Machine Learning (CSE 446): Bias and Fairness

Sham M Kakade

University of Washington
cse446-staff@cs.washington.edu

Training/Test Mismatch

1. x was an image of a tank; $y \in \{\text{Russian, American}\}$

Training/Test Mismatch

1. x was an image of a tank; $y \in \{\text{Russian, American}\}$ . . . actually learned “clear” vs. “blurry”

Training/Test Mismatch

1. x was an image of a tank; $y \in \{\text{Russian, American}\}$ ... actually learned "clear" vs. "blurry"
2. speech recognizers trained almost entirely on adult male speech

Training/Test Mismatch

1. x was an image of a tank; $y \in \{\text{Russian, American}\}$ ... actually learned "clear" vs. "blurry"
2. speech recognizers trained almost entirely on adult male speech ... performed badly for people who weren't men

Training/Test Mismatch

1. x was an image of a tank; $y \in \{\text{Russian, American}\}$ . . . actually learned "clear" vs. "blurry"
2. speech recognizers trained almost entirely on adult male speech . . . performed badly for people who weren't men
3. if we release a particular criminal, will they commit further crimes?

Training/Test Mismatch

1. x was an image of a tank; $y \in \{\text{Russian, American}\}$ ... actually learned "clear" vs. "blurry"
2. speech recognizers trained almost entirely on adult male speech ... performed badly for people who weren't men
3. if we release a particular criminal, will they commit further crimes? ... biased against racial minorities

Training/Test Mismatch

1. x was an image of a tank; $y \in \{\text{Russian, American}\}$ ... actually learned "clear" vs. "blurry"
2. speech recognizers trained almost entirely on adult male speech ... performed badly for people who weren't men
3. if we release a particular criminal, will they commit further crimes? ... biased against racial minorities
4. sentiment analysis: movie, restaurant, electronics reviews $\rightarrow$ political speech

Training/Test Mismatch

1. x was an image of a tank; $y \in \{\text{Russian, American}\}$ ... actually learned "clear" vs. "blurry"
2. speech recognizers trained almost entirely on adult male speech ... performed badly for people who weren't men
3. if we release a particular criminal, will they commit further crimes? ... biased against racial minorities
4. sentiment analysis: movie, restaurant, electronics reviews $\rightarrow$ political speech ... terrible performance

Training/Test Mismatch

1. x was an image of a tank; $y \in \{\text{Russian, American}\}$ ... actually learned "clear" vs. "blurry"
2. speech recognizers trained almost entirely on adult male speech ... performed badly for people who weren't men
3. if we release a particular criminal, will they commit further crimes? ... biased against racial minorities
4. sentiment analysis: movie, restaurant, electronics reviews $\rightarrow$ political speech ... terrible performance

Adaptation: what to do when you know your training and test data don't match?

Unsupervised Adaptation

$\mathcal{D}^{(\text{old})}$ is the distribution from which our labeled dataset $D^{(\text{old})} = \langle (x_n, y_n) \rangle_{n=1}^{N}$ is drawn.

$\mathcal{D}^{(\text{new})}$ is the distribution from which an unlabeled set $D^{(\text{new})} = \langle \breve{x}_m \rangle_{m=1}^{M}$ is drawn, and from which our **test data** are assumed to be drawn.

Reweighting

Let $\ell(x,y)$ be some loss function (true or surrogate).

$$
\begin{aligned}
\mathbb{E}_{(x,y)\sim\mathcal{D}^{(\text{new})}(x,y)}[\ell(x,y)] &= \sum_{x,y} \mathcal{D}^{(\text{new})}(x,y)\cdot\ell(x,y) \\
&= \sum_{x,y} \mathcal{D}^{(\text{new})}(x,y)\cdot\frac{\mathcal{D}^{(\text{old})}(x,y)}{\mathcal{D}^{(\text{old})}(x,y)}\cdot\ell(x,y) \\
&= \sum_{x,y} \mathcal{D}^{(\text{old})}(x,y)\cdot\frac{\mathcal{D}^{(\text{new})}(x,y)}{\mathcal{D}^{(\text{old})}(x,y)}\cdot\ell(x,y) \\
&= \mathbb{E}_{(x,y)\sim\mathcal{D}^{(\text{old})}(x,y)}\left[\frac{\mathcal{D}^{(\text{new})}(x,y)}{\mathcal{D}^{(\text{old})}(x,y)}\cdot\ell(x,y)\right]
\end{aligned}
$$

Reweighting

Let $\ell(x, y)$ be some loss function (true or surrogate).

$$\begin{aligned}
\mathbb{E}_{(x,y)\sim\mathcal{D}^{(\text{new})}(x,y)}[\ell(x,y)] &= \sum_{x,y} \mathcal{D}^{(\text{new})}(x,y) \cdot \ell(x,y) \\
&= \sum_{x,y} \mathcal{D}^{(\text{new})}(x,y) \cdot \frac{\mathcal{D}^{(\text{old})}(x,y)}{\mathcal{D}^{(\text{old})}(x,y)} \cdot \ell(x,y) \\
&= \sum_{x,y} \mathcal{D}^{(\text{old})}(x,y) \cdot \frac{\mathcal{D}^{(\text{new})}(x,y)}{\mathcal{D}^{(\text{old})}(x,y)} \cdot \ell(x,y) \\
&= \mathbb{E}_{(x,y)\sim\mathcal{D}^{(\text{old})}(x,y)}\left[\frac{\mathcal{D}^{(\text{new})}(x,y)}{\mathcal{D}^{(\text{old})}(x,y)} \cdot \ell(x,y)\right]
\end{aligned}$$

Challenge question: how to update SGD with *weighted* training examples?

Example Weights $\frac{\mathcal{D}^{(\text{new})}(x,y)}{\mathcal{D}^{(\text{old})}(x,y)}$

- Directly estimating the probabilities $\mathcal{D}$ is *really hard* (it's known as "density estimation").
- Instead, estimate the ratio.

Example Weights $\frac{\mathcal{D}^{(\text{new})}(x,y)}{\mathcal{D}^{(\text{old})}(x,y)}$

- Directly estimating the probabilities $\mathcal{D}$ is *really hard* (it's known as "density estimation").
- Instead, estimate the ratio.

Generative story for an (x, y) pair:

1. First, sample the pair from $\mathcal{D}^{(\text{base})}$.
2. Draw variable S, which ranges over $\{\text{old}, \text{new}\}$, according to $p(S \mid X = x)$.

Example Weights $\frac{\mathcal{D}^{(\text{new})}(x,y)}{\mathcal{D}^{(\text{old})}(x,y)}$

- Directly estimating the probabilities $\mathcal{D}$ is *really hard* (it's known as "density estimation").
- Instead, estimate the ratio.

Generative story for an (x, y) pair:

1. First, sample the pair from $\mathcal{D}^{(\text{base})}$.
2. Draw variable S, which ranges over $\{\text{old}, \text{new}\}$, according to $p(S \mid X = x)$.

This implies:

$$\mathcal{D}^{(\text{old})}(x, y) = \frac{\mathcal{D}^{(\text{base})}(x, y) \cdot p(S = \text{old} \mid X = x)}{\sum_{x', y'} \mathcal{D}^{(\text{base})}(x', y') \cdot p(S = \text{old} \mid X = x')}$$

$$\mathcal{D}^{(\text{new})}(x, y) = \frac{\mathcal{D}^{(\text{base})}(x, y) \cdot p(S = \text{new} \mid X = x)}{\sum_{x', y'} \mathcal{D}^{(\text{base})}(x', y') \cdot p(S = \text{new} \mid X = x')}$$

Example Weights $\frac{\mathcal{D}^{(\text{new})}(x,y)}{\mathcal{D}^{(\text{old})}(x,y)}$

- Directly estimating the probabilities $\mathcal{D}$ is *really hard* (it's known as "density estimation").
- Instead, estimate the ratio.

Generative story for an (x, y) pair:

1. First, sample the pair from $\mathcal{D}^{(\text{base})}$.
2. Draw variable S, which ranges over $\{\text{old}, \text{new}\}$, according to $p(S \mid X = x)$.

This implies:

$$\mathcal{D}^{(\text{old})}(x, y) \propto \mathcal{D}^{(\text{base})}(x, y) \cdot p(S = \text{old} \mid X = x)$$
$$\mathcal{D}^{(\text{new})}(x, y) \propto \mathcal{D}^{(\text{base})}(x, y) \cdot p(S = \text{new} \mid X = x)$$

$$\begin{aligned}\frac{\mathcal{D}^{(\mathsf{new})}(x,y)}{\mathcal{D}^{(\mathsf{old})}(x,y)} &\propto \frac{\mathcal{D}^{(\mathsf{base})}(x,y)\cdot p(\mathsf{new}\mid x)}{\mathcal{D}^{(\mathsf{base})}(x,y)\cdot p(\mathsf{old}\mid x)} \\ &= \frac{1-p(\mathsf{old}\mid x)}{p(\mathsf{old}\mid x)} \\ &= \frac{1}{p(\mathsf{old}\mid x)} - 1\end{aligned}$$

Unsupervised Adaptation Algorithm

Data: "old" data $\langle (x_n, y_n) \rangle_{n=1}^{N}$, "new" data $\langle \breve{x}_m \rangle_{m=1}^{M}$, learning algorithm $\mathcal{A}$ that takes a weighted training set

Result: classifier

$D^{(\text{distinguish})} = \langle (x_n, +1) \rangle_{n=1}^{N} \cup \langle (\breve{x}_m, -1) \rangle_{m=1}^{M}$;

train a probabilistic classifier $\hat{p}$ on $D^{(\text{distinguish})}$;

$D^{(\text{weighted})} = \left\langle \left(x_n, y_n, \frac{1}{\hat{p}(+1|x_n)} - 1 \right) \right\rangle_{n=1}^{N}$;

return $\mathcal{A}(D^{(\text{weighted})})$

Algorithm 1: SelectionAdaptation

Unsupervised Adaptation Algorithm

Data: "old" data $\langle (x_n, y_n) \rangle_{n=1}^{N}$, "new" data $\langle \breve{x}_m \rangle_{m=1}^{M}$, learning algorithm $\mathcal{A}$ that takes a weighted training set

Result: classifier

$D^{(\text{distinguish})} = \langle (x_n, +1) \rangle_{n=1}^{N} \cup \langle (\breve{x}_m, -1) \rangle_{m=1}^{M}$;

train a probabilistic classifier $\hat{p}$ on $D^{(\text{distinguish})}$;

$D^{(\text{weighted})} = \left\langle (x_n, y_n, \frac{1}{\hat{p}(+1|x_n)} - 1) \right\rangle_{n=1}^{N}$;

return $\mathcal{A}(D^{(\text{weighted})})$

Algorithm 2: SelectionAdaptation

Section 8.5 in **?** describes a theoretical result that makes conceptual use of something like $\hat{p}$.

Supervised Adaptation

"Old" labeled dataset $D^{(\mathsf{old})} = \langle (x_n, y_n) \rangle_{n=1}^{N}$.

"New" labeled dataset $D^{(\mathsf{new})} = \langle (\dot{x}_m, \dot{y}_m) \rangle_{m=1}^{M}$.

Test data is assumed to be from the same distribution as $D^{(\mathsf{new})}$.

Assume x_n is represented by $\mathbf{x}_n \in \mathbb{R}^d$ and $\dot{x}_m$ by $\dot{\mathbf{x}}_m \in \mathbb{R}^d$; the feature functions are the same.

Assume x_n is represented by $\mathbf{x}_n \in \mathbb{R}^d$ and $\dot{x}_m$ by $\dot{\mathbf{x}}_m \in \mathbb{R}^d$; the feature functions are the same.

Map:

$$\mathbf{x}_n \mapsto [\mathbf{x}_n; \mathbf{x}_n; \overbrace{0 \cdots 0}^{d \text{ zeroes}}]$$

Assume x_n is represented by $\mathbf{x}_n \in \mathbb{R}^d$ and $\dot{x}_m$ by $\dot{\mathbf{x}}_m \in \mathbb{R}^d$; the feature functions are the same.

Map:

$$\mathbf{x}_n \mapsto [\mathbf{x}_n; \mathbf{x}_n; \overbrace{0 \cdots 0}^{d \text{ zeroes}}]$$

Map:

$$\dot{\mathbf{x}}_m \mapsto [\dot{\mathbf{x}}_m; \overbrace{0 \cdots 0}^{d \text{ zeroes}}; \dot{\mathbf{x}}_m]$$

Data: "old" data $\langle(\mathbf{x}_n, y_n)\rangle_{n=1}^{N}$, "new" data $\langle\dot{\mathbf{x}}_m, \dot{y}_m\rangle_{m=1}^{M}$, learning algorithm $\mathcal{A}$
Result: classifier
$D = \langle([\mathbf{x}_n; \mathbf{x}_n; \mathbf{0}], y_n)\rangle_{n=1}^{N} \cup \langle([\dot{\mathbf{x}}_m; \mathbf{0}; \dot{\mathbf{x}}_m], \dot{y}_m)\rangle_{m=1}^{M}$;
return $\mathcal{A}(D)$

Algorithm 3: FeatureAugmentationAdaptation

Notes

- It may be a good idea to up-weight "new" data, especially if $N \gg M$.

Notes

- It may be a good idea to up-weight "new" data, especially if $N \gg M$.
- You can combine selection adaptation (first, on untransformed data) with feature augmentation.

Notes

- It may be a good idea to up-weight "new" data, especially if $N \gg M$.
- You can combine selection adaptation (first, on untransformed data) with feature augmentation.
- Always check these two baselines:
  1. train on union of all data (will work best if old and new are actually pretty close)
  2. train only on "new" data (will work best if old data is so distant as to be useless)

References I