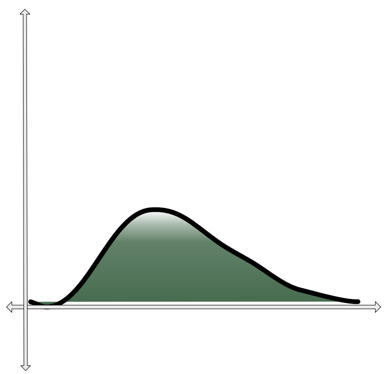
Learning from multiple data sources

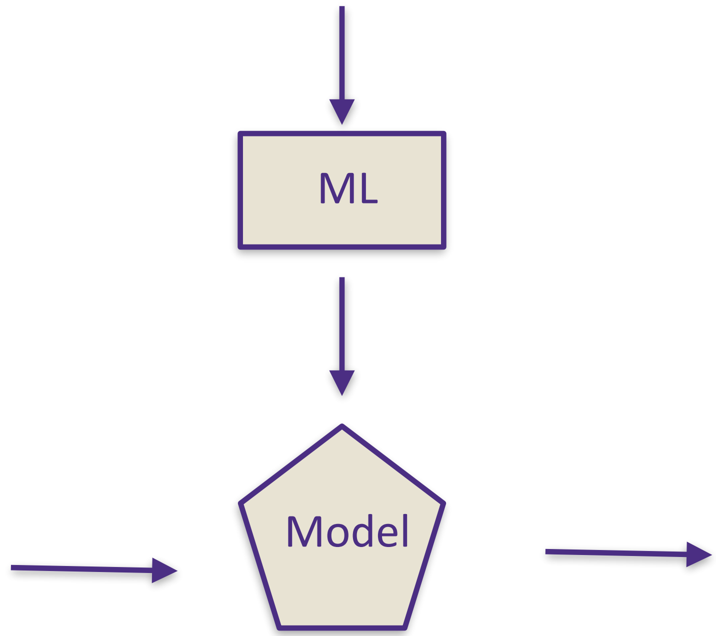
Jamie Morgenstern
joint work with
Chris Jung, Pranjal Awasthi

UNIVERSITY *of* WASHINGTON

Canonical learning paradigm

$$S = \{(x_1, y_1)\ldots(x_n, y_n) \mid (x_i, y_i) \sim_{iid} \mathscr{D}\}$$





$$S' = \{x'_1 \ldots x'_m \mid x'_i \sim_{iid} \mathscr{D}_{|X}\}$$

$$\{y'_1 \ldots y'_m\}$$

Are train and test identically distributed?

Merely the passage of time leads to drift

    On features, on labels…

And perhaps the distribution has changed *because* of the learning we've done

    We've chosen to sample from a non-iid distribution

    Data has "best responded" to our learning

Training data and test data are (virtually never) distributed equally.

Merely the passage of time leads to drift

- On features, on labels…

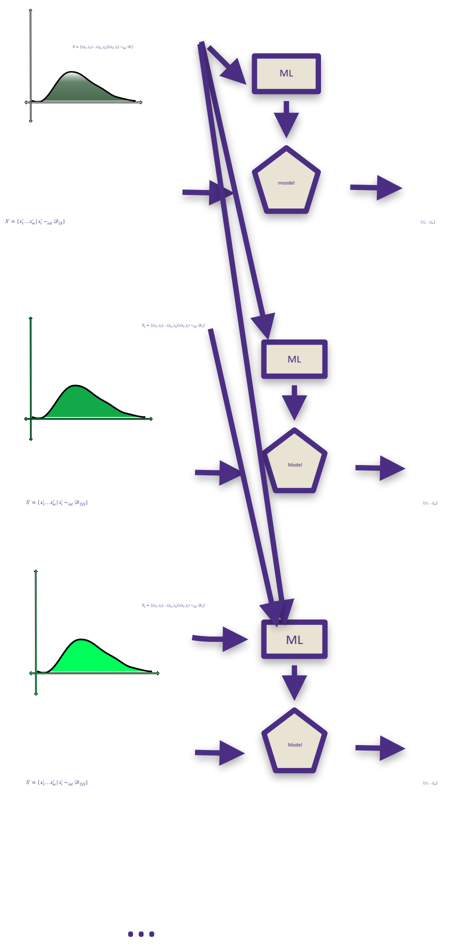
And perhaps the distribution has changed *because* of the learning we've done

- We've chosen to sample from a non-iid distribution
- Data has "best responded" to our learning

Often have multiple data sources, where some

- may be unlabelled
- may have auxilliary features
- likely follow different distributions

Instead, we often have



Where our test distribution may be unknown

and probably different from each training distribution

What should one do with i-non-i-d training data?

Standard uniform convergence isn't super relevant…

(at least) some of the training data cannot follow the test distribution

What differs between distributions?

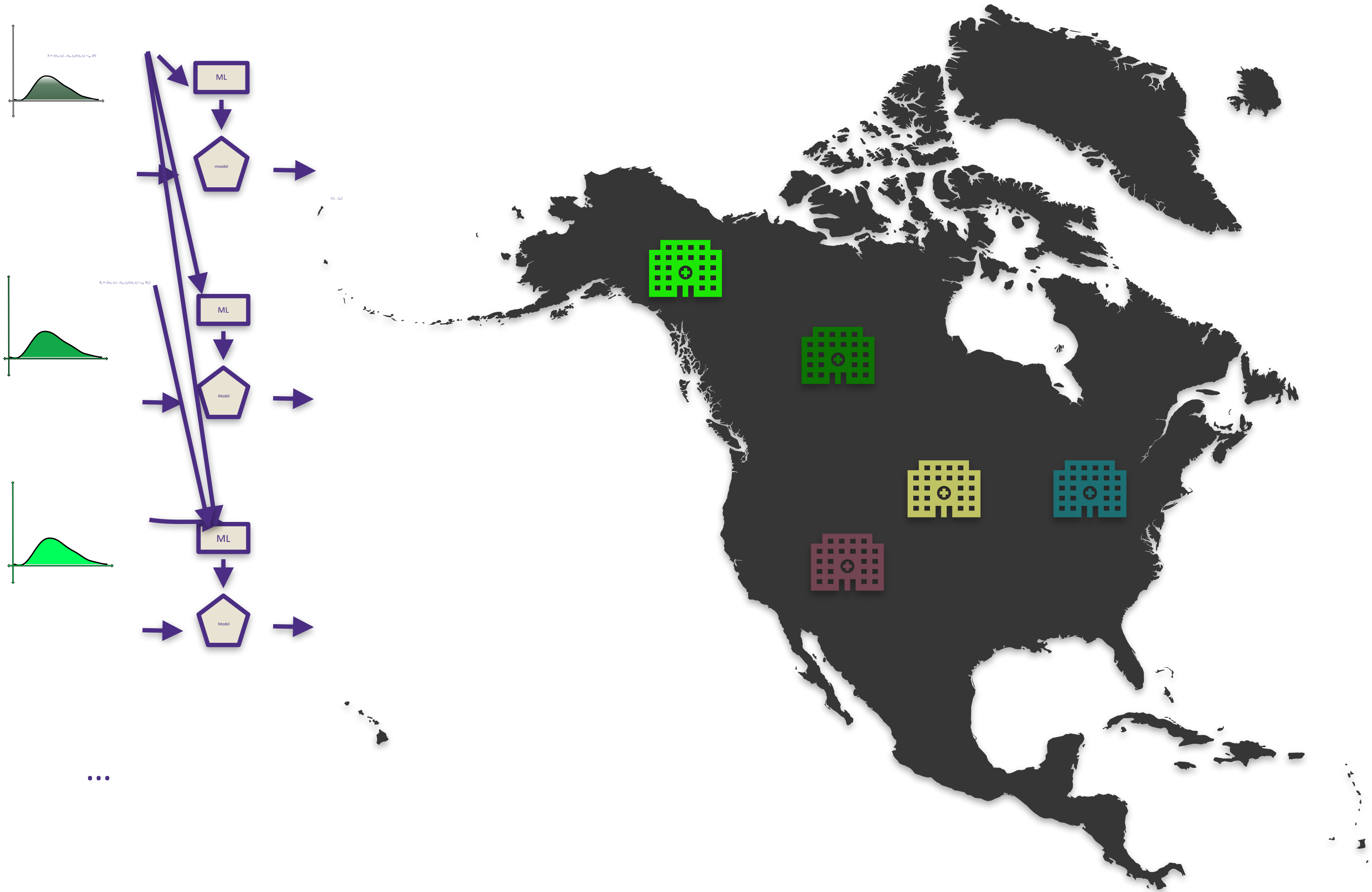
$\mathscr{D}_X$ —> covariate shift

$\mathscr{D}_{Y|X}$ —> model drift

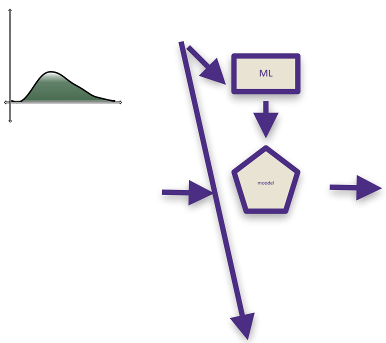
$X \to X \cup \{f_1, \ldots, f_t\}$ —> additional features

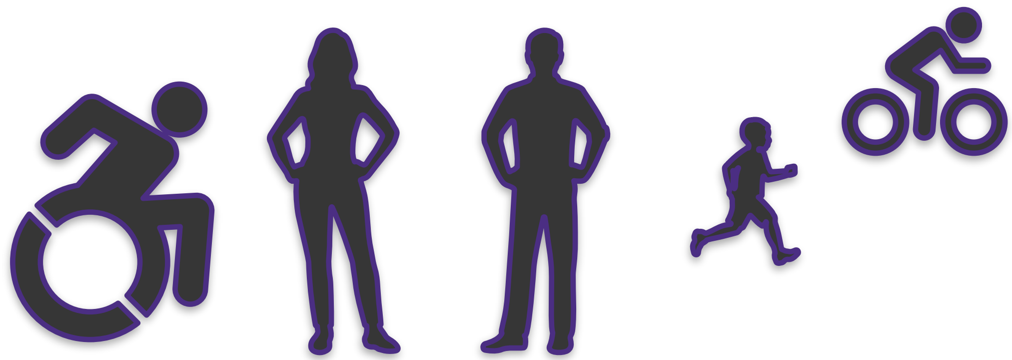
…

One application: learning from various hospitals



Another: ``patching” an unfair model





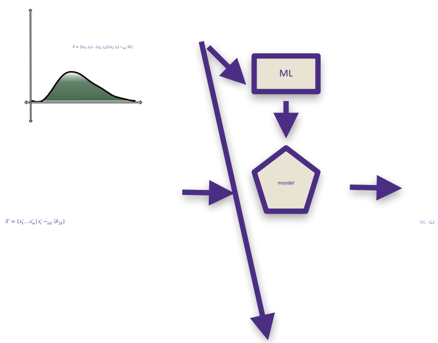
Observe high loss on some population P

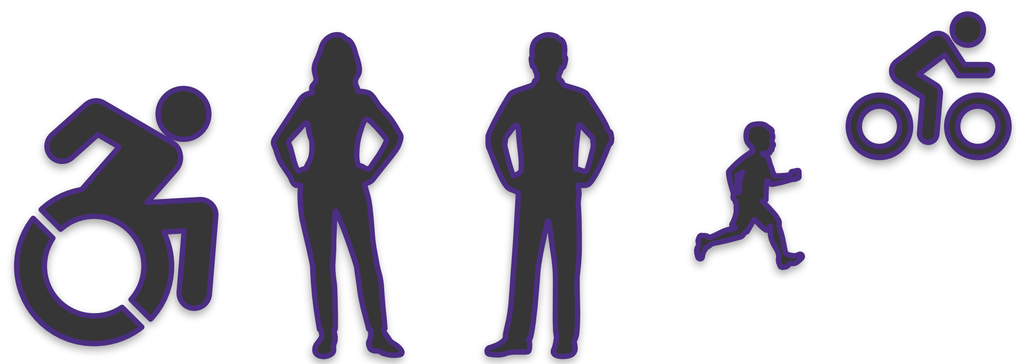
Some interventions:

- Dataset
    - Gather a brand new one / augment the existing one
    - possibly with demographic information
    - possibly with more folks from P
    - possibly with or without labels

- Retrain

Primary point





Reasoning about **training on multiple data sources** is super important, in particular with applications where one cares about equitable distribution of loss over different populations of people.

Related Work

Model subsumes:

DRO

[Scarf ’58, Záčková ‘66, Dupácová ‘87, Breton and El Hachem ‘95, Shapiro and Kleywegt ‘02, Shapiro and Ahmed ‘04]

Semi-supervised learning

Transfer learning/domain adaptation [...]

Generally assumes y|x might change, or x might change, rarely handles auxilliary features. Either assumes no information about test distribution or assumes sample access to labeled or unlabeled test data at training time.

DRO for fairness [HSNL’18]

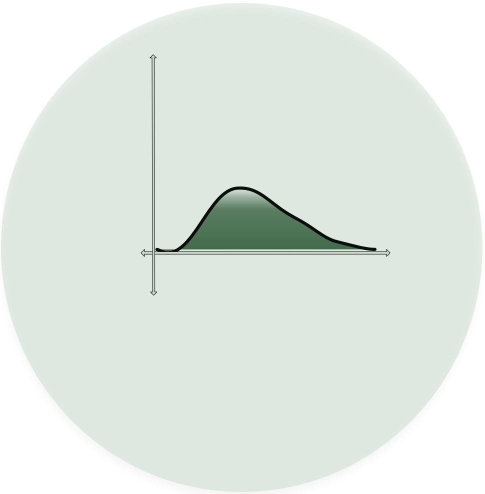
Multicalibration, omnipredictors [GKRSW’21, KKGR’22]

Outline

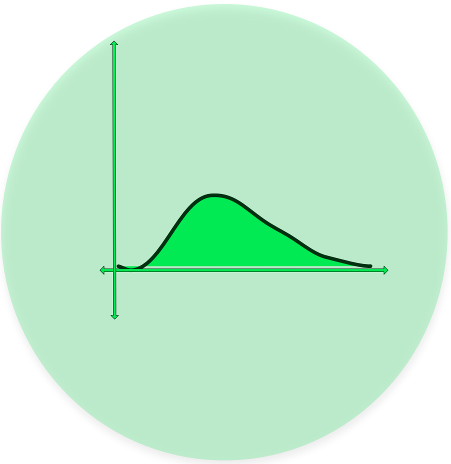
- Introduction and Motivation
- Our model
  - And a “fairness application”
- An algorithm designed for this problem
  - And a statement about its guarantees
- Experimental results

A formal model

D_P: features X, labels Y, radius r_P



D_A: features X, extra features A, radius r_A

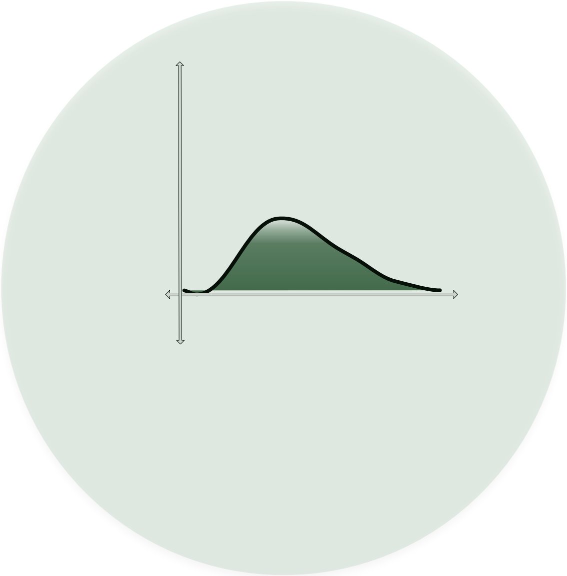


Build a linear model $\theta : (X \times A) \rightarrow Y$ which performs well on any distribution $D' \in B_{r_P}(D_P) \cap B_{r_A}(D_A)$

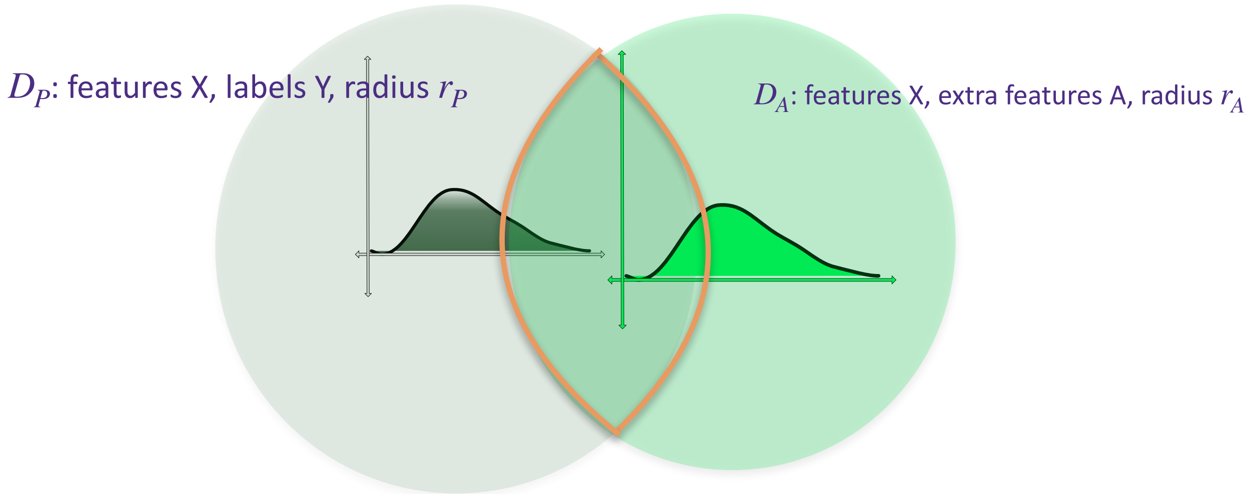
Distributionally robust optimization

DRO:

Find $\theta \in \text{argmin}_f \max_{D': d(D', D) \leq r} \ell(f, D')$



Main result

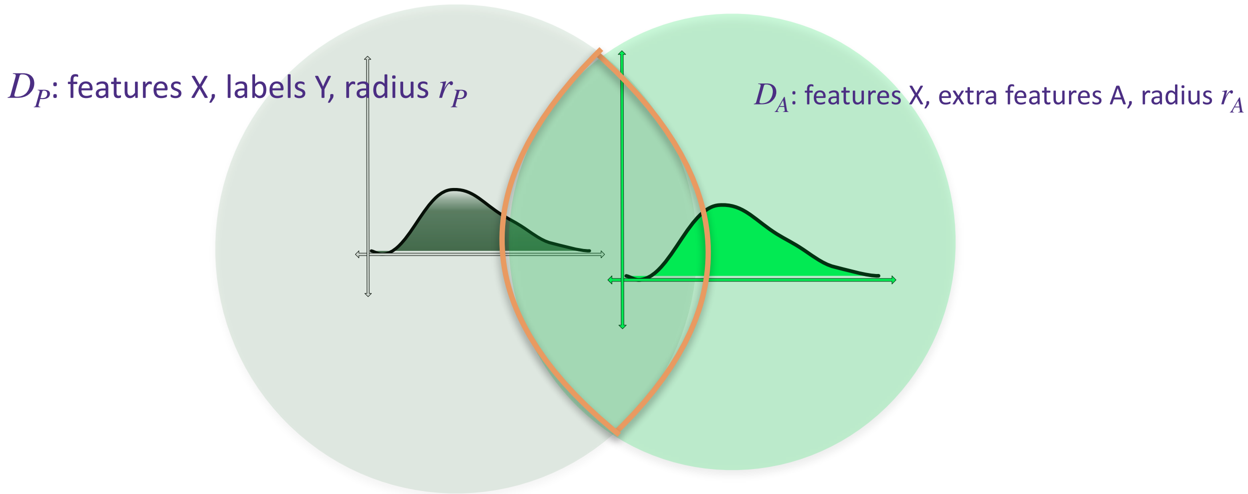


There is an algorithm which finds
linear $\theta : (X \times A) \rightarrow Y$ which performs well
on any distribution $D' \in B_{r_P}(D_P) \cap B_{r_A}(D_A)$

for logistic loss, with additive error O(min(r_P, r_A) * coupling cost)

Can utilize additional features to build a better predictor

Main ideas behind the algorithm

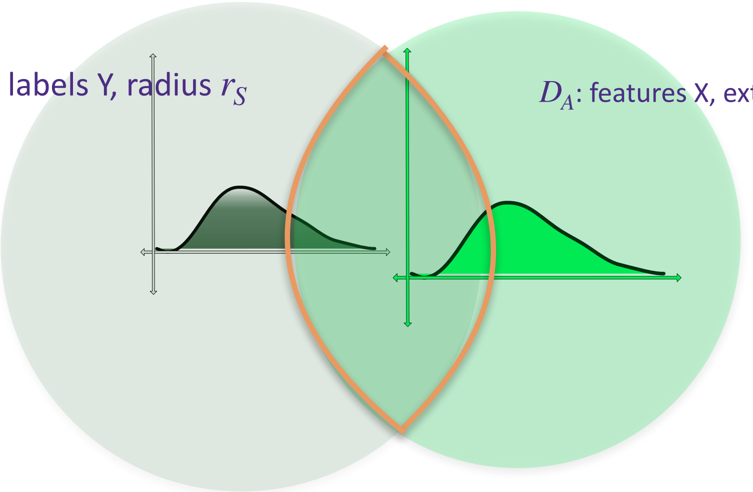


First, find a coupling between D_P, D_A and an unknown D’

Formulate an appropriate dual program

Solve dual using projected gradient descent

A corollary for equitable prediction



Suppose A contains demographic information.
Then, we can compute a model to minimize

$$\theta = \text{argmin}_{\theta} \ell(\theta, D') - \lambda \left(LTPD(\theta, D') \right)$$

where ℓ is log loss, and LTPD is log-probabilistic equalized opportunity

Outline

- Introduction and Motivation
- Our model
  - And a “fairness application”
- An algorithm designed for this problem
  - And a statement about its guarantees
- Experimental results

Experimental results: using all features

Datasets

Breast Cancer

$(|m_1, m_2| = \{(5, \quad 25), (25, \quad 5)\})$

Ionosphere dataset

$(|(m_1, m_2)| = \{(4, \quad 30), (25, \quad 9)\})$

Heart Disease dataset

$(|(m_1, m_2)| = \{(5, \quad 8)\})$

Handwritten Digits dataset (1 vs 8)

$(|(m_1, m_2)| = \{(32, \quad 32)\})$

Uniformly randomly split training data into S_P, S_A with v datapoints in both and filter.

https://archive.ics.uci.edu/ml/datasets/ionosphere

https://archive.ics.uci.edu/ml/datasets/Heart+Disease

https://scikit- learn.org/stable/modules/generated/sklearn.datasets.load_digits.html#sklearn.datasets.load_

gits: This is a copy of the test dataset from https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+

Experimental results: using all features

Compare DJ, our method, with

LR: Logistic regression trained on S_P

RLR: Regularized logistic regression on S_P

LRO: Logistic regression on overlapped v datapoints

RLRO: Regularized logistic regression on overlapped v datapoints

FULL training on unfiltered $S_A \cup S_P$

https://archive.ics.uci.edu/ml/datasets/ionosphere
https://archive.ics.uci.edu/ml/datasets/Heart+Disease
https://scikit- learn.org/stable/modules/generated/sklearn.datasets.load_digits.html#sklearn.datasets.load_
gits: This is a copy of the test dataset from https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+

Experimental results: using all features

DJ, our method

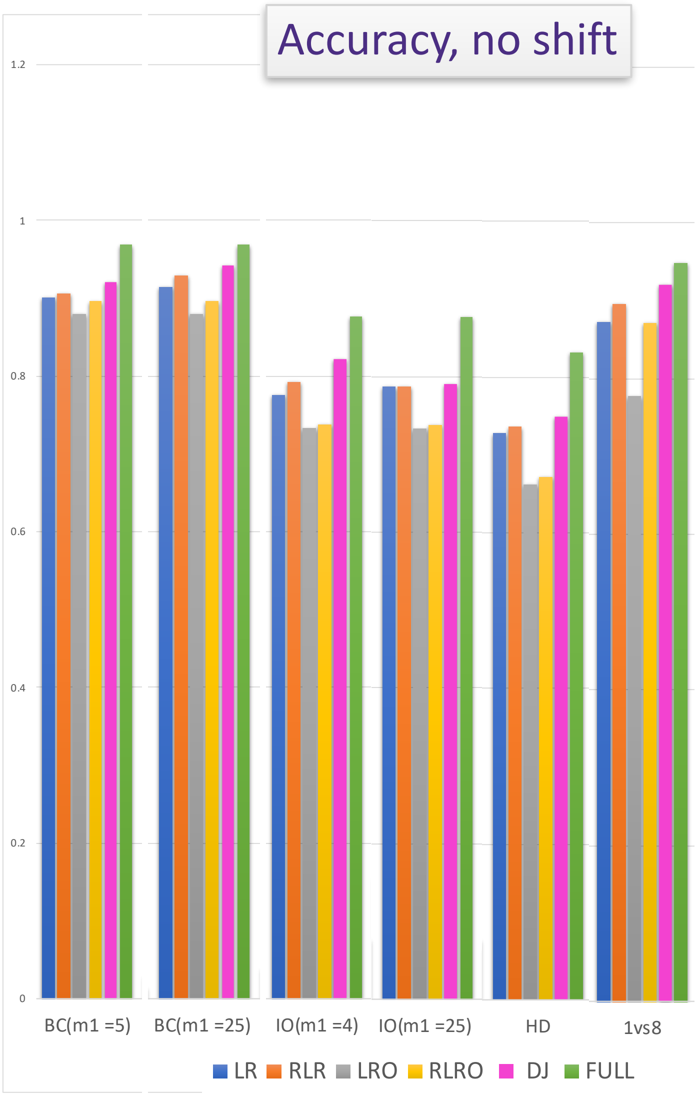
LR: Logistic reg on S_P

RLR: Regularized logistic regression on S_P

LRO: Logistic regression on overlapped v datapoints

RLRO: Regularized logistic regression on overlapped v datapoints

FULL training on unfiltered $S_A \cup S_P$ (labeled)



Remarks

DJ solves a harder problem than necessary here: it's robust to distribution shifts.

Other comparison points are not

Nonetheless, it's comparable to (and slightly better than) the best of the methods which don't have strictly more information.

Experimental results: covariate shift

preliminary evidence on synthetic datasets

with shift in $X, Y | X$

DJ also does quite well compared to LR, RLR, and DRLR on S_P

Still a lot more questions than answers here.

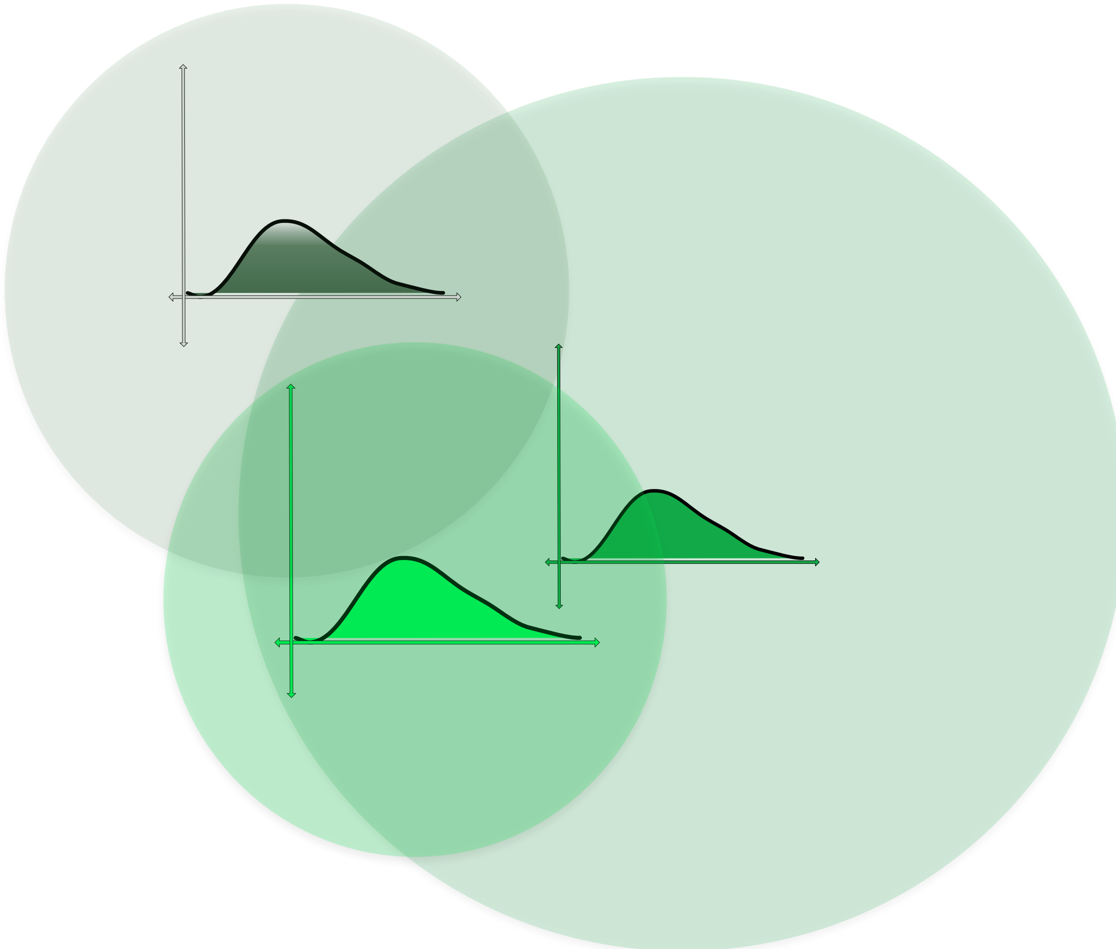
How much is due to a larger sample

versus optimizing over a smaller set?

Open questions

Is our additive loss necessary for efficient computation?

A generally interesting problem: multi-anchor DRO



DRO:

Find $\theta \in \text{argmin}_f \max_{D':d(D',D)\leq r} \ell(f, D')$

Multi-anchor DRO:

Find $\theta \in \text{argmin}_f \max_{D':d(D',D_1)\leq r_1 \wedge d(D',D_2)\leq r2 \wedge \ldots d(D',D_k)\leq r_k} \ell(f, D')$

Circumspection

This method minimizes the max of differences in log-probabilistic equalized opportunity.

Should not be considered an excuse to avoid the hard work of building good datasets

Benefits of using multiple sources

- Increased sample size
- For covariate shift, can learn more about “ground truth”—> unnatural experiment?
- Robustness to other distribution shifts

Part of a larger paradigm

Many of our ML ecosystems have additional structure/resources which may reduce the need to directly trade error minimization for equality of performance across demographics

— Additional active sampling [AAK**M**R’20]
— Models for predicting A from X [ABK**M’**21, AK**M**’20]
— Feature selection [STS**M**V’18, KA**M**’19]
— Correlation between (Y, A) and X