Discrimination in Decision Making: Humans vs. Machines

Muhammad Bilal Zafar, Isabel Valera,
Manuel Gomez-Rodriguez, Krishna P. Gummadi
Max Planck Institute for Software Systems

Machine decision making

- Refers to data-driven algorithmic decision making
  - By learning over data about past decisions

- To assist or replace human decision making

- Increasingly being used in several domains
  - Recruiting: Screening job applications
  - Banking: Credit ratings / loan approvals
  - Judiciary: Recidivism risk assessments
  - Journalism: News recommender systems

The concept of discrimination

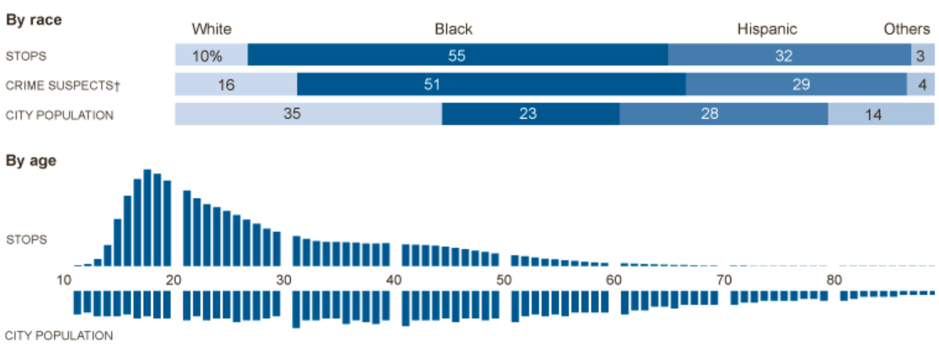
- Well-studied in social sciences
  - Political science
  - Moral philosophy
  - Economics
  - Law
    - Majority of countries have anti-discrimination laws
    - Discrimination recognized in several international human rights laws

- But, less-studied from a computational perspective

Why, a computational perspective?

1. Datamining is increasingly being used to detect discrimination in human decision making

- Examples: NYPD stop and frisk, Airbnb rentals



Why, a computational perspective?

2. Learning to avoid discrimination in data-driven (algorithmic) decision making

- Aren't algorithmic decisions inherently objective?
  - In contrast to subjective human decisions

- Doesn't that make them fair & non-discriminatory?

- Objective decisions can be unfair & discriminatory!

Why, a computational perspective?

- Learning to avoid discrimination in data-driven (algorithmic) decision making

  - *A priori* discrimination in biased training data
    - Algorithms will objectively learn the biases

  - Learning objectives target decision accuracy over all users
    - Ignoring outcome disparity for different sub-groups of users

Websites Vary Prices, Deals Based on Users' Information ...
online.wsj.com/.../SB100014241278873237772045... The Wall Street Journal
A Wall Street Journal investigation found that the **Staples** Inc. website displays different **prices** to people after estimating their **locations**. More than that, **Staples** ...

Our agenda: Two high-level questions

1. How to detect discrimination in decision making?
   - Independently of who makes the decisions
     - Humans or machines

2. How to avoid discrimination when learning?
   - Can we make algorithmic decisions more fair?
   - If so, algorithms could eliminate biases in human decisions
     - Controlling algorithms may be easier than retraining people

This talk

1. ~~How to detect discrimination in decision making?~~
   - ~~Independently of who makes the decisions~~
     - ~~Humans or machines~~

2. How to avoid discrimination when learning?
   - Can we make algorithmic decisions more fair?
   - If so, algorithms could eliminate biases in human decisions
     - Controlling algorithms may be easier than retraining people

The concept of discrimination

- A first approximate normative / moralized definition:

wrongfully impose a **relative disadvantage** on persons **based on** their membership in some **salient social group** e.g., race or gender

The devil is in the details

- What constitutes a salient social group?
  - A question for political and social scientists
- What constitutes relative disadvantage?
  - A question for economists and lawyers
- What constitutes a wrongful decision?
  - A question for moral-philosophers
- What constitutes **based on?**
  - A question for **computer scientists**

A computational perspective of decision making

- Binary classification based on user data (attributes)

	A_1	A_2	…	A_m
$User_1$	$x_{1,1}$	$x_{1,2}$	…	$x_{1,m}$
$User_2$	$x_{2,1}$			$x_{2,m}$
$User_3$	$x_{3,1}$			$x_{3,m}$
…	…			…
$User_n$	$x_{n,1}$	$x_{n,2}$	…	$x_{n,m}$

Decision
Accept
Reject
Reject
…
Accept

A computational perspective of decision making

- Binary classification based on user data (attributes)
  - Some of which are sensitive and others non-sensitive

	SA_1	NSA_2	…	NSA_m
$User_1$	$x_{1,1}$	$x_{1,2}$	…	$x_{1,m}$
$User_2$	$x_{2,1}$			$x_{2,m}$
$User_3$	$x_{3,1}$			$x_{3,m}$
…	…			…
$User_n$	$x_{n,1}$	$x_{n,2}$	…	$x_{n,m}$

Decision
Accept
Reject
Reject
…
Accept

A computational perspective of discrimination

- Decisions should not be based on sensitive attributes

	SA_1	NSA_2	…	NSA_m
$User_1$	$x_{1,1}$	$x_{1,2}$	…	$x_{1,m}$
$User_2$	$x_{2,1}$			$x_{2,m}$
$User_3$	$x_{3,1}$			$x_{3,m}$
…	…			…
$User_n$	$x_{n,1}$	$x_{n,2}$	…	$x_{n,m}$

Decision
Accept
Reject
Reject
…
Accept

What constitutes “based on”?

- Computationally, based on is a pattern of dependence between decision outputs & sensitive input attributes

- Examples: Three discrimination patterns

1. Disparate treatment $P(\hat{y}|\mathbf{x}, z) = P(\hat{y}|\mathbf{x})$
2. Disparate impact $P(\hat{y} = 1|z = 0) = P(\hat{y} = 1|z = 1)$
3. Disparate mistreatment $P(\hat{y} \neq y|z = 0) = P(\hat{y} \neq y|z = 1)$

A computational study of discrimination

- Define / identify interesting patterns of dependence

- Determine whether a pattern constitutes discrimination
  - Depends on context and is not a computational question

- Design tests to detect discriminatory patterns
  - By auditing human or algorithmic decision making

- Design learning methods to avoid discriminatory patterns

Learning to avoid discrimination

- Learning involves defining & optimizing a loss function
  - E.g., Hinge loss function for max. margin classification

$$\text{minimize} \sum_{i=1}^{N} max(0, 1 - y_i \mathbf{w}^{\mathbf{T}} \mathbf{x}_i)$$

  - Frequently, loss functions are defined to be convex
  - Allows for efficient optimization & learning

Learning to avoid discrimination

- Learning involves defining & optimizing a loss function
- Our strategy: Formulate discrimination patterns as constraints on learning process
- Optimize for accuracy under those constraints
  - No free lunch: Trade-off accuracy to avoid discrimination
- Key challenge: How to specify these constraints?
  - So that learning is efficient even under the constraints
    - i.e., loss function under constraints remains convex

Discrimination Pattern 1: Disparate Treatment

Pattern of disparate treatment

- Treat users with similar non-sensitive attributes, but different sensitive attributes similarly

$$P(\hat{y}|\mathbf{x}, z) = P(\hat{y}|\mathbf{x})$$

- Matches our intuitive notion of discrimination

Detecting disparate treatment

- Active situational testing
  - Check if changing a sensitive feature changes decision
    - Used for detecting implicit bias against women when hiring

- Passive k-NN (nearest neighbor) testing
  - Check if inputs with similar non-sensitive features received different decisions
    - Used for detecting racial discrimination in Airbnb rentals

Learning to avoid disparate treatment

- Remember our strategy?
- Express discrimination patterns as constraints on learning process
- Optimize for accuracy under those constraints

Learning hinge loss classifiers

$$\text{minimize} \sum_{i=1}^{N} max(0, 1 - y_i \mathbf{w}^{\mathbf{T}} \mathbf{x}_i)$$

Learning hinge loss classifiers without disparate treatment

$$\text{minimize} \sum_{i=1}^{N} max(0, 1 - y_i \mathbf{w}^{\mathbf{T}} \mathbf{x}_i)$$

subject to $$P(\hat{y}|\mathbf{x}, z) = P(\hat{y}|\mathbf{x})$$

- Train classifiers only on non-sensitive features
  - Constrain learning to not use sensitive features
  - Such training would pass situational testing

- Sufficient to handle biases in training data?

Training introduces indirect discrimination

	$\cancel{SA_1}$	NSA_2	…	NSA_m
$User_1$	$\cancel{x_{1,1}}$	$x_{1,2}$	…	$x_{1,m}$
$User_2$	$\cancel{x_{2,1}}$			$x_{2,m}$
$User_3$	$\cancel{x_{3,1}}$			$x_{3,m}$
…	~~…~~			…
$User_n$	$\cancel{x_{n,1}}$	$x_{n,2}$	…	$x_{n,m}$

Decision
Accept
Reject
Reject
…
Accept

- Sensitive features are stripped off in training data

Training introduces indirect discrimination

	$\cancel{SA_1}$	NSA_2	…	NSA_m
$User_1$	$\cancel{x_{1,1}}$	$x_{1,2}$	…	$x_{1,m}$
$User_2$	$\cancel{x_{2,1}}$			$x_{2,m}$
$User_3$	$\cancel{x_{3,1}}$			$x_{3,m}$
…	$\cancel{\ldots}$			…
$User_n$	$\cancel{x_{n,1}}$	$x_{n,2}$	…	$x_{n,m}$

Decision
Accept
Reject
Reject
…
Accept

- Lacking SA, NSAs correlated with sensitive features will be given more or less weights
  - Learning algorithm tries to compensate for lost data!

Training introduces indirect discrimination

	~~SA_1~~	NSA_2	…	NSA_m
$User_1$	~~$x_{1,1}$~~	$x_{1,2}$	…	$x_{1,m}$
$User_2$	~~$x_{2,1}$~~			$x_{2,m}$
$User_3$	~~$x_{3,1}$~~			$x_{3,m}$
…	~~…~~			…
$User_n$	~~$x_{n,1}$~~	$x_{n,2}$	…	$x_{n,m}$

Decision
Accept
Reject
Reject
…
Accept

- Exception: When sensitive & non-sensitive features are totally uncorrelated
  - Unlikely with big data with lots of features
  - Use of scalable learning algorithms

Indirect discrimination

- Also, observed in human decision making

- Indirectly discriminate against specific user groups using their correlated non-sensitive attributes
  - E.g., voter-id laws being passed in US states

- Notoriously hard to detect indirect discrimination
  - In decision making scenarios without ground truth

Doctrine of Disparate Impact

- A US law applied in employment & housing practices:

"practices..considered discriminatory and illegal if they have a disproportionate adverse impact on persons along the lines of a protected trait"

"A facially neutral employment practice is one that does not appear to be discriminatory on its face; rather it is one that is discriminatory in its application or effect"

Detecting disparate impact

- Proportionality tests over decision outcomes
  - E.g., in 70's and 80's, some US courts applied the 80% rule for employment practices
    - If 50% (P1%) of male applicants get selected at least 40% (P2%) of female applicants must be selected

- UK uses P1 – P2; EU uses (1-P1) / (1-P2)

- Different proportions may be considered fair in different domains

A controversial detection policy

- Critics: There exist scenarios where disproportional outcomes are justifiable
- Supporters: Provision for business necessity exists
- Law is necessary to detect indirect discrimination!

Discrimination Pattern 2: Disparate Impact

Disparate impact

- Users belonging to different sensitive attribute groups should have equal chance of getting selected

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

- Justification comes from desire to avoid indirect discrimination

Learning to avoid disparate impact

- Remember our strategy?

- Express discrimination patterns as constraints on learning process

- Optimize for accuracy under those constraints

Learning hinge loss classifiers

$$\text{minimize} \sum_{i=1}^{N} max(0, 1 - y_i \mathbf{w}^{\mathbf{T}} \mathbf{x}_i)$$

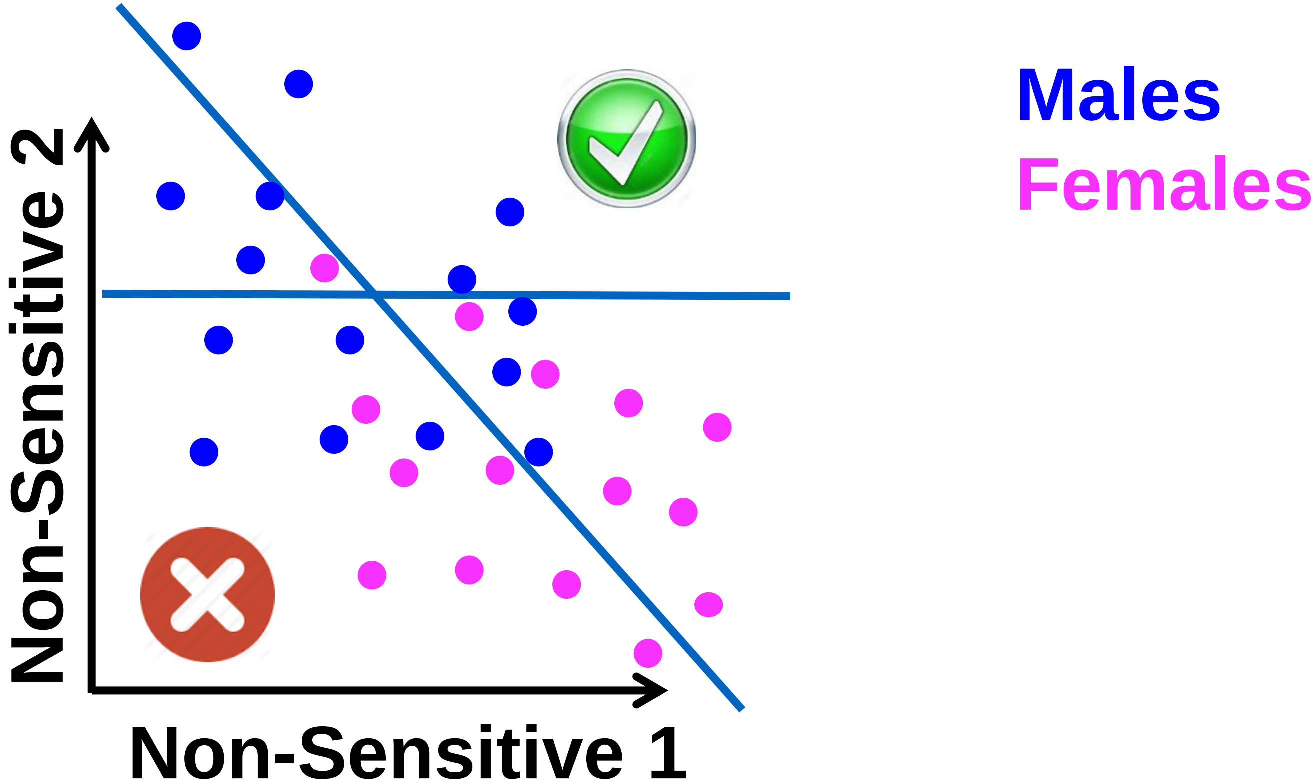
Learning hinge loss classifiers without disparate impact

$$\text{minimize} \sum_{i=1}^{N} max(0, 1 - y_i \mathbf{w}^{\mathbf{T}} \mathbf{x}_i)$$

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

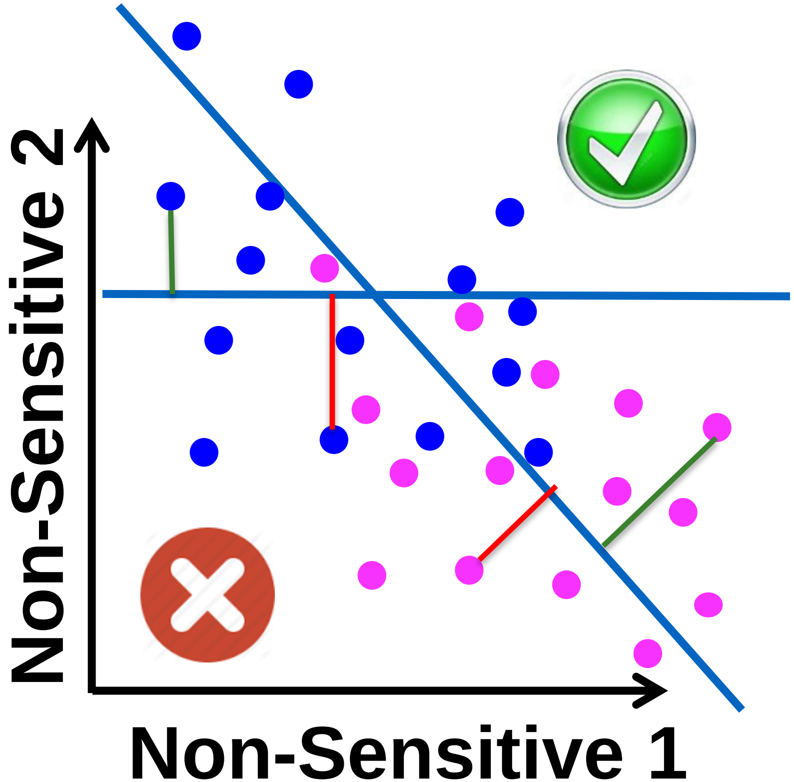
- Key challenge: How to specify these constraints?
  - So that learning is efficient even under the constraints

Disparate impact constraints: Intuition



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

Disparate impact constraints: Intuition



Males
Females

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

Specifying disparate impact constraints

- 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 hinge loss classifiers

$$\text{minimize} \sum_{i=1}^{N} max(0, 1 - y_i \mathbf{w}^{\mathbf{T}} \mathbf{x}_i)$$

Learning hinge loss classifiers without disparate impact

$$\text{minimize} \sum_{i=1}^{N} max(0, 1 - y_i \mathbf{w}^{\mathbf{T}} \mathbf{x}_i)$$

$$\text{subject to} \frac{1}{N} \sum_{i=1}^{N} (\mathbf{z}_i - \bar{\mathbf{z}}) \, \mathbf{w}^{\mathbf{T}} \mathbf{x}_i \leq \mathbf{c},$$

$$\frac{1}{N} \sum_{i=1}^{N} (\mathbf{z}_i - \bar{\mathbf{z}}) \, \mathbf{w}^{\mathbf{T}} \mathbf{x}_i \geq -\mathbf{c}.$$

Learning hinge loss classifiers without disparate impact

$$\text{minimize} \sum_{i=1}^{N} max(0, 1 - y_i \mathbf{w}^{\mathbf{T}} \mathbf{x}_i)$$

$$\text{subject to} \frac{1}{N} \sum_{i=1}^{N} (\mathbf{z}_i - \bar{\mathbf{z}}) \, \mathbf{w}^{\mathbf{T}} \mathbf{x}_i \leq \mathbf{c},$$

$$\frac{1}{N} \sum_{i=1}^{N} (\mathbf{z}_i - \bar{\mathbf{z}}) \, \mathbf{w}^{\mathbf{T}} \mathbf{x}_i \geq -\mathbf{c}.$$

Possible to solve this convex optimization efficiently!

Learning hinge loss classifiers without disparate impact

$$\text{minimize} \sum_{i=1}^{N} max(0, 1 - y_i \mathbf{w}^{\mathbf{T}} \mathbf{x}_i)$$

$$\text{subject to} \frac{1}{N} \sum_{i=1}^{N} (\mathbf{z}_i - \bar{\mathbf{z}}) \, \mathbf{w}^{\mathbf{T}} \mathbf{x}_i \leq \mathbf{c},$$

$$\frac{1}{N} \sum_{i=1}^{N} (\mathbf{z}_i - \bar{\mathbf{z}}) \, \mathbf{w}^{\mathbf{T}} \mathbf{x}_i \geq -\mathbf{c}.$$

Possible to solve this convex optimization efficiently!

Can be included in other decision-boundary classifiers

Learning logistic regression without disparate impact

$$p(y_i = 1|\mathbf{x}_i) = \frac{1}{1 + e^{-b_0 + \sum_j b_j x_{ij}}}$$

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{N} \log p(y_i|\mathbf{x}_i) \\ \text{subject to} & \frac{1}{N} \sum_{i=1}^{N} (\mathbf{z}_i - \bar{\mathbf{z}})\, \mathbf{b}^T [-1 \ \mathbf{x}_i] \leq \mathbf{c}, \\ & \frac{1}{N} \sum_{i=1}^{N} (\mathbf{z}_i - \bar{\mathbf{z}})\, \mathbf{b}^T [-1 \ \mathbf{x}_i] \geq -\mathbf{c} \end{array}$$

Possible to solve this convex optimization efficiently!

Evaluating discrimination constraints

- Tested it over UCI census income dataset
  - 45K users
  - 14 features
  - Non-sensitive: Education-level, # hours of work per week
  - Sensitive: Gender and race

- Classification task: Predict whether a user earns >50K (positive) and <50K (negative) per year

Income disparity for genders in dataset

Gender	<50K	>50K
Female	89%	11%
Male	69%	31%

0.35

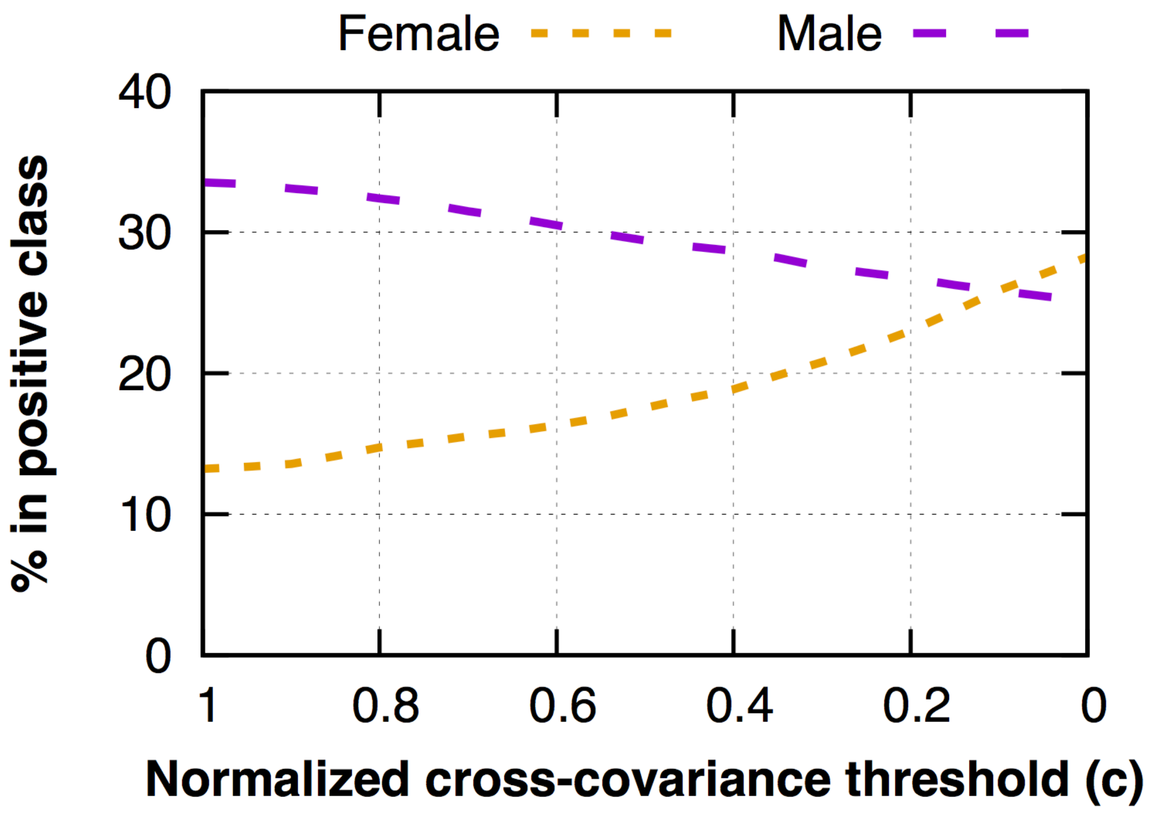
Logistic regression (with constraints)

- Introduce cross-covariance constraints

$$\left| \frac{1}{N} \sum_{i=1}^{N} \left(\mathbf{z}_i - \bar{\mathbf{z}}\right) \mathbf{b}^T [-1 \;\; \mathbf{x}_i] \right| \leq \mathbf{c}$$

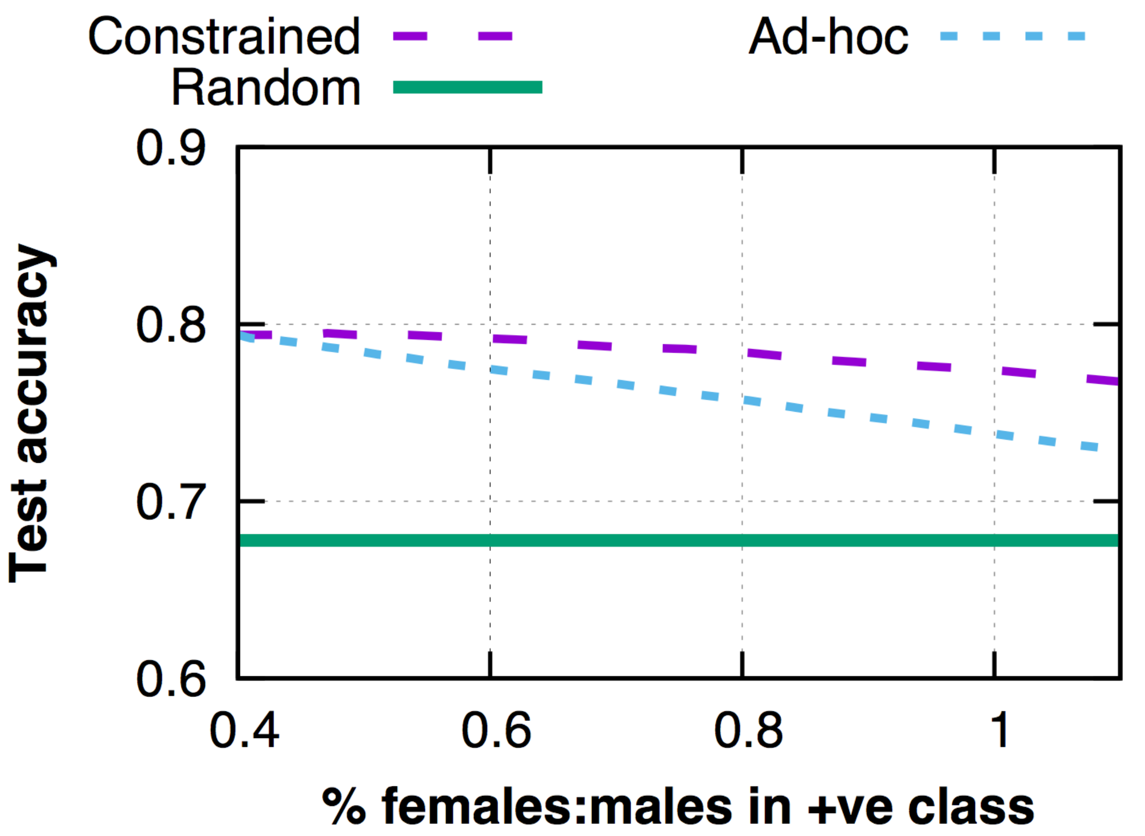
- Hypotheses to test / evaluate:
  - By varying the fairness threshold (c), we can alter the proportions of selected people in sensitive categories
  - Hopefully, without taking a huge hit in terms of accuracy

Reducing disparity with constraints



Tightening threshold reduces disparity in income estimates between men and women

Fairness vs. accuracy tradeoff



Loss in accuracy not too high!

Summary & Future Work

Summary: Discrimination through computational lens

- Define interesting patterns of dependence
  - Defined two patterns – disparate treatment & impact
  - Argued they correspond to direct and indirect discrimination
- Design tests to detect the discriminatory patterns
  - Such tests already exist: situational & proportionality tests
- Learning mechanisms to avoid discriminatory patterns
  - Proposed efficient learning methods for the above patterns

Ongoing work

- Discrimination beyond disparate treatment & impact

- Disparate **mis**treatment: Errors in classification for different groups of users should be same

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

- A better notion when training data is unbiased

- Defined constraints to avoid disparate mistreatment
  - Efficient solutions with convex-concave programming

Future work: Beyond binary classifiers

- How to learn
  - Non-discriminatory multi-class classification
  - Non-discriminatory regression
  - Non-discriminatory set selection
  - Non-discriminatory ranking

Zooming out:
The bigger picture

Fairness beyond discrimination

- Discrimination is one specific type of unfairness
- There may be other forms of "fairness patterns" desirable in decision-making scenarios
  - E.g., when performing college admissions, you might desire that an applicant's chance of getting admitted does not decrease with getting higher scores in specific exams
  - I.e., we can define a pattern of monotonic impact
- Need new ways to constrain learning algorithms!

Beyond fairness: FATE of Machine Decision Making

- Fairness: The focus of this talk
- Accountability: Assigning responsibility for decisions
  - Helps correct and improve decision making
- Transparency: Tracking the decision making process
  - Helps build trust in decision making
- Explainability: Interpreting (making sense of) decisions
  - Helps understand decision making

Thanks! Questions?

- For our works and other related works, check out:

  www.fatml.org
- Workshop on Fairness, Accountability, and Transparency in ML (2014, 2015, 2016)