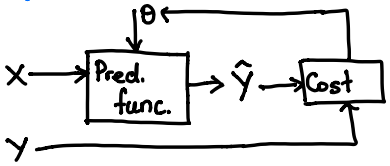
WEEK 1

SENTIMENT ANALYSIS WITH LOGISTIC REGRESSION.

•Supervised ML & Sentiment Analysis.



Positive/negative sentiment using logistic regression.

↳extract features → train LR → classify → positive.
→ negative.

•Vocabulary & Feature Extraction.

V = [I, am, happy, because, learning, NLP, ... hated, the, movie]

[1, 1, 1, 1, 1, 1, ... 0, 0, 0] → A lot of zeroes → sparse representation.

# of features = length of the vocabulary V.

LR needs to learn (n+1) parameters $[\theta_0, \theta_1, \theta_2, \ldots, \theta_n]$ → 1. large training time
→ 2. large prediction time

•Negative and Positive Frequencies and Feature Extraction.

Corpus:

I am happy because I am learning NLP.

I am happy.

I am sad, I am not learning NLP.

I am sad.

Vocabulary	PosFreq(1)	NegFreq(0)
I	3	3
am	3	3
happy	2	0
because	1	0
learning	1	1
NLP	1	1
sad	0	1
not	0	1

$$X_m = \left[1, \sum_w freqs(w,1), \sum_w freqs(w,0)\right]$$

1 → bias; $\sum_w freqs(w,1)$ → Sum Pos. Freqs.; $\sum_w freqs(w,0)$ → Sum Neg. Freqs.

$$X_m = [1, 8, 11] = [1, \underbrace{3+3+1+1}_{\text{(I) (am) (happy) (NLP)}}, \underbrace{3+3+2+1+1+1}_{\text{(I) (am) (sad) (not) (happy) (NLP)}}]$$

• Preprocessing.

stop words and punctuation

are, and, is, at, has, for, a

, . : ; ! " ' ...

~~@Mourri~~ and ~~@Andrew~~ are tuning a GREAT AI model at ~~http://.....~~ !!!

Stemming and lowercasing

tuning GREAT AI model. → → → [tun, great, ai, model]

tuning → tun ← tune, tuned, tuning

GREAT, Great, great → great

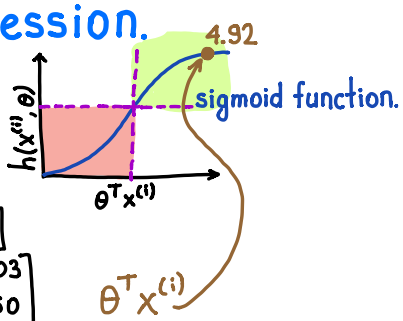
1. Eliminate handles and URLs.
2. Tokenize the string into words.
3. Remove stop words like "and, is, a, etc.".
4. Stemming: convert every word to its stem.
5. Convert all the words to lowercase.

$$\begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} \\ \vdots & \vdots & \vdots \\ 1 & x_1^{(m)} & x_2^{(m)} \end{bmatrix}$$

freqs = build_freqs(tweets, labels)

• Logistic Regression.

$$h(x^{(i)}, \theta) = \frac{1}{1+e^{-\theta^T x^{(i)}}}$$



[tun, ai, great, model]

$$X^{(i)} = \begin{bmatrix} 1 \\ 3476 \\ 245 \end{bmatrix} \quad \theta = \begin{bmatrix} 0.00003 \\ 0.00150 \\ -0.00120 \end{bmatrix}$$

θ (initialize)

h = h(X, θ) (predict)

$\nabla = \frac{1}{m} X^T (h - y)$ (get gradient)

$\theta = \theta - \alpha \nabla$ (update)

J(θ) (get loss)

• Cost function.

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \left[y^{(i)} \log h(x^{(i)}, \theta) + (1 - y^{(i)}) \log (1 - h(x^{(i)}, \theta) \right]$$

relevant when the label = 1

$y^{(i)}$	$h(x^{(i)},\theta)$	
0	any	0
1	0.99	~0
1	~0	-inf

$y^{(i)}$	$h(x^{(i)},\theta)$	
1	any	0
0	0.01	~0
0	~1	-inf

relevant when the label = 0