Bias and Fairness

Some ML issues in the real world

CMSC 422

MARINE CARPUAT
marine@cs.umd.edu

Many Cars Tone Deaf To Women's Voices

Female voices pose a bigger challenge for voice-activated technology than men's voices

Women have a tougher time using voice-command systems l...

http://www.autoblog.com/2011/05/31/women-voice-command-systems/

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

Prediction Fails Differently for Black Defendants

	WHITE	AFRICAN AMERICAN
Labeled Higher Risk, But Didn't Re-Offend	23.5%	44.9%
Labeled Lower Risk, Yet Did Re-Offend	47.7%	28.0%

Overall, Northpointe's assessment tool correctly predicts recidivism 61 percent of the time. But blacks are almost twice as likely as whites to be labeled a higher risk but not actually re-offend. It makes the opposite mistake among whites: They are much more likely than blacks to be labeled lower risk but go on to commit other crimes. (Source: ProPublica analysis of data from Broward County, Fla.)

https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Recall: Formal Definition of Binary Classification (from CIML)

TASK: BINARY CLASSIFICATION

Given:

1. An input space $\mathcal{X}$
2. An unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \{-1, +1\}$

Compute: A function f minimizing: $\mathbb{E}_{(x,y)\sim\mathcal{D}}\big[f(\boldsymbol{x}) \neq y\big]$

Train/Test Mismatch

- When working with real world data, training sample
  - reflects human biases
  - is influenced by practical concerns
    - e.g., what kind of data is easy to obtain

- Train/test distribution mismatch is frequent issue
  - aka covariate shift, sample selection bias, domain adaptation

Domain Adaptation

- What does it mean for 2 distributions to be related?
- When 2 distributions are related how can we build models that effectively share information between them?

Unsupervised adaptation

- **Goal:** learn a classifier f that achieves low expected loss under new distribution $\mathcal{D}^{new}$

- Given labeled training data from old distribution $\mathcal{D}^{\text{old}}$ $(x_1, y_1), \ldots, (x_N, y_N)$

- And unlabeled examples from new distribution $\mathcal{D}^{\text{new}}$: $z_1, \ldots, z_M$

Relation between test loss in new domain and old domain

$$
\begin{aligned}
&\text{test loss} && (8.1)\\
&= \mathbb{E}_{(x,y)\sim\mathcal{D}^{\text{new}}}\left[\ell(y, f(x))\right] && \text{definition} \quad (8.2)\\
&= \sum_{(x,y)} \mathcal{D}^{\text{new}}(x, y)\ell(y, f(x)) && \text{expand expectation} \quad (8.3)\\
&= \sum_{(x,y)} \mathcal{D}^{\text{new}}(x, y)\frac{\mathcal{D}^{\text{old}}(x, y)}{\mathcal{D}^{\text{old}}(x, y)}\ell(y, f(x)) && \text{times one} \quad (8.4)\\
&= \sum_{(x,y)} \mathcal{D}^{\text{old}}(x, y)\frac{\mathcal{D}^{\text{new}}(x, y)}{\mathcal{D}^{\text{old}}(x, y)}\ell(y, f(x)) && \text{rearrange} \quad (8.5)\\
&= \mathbb{E}_{(x,y)\sim\mathcal{D}^{\text{old}}}\left[\frac{\mathcal{D}^{\text{new}}(x, y)}{\mathcal{D}^{\text{old}}(x, y)}\ell(y, f(x))\right] && \text{definition} \quad (8.6)
\end{aligned}
$$

How can we estimate the ratio between Dnew and Dold?

Fixed base distribution

S = selection variable

$$\frac{\mathcal{D}^{\text{new}}(\boldsymbol{x},y)}{\mathcal{D}^{\text{old}}(\boldsymbol{x},y)} = \frac{\frac{1}{Z^{\text{new}}}\mathcal{D}^{\text{base}}(\boldsymbol{x},y)\,p(s=0 \mid \boldsymbol{x})}{\frac{1}{Z^{\text{old}}}\mathcal{D}^{\text{base}}(\boldsymbol{x},y)\,p(s=1 \mid \boldsymbol{x})} \quad \text{definition} \quad (8.9)$$

$$= \frac{\frac{1}{Z^{\text{new}}}p(s=0 \mid \boldsymbol{x})}{\frac{1}{Z^{\text{old}}}p(s=1 \mid \boldsymbol{x})} \quad \text{cancel base} \quad (8.10)$$

$$= Z\frac{p(s=0 \mid \boldsymbol{x})}{p(s=1 \mid \boldsymbol{x})} \quad \text{consolidate} \quad (8.11)$$

$$= Z\frac{1-p(s=1 \mid \boldsymbol{x})}{p(s=1 \mid \boldsymbol{x})} \quad \text{binary selection} \quad (8.12)$$

$$= Z\left[\frac{1}{p(s=1 \mid \boldsymbol{x})} - 1\right] \quad \text{rearrange} \quad (8.13)$$

We can estimate P(s=1|x) using a binary classifier!

Algorithm 23 $\textsc{SelectionAdaptation}(\langle(\boldsymbol{x}_n, y_n)\rangle_{n=1}^N, \langle \boldsymbol{z}_m \rangle_{m=1}^M, \mathcal{A})$

```
D^dist ← ⟨(x_n, +1)⟩_{n=1}^N ∪ ⟨(z_m, −1)⟩_{m=1}^M        // assemble data for distinguishing
                                                        // between old and new distributions
p̂ ← train logistic regression on D^dist
D^weighted ← ⟨(x_n, y_n, 1/p̂(x_n) − 1)⟩_{n=1}^N            // assemble weight classification
                                                        // data using selector
return A(D^weighted)                                    // train classifier
```

$D^{dist} \leftarrow \langle(\boldsymbol{x}_n, +1)\rangle_{n=1}^N \cup \langle(\boldsymbol{z}_m, -1)\rangle_{m=1}^M$ // assemble data for distinguishing // between old and new distributions

$\hat{p} \leftarrow$ train logistic regression on D^{dist}

$D^{weighted} \leftarrow \left\langle \left(\boldsymbol{x}_n, y_n, \frac{1}{\hat{p}(\boldsymbol{x}_n)} - 1\right) \right\rangle_{n=1}^N$ // assemble weight classification // data using selector

return $\mathcal{A}(D^{weighted})$ // train classifier

Supervised adaptation

- **Goal:** learn a classifier f that achieves low expected loss under new distribution $\mathcal{D}^{new}$

- Given labeled training data from old distribution $\mathcal{D}^{\text{old}}$ $\langle \boldsymbol{x}_n^{\text{(old)}}, y_n^{\text{(old)}} \rangle_{n=1}^{N}$

- And labeled examples from new distribution $\mathcal{D}^{\text{new}}: \langle \boldsymbol{x}_m^{\text{(new)}}, y_m^{\text{(new)}} \rangle_{m=1}^{M}$

One solution: feature augmentation

- Map inputs to a new augmented representation

$$
\begin{array}{rcllll}
 & & \text{shared} & \text{old-only} & \text{new-only} & \\
\boldsymbol{x}_n^{\text{(old)}} & \mapsto & \Big\langle\ \boldsymbol{x}_n^{\text{(old)}} & ,\ \boldsymbol{x}_n^{\text{(old)}} & ,\ \underbrace{0, 0, \ldots, 0}_{D\text{-many}} & \Big\rangle \\
\boldsymbol{x}_m^{\text{(new)}} & \mapsto & \Big\langle\ \boldsymbol{x}_m^{\text{(new)}} & ,\ \underbrace{0, 0, \ldots, 0}_{D\text{-many}} & ,\ \boldsymbol{x}_m^{\text{(new)}} & \Big\rangle
\end{array}
$$

One solution: feature augmentation

- Transform Dold and Dnew training examples
- Train a classifier on new representations
- Done!

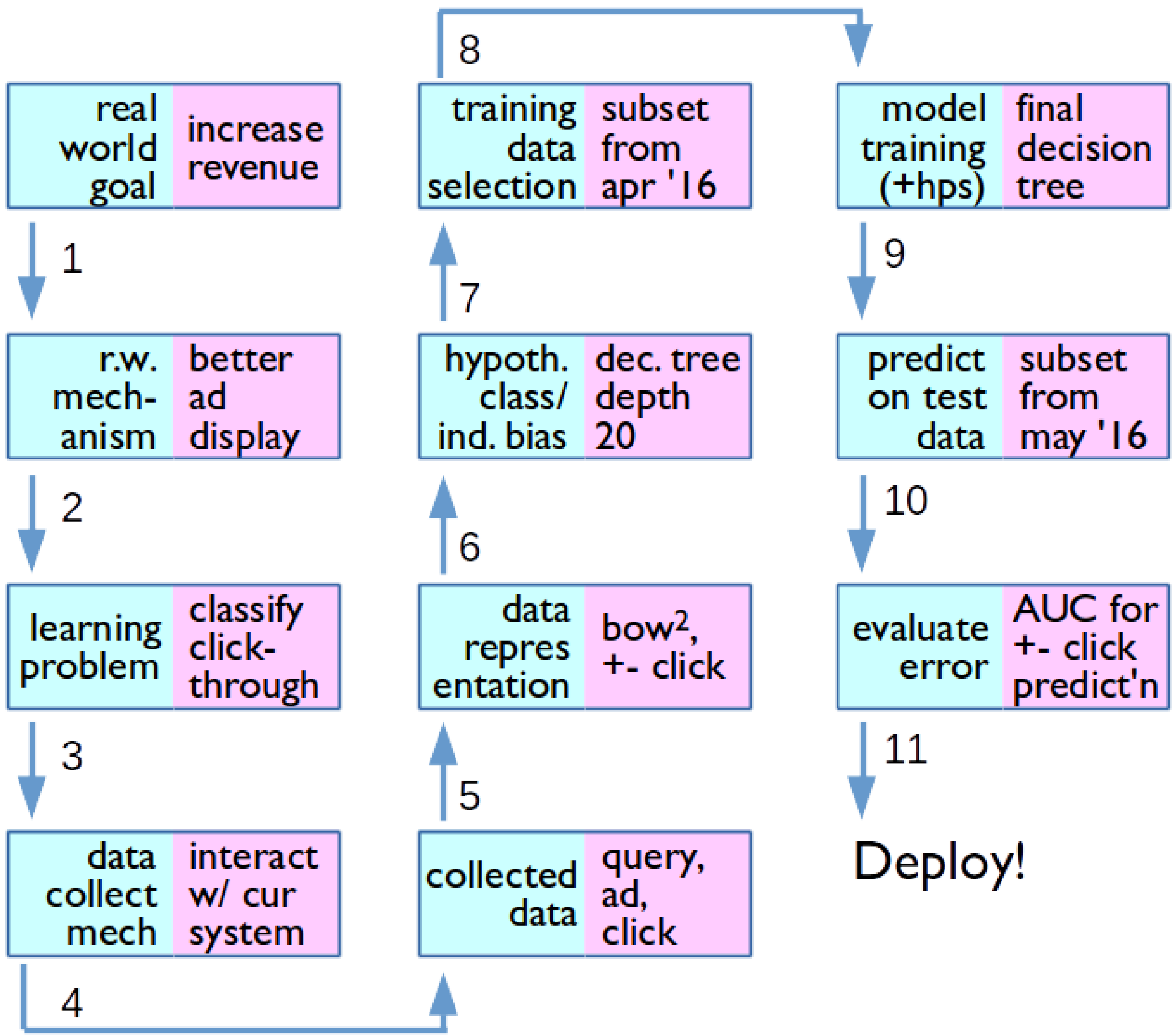
One solution: feature augmentation

- Adding instance weighting might be useful if N >> M
- Most effective when distributions are “not too close but not too far”
  - In practice, always try “old only”, “new only”, “union of old and new” as well!

Theorem 9 (Unsupervised Adaptation Bound). *Given a fixed representation and a fixed hypothesis space $\mathcal{F}$, let $f \in \mathcal{F}$ and let* $\epsilon^{(best)} = \min_{f^* \in \mathcal{F}} \frac{1}{2}\left[\epsilon^{(old)}(f^*) + \epsilon^{(new)}(f^*)\right]$*, then, for all $f \in \mathcal{F}$:*

$$\underbrace{\epsilon^{(new)}(f)}_{\text{error on } \mathcal{D}^{new}} \leq \underbrace{\epsilon^{(old)}(f)}_{\text{error on } \mathcal{D}^{old}} + \underbrace{\epsilon^{(best)}}_{\text{minimal avg error}} + \underbrace{d_{\mathcal{A}}(\mathcal{D}^{old}, \mathcal{D}^{new})}_{\text{distance}} \tag{8.27}$$

Typical Design Process for an ML Application



Bias is pervasive

– Bias in the labeling
– Sample selection bias
– Bias in choice of labels
– Bias in features or model structure
– Bias in loss function
– Deployed systems create feedback loops

ACM Code of Ethics

"To minimize the possibility of indirectly harming others, computing professionals must minimize malfunctions by following generally accepted standards for system design and testing. Furthermore, it is often necessary to assess the social consequences of systems to project the likelihood of any serious harm to others. If system features are misrepresented to users, coworkers, or supervisors, the individual computing professional is responsible for any resulting injury."

https://www.acm.org/about-acm/acm-code-of-ethics-and-professional-conduct

Bias and how to deal with it

- Train/test mismatch
- Unsupervised adaptation
- Supervised adaptation