CS 378 Lecture 2
Classification 1: Features, Perceptron

credit: Machine Learning Memes on Facebook

Announcements
- A0 solutions
- A1 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 $\bar{x} \in \mathbb{R}^n$ f: feature extractor

$f(\bar{x}) \in \mathbb{R}^n$

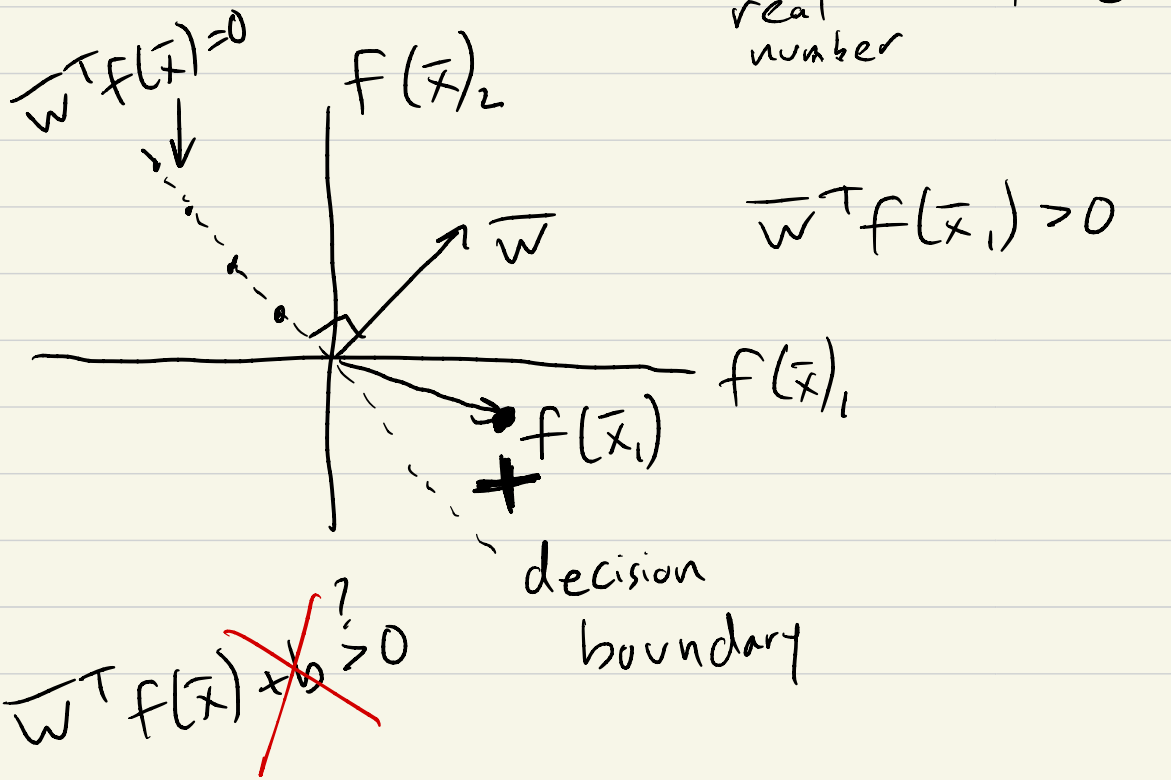
Label $y \in \{-1, +1\}$

Classifier maps $\bar{x} \to y$

Linear classifier: weight vector $\bar{w} \in \mathbb{R}^n$

θ

Decision rule: $\bar{w}^T f(\bar{x}) \overset{?}{>} 0$ $\Rightarrow$ $\underbrace{\bar{w} \cdot f(\bar{x})}_{\text{real number}}$ compare to 0



Sentiment

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

(1) Feature extraction

$\bar{x} \Rightarrow f(\bar{x})$

string $\quad \mathbb{R}^n$

(2) Learning

training set

$\{(f(\bar{x}^{(i)}), y^{(i)})\}_{i=1}^{D} \Rightarrow$ Classifier $\bar{w}$

D labeled examples

$(\bar{x}^{(i)}, y^{(i)})$

string $\quad +1/-1$

$[-1.2, +2.7, \ldots]$

Feature Extraction

$\bar{x}$ = the movie was great

Bag-of-words featurization

$\bar{x} \Rightarrow f(\bar{x})$ (count of the word?)

1 if sent $\bar{x}$ contains that word

$$[\ 1\ \ 0\ \ 0\ \ 0 \cdots\ 1\ \ 0 \cdots\ 1\ \ 0 \cdots 1\ ]$$

vocab the a of ... movie ... great ... was ...

10,000-dim

$f(\bar{x}) \in \mathbb{R}^{|V|}$ |V| vocab size

What are the weights? (learned in a bit)

$$\bar{w} = [\ \ldots\ +3.7\ \ldots\ -2.1\ \ldots\ ]$$

great terrible

posn 1124

Preprocessing

0. Vocab selection: what words are in the vector space? (common words: 10K or so)
   - ↘ if a word is unseen, replace with UNK
1. Tokenization

was great! [1 -- 1 --- 0]
was great! great

was great ! after tokenization

Tokenization: split on whitespace
split out punctuation
handle hyphenated words

2. Stopword filtering: throw out "a", "the", ...
3. Lowercasing

So far: unigram bag-of-words

Bigram bag-of-words

[1 0 1 ...]

the movie the fish was great

Unigram: 10K words

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

To manage the features: maintain an index

[the: 0
a: 1
⋮
movie: 47
⋮]

[⋮
was great: 742
⋮]

Machine Learning

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

$$\left(\bar{x}^{(i)}, y^{(i)}\right)_{i=1}^{D}$$

10K unigrams in BoW

Search over $\mathbb{R}^{10K}$ for the best $\bar{w}$

Need a training objective

objective: loss(dataset)

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

"if we use $\bar{w}$ as our weights, how badly do we screw up?"

Stochastic gradient descent

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

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

α ≈ step size

$\frac{\partial}{\partial \vec{w}}$ partial derivative

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

Perceptron

Initialize $\bar{w} = \bar{0}$

for t in range(0, epochs)

  for i in range(0, D)

$$y_{pred} \leftarrow \begin{cases} 1 & \text{if } \bar{w}^T f(\bar{x}^{(i)}) > 0 \\ -1 & \text{else} \end{cases}$$

$$\bar{w} \leftarrow \begin{cases} \bar{w} & \text{if } y_{pred} = y^{(i)} \quad \text{(correct)} \\ \bar{w} + \alpha f(\bar{x}^{(i)}) & \text{if } y^{(i)} = +1 \\ \bar{w} - \alpha f(\bar{x}^{(i)}) & \text{if } y^{(i)} = -1 \end{cases}$$

Say $\alpha = 1$ for now

At the end: output $\bar{w}$ as our weights

Why make this update?

$\bar{w} + f(\bar{x}^{(i)})$

Suppose we have:

$\bar{w}^T f(\bar{x}^{(i)}) \Rightarrow -1.3$ $\quad y^{(i)} = +1$

wrong

After update:

$\left(\bar{w} + f(\bar{x}^{(i)})\right)^T f(\bar{x}^{(i)})$ is now larger

dot product of $f(\bar{x}^{(i)})$ w/ itself is positive

good bad not

good = [1 0 0]

not good = [1 0 1]

bad = [0 1 0]

$\bar{w}$ [0 0 0] $\bar{w}$

$\bar{w}^T f(\bar{x}) \overset{?}{>} 0 \Rightarrow$ [1 0 0]