Multiplayer performative prediction

Maryam Fazel
Univ. of Washington

Joint work with:

Adhyyan Narang, Evan Faulkner, Dima Druvsyatskiy, Lillian Ratliff

Oaxaca Workshop on Optimization, Jan 9-13 2023

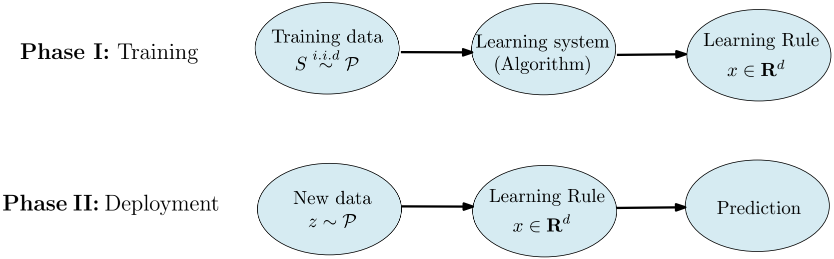
Thank you Steve!

IFDS
Institute for Foundations of Data Science

- NSF TRIPODS Phase II: Washington, Wisconsin, UC Santa Cruz, U Chicago
- Not possible (nor any fun!) without Steve... THANK YOU!

Pipeline of (classical) supervised learning

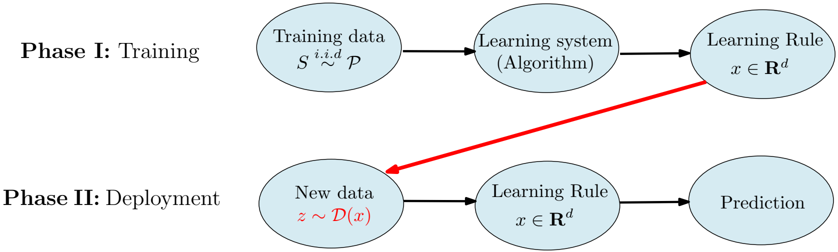
Assumption: Both "training data" and "test data" drawn from $\mathcal{P}$



Pipeline of supervised learning

Data distributions change due to

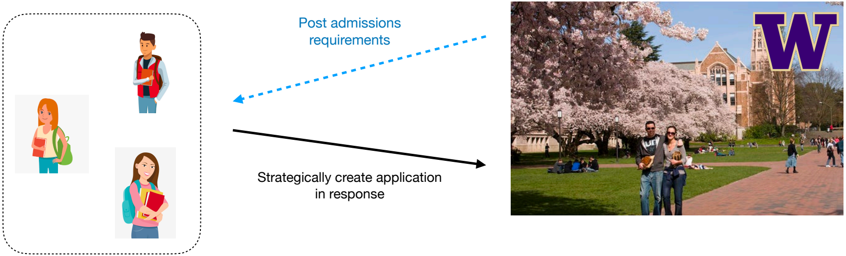
- time drift, dynamics (external effects)
- data generation itself *reacts* to learning rule



Prior work: performative prediction

[Perdomo,Zrnic,Dünner,Hardt, 2020] data z includes features+label; decision rule given by x

$$\min_{x \in \mathcal{X}} \mathbb{E}_{z \sim \mathcal{D}} \, \ell(z, x) \quad \longrightarrow \quad \min_{x \in \mathcal{X}} \mathbb{E}_{z \sim \mathcal{D}(x)} \, \ell(z, x)$$



Prior work: performative prediction

$$\min_{x \in \mathcal{X}} \mathbb{E}_{z \sim \mathcal{D}}\, \ell(z, x) \quad \longrightarrow \quad \min_{x \in \mathcal{X}} \mathbb{E}_{z \sim \mathcal{D}(x)}\, \ell(z, x)$$

Problem in x is nonconvex in general. Two paths forward:

Prior work: performative prediction

$$\min_{x \in \mathcal{X}} \mathbb{E}_{z \sim \mathcal{D}} \, \ell(z, x) \quad \longrightarrow \quad \min_{x \in \mathcal{X}} \mathbb{E}_{z \sim \mathcal{D}(x)} \, \ell(z, x)$$

Problem in x is nonconvex in general. Two paths forward:

- instead of optimality, check for *performative stablility* [Perdomo et al '20], [Mendler-Dunner et al '20],[Drusvyatskiy, Xiao '20]

$$\bar{x} = \arg\min_{x \in \mathcal{X}} \mathbb{E}_{z \sim \mathcal{D}(\bar{x})} \, \ell(z, x)$$

Prior work: performative prediction

$$\min_{x \in \mathcal{X}} \mathbb{E}_{z \sim \mathcal{D}} \, \ell(z, x) \quad \longrightarrow \quad \min_{x \in \mathcal{X}} \mathbb{E}_{z \sim \mathcal{D}(x)} \, \ell(z, x)$$

Problem in x is nonconvex in general. Two paths forward:

- instead of optimality, check for *performative stablility* [Perdomo et al '20], [Mendler-Dunner et al '20],[Drusvyatskiy, Xiao '20]

$$\bar{x} = \arg\min_{x \in \mathcal{X}} \mathbb{E}_{z \sim \mathcal{D}(\bar{x})} \, \ell(z, x)$$

  - describes fixed point of "retraining" (commonly used method)

Prior work: performative prediction

$$\min_{x \in \mathcal{X}} \mathbb{E}_{z \sim \mathcal{D}} \, \ell(z, x) \quad \longrightarrow \quad \min_{x \in \mathcal{X}} \mathbb{E}_{z \sim \mathcal{D}(x)} \, \ell(z, x)$$

Problem in x is nonconvex in general. Two paths forward:

- instead of optimality, check for *performative stablility* [Perdomo et al '20], [Mendler-Dunner et al '20],[Drusvyatskiy, Xiao '20]

$$\bar{x} = \arg\min_{x \in \mathcal{X}} \mathbb{E}_{z \sim \mathcal{D}(\bar{x})} \, \ell(z, x)$$

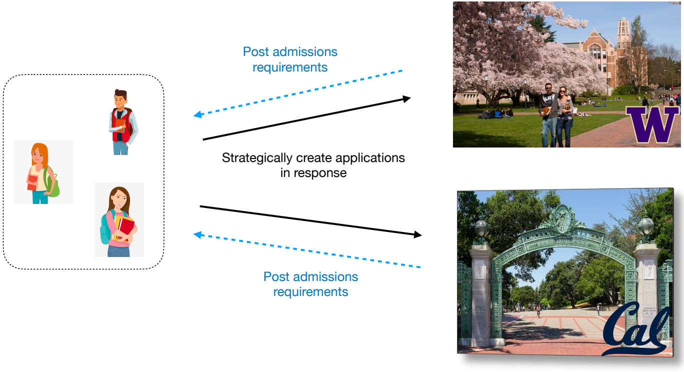
  - describes fixed point of “retraining” (commonly used method)
- identify conditions that make the problem convex [Miller et al 2021], then use convex optimization (e.g., [Izzo et al 2021])

Learning systems in real world: algorithms interact!

- **multiple** algorithms operate in an ecosystem
- population data reacts to the decisions of **all** algorithms (players)

Learning systems in real world: algorithms interact!

- **multiple** algorithms operate in an ecosystem
- population data reacts to the decisions of **all** algorithms (players)



other settings: ride-share platforms, driving-map apps, loan decisions, ...

This talk: Multi-player performative games

- model as an N-player game: each player solves for its own x_i (where x_{-i} denotes actions of other players):

$$\min_{x_i \in \mathcal{X}_i} \mathbb{E}_{z_i \sim \mathcal{D}_i(x_i, x_{-i})} \ell_i(z_i, x_i) \qquad i = 1 \ldots, N$$

This talk: Multi-player performative games

- model as an N-player game: each player solves for its own x_i (where x_{-i} denotes actions of other players):

$$\min_{x_i \in \mathcal{X}_i} \mathbb{E}_{z_i \sim \mathcal{D}_i(x_i, x_{-i})} \ell_i(z_i, x_i) \qquad i = 1 \ldots, N$$

- consider:
  - performatively stable points
  - Nash equilibria: no incentive to deviate unilaterally
- study **algorithms** that converge to these points—under suitable conditions
  - with access to different information/oracles, e.g., stochastic gradients

[Narang et al, AISTATS '22; arxiv], min-max: [Wood, Dall'Anese, '22]

Assumptions

convex $\mathcal{X}_i$ and

1. (Strong convexity, smoothness of losses)
   (i) $\ell_i(x, z_i)$ is α-strongly convex in x
   (ii) $z_i \mapsto \nabla_i \ell_i(x, z_i)$ is β_i-Lipschitz $\forall\, x \in \mathcal{X}$
2. (Lipschitz distributions) for some $\gamma_i > 0$,
$$\mathcal{W}_1(\mathcal{D}_i(x), \mathcal{D}_i(y)) \leq \gamma_i \, \|x - y\|, \qquad \forall x, y \in \mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_N,$$
   (Wasserstein-1 distance)
3. (Smoothness of distribution) for all $x \in \mathcal{X}$, the map $u_i \mapsto \mathbb{E}_{z_i \sim \mathcal{D}(u_i, x_{-i})} \ell_i(x, z_i)$ is differentiable at $u_i = x_i$ and its derivative is continuous

Challenge: two parts the gradient

Let's write the product rule for the gradient at x for a single player:

$$\min_x \mathbb{E}_{z\sim\mathcal{D}(x)}\, \ell(x,z)$$

$$\nabla\, \mathbb{E}_{z\sim\mathcal{D}(x)}\, \ell(x,z) = \mathbb{E}_{z\sim\mathcal{D}(x)}\, \nabla_x \ell(x,z) \;+\; \frac{d}{du}\mathbb{E}_{z\sim\mathcal{D}(u)}\, \ell(x,z)|_{u=x}$$

Challenge: two parts the gradient

Let's write the product rule for the gradient at x for a single player:

$$\min_x \mathbb{E}_{z\sim\mathcal{D}(x)}\, \ell(x,z)$$

$$\nabla\, \mathbb{E}_{z\sim\mathcal{D}(x)}\, \ell(x,z) = \underbrace{\mathbb{E}_{z\sim\mathcal{D}(x)}\, \nabla_x \ell(x,z)}_{\text{can compute by sampling}} + \underbrace{\frac{d}{du}\mathbb{E}_{z\sim\mathcal{D}(u)}\, \ell(x,z)\big|_{u=x}}_{\text{can't compute without knowing } \mathcal{D}}$$

- naive (myopic): ignore 2nd term, just retrain
- non-myopic: estimate the 2nd term

What does naive retraining converge to?

A fixed-point problem:

$$x^{t+1} = \text{argmin under } \mathcal{D}(x^t)$$

- when this map is a contraction, repeated retraining, repeated SGD, and variants converge (linearly) to fixed point $\bar{x}$
- contraction holds under **assumptions 1,2, and** $\rho < \mathbf{1}$ where $\rho := \frac{1}{\alpha}\sqrt{\sum_i (\beta_i \gamma_i)^2}$
- generalizes "performative stability" from single-player case

Non-myopic: Nash equilibrium for strongly monotone Games

- Definition: H is an α-strongly monotone map if

$$\langle H(z) - H(z'), z - z' \rangle \geq \alpha \|z - z'\|^2 \qquad \forall\, z, z' \in \mathbb{R}^d.$$

- in our setting, let $H_x(y) = (H_{1,x}(y), \ldots, H_{n,x}(y))$ where

$$H_{i,x}(y) := \frac{d}{du_i} \mathbb{E}_{z_i \sim \mathcal{D}(u_i, x_{-i})} \ell_i(y, z_i) \Big|_{u_i = x_i}$$

Theorem

With assumptions 1-3, $\rho < \frac{1}{2}$, and if $x \mapsto H_x(y)$ is monotone in x for each y, then the game is strongly monotone with parameter $(1 - 2\rho)\,\alpha$, and admits a unique Nash equlibrium.

- generalizes "mixture dominance" of distribution from single player case

Algorithms

For strongly monotone game, let $x^\star$ be Nash equilibrium

1. Derivative Free Method:
   - needs only samples from $\mathcal{D}(\hat{x}_i, x_{-i})$ and $\ell(z_i, \hat{x}_i)$ with random $\hat{x}_i$ on a sphere around x_i
   - complexity: $\mathbb{E}[\|x - x^\star\|^2] \leq \varepsilon$ after $O(\frac{d^2}{\varepsilon^2})$ iterations [Drusvyatskiy, F., Ratliff, 2022],[Bravo et al, 2018]
   - simple to use, but slow

Algorithms

For strongly monotone game, let $x^\star$ be Nash equilibrium

1. Derivative Free Method:
    - needs only samples from $\mathcal{D}(\hat{x}_i, x_{-i})$ and $\ell(z_i, \hat{x}_i)$ with random $\hat{x}_i$ on a sphere around x_i
    - complexity: $\mathbb{E}[\|x - x^\star\|^2] \leq \varepsilon$ after $O(\frac{d^2}{\varepsilon^2})$ iterations [Drusvyatskiy, F., Ratliff, 2022],[Bravo et al, 2018]
    - simple to use, but slow
2. Adaptive Method: (with parametric model for $\mathcal{D}_i$)
    - learn parameters from data: inject noise and query, update parameter estimates, update actions using estimated distribution
    - complexity: $O(\frac{d}{\varepsilon})$ iterations (for 'nice' distribution family)

Numerical example: rideshare platforms

Companies seek to maximize revenues by adjusting prices

- x_i: price adjustments across different locations for company i
- demand z_i seen by company i: $\quad z_i = \zeta_i + A_i x_i + A_{-i} x_{-i}$
  - ζ_i: empirical demands
  - x_i and x_{-i}: price adjustments
  - A_i, A_{-i} price elasticities
- Company i's loss: $\quad \ell_i(x_i, z_i) = -z_i^\top x_i + \frac{\lambda_i}{2}\|x_i\|^2$

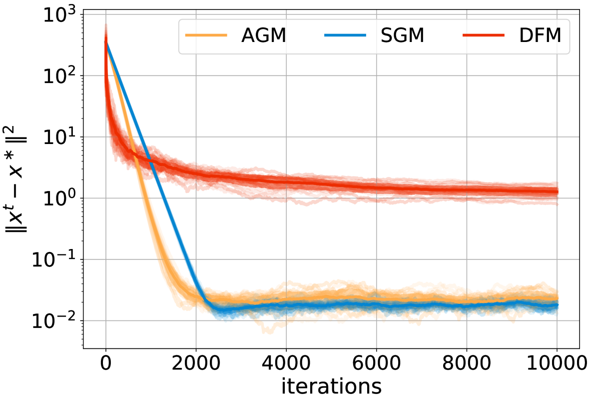
Numerical example: rideshare platforms

Companies seek to maximize revenues by adjusting prices

- x_i: price adjustments across different locations for company i
- demand z_i seen by company i: $z_i = \zeta_i + A_i x_i + A_{-i} x_{-i}$
  - ζ_i: empirical demands
  - x_i and x_{-i}: price adjustments
  - A_i, A_{-i} price elasticities
- Company i's loss: $\ell_i(x_i, z_i) = -z_i^\top x_i + \frac{\lambda_i}{2}\|x_i\|^2$
- data from Kaggle: Uber & Lyft, 1 month, Boston. ride data (location, time) and weather
- semi-synthetic experiments

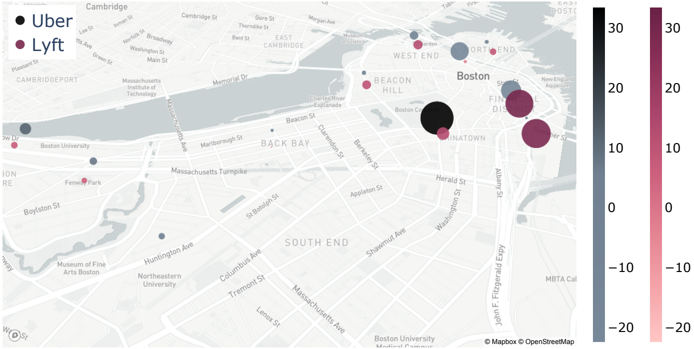
Numerical example: rideshare platforms

- Companies' price adjustments across locations given in x_i (for company i)
- Convergence to Nash for strongly monotone game



Numerical example: rideshare platforms

Revenue change by location over the myopic case (=not modeling performative term)



Summary & remarks

- In addition to 'indirect' coupling in distribution map $\mathcal{D}(x_i, x_{-i})$, can handle $\ell_i(x_i, x_{-i}, z_i)$
- Retraining algorithms converge to fixed points under mild assumptions
- Under stronger assumption of strongly monotone game, convergence to Nash (with different oracle settings)
- **Open directions:** non-Lipschitz distributions; more empirical studies