What is Machine Learning?

And why might it be unfair?

Aaron Roth

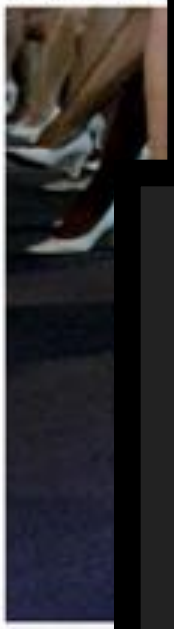
Computer and Information Sciences

University of Pennsylvania

PREDICTIVE POLICING: USING MACHINE LEARNING TO DETECT PATTERNS OF CRIME

Beauty contest judged by AI and the robots discriminate against dark skin

3 days ago | Pu



Sept9:Th
such as fa
this year,
intelligen
"human b

But when
winners:

Is an algorithm any less racist than a human?

Media Personal Finance Small Biz Luxury stock tickers

ta is driving inequality

etermining the
id and the
ecisions about

hat rely on data
Algorithms are
gorithms adjust
searchers in
human

ad for high-
to women, a
d.

t records were
tively black
Commission said
neighborhoods

Machine Bias

There's software used across the country to predict future criminals. And it's biased against blacks.

by Julia Angwin, Jeff Larson, Surya Mattu and Lauren Kirchner, ProPublica
May 23, 2016

ON A SPRING AFTERNOON IN 2014, Brisha Borden was running late to pick up her god-sister from school when she spotted an unlocked kid's blue Huffy bicycle and a silver Razor scooter. Borden and a friend grabbed the bike and scooter and tried to ride them down the street in the Fort Lauderdale suburb of Coral Springs.

Just as the 18-year-old girls were realizing they were too big for the tiny conveyances — which belonged to a 6-year-old boy — a woman came running after them saying, "That's my kid's stuff." Borden and her friend immediately dropped the bike and scooter and walked away.

But it was too late — a neighbor who witnessed the heist had already called the police. Borden and her friend were arrested and charged with burglary and petty theft for the

What is Machine Learning?

Its just statistics*.

*With a particular emphasis on *prediction*.

*And with an eye towards engineering in its design.

This Talk:
Focus on Supervised Classification

- And ignore 2 other major subfields of ML:
  - Unsupervised learning (Clustering)
  - Reinforcement learning (Control)

The Basic Setup

- Given: Data, consisting of d *features* and a *label*.
- Goal: Find a rule to predict *label* from *features*.

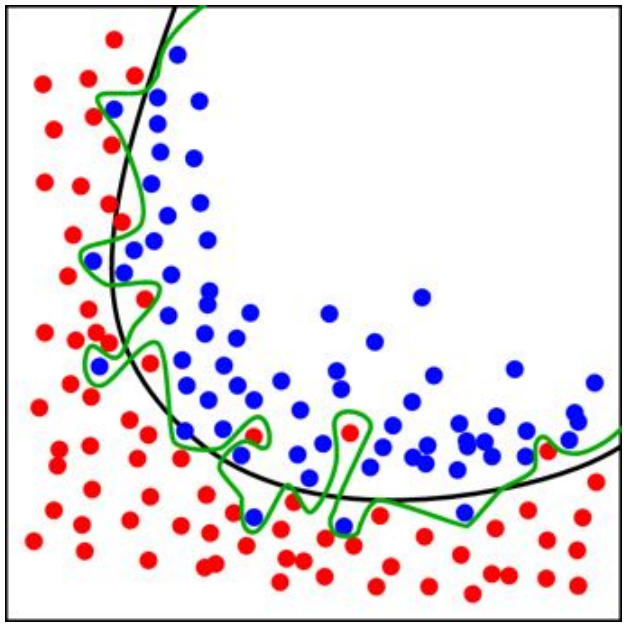
College?	Bankruptcy?	Tall?	Employed?	Homeowner?	Paid back loan?
Y	N	Y	Y	N	**Yes**
Y	N	N	Y	N	**Yes**
N	N	N	N	Y	**No**
Y	Y	Y	Y	N	**No**
N	N	Y	Y	N	**Yes**
N	N	N	N	Y	**No**
N	Y	Y	Y	Y	**Yes**

x y

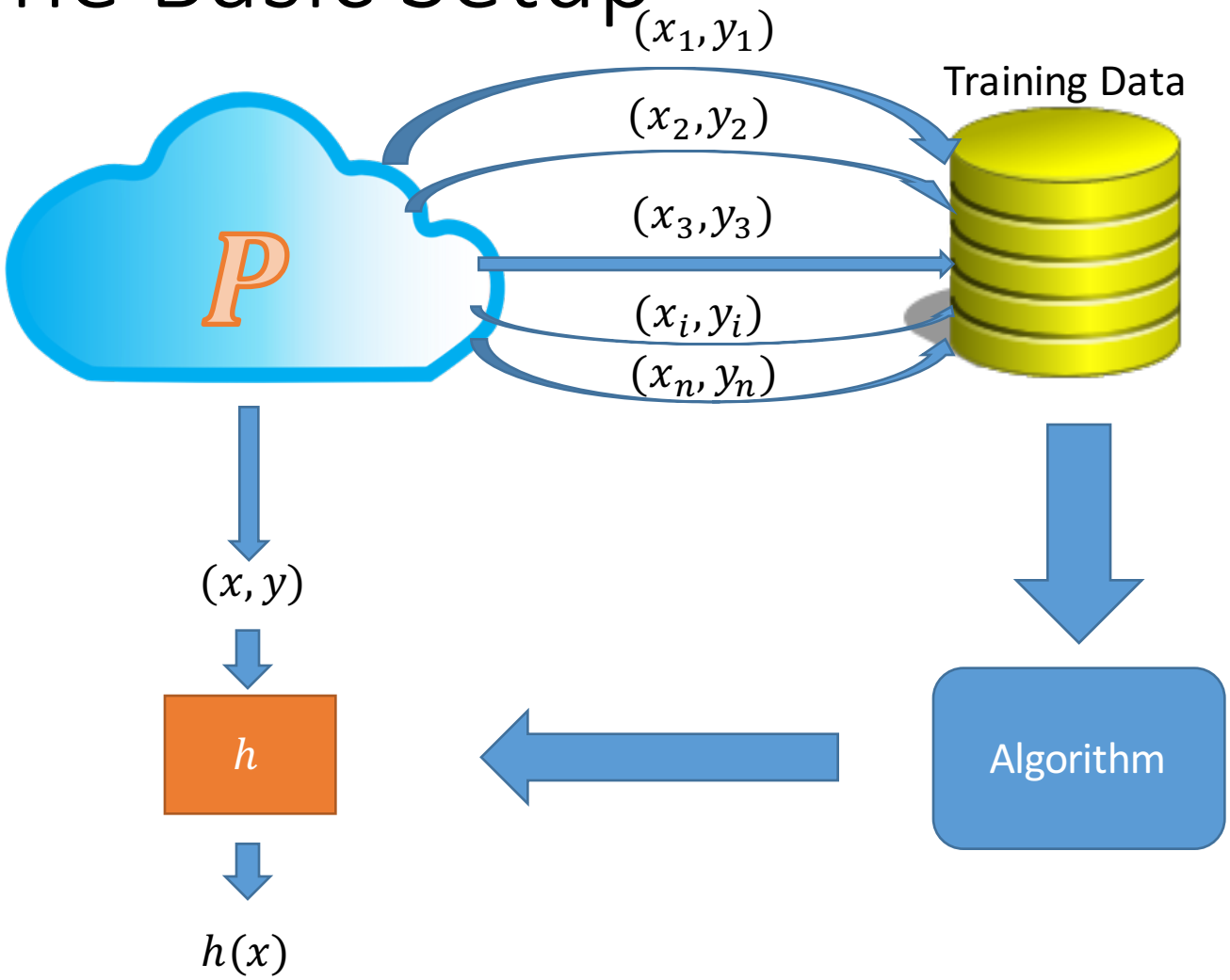
(College and **Employed** and not **Bankruptcy)** or **(Tall** and **Employed** and not **College)**

The Basic Setup

- Given data, select a *hypothesis* $h: \{Y, N\}^d \rightarrow \{Y, N\}$
- Goal is not prediction on the *training data*, but prediction on *new examples*.



The Basic Setup



The example is misclassified if $h(x) \neq y$.

The Basic Setup

- Goal: Find a classifier to minimize

$$err(h) = \Pr_{(x,y)\sim P}[h(x) \neq y]$$

We don't know P…

But we *can* minimize the empirical error:

$$\widehat{err}(h, D) = \frac{1}{n}|\{(x, y) \in D : h(x) \neq y\}|$$

The Basic Setup

College?	Bankruptcy ?	Tall?	Employed?	Homeowner?	Paid back loan?
Y	N	Y	Y	N	**Yes**
Y	N	N	Y	N	**Yes**
N	N	N	N	Y	**No**
Y	Y	Y	Y	N	**No**
N	N	Y	Y	N	**Yes**
N	N	N	N	Y	**No**
N	Y	Y	Y	Y	**Yes**

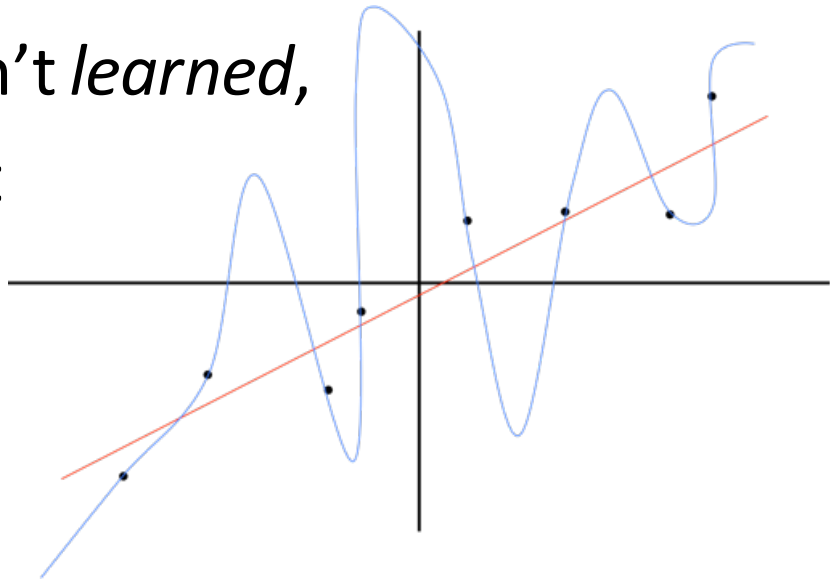
- Empirical Error Minimization:
  - Try a lookup table!

  $h(x) = Y$ if $x \in \{YNYYN, YNNYN, YNYYN, NYYYY\}$

  $h(x) = N$ otherwise.

  $\widehat{err}(h, D) = 0.$

- This would over-fit. We haven't *learned*, Just memorized. Learning must summarize.

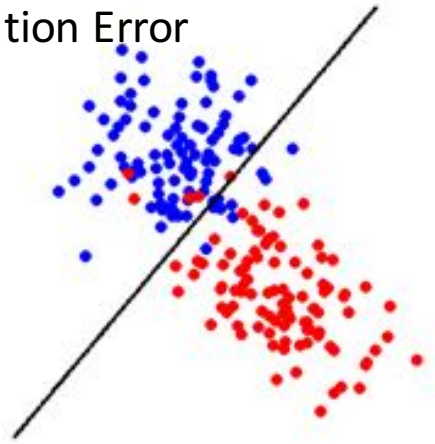


The Basic Setup

- Instead, limit hypotheses to come from a "simple" class C.
  - E.g. *linear* functions, or *small* decision trees, etc.

Compute $h^* = \arg\min_{h \in C} \widehat{err}(h, D)$

$$err(h^*) \leq \underbrace{\widehat{err}(h^*, D)}_{\text{Training Error}} + \underbrace{\max_{h \in C} |err(h) - \widehat{err}(h, D)|}_{\text{Generalization Error}}$$



The Basic Setup

- If you have sufficiently much data to guarantee:

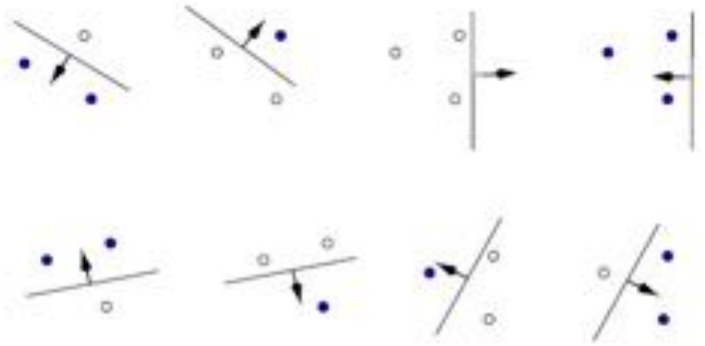
$$\text{generalization error} \leq \epsilon$$

Then you know $err(h^*) \leq OPT(C) + \epsilon$.

- How much is enough?
  - Depends on the complexity of the functions in C.

The Basic Setup

- VC-Dimension: “The largest number of points that functions in your class can label in all possible ways”



- VC-Dimension of 2 dimensional linear functions: 3
- VC-Dimension of d-dimensional linear functions: d+1
- “Rule of Thumb”: VC-Dimension ≈ “number of parameters”

The Basic Setup

- If $n \geq \frac{VCDIM(C)}{\epsilon^2}$, then with high probability:

$$\text{Generalization error} \leq \epsilon$$

$$\text{So, } err(h^*) \leq OPT(C) + \epsilon.$$

- Have to trade off complexity of C with generalization error…
  - Increasing complexity decreases $OPT(C)$, increases ϵ.
  - If you want both, need more data.

The Basic Setup

- Machine Learning as Optimization:

$$\text{Minimize } \sum_{i=1}^{n} \ell(h; x_i, y_i)$$
$$\text{such that } h \in C$$

- e.g. $\ell(h; x_i, y_i) = \begin{cases} 1, & h(x_i) \neq y_i \\ 0, & h(x_i) = y_i \end{cases}$

Caveat! Computational Hardness!

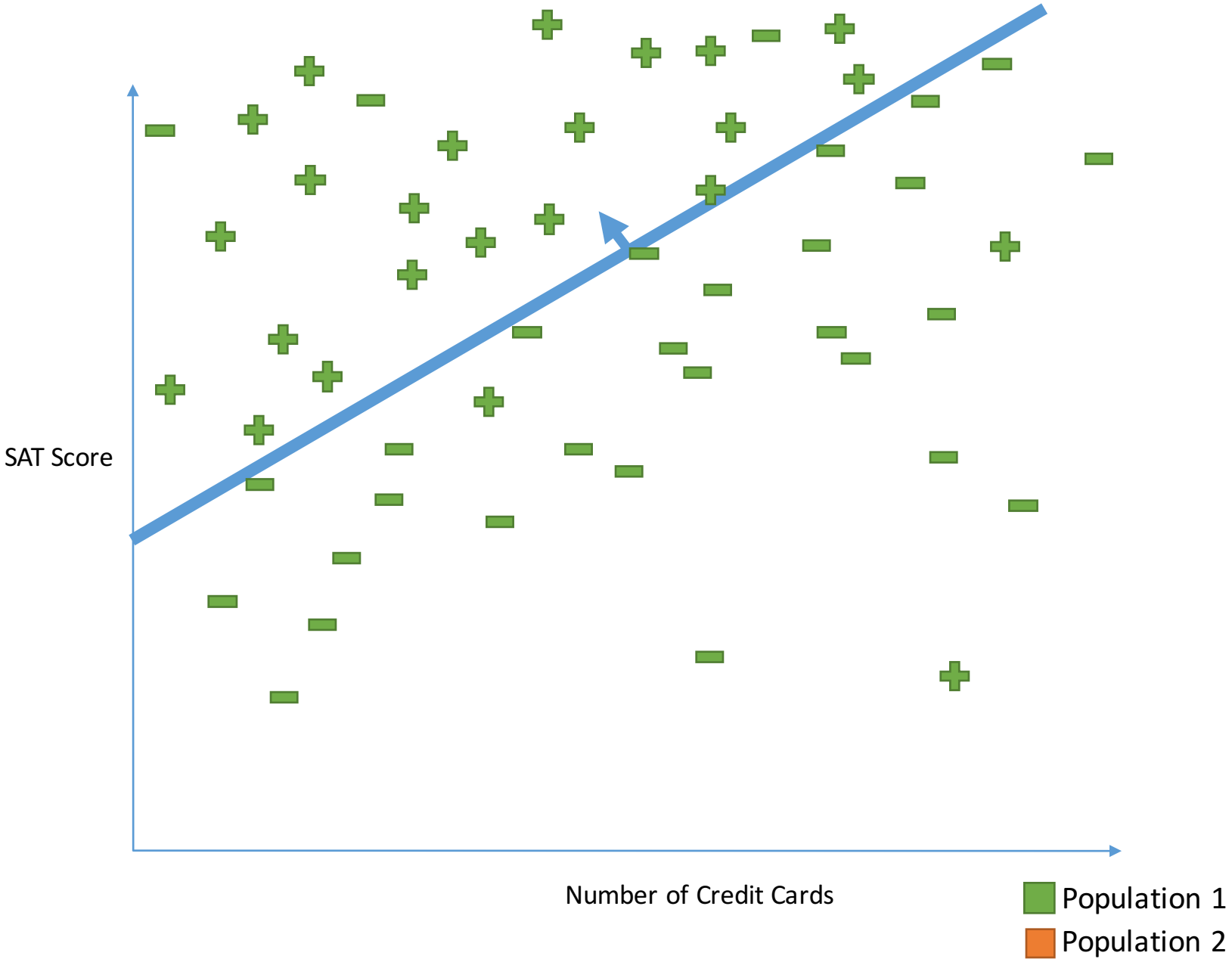
- Optimizing classification accuracy often intractable.
- Solutions:
  - Optimize a different *surrogate* loss function $\hat{\ell}(h; x, y)$
    - Hinge loss, squared error, etc.
  - Heuristically try and optimize
    - Might get stuck in local optima/fail to optimize globally
  - Both
- Can still talk sensibly about generalization.
- Still looks "fair" – can observe and debate objective function, e.g.

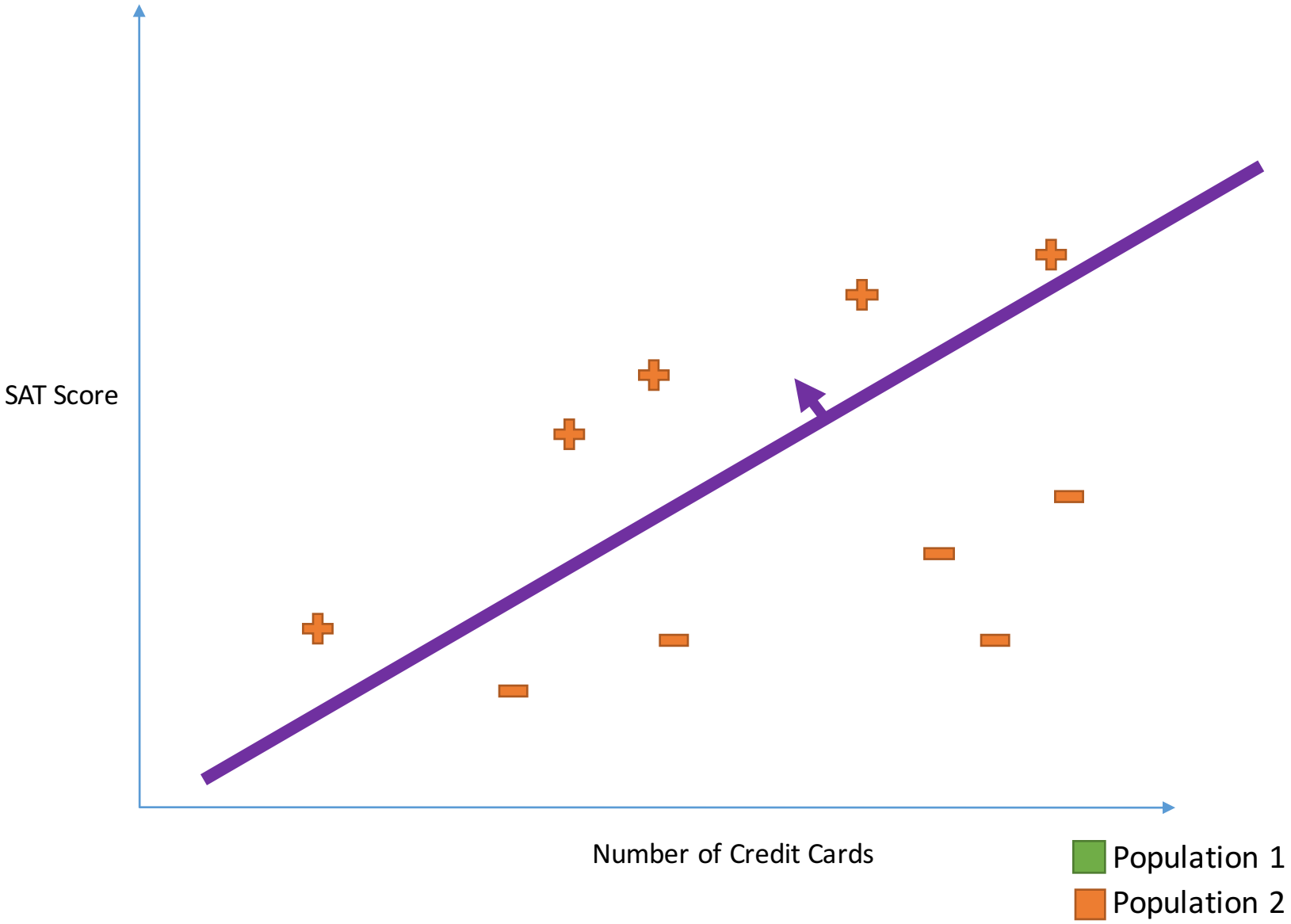
Why might machine learning be "unfair"?

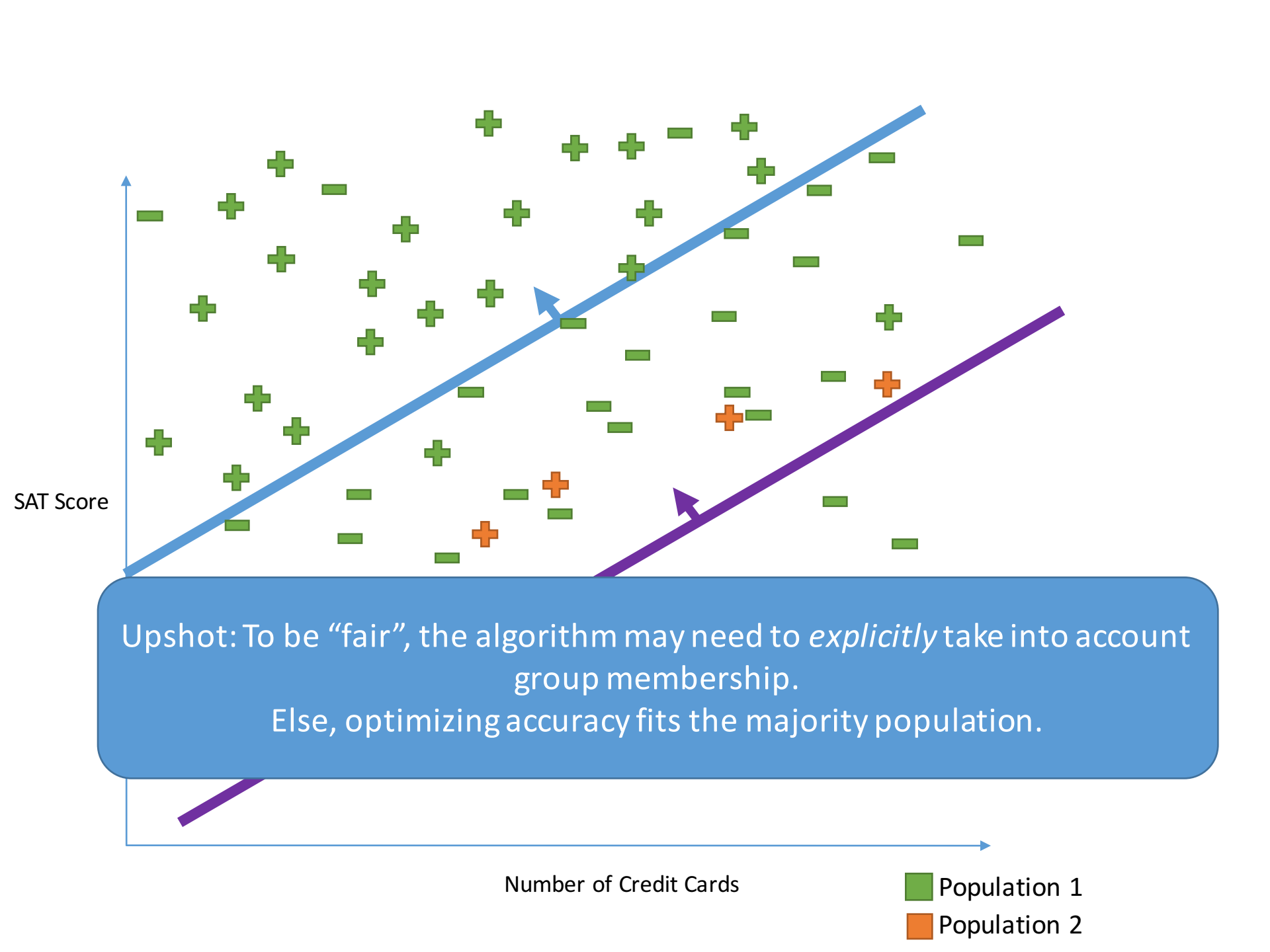
- Many reasons:
  - Data might encode existing biases.
    - E.g. labels are not "Committed a crime?" but "Was arrested."
  - Data collection feedback loops.
    - E.g. only observe "Paid back loan?" if the loan was granted.
  - Different populations with different properties.
    - E.g. "SAT score" might correlate with label differently in populations that employ SAT tutors.
  - Less data (by definition) about minority populations.

Why might machine learning be "unfair"?

- Many reasons:
  - Data might encode existing biases.
    - E.g. labels are not "Committed a crime?" but "Was arrested."
  - Data collection feedback loops.
    - E.g. only observe "Paid back loan?" if the loan was granted.
  - Different populations with different properties.
    - E.g. "SAT score" might correlate with label differently in populations that employ SAT tutors.
  - Less data (by definition) about minority populations.


SAT Score
Number of Credit Cards
Population 1
Population 2


SAT Score
Number of Credit Cards
Population 1
Population 2

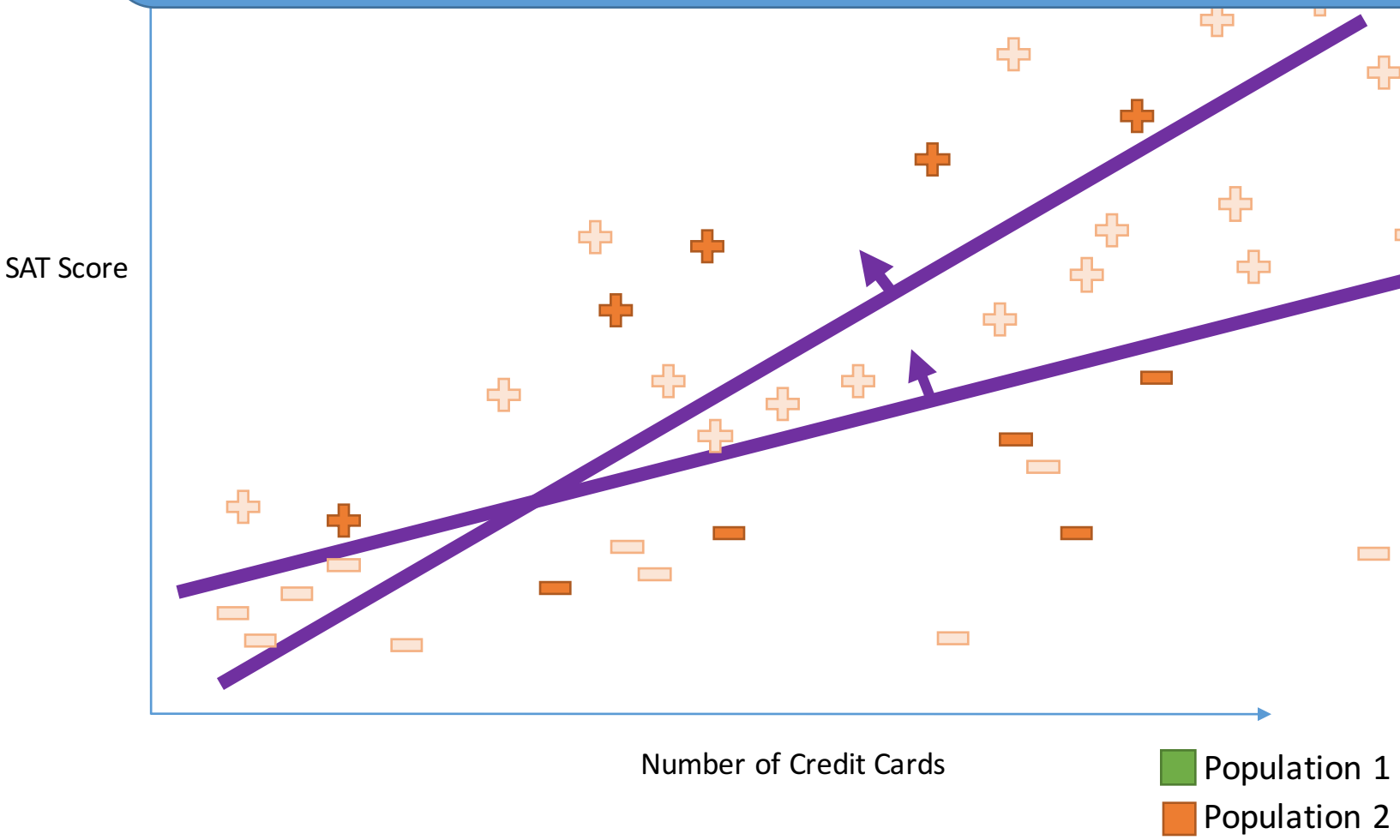

SAT Score
Number of Credit Cards
Upshot: To be "fair", the algorithm may need to *explicitly* take into account group membership.
Else, optimizing accuracy fits the majority population.
Population 1
Population 2

Why might machine learning be "unfair"?

- Many reasons:
  - Data might encode existing biases.
    - E.g. labels are not "Committed a crime?" but "Was arrested."
  - Data collection feedback loops.
    - E.g. only observe "Paid back loan?" if the loan was granted.
  - Different populations with different properties.
    - E.g. "SAT score" might correlate with label differently in populations that employ SAT tutors.
  - Less data (by definition) about minority populations.


SAT Score
Number of Credit Cards
Population 1
Population 2

Upshot: Algorithms trained on minority populations will be less accurate. Qualified individuals will be denied at a higher rate.



Why might machine learning be "unfair"?

- Many reasons:
  - Data might encode existing biases.
    - E.g. labels are not "Committed a crime?" but "Was arrested."
  - Data collection feedback loops.
    - E.g. only observe "Paid back loan?" if the loan was granted.
  - Different populations with different properties.
    - E.g. "SAT score" might correlate with label differently in populations that employ SAT tutors.
  - Less data (by definition) about minority populations.

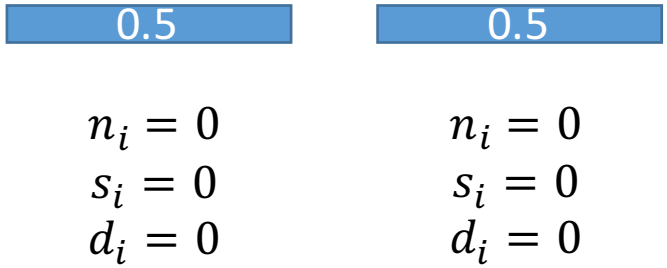
Toy Model

- Two kinds of loan applicants.
  - Type 1: Pays back loan with unknown probability p_1
  - Type 2: Pays back loan with unknown probability p_2
- Initially, bank believes p_1, p_2 uniform in $[0,1]$
- Every day, bank makes a loan to type most likely to pay it back according to posterior distribution on p_1, p_2
- Bank observes if it is repaid, but not counterfactuals. Bank then updates posteriors.

Toy Model

- Suppose bank has made n_i loans to population i, s_i have paid them back , d_i have defaulted. ($n_i = s_i + d_i$)
- Expected payoff of next loan is $\frac{s_i+1}{s_i+d_i+2}$

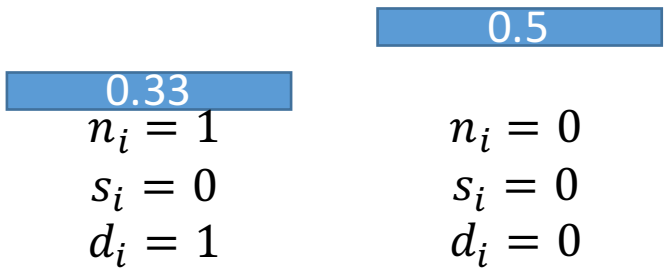
$$p_1 = p_2 = \frac{1}{2}$$



Toy Model

- Suppose bank has made n_i loans to population i, s_i have paid them back , d_i have defaulted. ($n_i = s_i + d_i$)
- Expected payoff of next loan is $\frac{s_i+1}{s_i+d_i+2}$

$$p_1 = p_2 = \frac{1}{2}$$



Toy Model

- Suppose bank has made n_i loans to population i, s_i have paid them back , d_i have defaulted. ($n_i = s_i + d_i$)
- Expected payoff of next loan is $\frac{s_i+1}{s_i+d_i+2}$

$$p_1 = p_2 = \frac{1}{2}$$

0.66

0.33

$n_i = 1$
$s_i = 0$
$d_i = 1$

$n_i = 1$
$s_i = 1$
$d_i = 0$

Toy Model

- Suppose bank has made n_i loans to population i, s_i have paid them back , d_i have defaulted. ($n_i = s_i + d_i$)
- Expected payoff of next loan is $\frac{s_i+1}{s_i+d_i+2}$

$$p_1 = p_2 = \frac{1}{2}$$

0.75

0.33

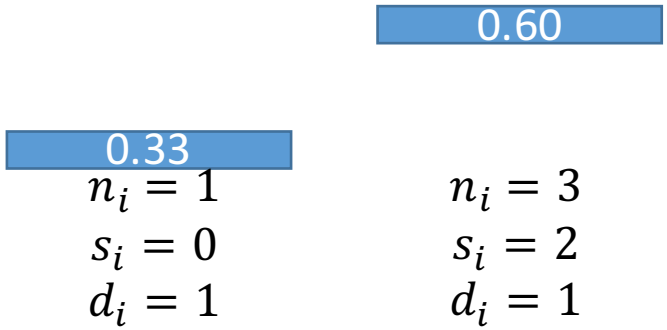
$n_i = 1$
$s_i = 0$
$d_i = 1$

$n_i = 2$
$s_i = 2$
$d_i = 0$

Toy Model

- Suppose bank has made n_i loans to population i, s_i have paid them back , d_i have defaulted. ($n_i = s_i + d_i$)
- Expected payoff of next loan is $\frac{s_i+1}{s_i+d_i+2}$

$$p_1 = p_2 = \frac{1}{2}$$



Toy Model

- Suppose bank has made n_i loans to population i, s_i have paid them back , d_i have defaulted. ($n_i = s_i + d_i$)
- Expected payoff of next loan is $\frac{s_i+1}{s_i+d_i+2}$

$$p_1 = p_2 = \frac{1}{2}$$

0.5

0.33

$n_i = 1$
$s_i = 0$
$d_i = 1$

$n_i = 4$
$s_i = 2$
$d_i = 2$

Toy Model

- Suppose bank has made n_i loans to population i, s_i have paid them back , d_i have defaulted. ($n_i = s_i + d_i$)
- Expected payoff of next loan is $\frac{s_i+1}{s_i+d_i+2}$

$$p_1 = p_2 = \frac{1}{2}$$

0.57

0.33

$n_i = 1$
$s_i = 0$
$d_i = 1$

$n_i = 5$
$s_i = 3$
$d_i = 2$

Toy Model

- Suppose bank has made n_i loans to population i, s_i have paid them back , d_i have defaulted. ($n_i = s_i + d_i$)
- Expected payoff of next loan is $\frac{s_i+1}{s_i+d_i+2}$

$$p_1 = p_2 = \frac{1}{2}$$

0.625

0.33

$n_i = 1$
$s_i = 0$
$d_i = 1$

$n_i = 6$
$s_i = 4$
$d_i = 2$

Toy Model

- Suppose bank has made n_i loans to population i, s_i have paid them back , d_i have defaulted. ($n_i = s_i + d_i$)
- Expected payoff of next loan is $\frac{s_i+1}{s_i+d_i+2}$

$$p_1 = p_2 = \frac{1}{2}$$

0.55

0.33

$n_i = 1$
$s_i = 0$
$d_i = 1$

$n_i = 7$
$s_i = 4$
$d_i = 3$

Toy Model

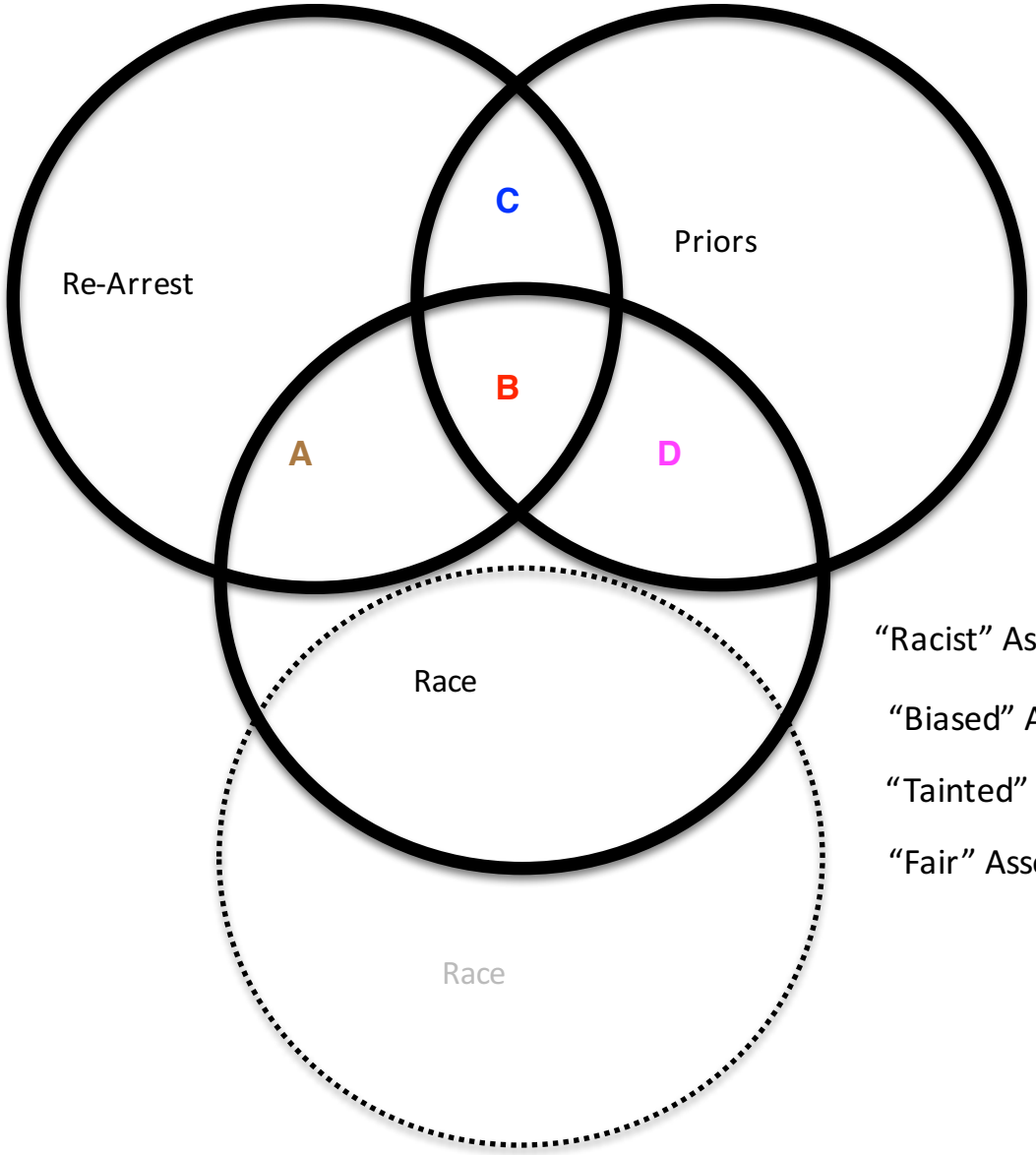
Upshot: Algorithms making myopically *optimal* decisions may forever discriminate against a qualified population because of unlucky prefix.

Less myopic algorithms balance *exploration* and *exploitation* – but this has its own unfairness issues.

$s_i = 0$ $\quad$ $s_i \approx x/2$

$d_i = 1$ $\quad$ $d_i \approx x/2$

A Venn Diagram for An Accuracy-Fairness Tradeoff



"Racist" Association = **A** + **B**

"Biased" Association = **A**

"Tainted" Association = **C** + **B**

"Fair" Association = **C**