CS 182: Ethics, Public Policy, and Technological Change

Rob Reich
Mehran Sahami
Jeremy Weinstein
Hilary Cohen
• Technical Assignment #1 on Algorithmic Decision-Making is out
  • See “Handouts” link on class webpage to get assignment handout and handouts on using Eclipse or PyCharm
  • See “Assignment” link on class web site to get starter code
  • Can use either Java (Eclipse) or Python (PyCharm) for the assignment
    • Starter code (identical functionality) available in Java and Python
    • See “Software” link on class webpage to download Eclipse/PyCharm
  • Assignment #1 due at 1:00pm on January 29th
    • Submission through Gradescope
      • Will submit code and write-up to questions separately
    • Gradescope CS182 Entry Code in Assignment #1 handout (at end)

• Sections start this week (on Thursday)
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
Many different forms of machine learning
  • We focus on the prediction (or classification) task

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 $\mathbf{X}$ for short (it’s a vector, but that’s not important)

Given observed $\mathbf{X}$, want to predict other variable $Y$
  • $Y$ called “output feature/variable”
  • Example 1: whether applicant should be issued a credit card
  • Example 2: whether defendant will commit a crime in the future

Seeking to “learn” a function $d(\mathbf{X})$ to predict $Y$:
\[
Y_{\text{prediction}} = d(\mathbf{X})
\]
Training a Learning Machine

• 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 $x = <x_1, x_2, ..., x_n>$

• 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
The Machine Learning Process

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
Recall: Promises/Perils

• Promises
  • Provide insights about domain
  • Improve accuracy of prediction compared to humans
    • Diminish/eliminate bias and inconsistency
  • Greater efficiency than human decision-making
    • Humans are slow and error-prone

• Perils
  • Encode existing biases and reduce fairness
  • Lack transparency and threaten due process
  • Increased efficiency is not always a benefit
**Basic Perceptron Algorithm**

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

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

if \text{prediction} != y: \quad \text{\textit{(incorrect prediction)}}
    if \text{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: \hspace{1cm} \text{(incorrect prediction)}
    for each weight \( w_i \) (where \( i = 1 \) to \( n \))
    \[
    w_i = w_i + (\text{error} \times x_i)
    \]
Batch Perceptron Pocket Algorithm

- **Batch**: for each pass through training data (called an "epoch")
  - 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 higher 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 Assignment #1
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
Legal Concepts Related to Fairness

- **Protected characteristics**
  - Some characteristics cannot be used to discriminate individuals in decision-making in particular circumstances
  - For example, in employment decisions, protected characteristics include: race, gender, and age (among others)
  - In medicine, however, it may make sense to prescribe different treatments to different genders

- **Disparate impact**
  - Impact of a policy is different between two groups distinguished by a protected characteristic
  - Does not require discriminatory intent
What is “Fair”? 

• There are many definitions of fairness
  • Narayanan (2018) provided 21 definitions of fairness
  • We will focus on some of the most commonly discussed definitions
  • Requires a bit of background in probability to formalize
    • So, here’s a working 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
  • $\Pr(X = 1 \mid Y = 1)$ Chance that variable $X = 1$ given that we know $Y = 1$
• **Anti-classification**: decisions do not consider “protected” characteristics (e.g., race, gender, age, etc.)
  • Consider only unprotected characteristics of two individuals X and X’
  • Implies: if the unprotected characteristics of X and X’ are the same, then the decision made for X and X’ should be the same

• **Classification parity**: Classification error is equivalent across groups defined by protected characteristics ($X_p$)
  • E.g., Parity of false positives: $\Pr(d(X) = 1 \mid Y = 0, X_p) = \Pr(d(X) = 1 \mid Y = 0)$
    • If you would not recidivate, then knowing your protected characteristics should not change the probability that we predict you will recidivate (chance of false positive prediction)
• **Calibration**: Outcomes should be independent of protected characteristics conditional on risk scores, $s(X)$
  - Formally: $\Pr(Y = 1 \mid s(X), X_p) = \Pr(Y = 1 \mid s(X))$
  - Given your risk score, the probability that you will recidivate should not change if we additionally knew your protected characteristics
  - In Perceptron, could think of the sum $\sum_{i=1}^{n} x_i \cdot w_i$ as a form of risk score that is then thresholded to make a prediction. (Risk scores often binned.)

• (Lack of) **Disparate impact**: impact of a policy should not be different between two groups (based on protected characteristic)
  - Recall, disparate impact does not require discriminatory intent, only that the impact is disparate between the two groups
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
    • Blacks who did not recidivate, % labeled high-risk: 45% \(\frac{B}{B+D}\)
    • Whites 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% \(\frac{C}{A+C}\)
COMPAS Algorithm

• Northpointe responds that algorithm is fair because risk scores are equally predictive of recidivism for both blacks and whites
  • Calibration: \( \Pr(Y = 1 \mid s(X), X_p) = \Pr(Y = 1 \mid s(X)) \)
• Algorithms require formalization of what should be optimized

• If we want to use algorithms to make decisions, it forces us to be precise about what we think "fairness" is and how we would define it

• How do you define fairness?

• What should the algorithm try to optimize?
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
    - Here come the lawyers...
- 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 in data
  • So higher proportion of black defendants are deemed medium or high risk
  • As a result, black defendants who do not reoffend are also more likely to be classified higher risk than white defendants who do not reoffend
Risk Distributions Differ

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)
Risk Distributions Differ

- Use different thresholds based on race to equalize error rates
  - Violates notion of anti-classification since you discriminate based on a protected characteristic (race)

Slide thanks to Sam Corbett-Davies
MACHINE BIAS

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

• Should we consider protected characteristics in an algorithm, if it can yield:
  • higher predictive accuracy?
  • error rate parity between different racial/gender groups?
  • correct for historical bias in the data?

• We want you to explore questions like this in your assignment

• Quick overview of assignment