CS 378 Lecture 2
Classification 1: Features, Perceptron

credit: Machine Learning Memes on Facebook

Announcements

- A1 released (due 9/8)
- Book notation diverges from lectures

Today
- Classification (linear, binary)
- Feature extraction
- ML basics + Perceptron

Classification

Points $\bar{x} \in \mathbb{R}^n$

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

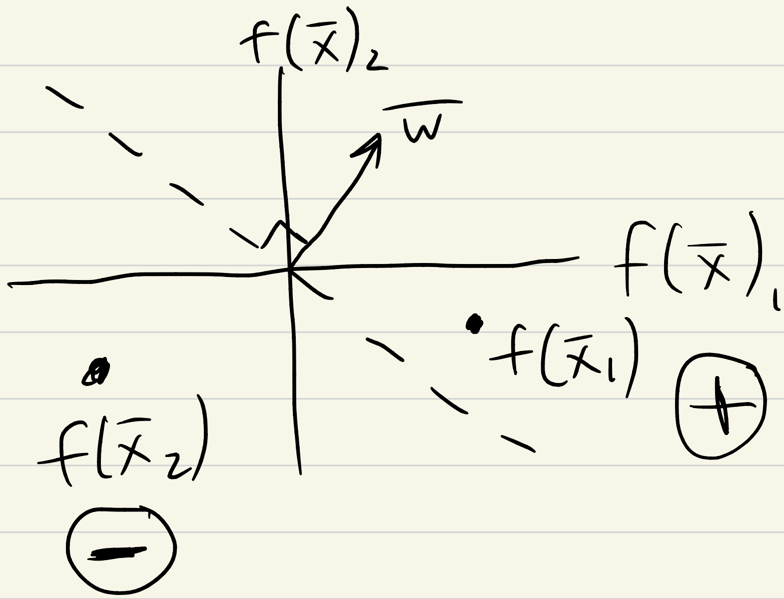
Label $y \in \{-1, +1\}$

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

Linear classifier: represented by a weight vector $\overline{w} \in \mathbb{R}^n$

Decision rule: $\overline{w}^T f(\overline{x}) \overset{?}{>} 0$

$\underbrace{\overline{w} \cdot f(\overline{x})}_{\text{a real number}}$ is greater than 0?



Sentiment Analysis

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

① Feature extraction

$$\underset{\text{string}}{\overline{x}} \Rightarrow \underset{\mathbb{R}^n}{f(\overline{x})}$$

② Learning

training set

$$\left\{ \left(f(\overline{x}^{(i)}), y^{(i)} \right) \right\}_{i=1}^{D} \Rightarrow \underset{\in \mathbb{R}^n}{\overline{w}}$$

D labeled examples

Feature Extraction

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

Bag-of-words featurization

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 1 & 1 & \cdots & 1 \end{bmatrix}$$
the a of ... movie great ... was ...

Vocabulary of ~10,000 words

value is the count of that word in $\bar{x}$

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

$$\begin{bmatrix} -0.1 & +0.2 & \cdots & +10 \end{bmatrix}$$
the a ... great

Preprocessing

⓪ Vocab selection: need a fixed set of words for the vector space

replace unseen words w/ UNK

① Tokenization

wasn't great! [1 (wasn't) 1 (great!) ... 0 (great)]

typical tokenization:

- break out punctuation
- break out contractions

was n't great !

② Stopword filtering:

- prepositions
- a, the
- pronouns (maybe for debiasing)

③ Lowercasing / Stemming

Fix typos?

So far: <u>unigram</u> BoW

Bigram BoW

[1 1 0 ...]

the movie | movie was | not good

"vocab" = vocab^2

Unigram: 10K   Bigram:

$(10K)^2$ in theory

1M in practice

Maintain an <u>index</u>

[the: 0
a: 1

movie: 47]

Can combine:

[the: 0
⋮
was great: 1172
⋮]

Machine Learning

Optimize parameters $\overline{w}$ to fit some training data

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

Find the best $\overline{w} \in \mathbb{R}^n$

Training objective: loss(dataset)

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

"if we use $\overline{w}$ as our weights, how badly do we screw up ex. (i)"

(sample i)

Stochastic gradient descent

for t in range(0, epochs)

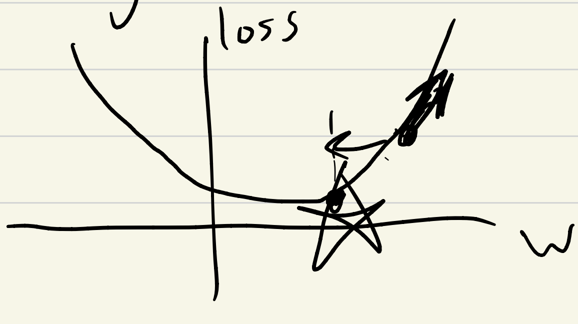
  for i in range(0, D)

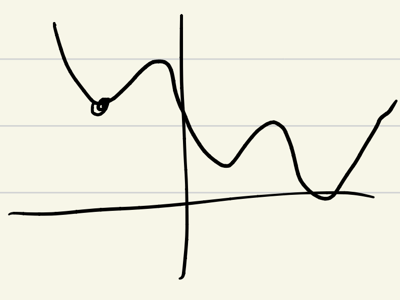
$$\overline{w} \leftarrow \overline{w} - \alpha \cdot \frac{\partial}{\partial \overline{w}} \text{loss}(\overline{x}^{(i)}, y^{(i)}, \overline{w})$$

α ↑ step size

≈ 1

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





Perceptron (instance of SGD)

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

for t in range(0, epochs)

for i in range(0, D) (shuffle exs each epoch)

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

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

Let $\alpha = 1$ for now

$\bar{w}$

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

Suppose

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

After update:

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

$\bar{w}^T f(\bar{x}^{(i)}) + \underbrace{f(\bar{x}^{(i)})^T f(\bar{x}^{(i)})}_{>0}$

larger than -1.3