Calibration and bias

COMS 4771 Fall 2023

Predicting conditional probabilities

Example: Click prediction for online ads

- X = features of (user, advertisement) pair
- Y = indicator that user will click on ad
- $\Pr(Y = 1 \mid X = x)$ is almost always near zero, but useful to know this probability, e.g., to compare ads, estimate revenue

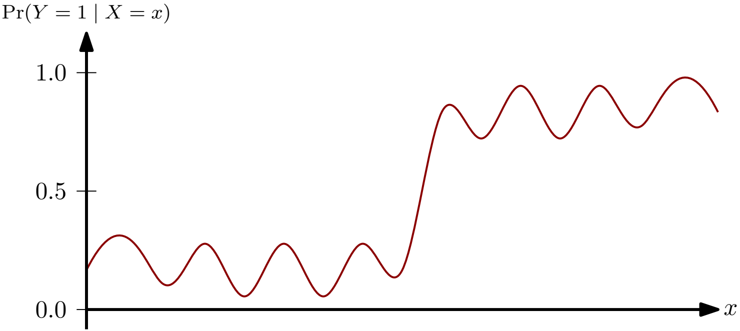
Example:

- If $\Pr(Y = 1 \mid X = x) \approx \Pr(Y = 0 \mid X = x)$, then perhaps classification mistake need not be counted

	Estimates $\Pr(Y = 1 \mid X = x)$
nearest neighbors	?
decision trees	?
generative models	$\checkmark$
logistic regression	$\checkmark$
Perceptron	no
SVM	no

Caution:

- Prediction/estimate of (conditional) probability is still a prediction
  - Some are accurate, some are inaccurate
  - Same goes for anything derived from these predictions
- At least as hard as learning to classify, and can be arbitrarily harder



(Please imagine a high-dimensional version of this picture)

Ultimately, need to validate accuracy of predictions of (conditional) probabilities

- **Challenge:** In many applications, only see one label y per feature vector x

Calibration

Prediction $\hat{p}(x)$ of $\Pr(Y = 1 \mid X = x)$ is (approximately) calibrated if

$$\Pr(Y = 1 \mid \hat{p}(X) = p) \approx p \quad \text{for all } p \in [0, 1]$$

Expected calibration error of $\hat{p}$ (assuming $\mathrm{range}(\hat{p})$ is finite set $\mathcal{P} \subset [0, 1]$):

$$\sum_{p \in \mathcal{P}} |\Pr(Y = 1 \wedge \hat{p}(X) = p) - p \times \Pr(\hat{p}(X) = p)|$$

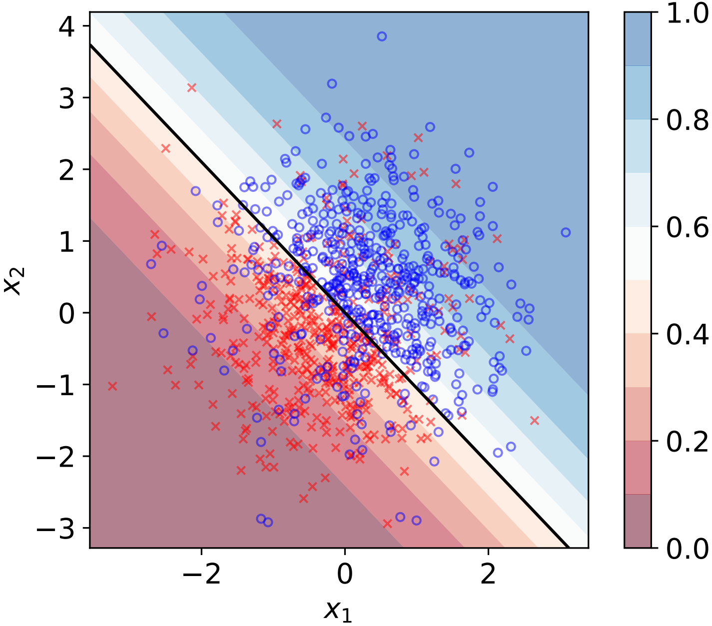
Possible to estimate this from test data if $\mathcal{P}$ is not too large

Synthetic example: $X = (X_1, X_2) \sim \mathrm{N}(0, I)$, and

$$\Pr(Y = 1 \mid X = x) = p^\star(x) = \begin{cases} 0.8 & \text{if } x_1 + x_2 > 0 \\ 0.2 & \text{otherwise} \end{cases}$$

Fit logistic regression model to 1000 training examples using MLE

- Error rate is 20.3%, which is nearly optimal
- However, expected calibration error of $\hat{p}$ is 0.13


x_2
x_1
4
3
2
1
0
−1
−2
−3
−2
0
2
1.0
0.8
0.6
0.4
0.2
0.0

Calibrating conditional probability predictions

Suppose you have real-valued "score" function $s \colon \mathbb{R}^d \to \mathbb{R}$

	Possible score $s(x)$
k-nearest neighbors	____________________
decision trees	______________________________
generative models	est. of $\Pr(Y = 1 \mid X = x)$
logistic regression	est. of $\Pr(Y = 1 \mid X = x)$
Perceptron	______
SVM	______

(many other possibilities)

Goal: obtain approximately calibrated predictor $\hat{p}(x)$ of $\Pr(Y = 1 \mid X = x)$

(Histogram) binning:

- Sort $s(x)$ from training/validation data into T bins
- Determine $T-1$ boundary values between the bins
- Let $\hat{p}^{(i)}$ be estimate of $\Pr(Y=1 \mid s(x) \in \text{bin } i)$
- Then define

$$\hat{p}(x) = \begin{cases} \hat{p}^{(1)} & \text{if } s(x) \text{ falls in bin } 1 \\ \hat{p}^{(2)} & \text{if } s(x) \text{ falls in bin } 2 \\ \vdots & \\ \hat{p}^{(T)} & \text{if } s(x) \text{ falls in bin } T \end{cases}$$

How can this possibly work?

- Key idea: score function turns problem into one with only a single feature
- No curse of dimension to worry about

Synthetic example: $X = (X_1, X_2) \sim \mathrm{N}(0, I)$, and

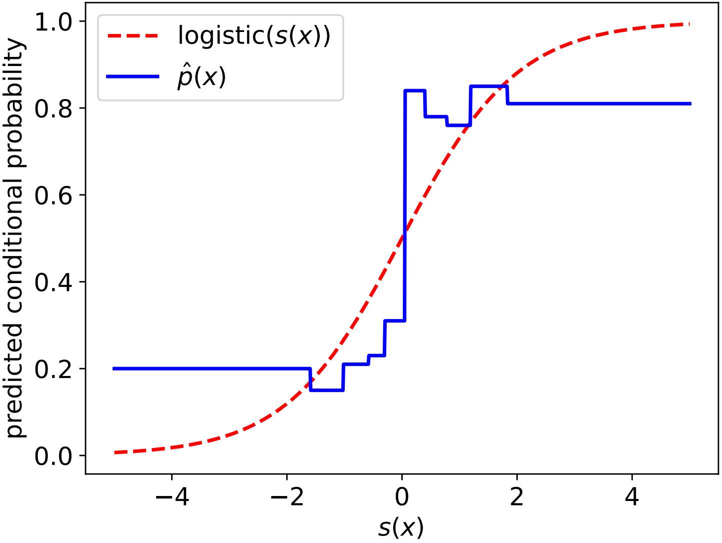
$$\Pr(Y = 1 \mid X = x) = p^\star(x) = \begin{cases} 0.8 & \text{if } x_1 + x_2 > 0 \\ 0.2 & \text{otherwise} \end{cases}$$

Fit logistic regression model to 1000 training examples using MLE

- **Apply binning to** $s(x) = \hat{w}^\mathsf{T} x$ (with $T = 10$ bins)
- Expected calibration error: 0.043 (down from 0.13)

Final predictor $\hat{p}(x)$:

range of $s(x)$	$\hat{p}(x)$
$s(x) < -1.591$	0.200
$-1.591 \leq s(x) < -1.024$	0.150
$-1.024 \leq s(x) < -0.578$	0.210
$-0.578 \leq s(x) < -0.296$	0.230
$-0.296 \leq s(x) < 0.055$	0.310
$0.055 \leq s(x) < 0.398$	0.840
$0.398 \leq s(x) < 0.777$	0.780
$0.777 \leq s(x) < 1.194$	0.760
$1.194 \leq s(x) < 1.835$	0.850
$1.835 \leq s(x)$	0.810



logistic(s(x))
p̂(x)
predicted conditional probability
0.0
0.2
0.4
0.6
0.8
1.0
−4
−2
0
2
4
s(x)

- Popular way to improve binning: enforce monotonicity (e.g., if you believe $\Pr(Y = 1 \mid s(x))$ is monotone in $s(x)$)

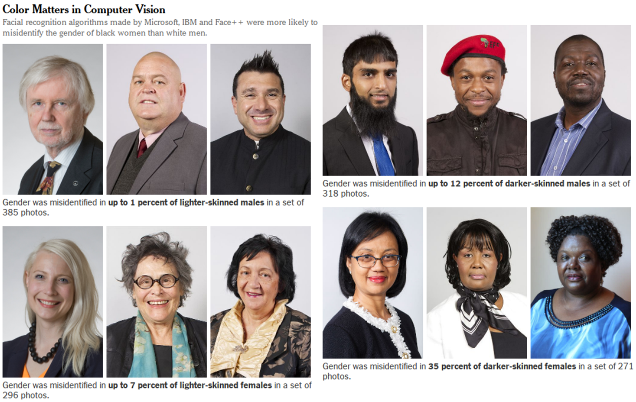
- Caution: a $\hat{p}$ with low expected calibration error does not necessarily give an accurate predict of Y from X
  - Only gives an accurate predictor of Y from $s(X)$
  - But perhaps $s(X)$ is constant!
  - In this case, suffices to predict the constant $\Pr(Y = 1)$

Calibration versus equalizing error rates

- Increasing use of predictive models in real-world applications (e.g., admissions, hiring, criminal justice)
- Do they offer "fair treatment" to individuals/groups?

Well-known example: **"Gender shades"** study (Buolamwini and Gebru, 2018)

- **Task:** predict gender from image of face
- **Major finding:** some commercial facial analysis software were less accurate for images of darker-skinned female individuals than for images of lighter-skinned male individuals



ProPublica **“Machine Bias”** study (Angwin et al, 2016)

- Judge needs to decide whether or not an arrested defendant should be released while awaiting trial
- Predictive model (“COMPAS”) predicts whether or not defendant will commit (violent) crime if released
- Study based data from Broward County, Florida argued that COMPAS treated black defendants unfairly in a certain sense

Setup for ProPublica study (highly simplified)

- X: feature vector specific to arrested defendant
- A: group membership attribute (e.g., race, sex, age; could be part of X)
- Y: outcome to predict (e.g., "will re-offend if released")
- $\hat{Y} = f_{\text{COMPAS}}(X)$: prediction of Y based on X
- For simplicity, assume $A, Y, \hat{Y}$ are all $\{0, 1\}$-valued

Types of errors:

- False positive rate: $\mathrm{FPR} = \Pr(\hat{Y} = 1 \mid Y = 0)$
- False negative rate: $\mathrm{FNR} = \Pr(\hat{Y} = 0 \mid Y = 1)$
- Per-group FPR and FNR: for each $a \in \{0, 1\}$,

$$\mathrm{FPR}_a = \Pr(\hat{Y} = 1 \mid Y = 0, A = a)$$
$$\mathrm{FNR}_a = \Pr(\hat{Y} = 0 \mid Y = 1, A = a)$$

Equalized odds: require that $\mathrm{FPR}_0 \approx \mathrm{FPR}_1$ and $\mathrm{FNR}_0 \approx \mathrm{FNR}_1$

- No group incurs errors (either type) at a higher rate than the other

ProPublica found: COMPAS software is very far from offering "equalized odds"

- $\mathrm{FPR}_0 = 45\%$, $\mathrm{FPR}_1 = 23\%$
- $\mathrm{FNR}_0 = 27\%$, $\mathrm{FNR}_1 = 48\%$

Response from Northpointe (creator of COMPAS)

- $f_{\mathrm{COMPAS}}(x) = \mathbb{1}\{\hat{p}(x) > t\}$ where $\hat{p}(x)$ is prediction of $\Pr(Y = 1 \mid X = x)$, and t is some suitable threshold parameter
- $\hat{p}$ approximately-calibrated, and also approximately-calibrated **for each group**

$$\Pr(Y = 1 \mid \hat{p}(X) = p, A = 0) \approx \Pr(Y = 1 \mid \hat{p}(X) = p, A = 1) \approx p$$

- So $\hat{p}$ has same probabilistic semantics for each group

Theorem (Chouldechova; Kleinberg-Mullainathan-Raghavan): Unless

$$\Pr(Y = 1 \mid A = 0) = \Pr(Y = 1 \mid A = 1) \quad \text{or} \quad \mathrm{FPR} = \mathrm{FNR} = 0,$$

it is impossible to simultaneously satisfy all of the following:

1. $\mathrm{FPR}_0 = \mathrm{FPR}_1$
2. $\mathrm{FNR}_0 = \mathrm{FNR}_1$
3. $\hat{p}$ is calibrated for group $A = 0$
4. $\hat{p}$ is calibrated for group $A = 1$

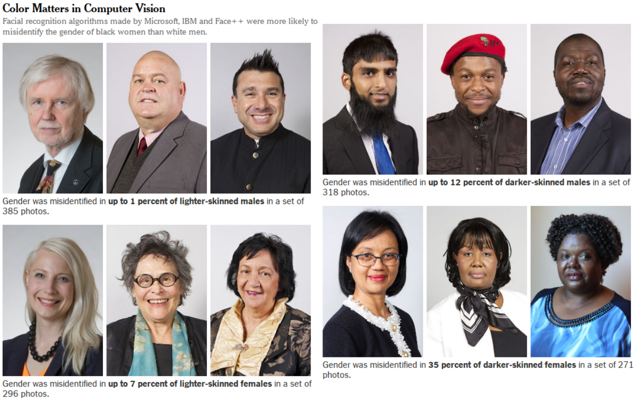
Distribution shift

Distribution shift (a.k.a. train/test mismatch, sample selection bias):

- Training data is sample from source distribution
- Care about (average) performance on data from target distribution
- Distribution shift: source $\neq$ target

Example: care about applying facial analysis software to images from general US population, but only train on images of light-skinned males

- Hardly any reason to expect things to work well . . .
- . . . unless you are "testing" only on images of light-skinned males



In many applications, training data is “dataset of convenience”

- Use whatever data you can get

All methods for addressing distribution shift require

- Either a lot of domain knowledge,
- Or additional data from target distribution
- (Often need both)

Example: re-weighting data

- Suppose you notice that, in training data,

$$\Pr(A = 0) \ll \Pr(A = 1)$$

  But you know that in target distribution, $A = 0$ and $A = 1$ equally often
- Use an importance weight of

$$\frac{1}{2\Pr(A = a)}$$

  for every example with $A = a$ in (empirical) expectation computations
- **Critical assumption:** conditional distribution of (X, Y) given A is the same in source and target; only marginal distribution of A differs

Importance-weighted test error rate

- Test data $(\tilde{X}^{(1)}, \tilde{Y}^{(1)}, \tilde{A}^{(1)}), \dots, (\tilde{X}^{(m)}, \tilde{Y}^{(m)}, \tilde{A}^{(m)}) \overset{\text{i.i.d.}}{\sim} (X, Y, A)$, from source distribution
- Define $p_a = \Pr(A = a)$ for each $a \in \{0, 1\}$
- Weighted test error rate:

$$\frac{1}{m} \sum_{i=1}^{m} \mathbb{1}\{f(\tilde{X}^{(i)}) \neq \tilde{Y}^{(i)}\} \times \frac{1}{2p_{\tilde{A}^{(i)}}}$$

Expected value of importance-weighted test error rate:

$$\mathbb{E}\left[\mathbb{1}\{f(X) \neq Y\} \times \frac{1}{2p_A}\right] = \underline{\hspace{10cm}}$$