How machines learn

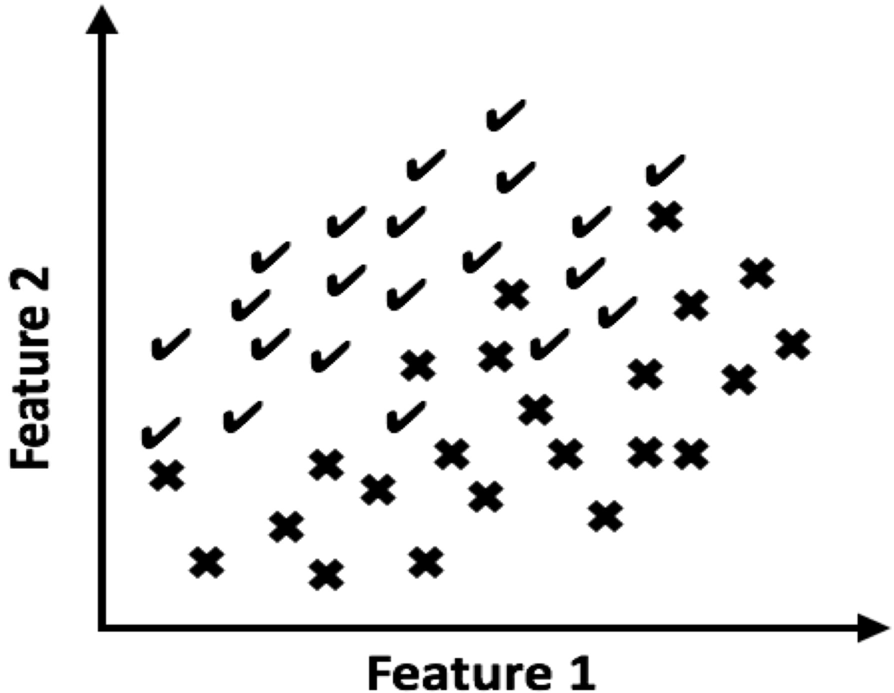
- By training over historical data
- Example task: Predict who will return loan

	F_1	F_2	…	F_m	Z
$User_1$	$x_{1,1}$	$x_{1,2}$	…	$x_{1,m}$	Z_1
$User_2$	$x_{2,1}$			$x_{2,m}$	Z_2
$User_3$	$x_{3,1}$			$x_{3,m}$	Z_3
…	…			…	…
$User_n$	$x_{n,1}$	$x_{n,2}$	…	$x_{n,m}$	Z_n

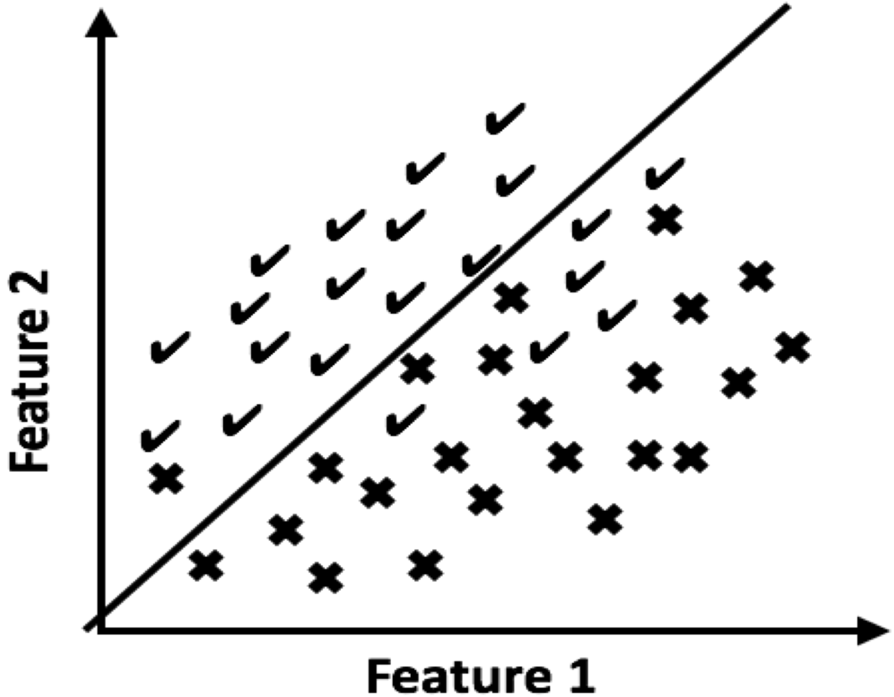
Past Outcomes
Returned
Defaulted
Defaulted
…
Returned

- Learning challenge: Learn a decision boundary (**W**) in the feature space separating the two classes

Predict who will return loans



Predict who will return loans



- Optimal (most accurate / least loss) linear boundary
- But, how do machines find (compute) it?

Learning (computing) the optimal boundary

- Define & optimize a loss (accuracy) function
  - The loss function captures inaccuracy in prediction

$$L(\mathbf{w}) = \sum_{i=1}^{N} (y_i - \mathbf{w}^T \mathbf{x}_i)^2 \qquad L(\mathbf{w}) = \sum_{i=1}^{N} -\log p(y_i | \mathbf{x}_i, \mathbf{w})$$

  - Minimize (optimize) it over all examples in training data

$$minimize\ L(\mathbf{w})$$

- Central challenge in machine learning
  - Finding loss function that capture prediction loss, yet be efficiently optimized
  - Many loss functions used in learning are convex

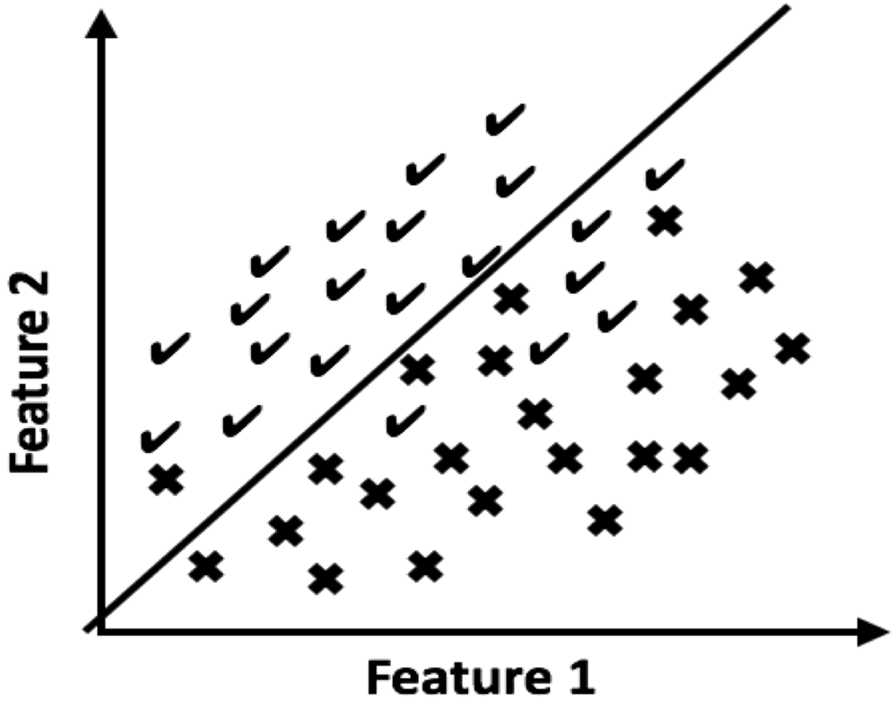
Convex-boundary based loss functions

Squared loss $$\sum_{i=1}^{N}(y_i - d_{\mathbf{w}}(\mathbf{x}_i))^2$$

Logistic loss $$-\sum_{i=1}^{N}\log(1 + e^{-y_i d_{\mathbf{w}}(\mathbf{x}_i)})$$

SVM loss $$||\mathbf{w}||^2 + C\sum_{i=1}^{N}\max(0, 1 - y_i d_{\mathbf{w}}(\mathbf{x}_i))$$

Predict who will return loans



- Optimal (most accurate / least loss) linear boundary
- But, how do machines find (compute) it?
  - The boundary was computed using $\min \sum_{i=1}^{N} (y_i - d_{\mathbf{w}}(\mathbf{x}_i))^2$

How to learn to avoid discrimination

- Specify discrimination measures as constraints on learning
- Optimize for accuracy under those constraints

$$\mathit{minimize}\ L(\mathbf{w})$$

$$\mathit{subject\ to}\quad P(\hat{y} \neq y | z = 0) = P(\hat{y} \neq y | z = 1)$$

- The constraints embed ethics & values when learning
- No free lunch: Additional constraints lower accuracy
  - Tradeoff between performance & ethics (avoid discrimination)

A few observations

- Any discrimination measure could be a constraint

$$\begin{aligned} &\mathit{minimize}\ L(\mathbf{w}) \\ &\mathit{subject\ to} \quad P(\hat{y}|\mathbf{x}, z) = P(\hat{y}|\mathbf{x}) \\ &\qquad\qquad\quad P(\hat{y}=1|z=0) = P(\hat{y}=1|z=1) \\ &\qquad\qquad\quad P(\hat{y} \neq y|z=0) = P(\hat{y} \neq y|z=1) \end{aligned}$$

- Might not need all constraints at the same time
  - E.g., drop disp. impact constraint when no bias in data
  - When avoiding disp. impact / mistreatment, we could achieve higher accuracy without disp. treatment

Key technical challenge

- How to learn efficiently under these constraints?

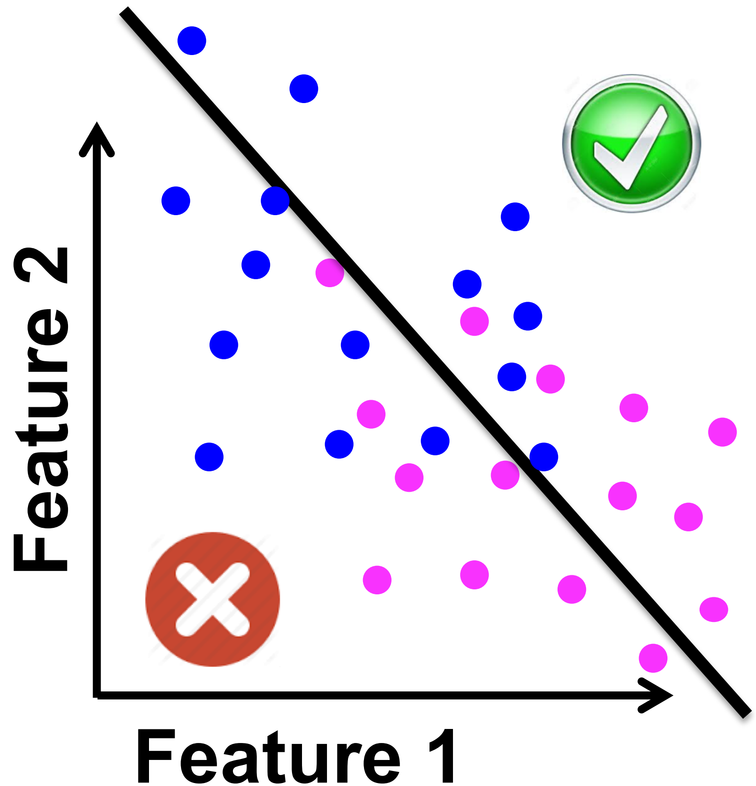
$$\begin{aligned}&\mathit{minimize}\ L(\mathbf{w})\\&\mathit{subject\ to}\quad P(\hat{y}=1|z=0)=P(\hat{y}=1|z=1)\end{aligned}$$

$$\begin{aligned}&\mathit{minimize}\ L(\mathbf{w})\\&\mathit{subject\ to}\quad P(\hat{y}\neq y|z=0)=P(\hat{y}\neq y|z=1)\end{aligned}$$

- Problem: The above formulations are not convex!
  - Can't learn them efficiently

- Need to find a better way to specify the constraints
  - So that loss function under constraints remains convex

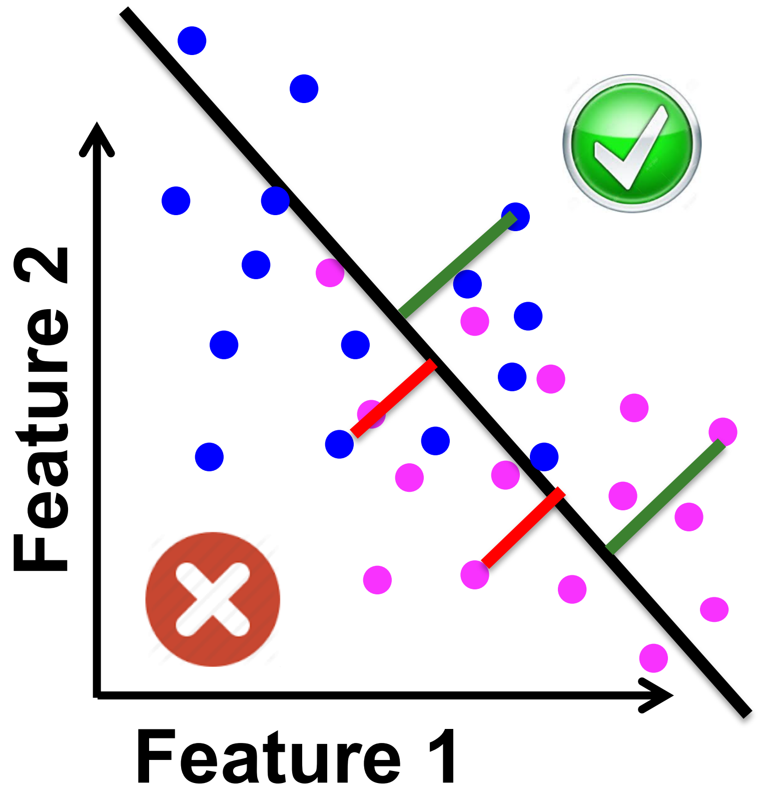
Disparate impact constraints: Intuition



$$P(\hat{y} = 1 | z = 0) = P(\hat{y} = 1 | z = 1)$$

Limit the differences in the acceptance (or rejection) ratios across members of different sensitive groups

Disparate impact constraints: Intuition



A proxy measure for $P(\hat{y} = 1 | z = 0) = P(\hat{y} = 1 | z = 1)$

Limit the differences in the average strength of acceptance and rejection across members of different sensitive groups

Specifying disparate impact constraints

- Instead of requiring: $P(\hat{y}=1|z=0) = P(\hat{y}=1|z=1)$

- Bound covariance between items' sensitive feature values and their signed distance from classifier's decision boundary to less than a threshold

$$\boxed{\left| \frac{1}{N} \sum_{i=1}^{N} (\mathbf{z}_i - \bar{\mathbf{z}})\, \mathbf{w}^{\mathbf{T}} \mathbf{x}_i \right| \leq \mathbf{c}}$$

Learning classifiers w/o disparate impact

- Previous formulation: Non-convex, hard-to-learn

$$\begin{aligned} &\textit{minimize} \; L(\mathbf{w}) \\ &\textit{subject to} \quad P(\hat{y}=1|z=0) = P(\hat{y}=1|z=1) \end{aligned}$$

- New formulation: Convex, easy-to-learn

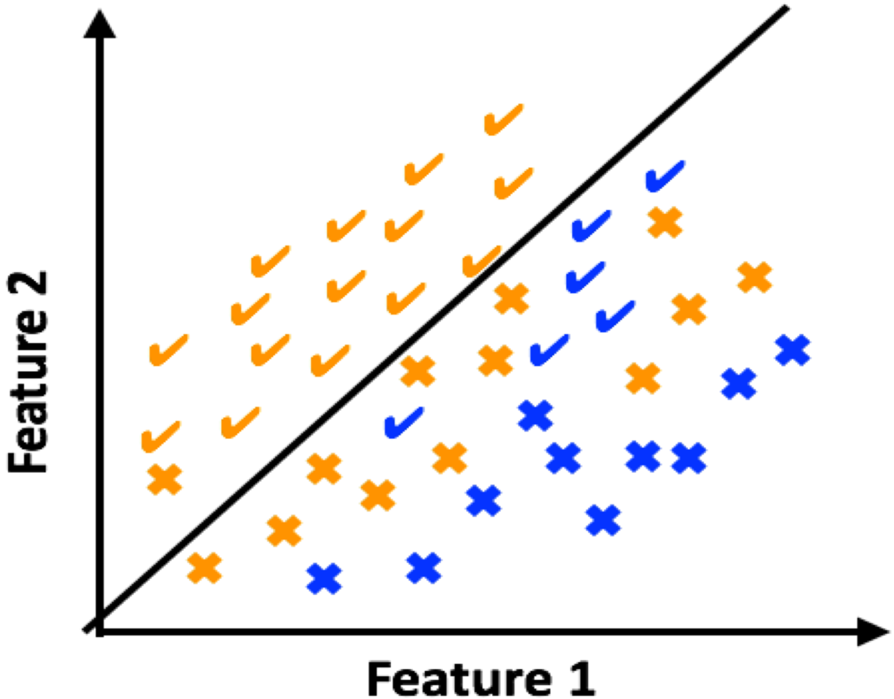
$$\begin{aligned} &\textit{minimize} \quad L(\mathbf{w}) \\ &\textit{subject to} \quad \frac{1}{N}\sum_{i=1}^{N} (\mathbf{z}_i - \bar{\mathbf{z}})\, \mathbf{w}^T \mathbf{x}_i \leq \mathbf{c} \\ &\qquad\qquad\quad \frac{1}{N}\sum_{i=1}^{N} (\mathbf{z}_i - \bar{\mathbf{z}})\, \mathbf{w}^T \mathbf{x}_i \geq -\mathbf{c} \end{aligned}$$

A few observations

- Our formulation can be applied to any convex-margin (loss functions) based classifiers
  - hinge-loss, logistic loss, linear and non-linear SVM

- Can easily change our formulation to optimize for fairness under accuracy constraints
  - Useful in practice, when you want to be fair but have business necessity to meet a certain accuracy threshold

Specifying mistreatment constraints



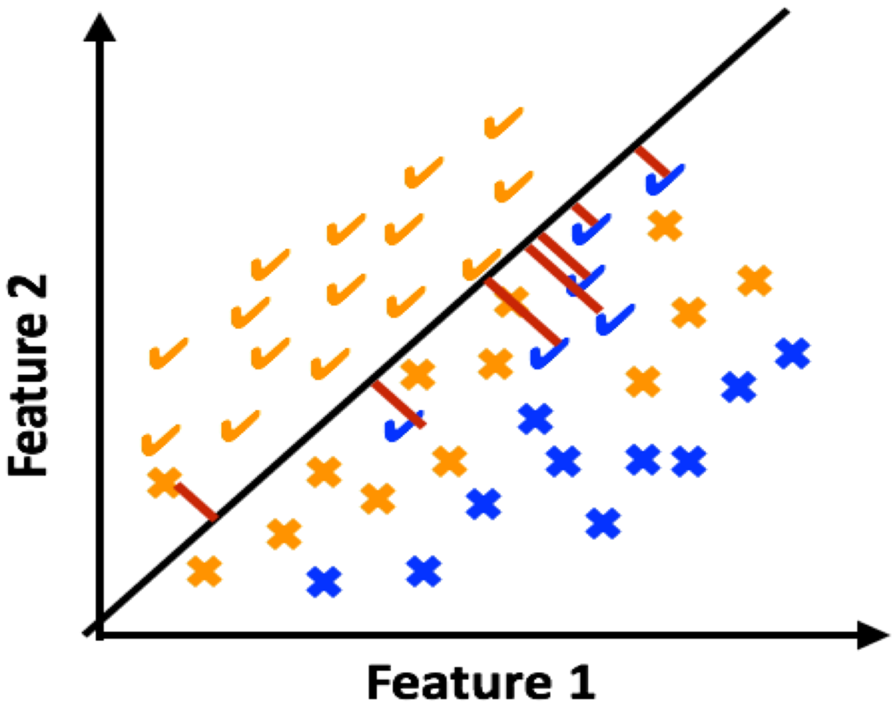
Idea: Avg. misclassification distance from boundary for both groups should be the same

Specifying mistreatment constraints

$$\underbrace{\min(0, y_i d_{\mathbf{w}}(\mathbf{x}_i))}_{}$$

Concave

($d_w(x)$ is affine)



Idea: Avg. misclassification distance from boundary for both groups should be the same

Rewriting mistreatment constraints

$$\min \quad \sum_{i=1}^{N} (y_i - d_{\mathbf{w}}(\mathbf{x}_i))^2$$

$$\text{s.t.} \qquad P(y_{true} \neq y_{pred} \mid ♀) = P(y_{true} \neq y_{pred} \mid ♂)$$

Rewriting mistreatment constraints

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{N} (y_i - d_{\mathbf{w}}(\mathbf{x}_i))^2 \\
\text{s.t.} \quad & -\epsilon \leq \underbrace{\frac{1}{|♂|} \sum_{♂} \min(0, y_i d_{\mathbf{w}}(\mathbf{x}_i))}_{\text{Concave}} - \underbrace{\frac{1}{|♀|} \sum_{♀} \min(0, y_i d_{\mathbf{w}}(\mathbf{x}_i))}_{\text{Concave}} \leq \epsilon
\end{aligned}
$$

$P(y_{true} \neq y_{pred} \mid ♂)$ $P(y_{true} \neq y_{pred} \mid ♀)$

- Can be solved efficiently
  - Using Disciplined Convex-Concave Programming
    - DCCP *[Shen, Diamond, Gu, Boyd, 2016]*

Learning classifiers w/o disparate mistreatment

- New formulation: Convex-concave, can learn efficiently using convex-concave programming

$$
\begin{array}{ll}
\textit{minimize} & L(\mathbf{w}) \\
\textit{subject to} & \frac{-N_1}{N} \sum_{i=1}^{N_0} g_{\mathbf{w}}(y_i, \mathbf{x}_i) + \frac{N_0}{N} \sum_{i=1}^{N_1} g_{\mathbf{w}}(y_i, \mathbf{x}_i) \leq \mathbf{c} \\
 & \frac{-N_1}{N} \sum_{i=1}^{N_0} g_{\mathbf{w}}(y_i, \mathbf{x}_i) + \frac{N_0}{N} \sum_{i=1}^{N_1} g_{\mathbf{w}}(y_i, \mathbf{x}_i) \geq -\mathbf{c}
\end{array}
$$

All misclassifications $g_{\mathbf{w}}(y, \mathbf{x}) = min(0, y d_{\mathbf{w}}(\mathbf{x})),$

False negatives $g_{\mathbf{w}}(y, \mathbf{x}) = min\left(0, \frac{1+y}{2} y d_{\mathbf{w}}(\mathbf{x})\right)$, or

False positives $g_{\mathbf{w}}(y, \mathbf{x}) = min\left(0, \frac{1-y}{2} y d_{\mathbf{w}}(\mathbf{x})\right),$

Evaluation: Recidivism risk estimates

- **Recidivism:** To re-offend within a certain time

- COMPAS risk assessment tool
  - Assign recidivism risk score to a criminal defendant
  - Score used to advise judges' decision

- ProPublica gathered COMPAS assessments
  - Broward Country, FL for 2013-14
  - Features: arrest charge, #prior offenses, age,...
  - Class label: 2-year recidivism

Key evaluation questions

- Do traditional classifiers suffer disparate mistreatment?
- Can our approach help avoid disparate mistreatment?

Disparity in mistreatment

- Trained logistic regression for recidivism prediction

Race	FPR	FNR
Black	34%	32%
White	15%	55%

- **False positive:** Non-recidivating person wrongly classified as recidivating

- **False negative:** Recidivating person wrongly classified as non-recidivating

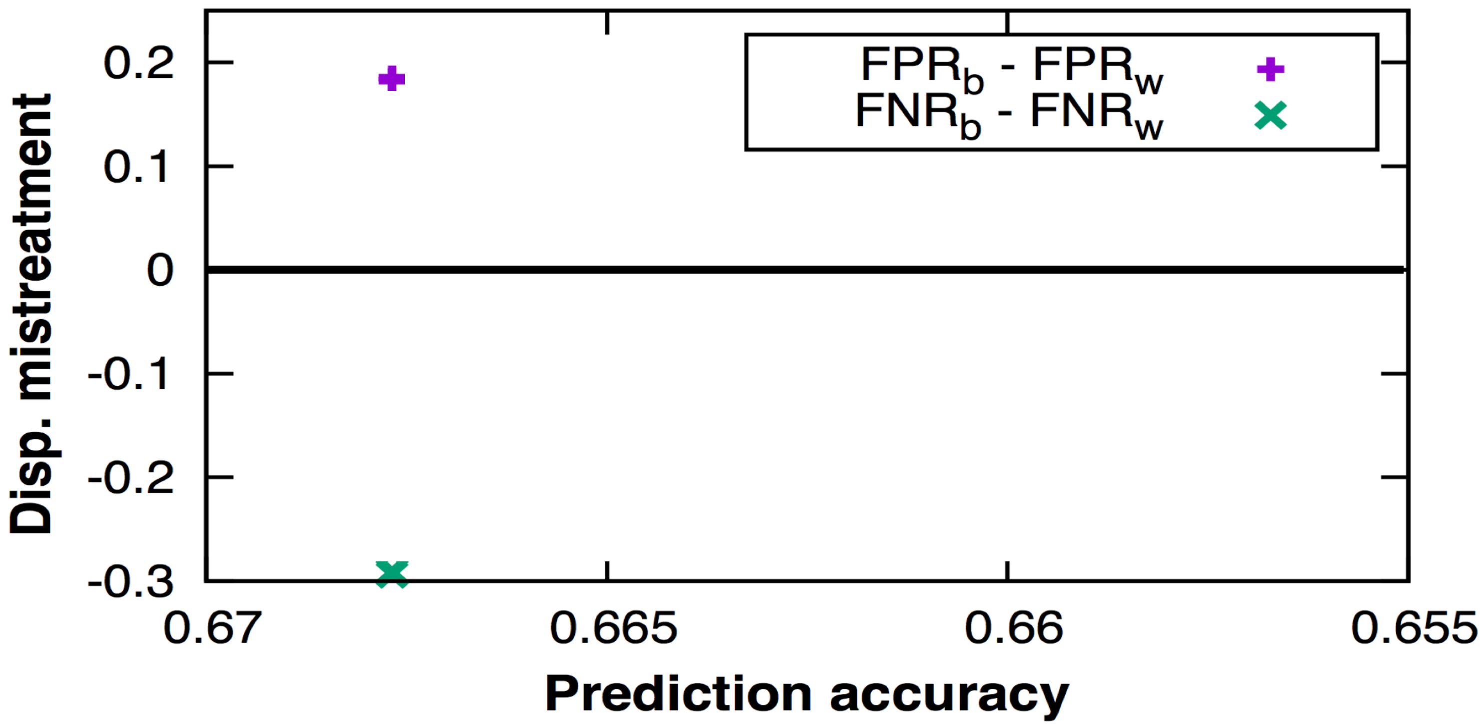
Key evaluation questions

- Do traditional classifiers suffer disparate mistreatment?
  - Yes! Considerable disparity in both FPR and FNR

- Can our approach help avoid disparate mistreatment?

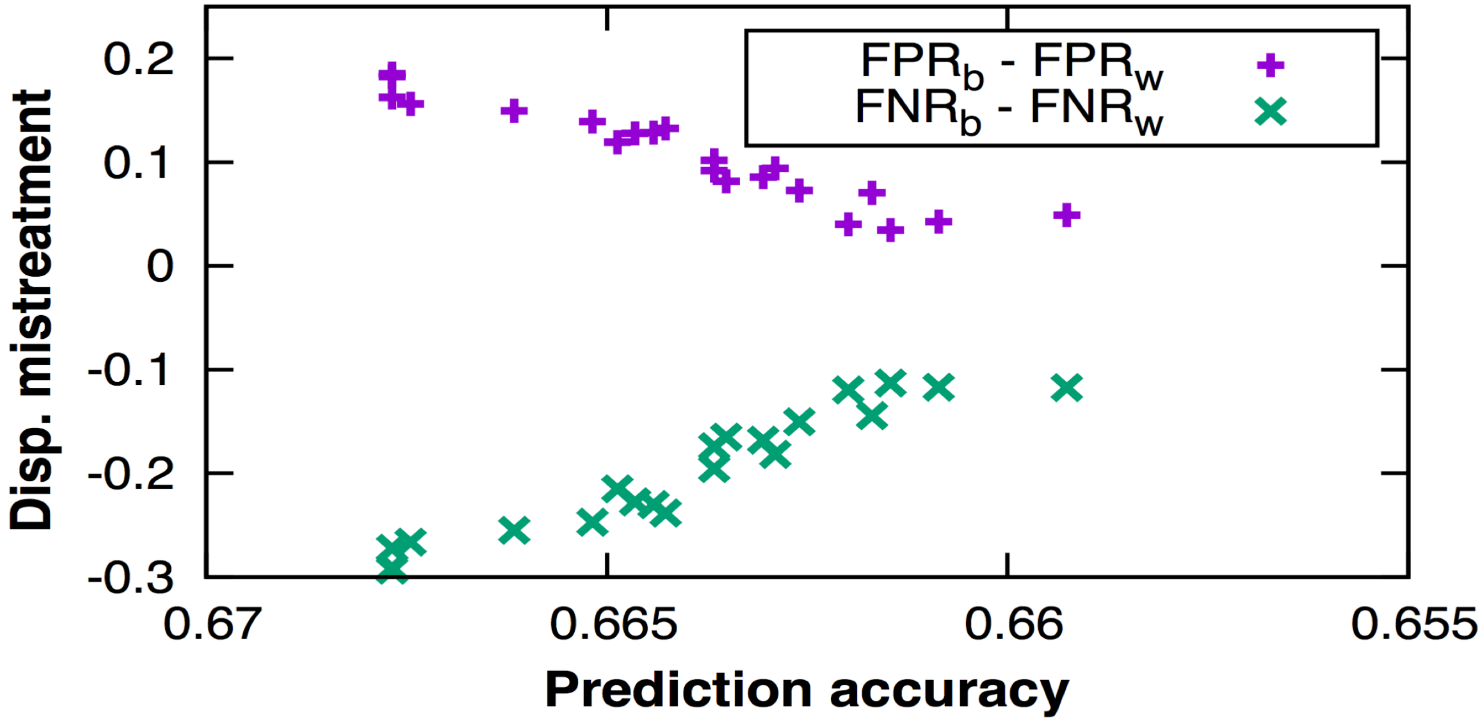
Removing disparate mistreatment

- Traditional classifiers without constraints



Removing disparate mistreatment

- Introducing our FPR and FNR Constraints



Key evaluation questions

- Do traditional classifiers suffer disparate mistreatment?
  - Yes! Considerable disparity in both FPR and FNR
- Can our approach help avoid disparate mistreatment?
  - Yes! For a small loss in accuracy

From Parity to Preference-based Discrimination Measures *[NIPS '17]*

Measures envy-free discrimination

- Preferred treatment allows group-conditional boundaries
- Yet, ensure they are envy-free
  - No lowering the bar to affirmatively select certain user groups
- Can be defined at individual or group-level
- More formally:

$$P(\hat{y} = 1 \mid X_{z=0}, W_{z=0}) \geq P(\hat{y} = 1 \mid X_{z=0}, W_{z=1})$$

$$P(\hat{y} = 1 \mid X_{z=1}, W_{z=1}) \geq P(\hat{y} = 1 \mid X_{z=1}, W_{z=0})$$

Learning preferred treatment classifiers

$$\textit{Minimize} \;\; L_{z=0}(W_{z=0}) + L_{z=1}(W_{z=1})$$

Subject to

$$P(\hat{y} = 1 \mid X_{z=0}, W_{z=0}) \geq P(\hat{y} = 1 \mid X_{z=0}, W_{z=1})$$
$$P(\hat{y} = 1 \mid X_{z=1}, W_{z=1}) \geq P(\hat{y} = 1 \mid X_{z=1}, W_{z=0})$$

- Preferred treatment subsumes parity treatment
  - Every parity treatment classifier offers preferred treatment
- Preferred treatment constraint is weaker than parity
  - Suffers lower cost of fairness

Measures bargained discrimination

- Preferred impact inspired by bargaining solutions in game-theory

- Disagreement (default) solution is parity!
  - Both groups try to avoid tragedy of parity

- Selects pareto-optimal boundaries over group accuracies

- More formally:

$$P(\hat{y} \neq y \mid X_{z=0}, W) \geq P(\hat{y} \neq y \mid X_{z=0}, W_{parity})$$
$$P(\hat{y} \neq y \mid X_{z=1}, W) \geq P(\hat{y} \neq y \mid X_{z=1}, W_{parity})$$