Distributionally Robust Feature Selection

Talk for R01 Team Meeting

Speaker: Maitreyi Swaroop

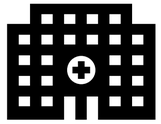
18 Sept, 2025

Flow

- Background + Motivation
- Problem statement
- Objective formulation
- Results

- [Disclaimer - in the course of this talk, I may use the words ***features***, ***variables*** and ***covariates*** interchangeably]

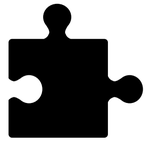
Background & Motivation



- **Clinical need**

- UPMC wants to screen patients for perinatal depression risk using survey responses +/ other patient records

  [collect patient features for diagnosis]



- **Challenges**

- Limited survey questions **[data acquisition]**
- Some Q/A may be irrelevant/prone to misreporting **[data quality/noise]**
- Centralized screeners may not be helpful across different populations **[fairness]**

- **Goal**

- From past data, identify the *optimal subset of features* needed to diagnose patients

Q1: ***What do we mean by optimal?***

Q2: ***How do we achieve this?***

Define optimal

- **Goal** 

- From past data, identify the *optimal subset of features* needed to diagnose patients

- **Desiderata**

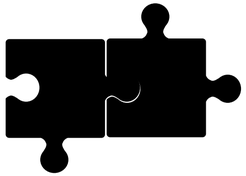
  - Selected features should be

  1. **Useful for diagnosing/prediction**
  2. **Informative across population groups**

How to achieve this?

- **Goal** 

- From past data, identify the *optimal subset of features* needed to diagnose patients where
- Selected features should be
  1. Useful for prediction
  2. Informative across population groups



Possible Solution

- Use regression - easy!
  - Predict the outcomes from the features,
  - Use regression coefficients as indicator of importance

Issues:

- Binds us to a specific prediction model
- Multiple-population setting

Setup

Problem Statement

- Known set of populations $P \in \mathscr{P}$ to which individuals may belong
- Each individual X_i has m features, and associated outcome Y_i
- Set the feature budget to $k < m$.
- **Goal**: to select a subset $\tilde{X}$ of $\leq k$ variables out of m, such that any model we train to predict Y using $\tilde{X}$ these variables performs well across different P.

Setup
Objective

- **Input** data $(X, Y) \sim P$
  - m features, $X \in \mathbb{R}^m$
  - Outcomes Y
- **Goal**: to select a subset $\tilde{X}$ of k variables out of m,
- $$\min_{i \in [1, \binom{m}{k}]} \quad \mathbb{E}[\text{Loss of a model trained on } (\tilde{X}_i, Y) \quad]$$

Enumerating subsets of size k from m

Setup

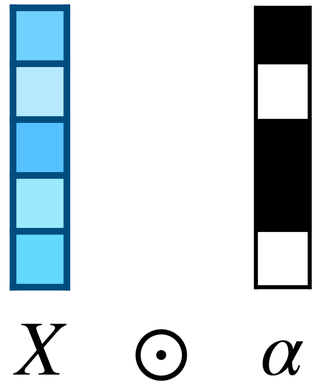
Objective

- **Input** data $(X, Y) \sim P$
  - m covariates, $X \in \mathbb{R}^m$
  - Outcomes Y
- **Goal**: to select a subset $\tilde{X}$ of k variables out of m, robust to distribution shifts
- $$\min_{i \in [1, \binom{m}{k}]} \max_{P_{test} \in \mathcal{P}} \mathbb{E}[\text{Loss of a model trained on } (\tilde{X}_i, Y) \sim P_{test}]$$

Choosing k out of m

Setup

Objective: Equivalent statement



- Problem of variable selection
  $\iff$ Problem of learning a mask $\alpha \in \{0,1\}^m$
  - $\min_{i \in [1, \binom{m}{k}]} \max_{P_{test} \in \mathcal{P}} \mathbb{E}[\text{Loss of a model trained on } (\tilde{X}_i, Y) \sim P_{test}]$
  - $\iff \min_{\alpha} \max_{P_{test} \in \mathcal{P}} \mathbb{E}[\text{Loss of a model trained on } (\alpha \odot X, Y) \sim P_{test}]$

    subject to the constraint $||\alpha||_0 = k$

    Basically means that only k entries of α can be 1, the remaining are 0
- **Issue:** combinatorial problem, tough to optimize

Setup

Objective: Equivalent statement

- Problem of learning a mask $\alpha \in \{0,1\}^m$
  - $$\min_{\alpha} \max_{P_{test} \in \mathscr{P}} \mathbb{E}[\text{Loss of a model trained on } (\alpha \odot X, Y) \sim P_{test}]$$

    subject to the constraint $||\alpha||_0 \leq k$
- **Issue:** combinatorial problem, tough to optimize

- **Need:** a *continuous* way of degrading the extent to which we use a variable

Setup

Objective: Equivalent statement

- Problem of learning a mask $\alpha \in \{0,1\}^m$
  - $$\min_{\alpha} \max_{P_{test} \in \mathscr{P}} \mathbb{E}[\text{Loss of a model trained on } (\alpha \odot X, Y) \sim P_{test}]$$

    subject to the constraint $||\alpha||_0 \leq k$
- **Issue:** combinatorial problem, tough to optimize

- **Need:** a *continuous* way of degrading the extent to which we use a variable

Setup

Objective: Equivalent statement

- Problem of learning a mask $\alpha \in [0,1]^m$ **"Relax" α to be a real number in the range [0,1]**
- $$\min_{\alpha} \max_{P_{test} \in \mathscr{P}} \mathbb{E}[\text{Loss of a model trained on } (\alpha \odot X, Y) \sim P_{test}]$$

  subject to the constraint $||\alpha||_0 \leq k$
- **Issue:** combinatorial problem, tough to optimize
- **Need:** a *continuous* way of degrading the extent to which we use a variable

Setup

Objective: Equivalent statement

- Problem of learning a mask $\alpha \in [0,1]^m$ **"Relax" α to be a real number in the range [0,1]**
  - $$\min_{\alpha} \max_{P_{test} \in \mathscr{P}} \mathbb{E}[\text{Loss of a model trained on } (\alpha \odot X, Y) \sim P_{test}]$$

    subject to the constraint $||\alpha||_1 \leq k$ **The sum of entries of α should be less than (or equal to) k**
- **Issue:** combinatorial problem, tough to optimize

- **Need:** a *continuous* way of degrading the extent to which we use a variable

Setup

Objective: Continuous relaxation

- Problem of learning a mask $\alpha \in [0,1]^m$ **"Relax" α to be a real number in the range [0,1]**
  - $$\min_{\alpha} \max_{P_{test} \in \mathscr{P}} \mathbb{E}[\text{Loss of a model trained on } (\alpha \odot X, Y) \sim P_{test}]$$

    subject to the constraint $||\alpha||_1 \leq k$ **i.e.** $\sum_{j=1}^{m} \alpha[j] \leq k$
  - $$\min_{\alpha} \left[\left[\max_{P_{test} \in \mathscr{P}} \mathbb{E}[\text{Loss of a model trained on } (\alpha \odot X, Y) \sim P_{test}] \right] + \lambda ||\alpha_1|| \right]$$

    Constraint is enforced by tweaking *hyperparameter* λ
- **Achieved:** a *continuous relaxation* of the original problem!

Setup

Objective: Continuous relaxation

- **Achieved:** a *continuous relaxation* of the original problem! **(Yay :))**
- $$\min_{\alpha} \left[\left[\max_{P_{test} \in \mathscr{P}} \mathbb{E}[\text{Loss of a model trained on } (\alpha \odot X, Y) \sim P_{test}] \right] + \lambda \|\alpha_1\| \right]$$
- **But there's still something missing here …**

Setup

Objective: Continuous relaxation

- **Achieved:** a *continuous relaxation* of the original problem! (Yay :))
- $$\min_{\alpha} \left[\left[\max_{P_{test} \in \mathcal{P}} \mathbb{E}[\text{Loss of a model trained on } (\alpha \odot X, Y) \sim P_{test}] \right] + \lambda \|\alpha_1\| \right]$$
- But there's still something missing here …
  - **What if the predictive model simply learns alpha to be a constant value here, say, $\frac{k}{m}$; then $\|\alpha_1\| = \sum_{j=1}^{m} \alpha[j] = k$!**

Each feature will have the same weight
$\Longrightarrow$no feature is really "selected"

Constraint is satisfied

Setup

Objective: Continuous relaxation

- **Achieved:** a *continuous relaxation* of the original problem!

- $$\min_{\alpha} \left[\left[\max_{P_{test} \in \mathscr{P}} \mathbb{E}[\text{Loss of a model trained on } (\alpha \odot X, Y) \sim P_{test}] \right] + \lambda \|\alpha_1\| \right]$$

- **We cannot apply α to X in a *deterministic way*.**
- ***Randomness needed*** to prevent all the indices of α from simply scaling all features
  - The randomness can be *controlled by α*

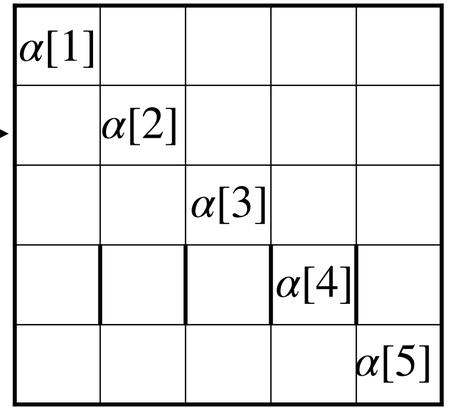
Setup

Objective: Continuous relaxation

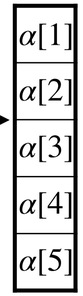
- **Achieved:** a *continuous relaxation* of the original problem!

- $$\min_{\alpha} \left[\left[\max_{P_{test} \in \mathscr{P}} \mathbb{E}[\text{Loss of a model trained on } (S(\alpha), Y) \sim P_{test}] \right] + \lambda \|\alpha_1\| \right]$$

- Replaced $\alpha \odot X$ with $S(\alpha) \sim \mathcal{N}(X, \text{Cov}_{\alpha})$



  - **The randomness is parameterised *by* α**



Setup

Objective: Continuous relaxation

- **Achieved:** a *continuous relaxation* of the original problem!
- $$\min_{\alpha} \left[\left[\max_{P_{test} \in \mathcal{P}} \mathbb{E}[\text{Loss of a model trained on } (S(\alpha), Y) \sim P_{test}] \right] + \lambda Reg(\alpha) \right]$$
- Replaced $\alpha \odot X$ with $S(\alpha) \sim \mathcal{N}(X, \text{Cov}_{\alpha})$
- **Final missing piece: what to put in place of Loss?**

Setup

Objective: Modelling [Loss of Model] Term

- $$\min_{\alpha} \left[\max_{P_{test} \in \mathcal{P}} \left[\mathbb{E}_{(S(\alpha), Y) \sim P_{test}} [\text{Loss of a model trained on } (S(\alpha), Y) \sim P_{test}] \right] + \lambda Reg(\alpha) \right]$$

- **Want:** our setting to be model-agnostic
  - Say, we have the best theoretical predictor, given our loss function and data
  - Bayes-optimal predictor!

Setup

Objective: Modelling [Loss of Model] Term

- $$\min_{\alpha} \left[\max_{P_{test} \in \mathcal{P}} \left[\mathbb{E}_{(S(\alpha), Y) \sim P_{test}} [\text{Loss of a model trained on } (S(\alpha), Y) \sim P_{test}] \right] + \lambda Reg(\alpha) \right]$$

- **Want:** our setting to be model-agnostic

- We can consider the loss of the **Bayes-optimal predictor**!

- Say the **loss function is mean-squared-error (MSE)**
  - Loss of Bayes-optimal predictor = Bias^2 + $\mathbb{V}[Y \mid S(\alpha)] = \mathbb{V}[Y \mid S(\alpha)]$ … Bias = 0

Setup

Objective: Modelling [Loss of Model] Term; TLDR

- $$\min_{\alpha} \left[\max_{P_{test} \in \mathcal{P}} \left[\mathbb{E}_{(S(\alpha),Y) \sim P_{test}} [\text{Loss of a model trained on } (S(\alpha), Y) \sim P_{test}] \right] + \lambda Reg(\alpha) \right]$$

- **Loss of a model trained on** $(S(\alpha), Y) \sim P_{test} = \mathbb{V}[Y \mid S(\alpha)]$

- $$\min_{\alpha} \left[\max_{P_{test} \in \mathcal{P}} \left[\mathbb{E}_{S(\alpha)} \left[\mathbb{V}[Y \mid S(\alpha)] \right] \right] + \lambda Reg(\alpha) \right]$$

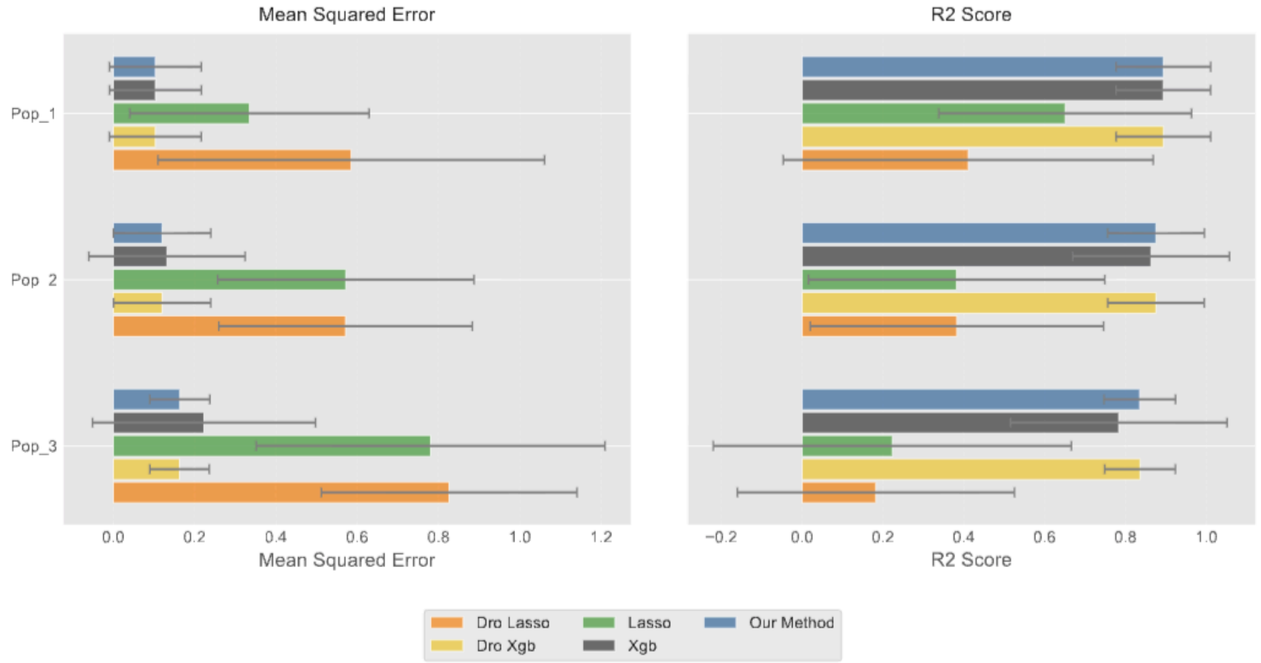
Additional concerns

Optimizing the objective

Results: It works!

On synthetic datasets

- **Synthetic example:**
- 3 populations
- 15 features each
- For each population,
  - draw 4 relevant features
  - 11 spurious or noise features.

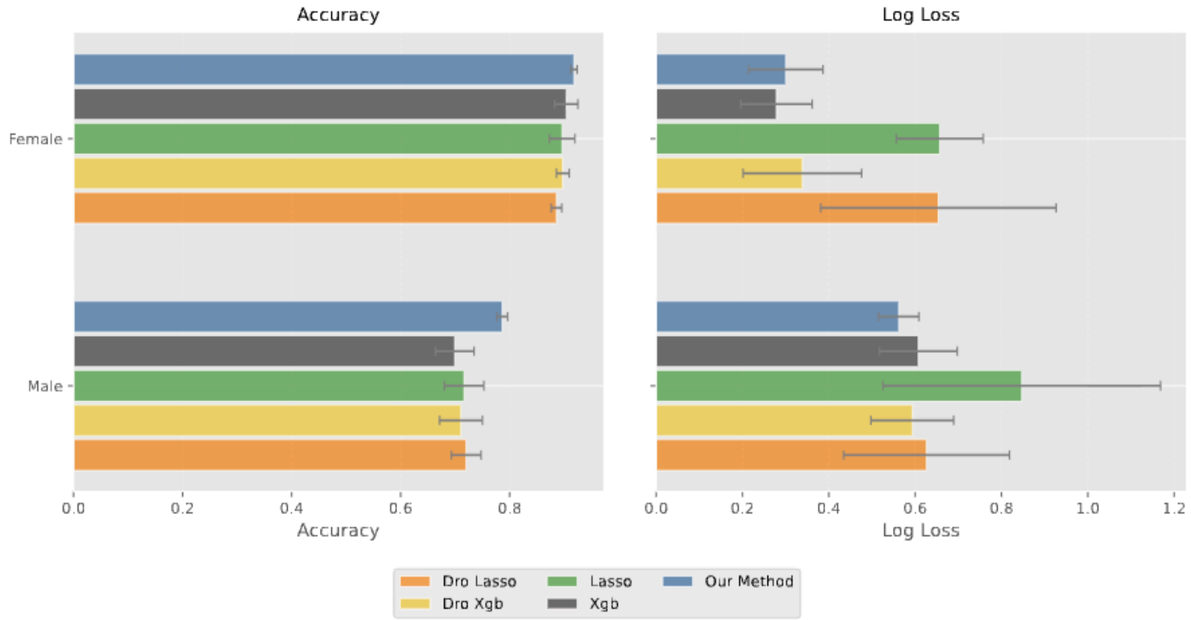


$$Y = w_1(X_{i_1} - 1)^2 + 0.2w_2 \exp\left(\frac{X_{i2}}{2}\right) + w_3\mathbb{I}[X_{i3} > 0.5] + w_4(X_{i4}^2 - 1) + \sum_{i \notin \{i_1 \ldots, i_4\}} X_i$$

Results: It works!

& semi-synthetic dataset

- **UCI Adult Income Dataset:**
- 2 populations : Male & Female
- 14 features each (44 after one-hot encoding)
- Target variable Y
  - Binary variable
  - Indicates whether an individual's income exceeds $50K per year



- [Shows that our method extends beyond MSE to classification based losses too!]

Next steps

- **Implement on the real dataset (welcome Shikha!:))**