Classification objectives

COMS 4721 Spring 2022
Daniel Hsu

Classification Errors

Motivation

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

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

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

Expected cost

Cost structure:

	$f(\vec{X}) = 0$	$f(\vec{X}) = 1$
$Y = 0$	0	c_{FP}
$Y = 1$	c_{FN}	0

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

Expected cost

Cost structure:

	$f(\vec{X}) = 0$	$f(\vec{X}) = 1$
$Y = 0$	0	c_{FP}
$Y = 1$	c_{FN}	0

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

So **expected cost** of f in statistical model is

$$\begin{aligned}\mathbb{E}[\mathrm{cost}(f)] &= \Pr(f(\vec{X}) = 1 \text{ and } Y = 0) \cdot c_{\mathrm{FP}} \;+\; \Pr(f(\vec{X}) = 0 \text{ and } Y = 1) \cdot c_{\mathrm{FN}} \\ &= \Pr(Y = 0) \cdot \mathrm{FPR}(f) \cdot c_{\mathrm{FP}} \;+\; \Pr(Y = 1) \cdot \mathrm{FNR}(f) \cdot c_{\mathrm{FN}}\end{aligned}$$

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 $\vec{x}$:

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

  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 $\vec{x}$:

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

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

- Reasoning based on each possible prediction:

Prediction upon $\vec{X} = \vec{x}$	Conditional expected cost
1	$(1 - \eta(\vec{x})) \cdot c_{\mathrm{FP}}$
0	$\eta(\vec{x}) \cdot c_{\mathrm{FN}}$

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)?

- 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:

Prediction upon $\vec{X} = \vec{x}$	Conditional expected cost
1	$(1 - \eta(\vec{x})) \cdot c_{\text{FP}}$
0	$\eta(\vec{x}) \cdot c_{\text{FN}}$

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

$$f^\star(\vec{x}) := \mathbb{1}\{\eta(\vec{x}) > \alpha\} \qquad \text{where } \alpha := \frac{c_{\text{FP}}}{c_{\text{FP}} + c_{\text{FN}}}$$

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 α

$$f(\vec{x}) := \mathbb{1}\{\hat{\eta}(\vec{x}) > \alpha\} \qquad \text{where } \alpha := \frac{c_{\mathrm{FP}}}{c_{\mathrm{FP}} + c_{\mathrm{FN}}}$$

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!)
- Construct classifier that thresholds the estimate $\hat{\eta}$ at α

$$f(\vec{x}) := \mathbb{1}\{\hat{\eta}(\vec{x}) > \alpha\} \qquad \text{where } \alpha := \frac{c_{\mathrm{FP}}}{c_{\mathrm{FP}} + c_{\mathrm{FN}}}$$

Modify training objective

Example:

- Original sum of logarithmic losses

$$J(\vec{w}) = \sum_{(\vec{x},y)\in\mathcal{S}} \ell_{\log}(y, p_{\vec{w}}(\vec{x}))$$

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 α

$$f(\vec{x}) := \mathbb{1}\{\hat{\eta}(\vec{x}) > \alpha\} \qquad \text{where } \alpha := \frac{c_{\mathrm{FP}}}{c_{\mathrm{FP}} + c_{\mathrm{FN}}}$$

Modify training objective

Example:

- Original sum of logarithmic losses

$$J(\vec{w}) = \sum_{(\vec{x},y)\in\mathcal{S}} \ell_{\log}(y, p_{\vec{w}}(\vec{x}))$$

- Instead, minimize *weighted* sum of logarithmic losses

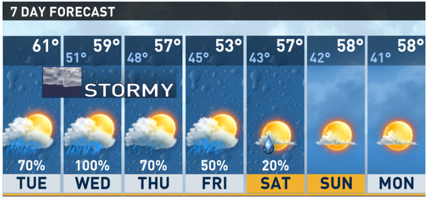
$$\widetilde{J}(\vec{w}) = \sum_{(\vec{x},y)\in\mathcal{S}} c(y)\, \ell_{\log}(y, p_{\vec{w}}(\vec{x}))$$

where $c(0) := c_{\mathrm{FP}}$ and $c(1) := c_{\mathrm{FN}}$, and construct classifier $f(\vec{x}) := \mathbb{1}\{p_{\vec{w}}(\vec{x}) > 1/2\}$

Calibration

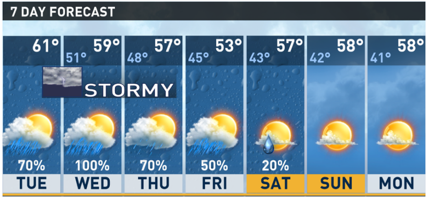
Probability calibration

What are semantics of the weather forecast “70% chance of rain”?



Probability calibration

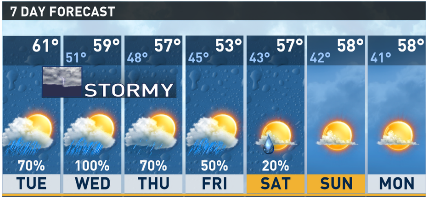
What are semantics of the weather forecast "70% chance of rain"?



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

Probability 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

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 $\eta(\vec{x}) = \Pr(Y = 1 \mid \vec{X} = \vec{x})$
  (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!)

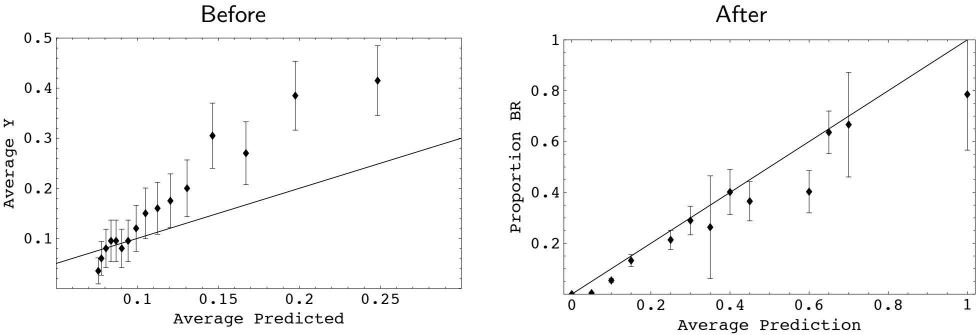
Post-processing approach

- Start with (possibly un-calibrated) scoring function $s(\vec{x})$ ... (e.g., $s(\vec{x}) = \vec{x} \cdot \vec{w}$)
  Then apply *calibration procedure*

Calibration example

Example from Foster & Stine, JASA 2004

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



(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 $s(\vec{x})$ ... estimate $\Pr(Y = 1 \mid s(\vec{X}) = s(\vec{x}))$ (!)

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

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

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

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

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, θ) of logistic regression model (with affine expansion) using $s(\vec{x})$ as scalar feature:

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

    $\rightarrow (\hat{m}, \hat{\theta})$
  - Return $\hat{p}(\vec{x}) := \mathrm{logistic}(\hat{m} \times s(\vec{x}) + \hat{\theta})$
- **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 $\eta(\vec{x}) = \Pr(Y = 1 \mid \vec{X} = \vec{x})$ 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!

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

Fairness in ML

Use of ML models in decision-making: "data-driven solutions"

Fairness in ML

Use of ML models in decision-making: "data-driven solutions"

Why study "fairness"?

Fairness in ML

Use of ML models in decision-making: "data-driven solutions"

Why study "fairness"?

- Automated decisions $\Rightarrow$ potential for high rate of harm

Fairness in ML

Use of ML models in decision-making: "data-driven solutions"

Why study "fairness"?

- Automated decisions $\Rightarrow$ potential for high rate of harm
- Metrics-driven $\Rightarrow$ potential for measurement / testing for harms

Fairness in ML

Use of ML models in decision-making: "data-driven solutions"

Why study "fairness"?

- Automated decisions $\Rightarrow$ potential for high rate of harm
- Metrics-driven $\Rightarrow$ potential for measurement / testing for harms
- Metrics-driven $\Rightarrow$ potential to delude about non-harm

Disparate treatment of subgroups by ML classifiers

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

Disparate treatment of subgroups by ML classifiers

ML classifiers in high-stakes applications $\Rightarrow$ 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 $\Rightarrow$ 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
  - $Y = \begin{cases} 1 & \text{if defendant will commit (violent) crime if released} \\ 0 & \text{otherwise} \end{cases}$

Disparate treatment of subgroups by ML classifiers

ML classifiers in high-stakes applications $\Rightarrow$ 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

Disparate treatment of subgroups by ML classifiers

ML classifiers in high-stakes applications $\Rightarrow$ 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

$$\mathrm{FPR}_0(f) = \Pr(f_{\text{COMPAS}}(\vec{X}) = 1 \mid Y = 0, A = 0)$$
$$\mathrm{FPR}_1(f) = \Pr(f_{\text{COMPAS}}(\vec{X}) = 1 \mid Y = 0, A = 1)$$

where A is race attribute (Black = 0, White = 1)

ProPublica's analysis

Let $\hat{Y} := f_{\textsc{compas}}(\vec{X})$

Black defendants

$(A = 0)$	$\hat{Y} = 0$	$\hat{Y} = 1$
$Y = 0$	0.27	0.22
$Y = 1$	0.14	0.37

White defendants

$(A = 1)$	$\hat{Y} = 0$	$\hat{Y} = 1$
$Y = 0$	0.46	0.14
$Y = 1$	0.19	0.21

ProPublica's analysis

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

Black defendants

$(A=0)$	$\hat{Y}=0$	$\hat{Y}=1$
$Y=0$	0.27	0.22
$Y=1$	0.14	0.37

White defendants

$(A=1)$	$\hat{Y}=0$	$\hat{Y}=1$
$Y=0$	0.46	0.14
$Y=1$	0.19	0.21

False positive rates for each subgroup:

$$\begin{aligned}
\text{FPR}_0(f) &= \Pr(\hat{Y}=1 \mid Y=0, A=0) = \frac{0.22}{0.27+0.22} = 45\% \\
\text{FPR}_1(f) &= \Pr(\hat{Y}=1 \mid Y=0, A=1) = \frac{0.14}{0.14+0.46} = 23\%
\end{aligned}$$

ProPublica's analysis

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

Black defendants

$(A=0)$	$\hat{Y}=0$	$\hat{Y}=1$
$Y=0$	0.27	0.22
$Y=1$	0.14	0.37

White defendants

$(A=1)$	$\hat{Y}=0$	$\hat{Y}=1$
$Y=0$	0.46	0.14
$Y=1$	0.19	0.21

False positive rates for each subgroup:

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

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

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"

$$f_{\textsc{compas}}(\vec{x}) = \mathbb{1}\{\mathrm{riskiness}(\vec{x}) > t\}$$

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

Northpointe's analysis

Northpointe's COMPAS classifier is based on thresholding a "riskiness score"

$$f_{\text{COMPAS}}(\vec{x}) = \mathbb{1}\{\text{riskiness}(\vec{x}) > t\}$$

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

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

$$\Pr(Y = 1 \mid \text{riskiness}(\vec{X}) = r, A = 0) = r$$
$$\text{and} \quad \Pr(Y = 1 \mid \text{riskiness}(\vec{X}) = r, A = 1) = r$$

(i.e., riskiness scores have same probabilistic interpretation for both subgroups)

Impossibility

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 $A = 1$

Impossibility

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