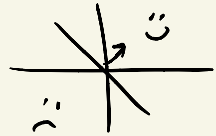
CS 371N Lecture 2



Classification 1: Features, Perceptron

Announcements

- A1 released, due in 2 weeks
- Reading notation != lecture notation

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

Classification

Points $\bar{x}$ — for us: strings

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

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

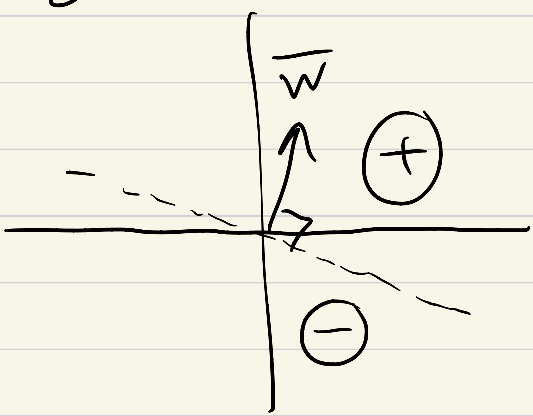
Classifier $\bar{x} \to y$

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

Decision rule: $\bar{w}^T f(\bar{x}) + b \overset{?}{\geq} 0$

if $> 0 \Rightarrow +1$ else $\Rightarrow -1$

$n = 2$



Sentiment Analysis

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

wow, that was soooo bad!

① Feature extraction:

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

string

② Learning algorithm   D exs

Training set $\{(f(\bar{x}^{(i)}), y^{(i)})\}_{i=1}^{D}$

$\Rightarrow ??? \Rightarrow \bar{w}$ learned weights vector

Feature Extraction

combos

- $\boxed{\text{what words are there}}$
- are the words "positive" or "negative"
- order
- punc. (! :( )
- "intensity" (good vs. great)
- context of words ("not great")

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

Our basic tool: bag-of-words features

$$\begin{array}{cccccccccccc} [\,1 & 0 & 1 & .\,0 & \ldots & 0 & 1 & \ldots & 0 & \ldots & 1\,] \\ \text{the} & \text{a} & \ldots & \text{movie} & \ldots & \text{good} & \text{great} & \ldots & & & \text{was} \ldots \end{array}$$

Vocabulary of n words $n \approx 10{,}000$

$\underline{1}$ if present (or count)

10K-dim vector, 4 1s 9996 0s
^
sparse

weight vector $\bar{w} \in \mathbb{R}^{10{,}000}$

$$\begin{array}{ccccc} [-0.1 & +0.2 \ldots & +0.3 \ldots & +10 \ldots & -0.1] \\ \text{the} & \text{a} & \text{movie} & \text{great} & \text{was} \end{array}$$

$$\bar{w}^T f(\bar{x}) = 10.1 = w_{the} \cdot 1 + w_{movie} \cdot 1$$
$$+ w_{was} \cdot 1 + w_{great} \cdot 1$$

Problems

1. "not great" $w_{not} + w_{great}$

   ↳ also "really great"

   ↳ no order of words

2. Static weights

   word senses (awesome, good)

3. Different weights for related words

   good, great

   +10 "never seen"

Preprocessing

1. Vocab selection

   vector space is fixed

   maybe we look at our training data

   $\overline{W} \in \mathbb{R}^n$, n doesn't change at test time

replace rare words in train w/UNK
learn a weight for it

that movie... really, it wasn't that great!

was n't   wasn't   was not

really   ,   [space]   great   !

$$\overline{w} = \{ \overset{+3}{\text{great}} \; \ldots \; \overset{+7}{\text{great!}} \; \ldots] \quad ???$$

<u>Tokenization</u>
- break out punc.
- break out contractions

(2) Remove <u>stopwords</u> (the, of, a, ...)
<u>optional</u>

(3) Lowercasing / Stemming → arrives → arrive

Not "not great" should be different from "great"

[the -- great --- not_great ----]
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad 1$

Bigram bag-of-words

$\left.\begin{array}{l}\text{Unigrams (each word)} \\ \text{Bigrams (each adjacent pair)}\end{array}\right\} \to V$

$$\{ \begin{array}{cccccc} \text{the} & \text{a} & \text{not\_good} & \text{movie} & \text{horror\_movie} & \ldots \\ U & U & B & U & B & \end{array}]$$

We can select what bigrams go in the vocab! Did we solve "not"?

1. No.. not + X for all X
2. "not very good"

Machine Learning

Optimize parameters $\bar{w}$ to fit training data

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

→ if we use $\bar{w}$ to predict on $x^{(i)}$, how badly do we mess up w.r.t. $y^{(i)}$?

(Stochastic) Gradient Descent

for t in range(0, epochs):

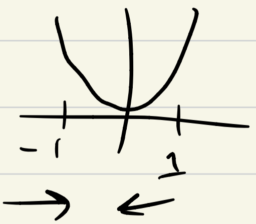
for i in range (0, D):

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

α step size for now $= 1$

subtract gradient of loss $\Rightarrow$
find $\overline{w}$ with lower loss

$\text{loss}(w) = w^2$



$w = 1$

$\frac{\partial}{\partial w} \text{loss} = 2$

$-\frac{\partial}{\partial w} \text{loss} = -2$

Perceptron

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

for t in range(0, epochs)

  for i in range(0, D)

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

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

Set $\alpha = 1$ for now

Suppose $\overline{w}^T f(\overline{x}^{(i)})$ was < 0

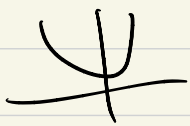
Let $\overline{w}' = \overline{w} + \alpha f(\overline{x}^{(i)})$

$$\overline{w}'^T f(\overline{x}) = \overline{w}^T f(\overline{x}^{(i)}) + \underbrace{\alpha f(\overline{x}^{(i)})^T f(\overline{x}^{(i)})}_{> 0}$$

Sparsity

If $f(\bar{x}^{(i)})$ only involves 4 features w/nonzero values, computing y_{pred} and the new $\bar{w}$ only involves those 4 features

Step size For w^2 case

 need $\alpha < 1$

In general: decrease α over training

One possibility: $\alpha = \frac{1}{t}$ t epochs

$\alpha = e^{-t}$ ← drops too fast

Do not randomly init. $\bar{w}$ on A1