• Section time preference sign-up extended to TODAY at 3pm. Sign-up at: http://cs181.stanford.edu

• Technical Assignment #1 on Algorithmic Decision-Making is out
  • Handout and starter code available on class website
  • You will need to use Eclipse (and Java) for the assignment
    • See the “Software” link on the class webpage to download Eclipse
    • New version of Eclipse (and plugin) this academic year, so if you downloaded Eclipse prior to Fall 2018, you’ll need new version
EVENT: Wednesday at 7pm

- On Wednesday (1/16) at 7:00 PM, we will be kicking off a Continuing Studies class that runs in parallel to this one. Our opening session will be a conversation with two formidable guests:
  - **Reid Hoffman**, Co-Founder and Executive Chairman, LinkedIn; Partner, Greylock Partners
  - **Nicole Wong**, Former Deputy Chief Technology Officer of the United States (2013-2014); Former Legal Director of Products, Twitter (2012-2013); Former Vice President and Deputy General Counsel, Google (2004-2011)

- We have reserved **100 seats for CS181 students**. Please RSVP using the link at [http://cs181.stanford.edu](http://cs181.stanford.edu). RSVP link will close 5pm Tuesday (tomorrow).
  - Seats will be granted on a lottery basis if more than 100 students RSVP.
  - If you are selected to attend, we will confirm your attendance by email by Wednesday morning.
Today’s Agenda

1. Introduction to machine learning and Perceptron algorithm
2. Definitions of “fairness” (with a brief intro. to probability)
3. Discussion of ProPublica analysis of COMPAS algorithm
4. Overview of technical assignment
What is Machine Learning?

- Many different forms of “Machine Learning”
  - We focus on the problem of prediction
- Want to make a prediction based on observations
  - Set of $n$ observed variables: $<X_1, X_2, ..., X_n>$
    - $X_1, X_2, ..., X_n$ are called “input features/variables”
    - For example: age, annual income, gender, education, etc.
    - Referred to as $X$ for short (it’s a vector, but that’s not important)
- Given observed $<X>$, want to predict other variable $Y$
  - $Y$ called “output feature/variable”
  - For example: whether someone with commit a crime in the future
- Seeking to “learn” a function $d(<X>)$ to predict $Y$:
  $$Y_{\text{prediction}} = d(<X>)$$
We are given set of $M$ “training” instances
- Each training instance is really a pair: $(<x_1, x_2, ..., x_n>, y)$
- Training instances are previously observed data
- Provides output value $y$ associated with each observed set of input values $<x_1, x_2, ..., x_n>$

Learning: use training data to specify $d(<X>)$
- Generally, first select a functional form for $d(<X>)$
- Then, determine parameters (weights) of model $d(<X>)$ using training data
Training data: set of $M$ pre-classified data instances

- $M$ training pairs: $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(M)}, y^{(M)})$
  - Use superscripts to denote $i$-th training instance

Learning algorithm: method for determining $d(X)$

- Given a new input observation of $X = <X_1, X_2, \ldots, X_n>$
- Use $d(X)$ to compute a corresponding output (prediction)
- When prediction is discrete, we call $d(X)$ a “classifier” and sometimes call the output the predicted “class” of the input
Basic Perceptron Algorithm

\[ \text{sum} = \sum_{i=1}^{n} x_i \cdot w_i \]

if (sum > 0) {
    prediction = 1
} else {
    prediction = 0
}

if (prediction != y) {  \hspace{1cm} (incorrect prediction)
    if (prediction == 1) {
        for each weight \( w_i \) (where \( i = 1 \) to \( n \))
        \[ w_i = w_i - x_i \]
    } else {
        for each weight \( w_i \) (where \( i = 1 \) to \( n \))
        \[ w_i = w_i + x_i \]
    }
}
Basic Perceptron Algorithm

\[ \text{sum} = \sum_{i=1}^{n} x_i \cdot w_i \]

if (sum > 0) {
    prediction = 1
} else {
    prediction = 0
}

// Mathematically equivalent, but more compact update rule
error = y - prediction

if (error != 0) {
    (incorrect prediction)
    for each weight \( w_i \) (where \( i = 1 \) to \( n \))
        \[ w_i = w_i + (error \times x_i) \]
}
Batch Perceptron Pocket Algorithm

- **Batch**: for each pass through training data
  - Compute what the change in weights would be for each instance
  - Average the changes over all instances (“average difference”)
  - Update weights with average difference

- **Pocket**: for each pass through training data
  - Compute number of correct predictions made with current weights
  - If number of correct predictions is more than any previous pass, save this set of weights in our “pocket”
  - After making some number of passes through the data for training, we use the set of weights in our “pocket” as the final model

- More details (and pseudocode) in the “Probability and Machine Learning” handout/reading
  - That is the algorithm implemented in the assignment
Today’s Agenda

1. Introduction to machine learning and Perceptron algorithm
2. Definitions of “fairness” (with a brief intro. to probability)
3. Discussion of ProPublica analysis of COMPAS algorithm
4. Overview of technical assignment
What is “Fairness”? 

- There are many definitions of fairness  
  - We’ll focus on just a few of them  
  - Requires a bit of background in probability  
  - So here’s the world’s fastest* introduction to probability

- Probability: Chance that something will happen  
  - Coin flip can be heads or tails. Set X = 1 if heads, 0 otherwise  
  - Pr(X = 1) Chance that variable X = 1 (flipped “heads”)

- Conditional probability: Probability that something will happen given that something else has been observed  
  - Pr(X = 1 | Y = 1) Chance that variable X = 1 given that we know Y = 1

*The claim of “world’s fastest” only exists in Mehran’s head and has not been substantiated by an independent outside source. Then again, that may be true of everything he’s saying here. But you trust him. He’s a nice guy. Besides, he has a lightsaber and no one with a lightsaber can be bad. Except that Darth Vader guy (and he was kinda good in the end). And Emperor Palpatine. And Darth Maul. Oh, nevermind...
Some Definitions Related to “Fairness”

- **Anti-classification**: decisions do not consider “protected” characteristics (e.g., race, gender, age, etc.)
  - Decision made for individual X is same as that for individual X’ if we only consider unprotected characteristics of X and X’

- **Classification parity**: Classification error is equivalent across groups defined by protected characteristics \(X_p\)
  - E.g., Parity of false positives: \(\Pr(d(X) = 1 \mid Y = 0, X_p) = \Pr(d(X) = 1 \mid Y = 0)\)

- **Calibration**: Outcomes should be independent of protected characteristics conditional on risk scores, \(s(X)\)
  - Formally: \(\Pr(Y = 1 \mid s(X), X_p) = \Pr(Y = 1 \mid s(X))\)

- (Lack of) **Disparate impact**: impact of a policy should not be different between two groups (based on protected characteristic)
  - Does not require discriminatory intent
Today’s Agenda

1. Introduction to machine learning and Perceptron algorithm
2. Definitions of “fairness” (with a brief intro. to probability)
3. Discussion of ProPublica analysis of COMPAS algorithm
4. Overview of technical assignment
COMPAS Algorithm

- “Black box” model by Northpointe to assess risk of recidivism
  - Predicts a risk score of recidivism based on features of individual
  - Race is not one of the input features to the model

<table>
<thead>
<tr>
<th>Contingency Table</th>
<th>Recidivated</th>
<th>Did not recidivate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Labeled High-risk</td>
<td>True positive (A)</td>
<td>False positive (B)</td>
</tr>
<tr>
<td>Labeled Low-risk</td>
<td>False negative (C)</td>
<td>True negative (D)</td>
</tr>
</tbody>
</table>

- ProPublica analysis (no classification parity)
  - Score correctly predicted recidivism: 61% \(\frac{A}{A+B}\)
    - Correct for white defendants: 59% \(\frac{A}{A+B}\)
    - Correct for black defendants: 63% \(\frac{A}{A+B}\)
  - But, the way misclassification were made were different
    - Black who did not recidivate, % labeled high-risk: 45% \(\frac{B}{B+D}\)
    - White who did not recidivate, % labeled high-risk: 23% \(\frac{B}{B+D}\)
    - Blacks who recidivated, % labeled low-risk: 28% \(\frac{C}{A+C}\)
    - Whites who recidivated, % labeled low-risk: 48% \(\frac{C}{A+C}\)
Northpointe responds that algorithm is fair because risk scores are equally predictive of recidivism for both blacks and whites.

- Calibration: \( \Pr(Y = 1 \mid s(X), X_p) = \Pr(Y = 1 \mid s(X)) \)
Here Come the Computer Scientists

• Can’t we have just have all definitions of fairness
  • Let me just crank up my deep neural network...
• Sorry, Kleinberg et al (2017) prove you can’t (generally) get both calibration and classification parity
• And, you can have proxies for protected characteristics
  • (Sets of) features that are not protected, but correlate strongly with protected features
  • And it can be hard to determine which such features should be allowed
• And, there can be historical bias or disproportionality in the data that will be reflected in results of machine learning algorithms
  • E.g., A classifier built to predict a condition that only occurs in 0.5% of the population is 99.5% accurate if it always predicts that no one has condition
• And, there’s the problem of infra-marginality
  • Say what?!
Risk Distributions Differ

- Distribution of defendants across risk categories by race (Corbett-Davies et al, 2016):

- Black defendants recidivism rate is higher than whites
  - So higher proportion of black defendants are deemed medium or high risk
  - As a result, blacks who do not reoffend are also more likely to be classified higher risk than whites who do not reoffend
Risk Distributions Differ

![Graph showing risk distributions for different groups.](image)

Density vs Probability of reoffending for Black and White groups.

Slide thanks to Sam Corbett-Davies
Risk Distributions Differ

• Use a single threshold based on risk scores for detention
  • Might produce disparate false positive rates (what ProPublica found)

Slide thanks to Sam Corbett-Davies
• Use different thresholds based on race to equalize error rates
  • Violates notion of anti-classification since you discriminate based on a protected characteristic (race)
MACHINE BIAS

Bias in Criminal Risk Scores Is Mathematically Inevitable, Researchers Say
Overview of Assignment

- Now that you’re sufficiently disturbed, let’s think happier thoughts... like your assignment.