Classification 1: Features, Perceptron

Announcements
- AO solutions
- AI released
- Polls from lec 1
- Lecture notes on course website (this PDF + typed note)
- Book notation diverges from lectures
- Greg OH’s Change

Social impact:
- generate hate speech
- Surveillance
- generate fake news/science
- deepfakes
- resume parser
- anything on a biased dataset
- people put trust in flawed systems (mistranslations, etc.)
Classification Points \( \mathbf{x} \in \mathbb{R}^n \) \( f: \) feature extractor

Label \( y \in \{-1, +1\} \)

Classifier maps \( \mathbf{x} \rightarrow y \)

Linear classifier: weight vector \( \mathbf{w} \in \mathbb{R}^n \)

Decision rule:
\[
\mathbf{w}^T f(\mathbf{x}) > 0 \quad \text{or} \quad \mathbf{w}^T f(\mathbf{x}) < 0
\]

\( \mathbf{w}^T f(\mathbf{x}) = 0 \)

```
\begin{figure}
\centering
\includegraphics[width=\textwidth]{linear_classifier_diagram.png}
\end{figure}
```

\( \mathbf{w}^T f(\mathbf{x}_1) > 0 \)

\( \mathbf{w}^T f(\mathbf{x}_2) < 0 \)

Decision boundary

```
\begin{align*}
\mathbf{w}^T f(\mathbf{x}) &> 0 \\
\mathbf{w}^T f(\mathbf{x}) &< 0
\end{align*}
```
Sentiment

$x = \text{the movie was great! would watch again!}$

1) Feature extraction

\[ x \Rightarrow f(x) \]

string $\mathbb{R}^n$

2) Learning

\[ \{(f(x^{(i)}), y^{(i)})\}_{i=1}^D \Rightarrow \text{classifier } \overline{w} \]

$D$ labeled examples

\[ (\overline{x}^{(i)}, y^{(i)}) \]

string $\pm 1$ \[ [-1.2, +2.7, \ldots] \]
Feature Extraction

\( \overline{x} = \text{the movie was great} \)

Bag-of-words featurization

\( \overline{x} \Rightarrow f(\overline{x}) \) if sent \( \overline{x} \) contains that word

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 1 & 0 & \cdots & 1 & 0 & \cdots & 1 & 0 & \cdots & 1
\end{bmatrix}
\]

10,000-dim

\( f(\overline{x}) \in \mathbb{R} \)

1001 vocab size

What are the weights?

\( \overline{w} = \begin{bmatrix}
\ldots & +3.7 & \ldots & -2.1 & \ldots
\end{bmatrix}
\)

great

terrible

posn 1124

(cleaned in a bit)
Preprocessing

① Vocab selection: what words are in the vector space?
(common words: 10K or so)

② Tokenization
   if a word is unseen, replace with UNK

was great! [ 1 ... 1 -- 0 ]
was great! great
was great! after tokenization

Tokenization: split on whitespace
   split out punctuation
   handle hyphenated words

② Stop word filtering: throw out "a" / "the" / ...

③ Lowercasing
So far: unigram bag-of-words

Bigram bag-of-words

\[
\begin{bmatrix}
1 & 0 & 1 & \ldots \\
\end{bmatrix}
\]

the movie the fish was great

Unigram: 10K words

Bigram: \((10K)^2\) \(\Rightarrow\) maybe \(-1M\)
pairs observed

To manage the features:
Maintain an index

\[
\begin{bmatrix}
\text{the} : 0 \\
\text{a} : 1 \\
\vdots \\
\text{movie} : 47 \\
\vdots \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
\vdots \\
\text{was great: 742} \\
\vdots \\
\end{bmatrix}
\]
Machine Learning

Parameters $\overline{w}$ to optimize to fit some training data $D$

$(x^{(i)}, y^{(i)})_{i=1}^{10k}$ unigrams in Bow

Search over $\mathbb{R}^{10k}$ for the best $\overline{w}$

Need a training objective

Objective: $\text{loss} (\text{dataset})$

$\text{loss} = \sum_{i=1}^{D} \text{loss} (x^{(i)}, y^{(i)}, \overline{w})$

"if we use $\overline{w}$ as our weights, how badly do we screw up?"
Stochastic gradient descent

for $t$ in range (0, epochs)
for $i$ in range (0, D)

$\overline{w} \leftarrow \overline{w} - \alpha \cdot \frac{2}{2w} \text{loss} \left( \overline{x}^{(i)}, y^{(i)}; \overline{w} \right)$

update $\overline{w}$ by subtracting gradient of the loss
Perceptron

Initialize $\overline{w} = 0$

for $t$ in range (0, epochs)

for $i$ in range (0, D)

$Y_{pred} = \begin{cases} 1 & \text{if } \overline{w} + f(\overline{x}(i)) > 0 \\ -1 & \text{else} \end{cases}$

$\overline{w} = \begin{cases} \overline{w} & \text{if } Y_{pred} = Y(i) \\ \overline{w} + \alpha f(\overline{x}(i)) & \text{if } Y(i) = +1 \\ \overline{w} - \alpha f(\overline{x}(i)) & \text{if } Y(i) = -1 \end{cases}$

Say $\alpha = 1$ for now

At the end: output $\overline{w}$ as our weights
Why make this update? 
\[ \overrightarrow{\mathbf{w}} + f(\overrightarrow{x}^{(i)}) \]

Suppose we have:
\[ \overrightarrow{\mathbf{w}}^T f(\overrightarrow{x}^{(i)}) \Rightarrow -1.3 \quad y^{(i)} = +1 \quad \text{wrong} \]

After update:
\[ (\overrightarrow{\mathbf{w}} + f(\overrightarrow{x}^{(i)}))^T f(\overrightarrow{x}^{(i)}) \] is now larger

\[ \text{dot product of } f(\overrightarrow{x}^{(i)}) \text{ with itself is positive} \]
good \ bad \ not

good = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}

not good = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix}

bad = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}

\vec{w} \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}

\vec{w}^T f(\vec{x}) > 0 \Rightarrow \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}