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

<table>
<thead>
<tr>
<th>Estimates $\Pr(Y = 1 \mid X = x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>nearest neighbors</td>
</tr>
<tr>
<td>decision trees</td>
</tr>
<tr>
<td>generative models</td>
</tr>
<tr>
<td>logistic regression</td>
</tr>
<tr>
<td>Perceptron</td>
</tr>
<tr>
<td>SVM</td>
</tr>
</tbody>
</table>
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

![Graph showing $\Pr(Y = 1 \mid X = x)$]({attachment:graph.png})

(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 \textit{(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 range$(\hat{p})$ is finite set $\mathcal{P} \subset [0, 1]$):

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

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

---

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

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

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
Calibrating conditional probability predictions
Suppose you have real-valued “score” function \( s: \mathbb{R}^d \rightarrow \mathbb{R} \)

<table>
<thead>
<tr>
<th></th>
<th>Possible score ( s(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k )-nearest neighbors</td>
<td>[ \text{est. of } \Pr(Y = 1 \mid X = x) ]</td>
</tr>
<tr>
<td>decision trees</td>
<td>[ \text{est. of } \Pr(Y = 1 \mid X = x) ]</td>
</tr>
<tr>
<td>generative models</td>
<td>[ \text{est. of } \Pr(Y = 1 \mid X = x) ]</td>
</tr>
<tr>
<td>logistic regression</td>
<td>[ \text{est. of } \Pr(Y = 1 \mid X = x) ]</td>
</tr>
<tr>
<td>Perceptron</td>
<td>[ \text{est. of } \Pr(Y = 1 \mid X = x) ]</td>
</tr>
<tr>
<td>SVM</td>
<td>[ \text{est. of } \Pr(Y = 1 \mid X = x) ]</td>
</tr>
</tbody>
</table>

(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: $X = (X_1, X_2) \sim N(0, I)$, and

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

Fit logistic regression model to 1000 training examples using MLE

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

<table>
<thead>
<tr>
<th>range of $s(x)$</th>
<th>$\hat{p}(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s(x) &lt; -1.591$</td>
<td>0.200</td>
</tr>
<tr>
<td>$-1.591 \leq s(x) &lt; -1.024$</td>
<td>0.150</td>
</tr>
<tr>
<td>$-1.024 \leq s(x) &lt; -0.578$</td>
<td>0.210</td>
</tr>
<tr>
<td>$-0.578 \leq s(x) &lt; -0.296$</td>
<td>0.230</td>
</tr>
<tr>
<td>$-0.296 \leq s(x) &lt; 0.055$</td>
<td>0.310</td>
</tr>
<tr>
<td>$0.055 \leq s(x) &lt; 0.398$</td>
<td>0.840</td>
</tr>
<tr>
<td>$0.398 \leq s(x) &lt; 0.777$</td>
<td>0.780</td>
</tr>
<tr>
<td>$0.777 \leq s(x) &lt; 1.194$</td>
<td>0.760</td>
</tr>
<tr>
<td>$1.194 \leq s(x) &lt; 1.835$</td>
<td>0.850</td>
</tr>
<tr>
<td>$1.835 \leq s(x)$</td>
<td>0.810</td>
</tr>
</tbody>
</table>

![Graph showing predicted conditional probability]
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**: $FPR = \Pr(\hat{Y} = 1 \mid Y = 0)$
- **False negative rate**: $FNR = \Pr(\hat{Y} = 0 \mid Y = 1)$
- Per-group FPR and FNR: for each $a \in \{0, 1\}$,
  
  $FPR_a = \Pr(\hat{Y} = 1 \mid Y = 0, A = a)$
  
  $FNR_a = \Pr(\hat{Y} = 0 \mid Y = 1, A = a)$

**Equalized odds**: require that $FPR_0 \approx FPR_1$ and $FNR_0 \approx FNR_1$
- No group incurs errors (either type) at a higher rate than the other

**ProPublica found**: COMPAS software is very far from offering “equalized odds”
- $FPR_0 = 45\%$, $FPR_1 = 23\%$
- $FNR_0 = 27\%$, $FNR_1 = 48\%$
Response from Northpointe (creator of COMPAS)

- \( f_{COMPAS}(x) = 1 \{ \hat{p}(x) > t \} \) 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

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

it is impossible to simultaneously satisfy all of the following:

1. \( \text{FPR}_0 = \text{FPR}_1 \)
2. \( \text{FNR}_0 = \text{FNR}_1 \)
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 \((\tilde{X}^{(1)}, \tilde{Y}^{(1)}, \tilde{A}^{(1)}), \ldots, (\tilde{X}^{(m)}, \tilde{Y}^{(m)}, \tilde{A}^{(m)})\) 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:

$$\mathbb{E}\left[ 1\{f(X) \neq Y\} \times \frac{1}{2p_A} \right] = \frac{29}{29}$$