Introduction to Machine Learning

Fairness in Machine Learning

Mingchen Gao

Computer Science & Engineering
State University of New York at Buffalo
Buffalo, NY, USA
mgao8@buffalo.edu
Slides adapted from Varun Chandola

University at Buffalo
Department of Computer Science and Engineering
School of Engeering and Applied Sciences

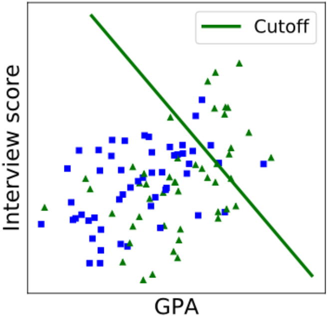
Outline

Introduction

- Main text - `https://fairmlbook.org` [1]
  - Solon Barocas, Moritz Hardt, Arvind Narayanan
- Other recommended resources:
  - Fairness in machine learning (NeurIPS 2017)
  - 21 fairness definitions and their politics (FAccT 2018)
  - Machine Bias - COMPAS Study
  - Mehrabi, Ninareh, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. **"A survey on bias and fairness in machine learning."** ACM computing surveys (CSUR) 54, no. 6 (2021): 1-35.

Toy Example

- *Task*: Learn a ML based job hiring algorithm
- *Inputs*: GPA, Interview Score
- *Target*: Average performance review
- *Sensitive attribute*: Binary (denoted by □ and Δ), represents some demographic group
  - We note that GPA is correlated with the sensitive attribute



Process

1. Regression model to predict target
2. Apply a threshold (denoted by green line) to select candidates

Toy Example

- ML models does not use sensitive attribute
- Does it mean it is fair?

Toy Example

- ML models does not use sensitive attribute
- Does it mean it is fair?
- It depends on the definition of fairness

Toy Example

- ML models does not use sensitive attribute
- Does it mean it is fair?
- It depends on the definition of fairness

Fairness-as-blindness notion

- Two individuals with similar features get similar treatment
- This model is fair

What about a different definition of fairness?

- Are candidates from the two groups equally likely to be hired?

What about a different definition of fairness?

- Are candidates from the two groups equally likely to be hired?
- No - triangles are more likely to be hired than squares
- Why did the model become unfair because of this definition?

Why this disparity in the data?

- Many factors could have led to this:
  - Managers who score employee's performance might have a bias
  - Workplace might be biased against one group
  - Socio-economic background of one group might have resulted in poor educational outcomes
  - Combination of these factors
- Let us assume that this disparity that was learnt by the ML model is unjustified
- How do we get rid of this?

Making ML model bias-free

- Option 1: ignore GPA as a feature
  - Might result in poor accuracy of the model

Making ML model bias-free

- Option 1: ignore GPA as a feature
  - Might result in poor accuracy of the model
- Option 2: pick different thresholds for each sub-group
  - Model is no longer "blind"

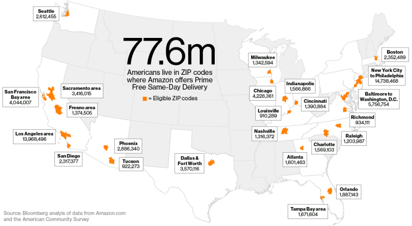
Making ML model bias-free

- Option 1: ignore GPA as a feature
  - Might result in poor accuracy of the model
- Option 2: pick different thresholds for each sub-group
  - Model is no longer "blind"
- Option 3: add a diversity reward to the objective function
  - Could still result in poor accuracy

Why fairness?

- We want/expect everything to be fair and bias-free
- Machine learning driven systems are everywhere : admissions, job offers, bail granting, loan approvals
- Obviously we want them to be fair as well
  - Closely related are issues of ethics, trust, and accountability

- A data-driven system to determine neighborhoods to offer *same-day delivery* service



- In many U.S. cities, white residents were more than twice as likely as black residents to live in one of the qualifying neighborhoods.
- *Src:* - `https://www.bloomberg.com/graphics/2016-amazon-same-day/`

ML - Antithesis to fairness

- Machine learning algorithms are based on *generalization*
- Trained on historical data which can be unfair
  - Our society has always been unfair
- Can perpetuate historical prejudices

Continuing with the Amazon example

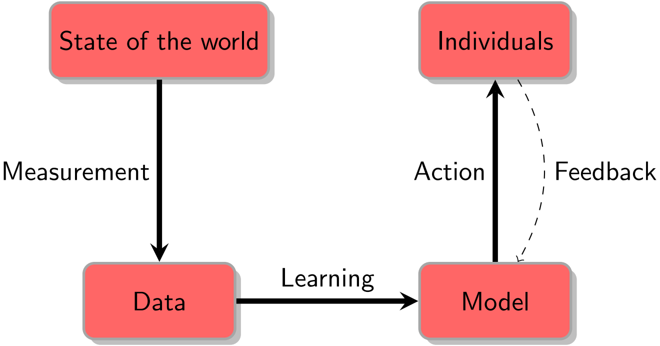
- Amazon claims that *race* was not a factor in their model (not a feature)
- Was designed based on efficiency and cost considerations
- Race was *implicitly* coded

What do we want to do?

- Make machine learning algorithms fair
- Need a quantifiable fairness metric
  - Similar to other performance metrics such as precision, recall, accuracy, etc.
- Incorporate the fairness metric in the learning process
- Often leads to a tension with other metrics

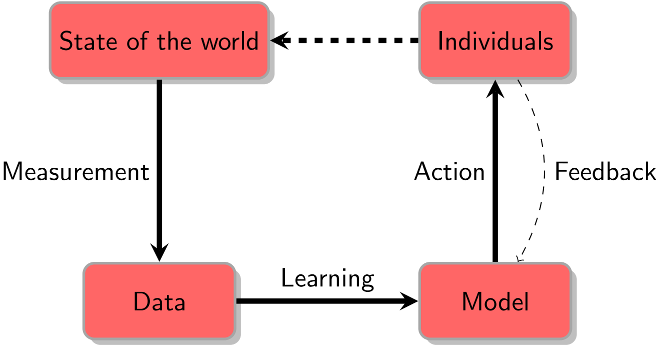
How does an ML algorithm becomes unfair?

- The “ML for People” Pipeline



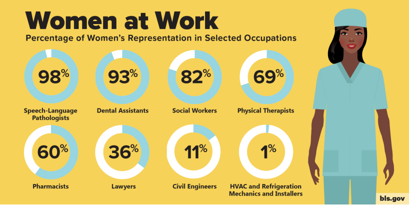
How does an ML algorithm becomes unfair?

- The "ML for People" Pipeline



Issues with the state of the society

- Most ML applications are about people
  - Even a pothole identification algorithm
- Demographic disparities exist in society
- These get embedded into the training data
- As ML practitioners we are not focused on removing these disparities
- We do not want ML to reinforce these disparities
- The dreaded **feedback loops** [3]



Measurement Issues

- Measurement of data is fraught with subjectivity and technical issues
- Measuring race, or any categorical variable, depends on how the categories are defined
- Most critical - defining the target variable
  - Often this is "made up" rather than measured objectively
  - credit-worthiness of a loan applicant
  - attractiveness of a face (beauty.ai, FaceApp)

Criminal Risk Assessment

1. Target variable - bail or not?
2. Target variable - will commit a crime later or not (recidivism)?

Measurement Issues

- Technical issues can often lead to bias
  - Default settings of cameras are usually optimized for lighter skin tones [5]

- Most images data sets used to train object recognition systems are biased relative to each other
  - http://people.csail.mit.edu/torralba/research/bias/

How to fix the measurement bias?

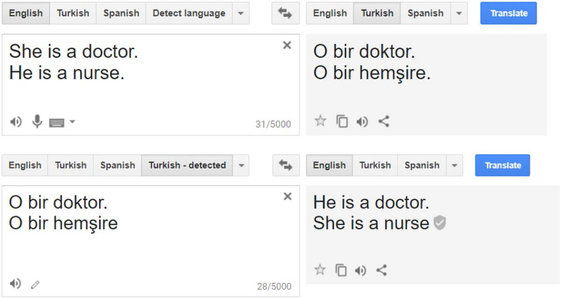
- Understand the provenance of the data
  - Even though you (ML practitioner) are working with data “given” to you
- “Clean” the data

Issues with models

- We know the training data can have biases
- Will the ML model preserve, mitigate or exacerbate these biases?
- ML model will learn a pattern in the data that assists in optimizing the objective function
- Some patterns are useful - *smoking is associated with cancer*, some are not - *girls like pink and boys like blue*
- But ML algorithm has not way of distinguishing between these two types of patterns
  - established by social norms and moral judgements
- Without a specific intervention, the ML algorithm will extract stereotypes

An Example

- Machine translation



How to make the ML model more fair

- Model reflects biases in the data
- Withold sensitive attributes (gender, race, . . .)
- Is that enough?

How to make the ML model more fair

- Model reflects biases in the data
- Withold sensitive attributes (gender, race, . . .)
- Is that enough?

Unfortunately not

- There could be *proxies* or *redundant encodings*
- Example - Using "programming experience in years" might indirectly encode gender bias
  - Age at which someone starts programming is well-known to be correlated with gender

How to make the ML model more fair

- Better objective functions that are fair to all sub-groups
- Ensure equal error rate for all sub-groups

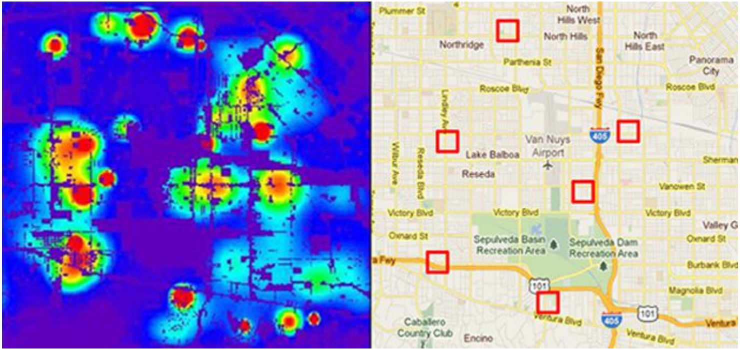
The Nymwars Controversy

- Google, Facebook and other companies blocking users with uncommon names (presumably *fake*)
- Higher error rate for cultures with a diverse set of names

The pitfalls of action

- While as ML practitioners our world ends after we have trained a *good* model
- But this model will impact people
- Need to understand that impact in the larger socio-technical system
  - Are there disparities in the error across different sub-groups?
  - How do these disparities change over time (drift)?
  - What is the perception of society about the model?
    - Ethics, trustworthiness, accountability
    - Explainability and interpretability
    - **Correlation is not causation**

The perils of feedback loops



- The "actions" made by individuals based on the predictions of the ML model could be fed back into the system, either explicitly or implicitly
  - Self-fulfilling predictions
  - Predictions impacting the training data
  - Predictions impacting the society

Problem Setup

Notation

- Predict Y given $\mathbf{X}$
- Y is our target class $Y \in \{0, 1\}$
- $\mathbf{X}$ represents the input feature vector

Example

- Y - Will an applicant pay the loan back?
- $\mathbf{X}$ - Applicant characteristics - credit history, income, etc.

Supervised Learning

- Given training data: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$
- Either learn a function f, such that:

$$y^* = f(\mathbf{x}^*)$$

- Or, assume that the data was drawn from a probability distribution
- In either case, we can consider the classification output as a random variable $\hat{Y}$
- Now we have three random variables:

$$\mathbf{X}, Y, \hat{Y}$$

- We are going to ignore how we get $\hat{Y}$ from $\mathbf{X}$ for these discussions

How do we measure the quality of a classifier?

- So far we have been looking at accuracy

A different way to look at accuracy

$$\text{Accuracy} \equiv P(Y = \hat{Y})$$

- Probability of the predicted label to be equal to the true label
- How do we calculate this?

Accuracy is not everyting!

- Consider a test data set with 90 examples with true class 1 and 10 examples with true class 0
- A *degenerate* classifier that classifies everything as label 1, would still have a 90% accuracy on this data set

Other evaluation criteria

Event	**Condition**	**Metric**
$\hat{Y} = 1$	$Y = 1$	True positive rate (recall on positive class)
$\hat{Y} = 0$	$Y = 1$	False negative rate
$\hat{Y} = 1$	$Y = 0$	False positive rate
$\hat{Y} = 0$	$Y = 0$	True negative rate (recall on negative class)

- Here we are treating class label 1 as the positive class and class label 0 as the negative class.

We can swap the condition and the event

Event	**Condition**	**Metric**
$Y = 1$	$\hat{Y} = 1$	precision (on positive class)
$Y = 0$	$\hat{Y} = 0$	precision (on negative class)

Score Functions

- Often classification involves computing a **score** and then applying a threshold
- E.g., Logistic regression: first calculate $P(Y = 1|\mathbf{X} = \mathbf{x})$, then apply a threshold of 0.5
- Or, Support Vector Machine: first calculate $\mathbf{w}^\top \mathbf{x}$ and then apply a threshold of 0

Conditional Expectation

$$r(\mathbf{x}) = \mathbb{E}[Y|\mathbf{X} = \mathbf{x}]$$

- We can treat it as a random variable too $R = \mathbb{E}[Y|\mathbf{X}]$
- This is what logistic regression uses.

From scores to classification

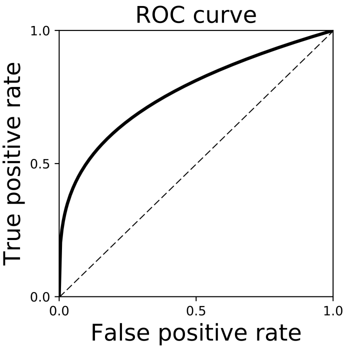
- Use a threshold t

$$y = \begin{cases} 1 & \text{if } r(\mathbf{x}) \geq t, \\ 0 & \text{otherwise} \end{cases}$$

- What threshold to choose?
  - If t is high, only few examples with very high score will be classified as 1 (accepted)
  - If t is low, only few examples with very low score will be classified as 0 (rejected)

The *Reciever Operating Characteristic* (ROC) Curve

- Exploring the entire range of t
- Each point on the plot is the FPR and TPR for a given value of t
- Area under the ROC curve or AUC is a quantitative metric derived from ROC curve

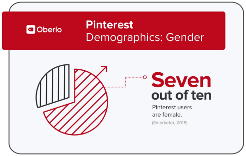


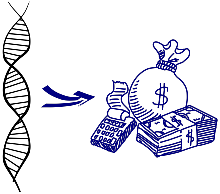
Sensitive Attributes

- Let A denote the attribute representing the sensitive characteristic of an individual
- There could be more than one sensitive attributes

Things to remember

- It is not always easy to identify A and differentiate it from $\mathbf{X}$
- Removing the sensitive attribute from $\mathbf{X}$ does not guarantee fairness
- Removing the sensitive attribute could make the classifier less accurate
- Not always a good idea to remove the impact of sensitive attributes





Quantifying Fairness

- Let us define some reasonable ways of measuring fairness
  - There are several ways to do this
  - All are debatable
- Three different categories

Independence	**Separation**	**Sufficiency**
$\hat{Y} \perp\!\!\!\perp A$	$\hat{Y} \perp\!\!\!\perp A \mid Y$	$Y \perp\!\!\!\perp A \mid \hat{Y}$

- Y - True label; $\hat{Y}$ - Predicted label; A - Sensitive attribute;

Conditional Independence

$$A \perp\!\!\!\perp B|C \Leftarrow P(A, B|C) = P(A|C)P(B|C)$$

- Amount of Speeding fine $\perp\!\!\!\perp$ Type of Car | Speed

Independence

$$P(\hat{Y} = 1 | A = a) = P(\hat{Y} = 1 | A = b), \forall a, b \in A$$

- Referred to as *demographic parity*, *statistical parity*, *group fairness*, *disparate impact*, etc.
- Probability of an individual to be assigned a class is equal for each group

Disparate Impact Law

$$\frac{P(\hat{Y} = 1 | A = a)}{P(\hat{Y} = 1 | A = b)} \geq 1 - \epsilon$$

For $\epsilon = 0.2$ - *80 percent rule*

Issues with independence measures

- *The self fulfilling prophecy* [2]
- Consider the hiring scenario where the model picks p excellent candidates from group a and p poor quality candidates from group b
  - Meets the independence criteria
  - However, it is still unfair

How to satisfy fairness criteria?

1. **Pre-processing phase**: Adjust the feature space to be uncorrelated with the sensitive attribute.
2. **Training phase**: Build the constraint into the optimization process for the classifier.
3. **Post-processing phase**: Adjust a learned classifier so that it is uncorrelated to the sensitive attribute

Separation

$$\hat{Y} \perp\!\!\!\perp A | Y$$

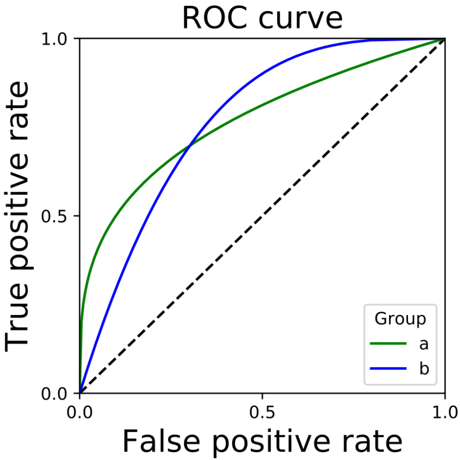
- Alternatively, the true positive rate and the false positive rate is equal for any pair of groups:

$$\begin{aligned} P(\hat{Y} = 1 | Y = 1, A = a) &= P(\hat{Y} = 1 | Y = 1, A = b) \\ P(\hat{Y} = 1 | Y = 0, A = a) &= P(\hat{Y} = 1 | Y = 0, A = b) \\ \forall a, b \in A \end{aligned}$$

- Can handle the discrepancy with the independence metric mentioned earlier

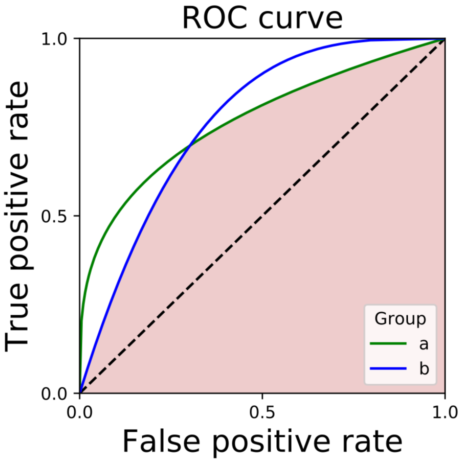
How to achieve separation

- Apply post-processing step using the ROC Curve
- Plot ROC curve for each group
- Within the constraint region (overlap), pick a classifier that minimizes the given cost



How to achieve separation

- Apply post-processing step using the ROC Curve
- Plot ROC curve for each group
- Within the constraint region (overlap), pick a classifier that minimizes the given cost



Sufficiency

$$Y \perp\!\!\!\perp A | R$$

- Alternatively, the precision is equal for any pair of groups:

$$P(Y = 1 | R = r, A = a) \quad = \quad P(Y = 1 | R = r, A = b)$$
$$\forall r \in dom(R) \text{ and } a, b \in A$$

Achieving sufficieny by calibration

What is calibration?

- Let us revert back to the score R
  - Recall that $\hat{Y}$ was obtained by applying a threshold on R
- R is *calibrated*, if for all r in the domain of R:

$$P(Y = 1|R = r) = r$$

- Of course, this means that R should be between 0 and 1
- *Platt Scaling*: Converts an uncalibrated score to a calibrated score [4]

- Calibration by group implies sufficiency
  - Apply Platt scaling to each group defined by the sensitive attribute

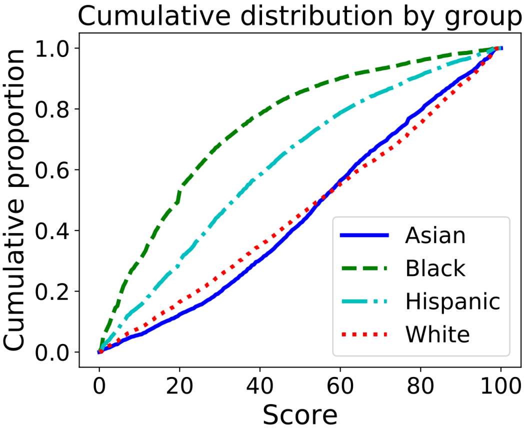
Case Study: Credit Scoring

- Extend loan or not - based on the risk that a loan applicant will default on a loan
- Data from the *Federal Reserve*
  - A - Demographic information (race)
  - R - Credit score
  - Y - Default or not (defined by credit bureau)

Table: Credit score distribution by race

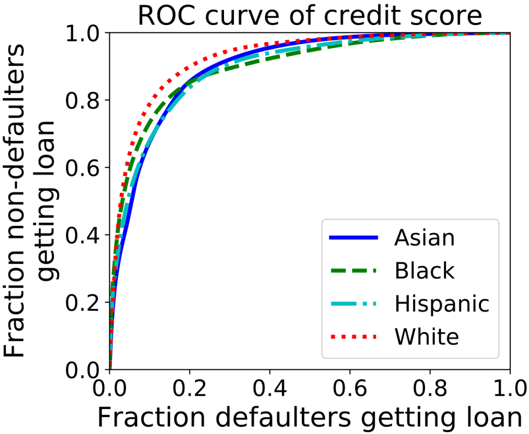
Race or ethnicity	Samples with both score and outcome
White	133,165
Black	18,274
Hispanic	14,702
Asian	7,906
Total	174,047

Group-wise distribution of credit score



- Strongly depends on the group

Using credit score for classification



- How make the classifier fair?

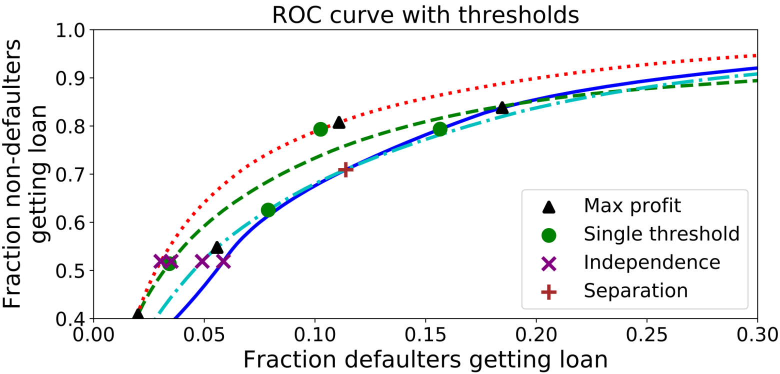
Four Strategies

1. *Maximum profit*: Pick group-dependent score thresholds in a way that maximizes profit
2. *Single threshold*: Pick a single uniform score threshold for all groups in a way that maximizes profit
3. *Separation*: Achieve an equal true/false positive rate in all groups. Subject to this constraint, maximize profit.
4. *Independence*: Achieve an equal acceptance rate in all groups. Subject to this constraint, maximize profit.

What is the profit?

- Need to assume a reward for a true positive classification and a cost/penalty for a false positive classification
- We will assume that cost of a false positive is 6 times greater than the reward for a true positive.

Comparing different criteria



References I

S. Barocas, M. Hardt, and A. Narayanan.
Fairness and Machine Learning.
fairmlbook.org, 2019.
http://www.fairmlbook.org.

C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel.
Fairness through awareness.
In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS 12, page 214226, New York, NY, USA, 2012. Association for Computing Machinery.

D. Ensign, S. A. Friedler, S. Neville, C. Scheidegger, and S. Venkatasubramanian.
Runaway feedback loops in predictive policing.
In *Conference on Fairness, Accountability and Transparency, FAT 2018, 23-24 February 2018, New York, NY, USA*, volume 81 of *Proceedings of Machine Learning Research*, pages 160–171. PMLR, 2018.

References II

J. Platt.
Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods.
Adv. Large Margin Classif., 10, 06 2000.

L. Roth.
Looking at shirley, the ultimate norm: Colour balance, image technologies, and cognitive equity.
Canadian Journal of Communication, 34:111–136, 2009.