Classification objectives

COMS 4771 Section 2 Fall 2021
Daniel Hsu
Classification Errors
Different types of classification errors may have different practical consequences

When errors are inevitable, how does one manage trade-offs?
Types of mistakes in binary classification

Types of mistakes:

- **False positive**: Predict $f(\vec{x}) = 1$ but true label is $y = 0$
- **False negative**: Predict $f(\vec{x}) = 0$ but true label is $y = 1$
Types of mistakes in binary classification

Types of mistakes:

- **False positive**: Predict $f(\vec{x}) = 1$ but true label is $y = 0$
- **False negative**: Predict $f(\vec{x}) = 0$ but true label is $y = 1$

Statistical model for future data tells us how often these mistakes are committed

- Outcome/label is a Bernoulli random variable $Y$
- Feature vector is a vector of $d$ random variables $\vec{X} := (X_1, \ldots, X_d)$
- Joint distribution of $(\vec{X}, Y)$ reflects the population of examples we anticipate encountering in the future for the present application
Types of mistakes in binary classification

Types of mistakes:

- **False positive**: Predict $f(\vec{x}) = 1$ but true label is $y = 0$
- **False negative**: Predict $f(\vec{x}) = 0$ but true label is $y = 1$

Statistical model for future data tells us how often these mistakes are committed

- Outcome/label is a Bernoulli random variable $Y$
- Feature vector is a vector of $d$ random variables $\vec{X} := (X_1, \ldots, X_d)$
- Joint distribution of $(\vec{X}, Y)$ reflects the population of examples we anticipate encountering in the future for the present application

- **False positive rate**: $\text{FPR}(f) := \Pr(f(\vec{X}) = 1 \mid Y = 0)$
- **False negative rate**: $\text{FNR}(f) := \Pr(f(\vec{X}) = 0 \mid Y = 1)$
Types of mistakes in binary classification

Types of mistakes:

- **False positive:** Predict $f(\vec{x}) = 1$ but true label is $y = 0$
- **False negative:** Predict $f(\vec{x}) = 0$ but true label is $y = 1$

Statistical model for future data tells us how often these mistakes are committed

- Outcome/label is a Bernoulli random variable $Y$
- Feature vector is a vector of $d$ random variables $\vec{X} := (X_1, \ldots, X_d)$
- Joint distribution of $(\vec{X}, Y)$ reflects the population of examples we anticipate encountering in the future for the present application

- **False positive rate:** $\text{FPR}(f) := \Pr(f(\vec{X}) = 1 \mid Y = 0)$
- **False negative rate:** $\text{FNR}(f) := \Pr(f(\vec{X}) = 0 \mid Y = 1)$

**Error rate:** $\text{err}(f) := \Pr(f(\vec{X}) \neq Y) = \Pr(Y = 0) \cdot \text{FPR}(f) + \Pr(Y = 1) \cdot \text{FNR}(f)$
Expected cost

Cost structure:

<table>
<thead>
<tr>
<th></th>
<th>$f(\vec{X}) = 0$</th>
<th>$f(\vec{X}) = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y = 0$</td>
<td>0</td>
<td>$c_{FP}$</td>
</tr>
<tr>
<td>$Y = 1$</td>
<td>$c_{FN}$</td>
<td>0</td>
</tr>
</tbody>
</table>

for some $c_{FP} > 0$ and $c_{FN} > 0$
Expected cost

Cost structure:

<table>
<thead>
<tr>
<th>$f(\vec{X}) = 0$</th>
<th>$f(\vec{X}) = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y = 0$</td>
<td>0</td>
</tr>
<tr>
<td>$Y = 1$</td>
<td>$c_{FP}$</td>
</tr>
</tbody>
</table>

for some $c_{FP} > 0$ and $c_{FN} > 0$

So expected cost of $f$ in statistical model is

$$
\mathbb{E}[\text{cost}(f)] = \Pr(f(\vec{X}) = 1 \text{ and } Y = 0) \cdot c_{FP} + \Pr(f(\vec{X}) = 0 \text{ and } Y = 1) \cdot c_{FN}
$$

$$
= \Pr(Y = 0) \cdot \text{FPR}(f) \cdot c_{FP} + \Pr(Y = 1) \cdot \text{FNR}(f) \cdot c_{FN}
$$
Structure of binary classifiers that minimize expected cost

**Question:** What are the predictions made by the classifier of *smallest expected cost* (according to cost structure on previous slide)?
Structure of binary classifiers that minimize expected cost

**Question:** What are the predictions made by the classifier of smallest expected cost (according to cost structure on previous slide)?

- For each possible feature vector $\vec{x}$, conditional distribution of $Y$ given $\vec{X} = \vec{x}$ is Bernoulli with “success probability” parameter that may be specific to $\vec{x}$:

$$ (Y | \vec{X} = \vec{x}) \sim \text{Bernoulli}(\eta(\vec{x})) $$

The $\vec{x}$-specific parameter $\eta(\vec{x})$ is a number between 0 and 1.
Structure of binary classifiers that minimize expected cost

**Question:** What are the predictions made by the classifier of smallest expected cost (according to cost structure on previous slide)?

- For each possible feature vector $\vec{x}$, conditional distribution of $Y$ given $\vec{X} = \vec{x}$ is Bernoulli with “success probability” parameter that may be specific to $\vec{x}$:

  $$(Y \mid \vec{X} = \vec{x}) \sim \text{Bernoulli}(\eta(\vec{x}))$$

  The $\vec{x}$-specific parameter $\eta(\vec{x})$ is a number between 0 and 1.

- Reasoning based on each possible prediction:

<table>
<thead>
<tr>
<th>Prediction upon $\vec{X} = \vec{x}$</th>
<th>Conditional expected cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$(1 - \eta(\vec{x})) \cdot c_{FP}$</td>
</tr>
<tr>
<td>0</td>
<td>$\eta(\vec{x}) \cdot c_{FN}$</td>
</tr>
</tbody>
</table>
Structure of binary classifiers that minimize expected cost

Question: What are the predictions made by the classifier of smallest expected cost (according to cost structure on previous slide)?

▶ For each possible feature vector $\vec{x}$, conditional distribution of $Y$ given $\vec{X} = \vec{x}$ is Bernoulli with “success probability” parameter that may be specific to $\vec{x}$:

$$(Y \mid \vec{X} = \vec{x}) \sim \text{Bernoulli}(\eta(\vec{x}))$$

The $\vec{x}$-specific parameter $\eta(\vec{x})$ is a number between 0 and 1

▶ Reasoning based on each possible prediction:

<table>
<thead>
<tr>
<th>Prediction upon $\vec{X} = \vec{x}$</th>
<th>Conditional expected cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$(1 - \eta(\vec{x})) \cdot c_{FP}$</td>
</tr>
<tr>
<td>0</td>
<td>$\eta(\vec{x}) \cdot c_{FN}$</td>
</tr>
</tbody>
</table>

▶ So, prediction that minimizes conditional expected cost given $\vec{X} = \vec{x}$ is:

$$f^*(\vec{x}) := 1\{\eta(\vec{x}) > \alpha\} \quad \text{where} \quad \alpha := \frac{c_{FP}}{c_{FP} + c_{FN}}$$
Leveraging cost structure in training

Plug-in approach

- Directly estimate $\eta(\vec{x}) = \Pr(Y = 1 \mid \vec{X} = \vec{x})$ (denote the estimate by $\hat{\eta}$) (This is a kind of regression problem!)
Leveraging cost structure in training

Plug-in approach

- Directly estimate \( \eta(\vec{x}) = \Pr(Y = 1 \mid \vec{X} = \vec{x}) \) (denote the estimate by \( \hat{\eta} \))
  (This is a kind of regression problem!)
- Construct classifier that thresholds the estimate \( \hat{\eta} \) at \( \alpha \)

\[
f(\vec{x}) := 1 \{ \hat{\eta}(\vec{x}) > \alpha \} \quad \text{where } \alpha := \frac{c_{FP}}{c_{FP} + c_{FN}}
\]

Modify training objective

Example:

- Original sum of logarithmic losses
  \[
  J(\vec{w}) = \sum_{(\vec{x}, y) \in S} \ell(\log(y, p_{\vec{w}}(\vec{x})))
  \]
- Instead, minimize weighted sum of logarithmic losses
  \[
  eJ(\vec{w}) = \sum_{(\vec{x}, y) \in S} c(y) \ell(\log(y, p_{\vec{w}}(\vec{x})))
  \]
  where \( c(0) := c_{FP} \) and \( c(1) := c_{FN} \), and construct classifier
  \[
f(\vec{x}) := 1 \{ p_{\vec{w}}(\vec{x}) > \frac{1}{2} \}
  \]
Leveraging cost structure in training

Plug-in approach

- Directly estimate \( \eta(x) = \Pr(Y = 1 \mid X = x) \) (denote the estimate by \( \hat{\eta} \))
  (This is a kind of regression problem!)
- Construct classifier that thresholds the estimate \( \hat{\eta} \) at \( \alpha \)

\[
f(x) := 1\{\hat{\eta}(x) > \alpha\}
\]

where \( \alpha := \frac{c_{FP}}{c_{FP} + c_{FN}} \)

Modify training objective

Example:

- Original sum of logarithmic losses

\[
J(\vec{w}) = \sum_{(\vec{x}, y) \in S} \ell_{\text{log}}(y, p_{\vec{w}}(\vec{x}))
\]
Leveraging cost structure in training

Plug-in approach
- Directly estimate \( \eta(\vec{x}) = \Pr(Y = 1 \mid \vec{X} = \vec{x}) \) (denote the estimate by \( \hat{\eta} \))
  (This is a kind of regression problem!)
- Construct classifier that thresholds the estimate \( \hat{\eta} \) at \( \alpha \)
  \[
  f(\vec{x}) := 1\{\hat{\eta}(\vec{x}) > \alpha\} \quad \text{where} \quad \alpha := \frac{c_{FP}}{c_{FP} + c_{FN}}
  \]

Modify training objective
Example:
- Original sum of logarithmic losses
  \[
  J(\vec{w}) = \sum_{(\vec{x},y) \in S} \ell_{\log}(y, p_{\vec{w}}(\vec{x}))
  \]
- Instead, minimize weighted sum of logarithmic losses
  \[
  \tilde{J}(\vec{w}) = \sum_{(\vec{x},y) \in S} c(y) \ell_{\log}(y, p_{\vec{w}}(\vec{x}))
  \]
  where \( c(0) := c_{FP} \) and \( c(1) := c_{FN} \), and construct classifier \( f(\vec{x}) := 1\{p_{\vec{w}}(\vec{x}) > 1/2\} \)
Calibration
What are semantics of the weather forecast “70% chance of rain”?

Probability calibration

Ideally, among all days where forecaster says “70% chance of rain”, should have:

- ≈ 70% with rain
- ≈ 30% with no rain

This property is called calibration
What are semantics of the weather forecast “70% chance of rain”?  

- Ideally, among all days where forecaster says “70% chance of rain”, should have:
  - \( \approx 70\% \) with rain
  - \( \approx 30\% \) with no rain
What are semantics of the weather forecast “70% chance of rain”? 

- Ideally, among all days where forecaster says “70% chance of rain”, should have:
  - $\approx 70\%$ with rain
  - $\approx 30\%$ with no rain

This property is called **calibration**

$$\Pr(Y = 1 \mid p(\vec{X})) \approx p(\vec{X})$$
How to get calibrated probability predictions?

**Direct approach**

- Directly estimate $\eta(\bar{x}) = \Pr(Y = 1 \mid \bar{X} = \bar{x})$
  
  (This is a kind of regression problem!)
How to get calibrated probability predictions?

**Direct approach**
- Directly estimate $\eta(\vec{x}) = \Pr(Y = 1 \mid \vec{X} = \vec{x})$
  
  (This is a kind of *regression* problem!)

**Post-processing approach**
- Start with (possibly un-calibrated) scoring function $s(\vec{x}) \ldots$ (e.g., $s(\vec{x}) = \vec{x} \cdot \vec{w}$)
  
  Then apply *calibration procedure*
Example from Foster & Stine, JASA 2004

- Horizontal axis: predicted probability of bankruptcy
- Vertical axis: actual proportion of bankruptcy

Before

After

(As judged on test data)
Some calibration procedures

Starting with (possibly un-calibrated) scoring function $s(\vec{x})$ . . . estimate $\Pr(Y = 1 \mid s(\vec{X}) = s(\vec{x}))$ (!)
Some calibration procedures

Starting with (possibly un-calibrated) scoring function $s(x)$ . . . estimate $Pr(Y = 1 \mid s(X) = s(x))$ (!)

▶ Platt scaling

▶ Estimate parameters $(m, \theta)$ of logistic regression model (with affine expansion) using $s(x)$ as scalar feature:

$$Pr_{(m, \theta)}(Y = 1 \mid s(X) = s(x)) = \text{logistic}(m \times s(x) + \theta)$$

$\rightarrow (\hat{m}, \hat{\theta})$

▶ Return $\hat{p}(x) := \text{logistic}(\hat{m} \times s(x) + \hat{\theta})$
Some calibration procedures

Starting with (possibly un-calibrated) scoring function $s(x)$... estimate $\Pr(Y = 1 \mid s(\tilde{X}) = s(x))$ (!)

- **Platt scaling**
  - Estimate parameters $(m, \theta)$ of logistic regression model (with affine expansion) using $s(x)$ as scalar feature:

  $$\Pr_{(m, \theta)}(Y = 1 \mid s(\tilde{X}) = s(x)) = \text{logistic}(m \times s(x) + \theta)$$

  $\rightarrow (\hat{m}, \hat{\theta})$

- Return $\hat{p}(x) := \text{logistic}(\hat{m} \times s(x) + \hat{\theta})$

- **Binning**
  - Divide range of $s(x)$ into several bins $B_1, B_2, \ldots$ (how many???)
  - Estimate $\Pr(Y = 1 \mid s(\tilde{X}) \in B_i)$ for each $i$ $\rightarrow (\hat{p}_1, \hat{p}_2, \ldots)$

- Return $\hat{p}(x) := \begin{cases} \hat{p}_1 & \text{if } s(x) \in B_1 \\ \hat{p}_2 & \text{if } s(x) \in B_2 \\ \vdots & \vdots \end{cases}$

Typically, use separate data for training $s(x)$ and for post-processing calibration.
Some calibration procedures

Starting with (possibly un-calibrated) scoring function $s(\vec{x})$ . . . estimate $\Pr(Y = 1 \mid s(\vec{X}) = s(\vec{x}))$ (!)

- **Platt scaling**
  - Estimate parameters $(m, \theta)$ of logistic regression model (with affine expansion) using $s(\vec{x})$ as scalar feature:
    \[
    \Pr_{(m,\theta)}(Y = 1 \mid s(\vec{X}) = s(\vec{x})) = \text{logistic}(m \times s(\vec{x}) + \theta)
    \]
    $\rightarrow (\hat{m}, \hat{\theta})$
  - Return $\hat{p}(\vec{x}) := \text{logistic}(\hat{m} \times s(\vec{x}) + \hat{\theta})$

- **Binning**
  - Divide range of $s(\vec{x})$ into several bins $B_1, B_2, \ldots$ (how many???)
  - Estimate $\Pr(Y = 1 \mid s(\vec{X}) \in B_i)$ for each $i \rightarrow (\hat{p}_1, \hat{p}_2, \ldots)$
  - Return $\hat{p}(\vec{x}) := \begin{cases} 
  \hat{p}_1 & \text{if } s(\vec{x}) \in B_1 \\
  \hat{p}_2 & \text{if } s(\vec{x}) \in B_2 \\
  \vdots & \vdots
  \end{cases}$

- . . . and many others

Typically, use separate data for training $s(\vec{x})$ and for post-processing calibration
Caveats

- Estimating $\eta(\vec{x}) = \Pr(Y = 1 \mid \vec{X} = \vec{x})$ can be harder than learning good classifier!
Caveats

- Estimating $\eta(\vec{x}) = \Pr(Y = 1 \mid \vec{X} = \vec{x})$ can be harder than learning good classifier!

  E.g., even if affine expansion is good enough for linear classifier, may need higher-degree polynomial expansion for good estimate of $\eta(\vec{x})$
Caveats

- Estimating $\eta(\vec{x}) = \Pr(Y = 1 | \vec{X} = \vec{x})$ can be harder than learning good classifier!

  E.g., even if affine expansion is good enough for linear classifier, may need higher-degree polynomial expansion for good estimate of $\eta(\vec{x})$

- Good calibration does not imply good classification!
Caveats

- Estimating $\eta(\vec{x}) = \Pr(Y = 1 \mid \vec{X} = \vec{x})$ can be harder than learning good classifier!

  E.g., even if affine expansion is good enough for linear classifier, may need higher-degree polynomial expansion for good estimate of $\eta(\vec{x})$

- Good calibration does not imply good classification!

  **Easiest way to get calibration:** Ignore $\vec{x}$; just always output your estimate $\hat{p}_0$ of $\Pr(Y = 1)$

    $$\hat{p}(\vec{x}) \equiv \hat{p}_0$$

  Ignoring $\vec{x}$ is usually a bad idea if you care about classification accuracy
Fairness: COMPAS Case Study
Use of ML models in decision-making: “data-driven solutions”
Use of ML models in decision-making: “data-driven solutions”

Why study “fairness”? 
Use of ML models in decision-making: “data-driven solutions”

Why study “fairness”?

- Automated decisions ⇒ potential for high rate of harm
Use of ML models in decision-making: “data-driven solutions”

Why study “fairness”? 

▶ Automated decisions ⇒ potential for high rate of harm
▶ Metrics-driven ⇒ potential for measurement / testing for harms
Use of ML models in decision-making: “data-driven solutions”

Why study “fairness”?

- Automated decisions ⇒ potential for high rate of harm
- Metrics-driven ⇒ potential for measurement / testing for harms
- Metrics-driven ⇒ potential to delude about non-harm
Disparate treatment of subgroups by ML classifiers

ML classifiers in high-stakes applications ⇒ Potential for (harmful) disparate treatment of subgroups
Disparate treatment of subgroups by ML classifiers

ML classifiers in high-stakes applications ⇒ Potential for (harmful) disparate treatment of subgroups

ProPublica study of ML classifier used in pre-trial detention (Angwin, Larson, Mattu, Kirchner, 2016)

- Judge must decide whether an arrested defendant should be released while awaiting trial
Disparate treatment of subgroups by ML classifiers

ML classifiers in high-stakes applications ⇒ Potential for (harmful) disparate treatment of subgroups

ProPublica study of ML classifier used in pre-trial detention (Angwin, Larson, Mattu, Kirchner, 2016)

- Judge must decide whether an arrested defendant should be released while awaiting trial
- Binary classification problem:
  - $\vec{X} =$ “features” of defendant, available at time of judge’s decision
  - $Y = \begin{cases} 1 & \text{if defendant will commit (violent) crime if released} \\ 0 & \text{otherwise} \end{cases}$

$FPR_0(f) = \Pr(f_{\text{COMPAS}}(\vec{X}) = 1 | Y = 0, A = 0)$

$FPR_1(f) = \Pr(f_{\text{COMPAS}}(\vec{X}) = 1 | Y = 0, A = 1)$

where $A$ is race attribute (Black = 0, White = 1)
Disparate treatment of subgroups by ML classifiers

ML classifiers in high-stakes applications ⇒ Potential for (harmful) disparate treatment of subgroups

ProPublica study of ML classifier used in pre-trial detention (Angwin, Larson, Mattu, Kirchner, 2016)

- Judge must decide whether an arrested defendant should be released while awaiting trial
- Binary classification problem:
  - $\vec{X} =$ “features” of defendant, available at time of judge’s decision
  - $Y = \begin{cases} 1 & \text{if defendant will commit (violent) crime if released} \\ 0 & \text{otherwise} \end{cases}$
- Studied classifier $f_{COMPAS}$ developed by company called Northpointe

Finding: Very different false positive rates for different subgroups defined by race
Disparate treatment of subgroups by ML classifiers

ML classifiers in high-stakes applications ⇒ Potential for (harmful) disparate treatment of subgroups

ProPublica study of ML classifier used in pre-trial detention (Angwin, Larson, Mattu, Kirchner, 2016)

- Judge must decide whether an arrested defendant should be released while awaiting trial
- Binary classification problem:
  - $\vec{X}$ = "features" of defendant, available at time of judge's decision
  - $Y = \begin{cases} 1 & \text{if defendant will commit (violent) crime if released} \\ 0 & \text{otherwise} \end{cases}$
- Studied classifier $f_{COMPAS}$ developed by company called Northpointe
- Finding: Very different false positive rates for different subgroups defined by race

\[
FPR_0(f) = \Pr(f_{COMPAS}(\vec{X}) = 1 \mid Y = 0, A = 0) \\
FPR_1(f) = \Pr(f_{COMPAS}(\vec{X}) = 1 \mid Y = 0, A = 1)
\]

where $A$ is race attribute (Black = 0, White = 1)
ProPublica’s analysis

Let $\hat{Y} := f_{\text{COMPAS}}(\vec{X})$

<table>
<thead>
<tr>
<th></th>
<th>Black defendants</th>
<th></th>
<th>White defendants</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$(A = 0)$</td>
<td>$\hat{Y} = 0$</td>
<td>$\hat{Y} = 1$</td>
</tr>
<tr>
<td>$Y = 0$</td>
<td>0.27</td>
<td>0.22</td>
<td>$Y = 0$</td>
</tr>
<tr>
<td>$Y = 1$</td>
<td>0.14</td>
<td>0.37</td>
<td>$Y = 1$</td>
</tr>
</tbody>
</table>
ProPublica’s analysis

Let $\hat{Y} := f_{\text{COMPAS}}(\vec{X})$

<table>
<thead>
<tr>
<th>Black defendants</th>
<th>$A = 0$</th>
<th>$\hat{Y} = 0$</th>
<th>$\hat{Y} = 1$</th>
<th>White defendants</th>
<th>$A = 1$</th>
<th>$\hat{Y} = 0$</th>
<th>$\hat{Y} = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y = 0$</td>
<td></td>
<td>0.27</td>
<td>0.22</td>
<td>$Y = 0$</td>
<td></td>
<td>0.46</td>
<td>0.14</td>
</tr>
<tr>
<td>$Y = 1$</td>
<td>0.14</td>
<td>0.37</td>
<td></td>
<td>$Y = 1$</td>
<td>0.19</td>
<td>0.21</td>
<td></td>
</tr>
</tbody>
</table>

False positive rates for each subgroup:

$$FPR_0(f) = \Pr(\hat{Y} = 1 \mid Y = 0, A = 0) = \frac{0.22}{0.27 + 0.22} = 45\%$$

$$FPR_1(f) = \Pr(\hat{Y} = 1 \mid Y = 0, A = 1) = \frac{0.14}{0.14 + 0.46} = 23\%$$
ProPublica’s analysis

Let \( \hat{Y} := f_{\text{COMPAS}}(X) \)

<table>
<thead>
<tr>
<th></th>
<th>( A = 0 )</th>
<th>( \hat{Y} = 0 )</th>
<th>( \hat{Y} = 1 )</th>
<th>( A = 1 )</th>
<th>( \hat{Y} = 0 )</th>
<th>( \hat{Y} = 1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Y = 0 )</td>
<td></td>
<td>0.27</td>
<td>0.22</td>
<td></td>
<td>0.46</td>
<td>0.14</td>
</tr>
<tr>
<td>( Y = 1 )</td>
<td>0.14</td>
<td>0.37</td>
<td></td>
<td>0.19</td>
<td>0.21</td>
<td></td>
</tr>
</tbody>
</table>

False positive rates for each subgroup:

\[
\text{FPR}_0(f) = \Pr(\hat{Y} = 1 \mid Y = 0, A = 0) = \frac{0.22}{0.27 + 0.22} = 45\%
\]

\[
\text{FPR}_1(f) = \Pr(\hat{Y} = 1 \mid Y = 0, A = 1) = \frac{0.14}{0.14 + 0.46} = 23\%
\]

Also rather unequal false negative rates between subgroups (27% vs 48%)
Northpointe’s analysis

Northpointe’s COMPAS classifier is based on thresholding a “riskiness score”

\[ f_{\text{COMPAS}}(\vec{x}) = 1\{\text{riskiness}(\vec{x}) > t\} \]

where \( \text{riskiness}(\vec{x}) \in [0, 1] \) is intended to estimate \( \Pr(Y = 1 | \vec{X} = \vec{x}) \)
Northpointe’s analysis

Northpointe’s COMPAS classifier is based on thresholding a “riskiness score”

\[ f_{\text{COMPAS}}(\vec{x}) = 1\{\text{riskiness}(\vec{x}) > t}\]

where \(\text{riskiness}(\vec{x}) \in [0, 1]\) is intended to estimate \(\Pr(Y = 1 | \vec{X} = \vec{x})\)

Northpointe shows that riskiness scores are (roughly) calibrated for each subgroup: For each \(r \in [0, 1]\),

\[
\Pr(Y = 1 | \text{riskiness}(\vec{X}) = r, A = 0) = r
\]

and \(\Pr(Y = 1 | \text{riskiness}(\vec{X}) = r, A = 1) = r\)

(i.e., riskiness scores have same probabilistic interpretation for both subgroups)
Theorem (Chouldechova, 2016; Kleinberg, Mullainathan, Raghavan, 2017). Unless

\[ \Pr(Y = 1 \mid A = 0) = \Pr(Y = 1 \mid A = 1), \]

or \( f(\vec{x}) := 1\{\text{riskiness}(\vec{x}) > t\} \) satisfies

\[ \text{FPR}(f) = \text{FNR}(f) = 0, \]

it is impossible to satisfy all of the following:

1. \( \text{FPR}_0(f) = \text{FPR}_1(f) \)
2. \( \text{FNR}_0(f) = \text{FNR}_1(f) \)
3. riskiness is calibrated for group \( A = 0 \)
4. riskiness is calibrated for group \( A = 1 \)
**Theorem (Chouldechova, 2016; Kleinberg, Mullainathan, Raghavan, 2017).** Unless

\[ \Pr(Y = 1 \mid A = 0) = \Pr(Y = 1 \mid A = 1), \]

or \( f(\vec{x}) := \mathbb{1}\{\text{riskiness}(\vec{x}) > t\} \) satisfies

\[ \text{FPR}(f) = \text{FNR}(f) = 0, \]

it is impossible to satisfy all of the following:

1. \( \text{FPR}_0(f) = \text{FPR}_1(f) \)
2. \( \text{FNR}_0(f) = \text{FNR}_1(f) \)
3. riskiness is calibrated for group \( A = 0 \)
4. riskiness is calibrated for group \( A = 1 \)

Even this narrow interpretation of the pre-trial detention decision problem reveals how domain expertise is **required** to evaluate a potential ML-based solution.
Recap

- Concerns in classification problems often go beyond error rate
- Potential for disparate treatment across subgroups is hazard of classification systems