy stabilizing for all the systems? If so, under what conditions?**

2. **How far is the learned common policy from each agent's locally optimal policy?**

3. **What is the (sample complexity) benefit to each participating agent?**

Questions & Challenges

1. **Is this common policy stabilizing for all the systems? If so, under what conditions?**

2. **How far is the learned common policy from each agent's locally optimal policy?**

3. **What is the (sample complexity) benefit to each participating agent?**

4. **Can the optimal controller be applied and fine-tuned on unseen systems? [meta-learning, see later]**

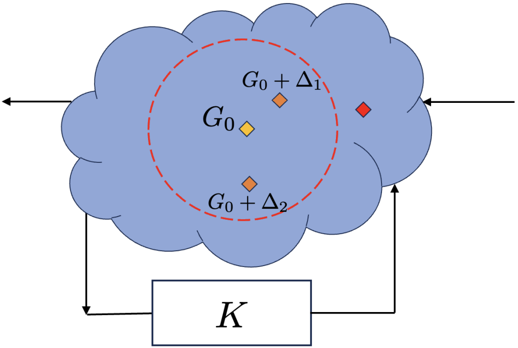
System Heterogeneity

we cannot expect a solution to Fed-LQR problem for arbitrary systems

- system heterogeneity

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

- contrast to classical robust control of nominal+perturbation



Low-Heterogeneity Regime

Scalar Example

consider a simple 2 system setting, with

$$x_{t+1}^{(1)} = \alpha x_t^{(1)} + u_t^{(1)}, \qquad x_{t+1}^{(2)} = -\alpha x_t^{(2)} + u_t^{(2)}$$

and controller $u_t^{(i)} = K x_t^{(i)}$ for $i = 1, 2$

Low-Heterogeneity Regime

Scalar Example

consider a simple 2 system setting, with

$$x_{t+1}^{(1)} = \alpha x_t^{(1)} + u_t^{(1)}, \qquad x_{t+1}^{(2)} = -\alpha x_t^{(2)} + u_t^{(2)}$$

and controller $u_t^{(i)} = K x_t^{(i)}$ for $i = 1, 2$

- $\epsilon_A = 2\alpha$ and $\epsilon_B = 0$
- $\epsilon_A > 2 \implies \alpha > 1 \implies$ both systems unstable
- for stability require $|\alpha - K| < 1$ **and** $|\alpha + K| < 1$

Takeaway

we will need to impose some bound on the degree of heterogeneity

Aside: Quantifying System Heterogeneity

recall our definition:

$$\max_{i,j} \|A^{(i)} - A^{(j)}\| \leq \epsilon_A, \quad \text{and} \quad \max_{i,j} \|B^{(i)} - B^{(j)}\| \leq \epsilon_B, \quad \text{for all } i, j$$

are these systems really similar?

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

possible fixes:

- $\mu(M, \Delta)$
- ν-gap
- Lyapunov functions

Algorithm 2: Model-free Federated Policy Learing for LQR (`FedLQR`)

Input: no. of periods N, period length L, stepsizes η_l, η_g, initial policy K_0

for $n = 0, 1, \dots, N-1$ **do**
    `// server operations`
    broadcast K_n to all clients
    for *each client* $i \in [M]$ **in parallel do**
        $K_{n,0}^{(i)} \leftarrow K_n$
        for $l = 0, 1, \dots, L-1$ **do**
            $\widehat{\nabla C^{(i)}}(K_{n,l}^{(i)}) \leftarrow \texttt{ZeroOrder}(K_{n,l}^{(i)}, r, \tau)$ `// estimate gradient`
            $K_{n,l+1}^{(i)} \leftarrow K_{n,l}^{(i)} - \eta_l \widehat{\nabla C^{(i)}}(K_{n,l}^{(i)})$ `// update policy`
        send $\Delta_n^{(i)} \leftarrow K_{n,L}^{(i)} - K_n$ to server `// new − old`
    aggregate updates $K_{n+1} \leftarrow K_n + \frac{\eta_g}{M} \sum_{i \in \mathcal{S}_k} \Delta_n^{(i)}$ `// global update`

Model-Based Results: Bounded Gradient Difference

with access to $(A^{(i)}, B^{(i)})$ and Q, R, the global update for the controller is

$$K_{n+1} = K_n - \frac{L\eta_l\eta_g}{ML}\sum_{i=1}^{M}\sum_{l=0}^{L-1}\nabla C(K_{n,l}^{(i)})$$

- if ϵ_A, ϵ_B are small then their policy gradient directions "should be" close
- for any i, j, we have

$$\|\nabla C^{(i)}(K) - \nabla C^{(j)}(K)\| \leq \underbrace{\epsilon_A h_1(K) + \epsilon_B h_2(K)}_{\mathcal{O}(\epsilon_A+\epsilon_B)}$$

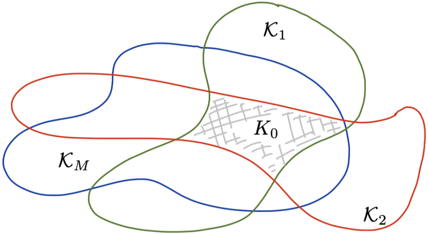
  where h_1, h_2 are bounded polynomials of the problem data

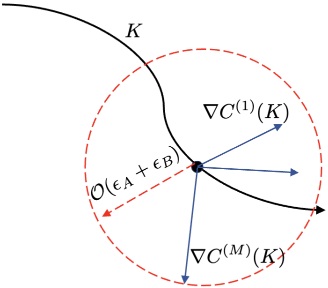
- gradient of agent i can be approximated by gradient of agent j

Model-Based Results: Bounded Gradient Difference

Bounded Policy Gradients

$$\|\nabla C^{(i)}(K) - \nabla C^{(j)}(K)\| \le \underbrace{\epsilon_A h_1(K) + \epsilon_B h_2(K)}_{\mathcal{O}(\epsilon_A + \epsilon_B)}$$





Model-Based Results: Agent Optimality

Distance between K_N and $K_i^\star$: [informal]

For each agent, after N rounds, if $\underbrace{(\epsilon_A g_1 + \epsilon_B g_2)^2 < g_3}_{\text{low heterogeneityregime}}$, then

$$C^{(i)}(K_N) - C^{(i)}(K_i^\star) \leq \underbrace{\left(1 - \eta\mu^2 C_1\right)^N}_{<1} \underbrace{\left(C^{(i)}(K_0) - C^{(i)}(K_i^{(\star)})\right)}_{\text{initial optimaility gap}} + \underbrace{C_u \mathcal{B}(\epsilon_A, \epsilon_B)}_{\text{bias}}$$

moreover K_N is stabilizing for all N.

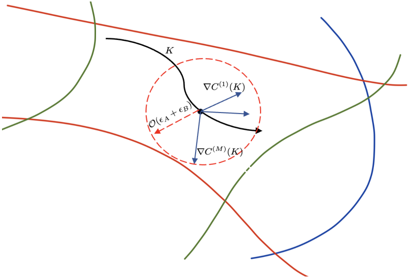
Model-Based Results: Agent Optimality

Distance between K_N and $K_i^\star$: [informal]

For each agent, after N rounds, if $\underbrace{(\epsilon_A g_1 + \epsilon_B g_2)^2 < g_3}_{\text{low heterogeneityregime}}$, then

$$C^{(i)}(K_N) - C^{(i)}(K_i^\star) \leq \underbrace{\left(1 - \eta\mu^2 C_1\right)^N}_{<1} \underbrace{\left(C^{(i)}(K_0) - C^{(i)}(K_i^{(\star)})\right)}_{\text{initial optimaility gap}} + \underbrace{C_u \mathcal{B}(\epsilon_A, \epsilon_B)}_{\text{bias}}$$

moreover K_N is stabilizing for all N.



Model-Based Results: Agent Optimality

Distance between K_N and $K_i^\star$: [informal]

For each agent, after N rounds, if $(\epsilon_A g_1 + \epsilon_B g_2)^2 < g_3$, then

$$C^{(i)}(K_N) - C^{(i)}(K_i^\star) \leq \underbrace{\left(1 - \eta\mu^2 C_1\right)^N}_{<1} \underbrace{C^{(i)}(K_0) - C^{(i)}(K_i^{(\star)})}_{\text{initial gap}} + \underbrace{C_u \mathcal{B}(\epsilon_A, \epsilon_B)}_{\text{bias}}$$

moreover K_N is stabilizing for all N.

Distance between $K^\star$ and $K_i^\star$: [informal]

For all agents

$$C^{(i)}(K^\star) - C^{(i)}(K_i^\star) = \mathcal{O}((\epsilon_A + \epsilon_B)^2).$$

Model-Free Results:

Variance Reduction

provided n_s and τ large enough, and r small enough, then w.h.p

$$\left\| \frac{1}{ML} \sum_{i=1}^{M} \sum_{l=0}^{L-1} \left[\widehat{\nabla C^{(i)}(K_{n,l}^{(i)})} - \nabla C^{(i)}(K_{n,l}^{(i)}) \right] \right\|_F \leq \epsilon$$

Model-Free Results:

Variance Reduction

provided n_s and τ large enough, and r small enough, then w.h.p

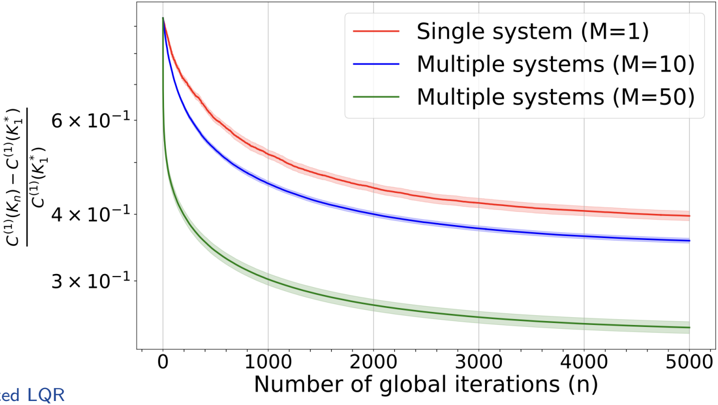
$$\left\| \frac{1}{ML} \sum_{i=1}^{M} \sum_{l=0}^{L-1} \left[\widehat{\nabla C^{(i)}(K_{n,l}^{(i)})} - \nabla C^{(i)}(K_{n,l}^{(i)}) \right] \right\|_F \leq \epsilon$$

- each agent obtains a $\frac{1}{ML}$ speed up per iteration relative to centralized case
- all results from model-based setting carry through!
- overall sample complexity improved by a factor $\tilde{\mathcal{O}}(\frac{1}{M})$
- each agent's sample cost improved from $\tilde{\mathcal{O}}(\frac{1}{\epsilon^2})$ to $\tilde{\mathcal{O}}(\frac{1}{M\epsilon^2})$

Numerics

Performance as a function of number of agents:

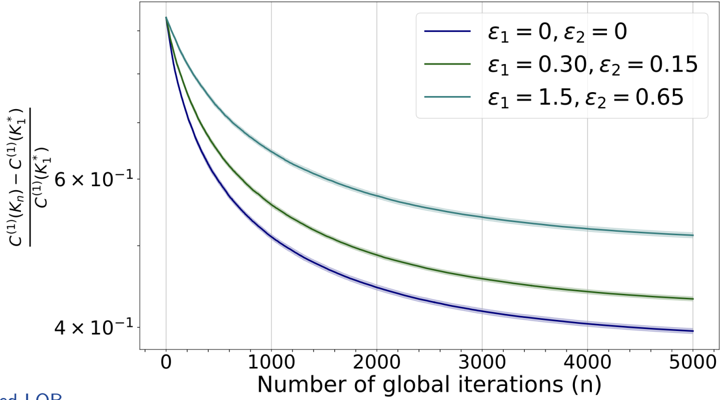
- **System:** 3 states, 3 inputs
- **Heterogeneity:** $\epsilon_A = \epsilon_B = \frac{1}{2}$
- ZO **Parameters:** $n_s = 5$, $\tau = 15$, $r = 0.1$



Numerics

Performance as a function of number of agents:

- **System:** 3 states, 3 inputs
- **No. Systems:** $M = 10$
- ZO **Parameters:** $n_s = 5$, $\tau = 15$, $r = 0.1$



Questions & Challenges

1. **Is this common policy stabilizing for all the systems? If so, under what conditions?**
2. **How far is the learned common policy from each agent's locally optimal policy?**
3. **What is the (sample complexity) benefit to each participating agent?**
4. **Can the optimal controller be applied and fine-tuned on unseen systems?**

For full details...

Proceedings of Machine Learning Research vol 242:902–915, 2024 6th Annual Conference on Learning for Dynamics and Control

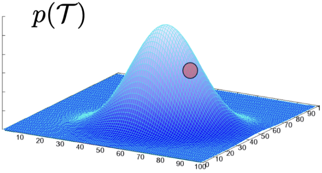
Meta-Learning Linear Quadratic Regulators: A Policy Gradient MAML Approach for Model-free LQR

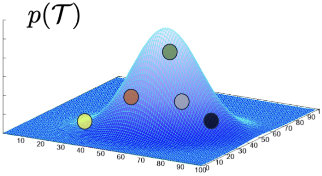
Leonardo F. Toso LT2879@COLUMBIA.EDU
Donglin Zhan DZ2478@COLUMBIA.EDU
James Anderson JAMES.ANDERSON@COLUMBIA.EDU
Han Wang HW2786@COLUMBIA.EDU
Columbia University, New York, NY

Meta-Learning for Control

learn a controller that is efficiently adaptable to all tasks in a distribution

$$x_{t+1} = \underbrace{\begin{bmatrix} 2 & -1 & \alpha \\ \beta & \delta & 4 \\ 0 & \gamma & -2 \end{bmatrix} x_t + \begin{bmatrix} -1 \\ \kappa \\ 2 \end{bmatrix}}_{\bigcirc} u_t$$

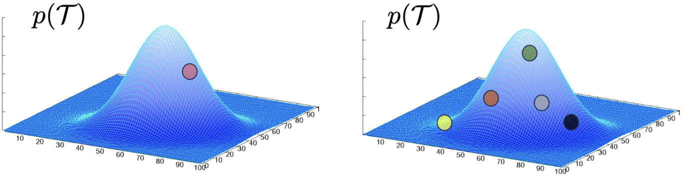




Meta-Learning for Control

learn a controller that is efficiently adaptable to all tasks in a distribution

$$x_{t+1} = \underbrace{\begin{bmatrix} 2 & -1 & \alpha \\ \beta & \delta & 4 \\ 0 & \gamma & -2 \end{bmatrix} x_t + \begin{bmatrix} -1 \\ \kappa \\ 2 \end{bmatrix}}\, u_t$$



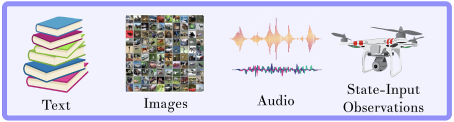
- the task: $\mathcal{T}^{(i)} := (A^{(i)}, B^{(i)}, Q^{(i)}, R^{(i)})$
- task objective:

$$C^{(i)}(K) \triangleq \left[\sum_{t=0}^{\infty} x^{(i)T} \left(Q^{(i)} + K^\top R^{(i)} K \right) x_t^{(i)} \right]$$

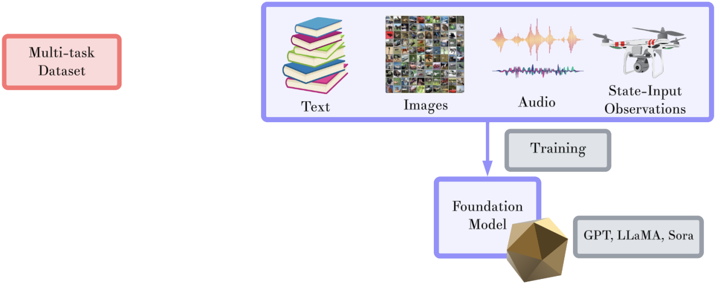
Meta-LQR

Meta-Learning: Learning to Learn

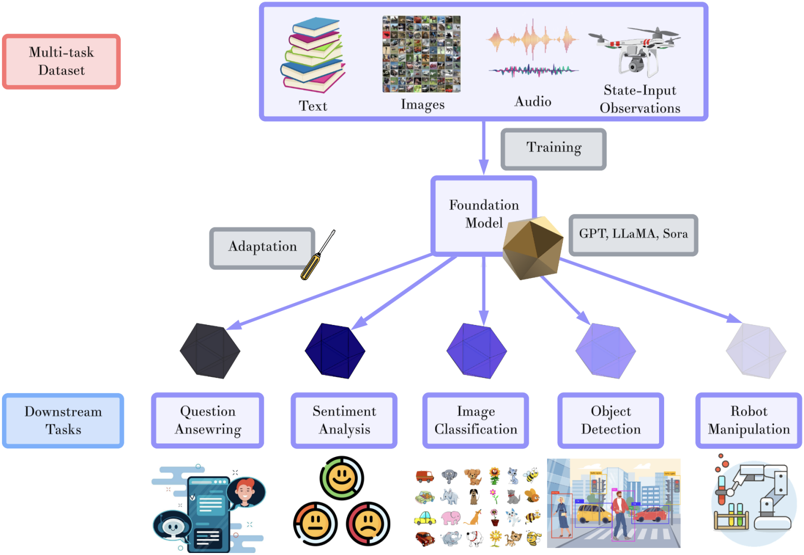
Multi-task Dataset



Meta-Learning: Learning to Learn



Meta-Learning: Learning to Learn



Meta-Learning for Control

$$x_{t+1} = A^{(1)} x_t + B^{(1)} u_t \text{ (sys)}$$

$(Q^{(1)}, R^{(1)})$ (env)

Design: $\{u_t\}_{t \geq 0}$ such that

$\min_{u_t} \mathbb{E} \sum_{t=0}^{\infty} c(\text{sys}, \text{env})$

Meta-Learning for Control

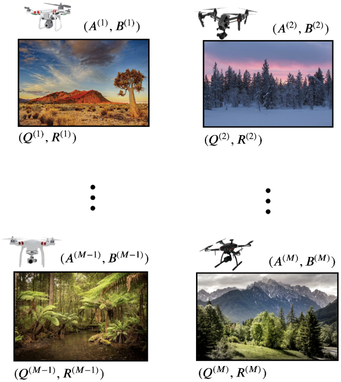
$x_{t+1} = A^{(1)} x_t + B^{(1)} u_t$ (sys)

$(Q^{(1)}, R^{(1)})$ (env)

Design: $\{u_t\}_{t\geq 0}$ such that

$\min_{u_t} \mathbb{E} \sum_{t=0}^{\infty} c(\text{sys}, \text{env})$

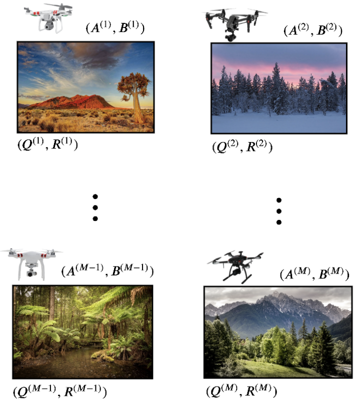
Training Tasks



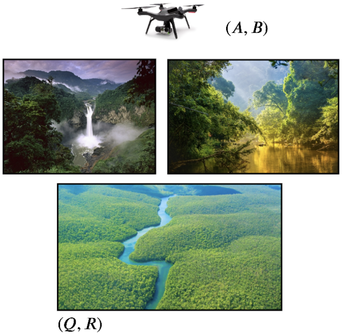
Downstream Applications

Design $\{u_t\}_t$ that adapts to different sys and env

Training Tasks



Downstream Tasks



Model Agnostic Meta Learning: MAML

consider the setting where tasks $\tau^{(i)} \sim p(\mathcal{T})$, $i \in \{1, \ldots, M\}$

- Features $x_{\tau^{(i)}}$
- Labels $y_{\tau^{(i)}}$
- Dataset $\mathcal{D}_{\tau^{(i)}} = \{x_{n,\tau^{(i)}}, y_{n,\tau^{(i)}}\}_{n=1}^{N}$
- Cost $\ell_{\tau^{(i)}}(\theta, \mathcal{D}_{\tau^{(i)}})$, for some model parameter $\theta \in \Theta$

Model Agnostic Meta Learning: MAML

consider the setting where tasks $\tau^{(i)} \sim p(\mathcal{T})$, $i \in \{1, \ldots, M\}$

- Features $x_{\tau^{(i)}}$
- Labels $y_{\tau^{(i)}}$
- Dataset $\mathcal{D}_{\tau^{(i)}} = \{x_{n,\tau^{(i)}}, y_{n,\tau^{(i)}}\}_{n=1}^{N}$
- Cost $\ell_{\tau^{(i)}}(\theta, \mathcal{D}_{\tau^{(i)}})$, for some model parameter $\theta \in \Theta$

Goal: Learn an initialization $\theta_0 \in \Theta$ that solves

$$\min_{\theta_0 \in \Theta} \frac{1}{M} \sum_{i=1}^{M} \ell_{\tau^{(i)}}(\hat{\theta}_0, \mathcal{D}_{\tau^{(i)}}),$$

$$\text{subject to } \hat{\theta}_0 = \underbrace{\theta_0 - \eta_l \nabla_{\theta_0} \ell_{\tau^{(i)}}(\theta_0, \mathcal{D}_{\tau^{(i)}})}_{\text{1 step PG}} \quad \text{(MAML)}$$

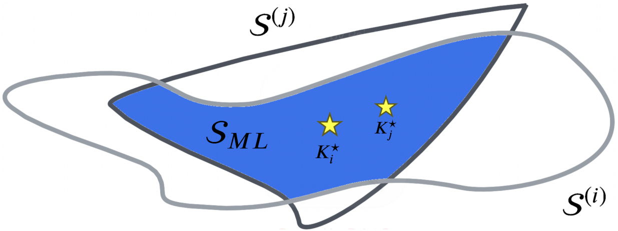
[Finn, Abeel, Levine, ICML, 2017]

Model Agnostic Meta-Learning

MAML-LQR Objective: Design a controller $K^\star_{ML}$ that can efficiently adapt to any task drawn from $p(\mathcal{T})$, i.e.,

$$K^\star_{\text{ML}} = \text{argmin}_{K \in \mathcal{S}_{ML}} C_{\text{ML}}(K) := \frac{1}{M} \sum_{i=1}^{M} C^{(i)} \underbrace{\left(K - \eta_l \nabla C^{(i)}(K)\right)}_{\text{1 step PG}}$$

subject to $\{(\text{sys-dyn})\}_{i=1}^{M}$, with $\mathcal{S}_{ML} \triangleq \cap_{i\in[M]} \mathcal{S}^{(i)}$



[Molybog & Lavaei, CCTA, 2021],[Musavi & Dullerud, CDC, 2023]

Algorithm 3: Model-free Federated Policy Learing for LQR (FedLQR)

Input: no. of periods N, stepsizes η_l, η_g, initial policy K_0, tasks $\mathcal{T}$

for $n = 0, 1, \ldots, N-1$ **do**
  broadcast K_n to all clients
  for *each task* $i \in [M]$ **in parallel do**
    $K_0^{(i)} \leftarrow K_n$
    // estimate gradient
    $[\nabla \widehat{C^{(i)}(K_n)}, \nabla^2 \widehat{C^{(i)}(K_n)}] \leftarrow \texttt{ZeroOrder2}(K_n, r, \tau, n_s)$
    // update policy
    $K_n^{(i)} \leftarrow K_n - \eta_l \nabla \widehat{C^{(i)}(K_n)}, \quad H^{(i)} \leftarrow I - \eta_l \nabla^2 \widehat{C^{(i)}(K_n)}$
  $\nabla C^{(i)}(K_n^{(i)}) \leftarrow \texttt{ZeroOrder2}(K_n, r, \tau, n_s)$ // update task gradients
  $K_{N+1} \leftarrow K_N - \frac{\eta_g}{M} \sum_{i=1}^{M} H^{(i)} \nabla C^{(i)}(K_N^{(i)}) H^{(i)}$ // update MAML

MAML-LQR Properties

For appropriately chosen parameters, we have:

- every iteration of the algorithm produces a stabilizing controller
- for all tasks: $C^{(i)}(K_N) - C^{(i)}(K_i^\star) \leq \epsilon + c_1(\bar{\epsilon})$
- for all tasks: $C^{(i)}(K_{\mathsf{ML}}^\star) - C^{(i)}(K_i^\star) \leq c_2(\bar{\epsilon})$

where $\bar{\epsilon}$ defines the task heterogeneity

Numerics

- nominal system: unstable Boeing aircraft

$$A = \begin{bmatrix} 1.22 & 0.03 & -0.02 & -0.32 \\ 0.01 & 0.47 & 4.70 & 0 \\ 0.02 & -0.06 & 0.40 & 0 \\ 0.01 & -0.04 & 0.72 & 1.55 \end{bmatrix}, \quad B = \begin{bmatrix} 0.01 & 0.99 \\ -3.44 & 1.66 \\ -0.83 & 0.44 \\ -0.47 & 0.25 \end{bmatrix}$$

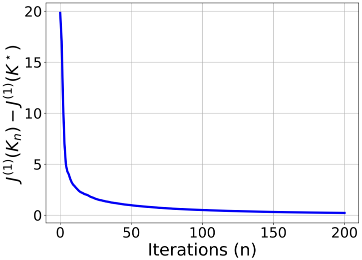
- initial stabilzing controller

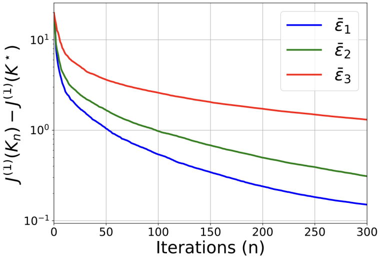
$$K_0 = \begin{bmatrix} 0.613 & -1.535 & 0.303 & 0.396 \\ 0.888 & 0.604 & -0.147 & -0.582 \end{bmatrix}$$

- heterogeneity ($\times 10^{-3}$): $M = 50$ tasks, with:

$$\epsilon_A = 1.2 \quad \epsilon_B = 1.1 \quad \epsilon_Q = 1.4 \quad \epsilon_R = 1.2$$

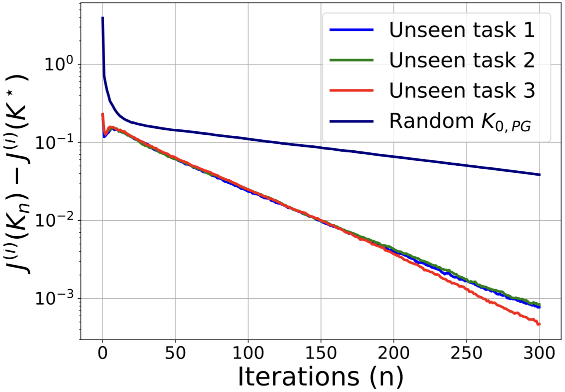
Numerics





- **Left:** gap between nominal task and MAML controller
- **Right:** varying levels of heterogeneity

Numerics



- three unseen tasks initiated from K^*
- one task initiated from K_0

Final Thoughts

- demonstrated that federated learning can be applied to optimal control
- proven sample and computational complexity performance boost as a function of number of agents and heterogeneity
- demonstrated that MAML can provably produce efficiently adaptable controllers
- bounded the optimality gap

james.anderson@columbia.edu