Week 1

Preprocessing: stop words and punctuation

@YMourri and @AndrewYNg are tuning a GREAT AI model at https://deeplearning.ai!!!

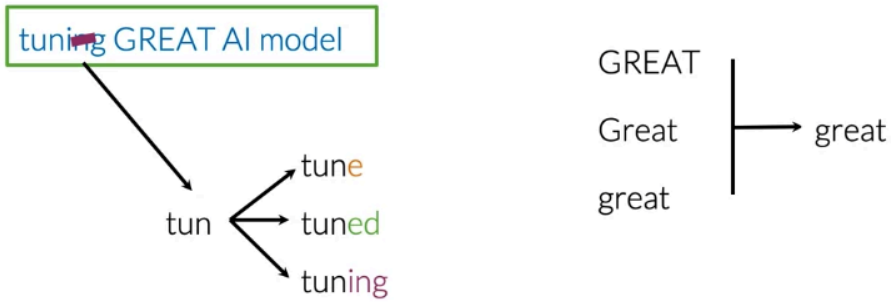
Stop words
and
is
are
at
has
for
a

Punctuation
,
.
:
!
“
‘

Preprocessing

1.50

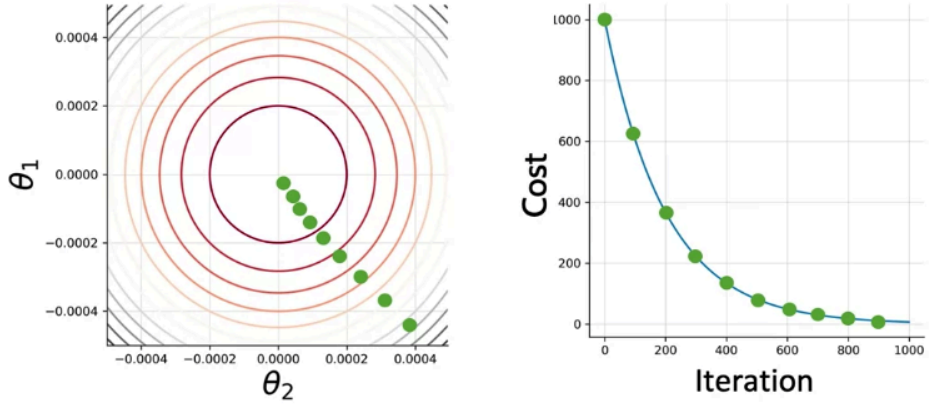
Preprocessing: Stemming and lowercasing



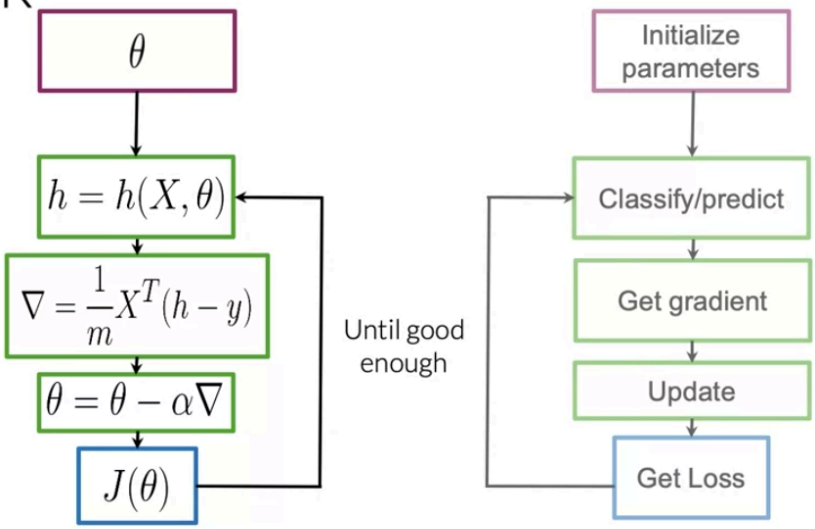
Logistic Regression: Training

1.50

Training LR



Training LR



Cost function for logistic regression

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

1.50

Testing logistic regression

- X_{val} Y_{val} θ

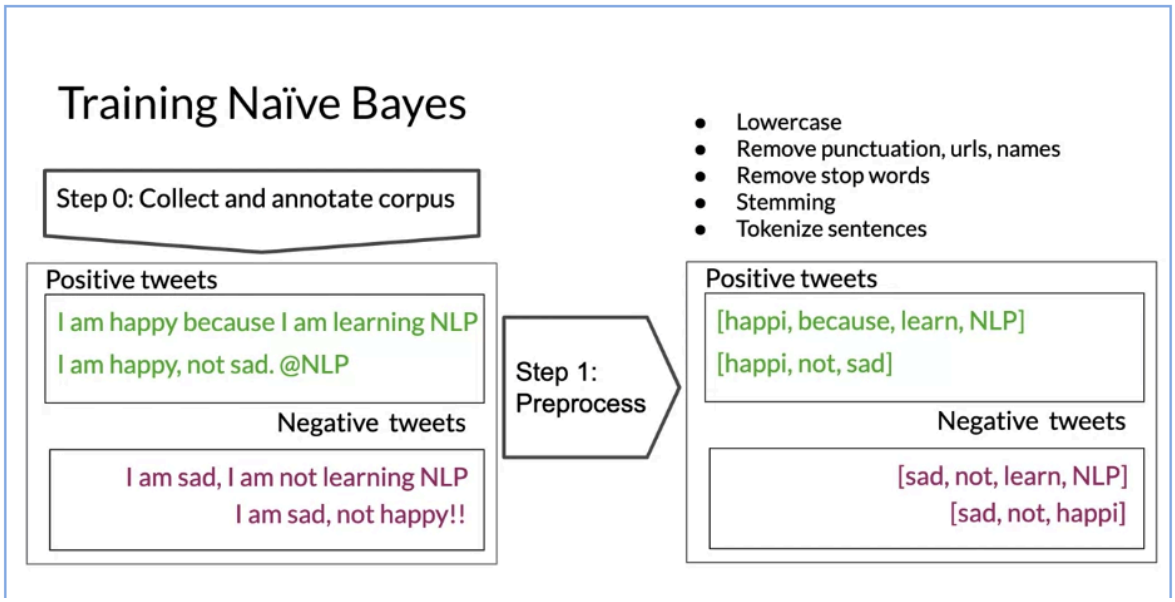
  $h(X_{val}, \theta)$

  $pred = h(X_{val}, \theta) \geq 0.5$

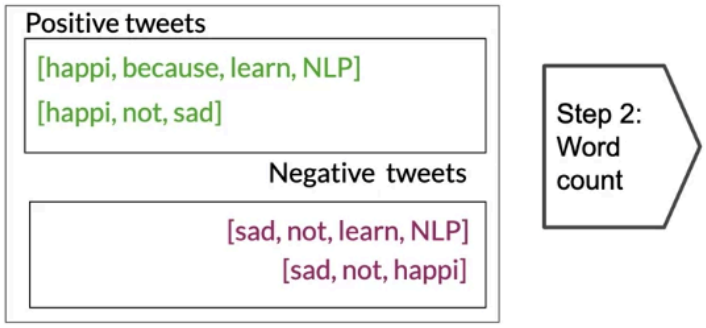
$$\begin{bmatrix} 0.3 \\ 0.8 \\ 0.5 \\ \vdots \\ h_m \end{bmatrix} \geq 0.5 = \begin{bmatrix} 0.3 \geq 0.5 \\ 0.8 \geq 0.5 \\ 0.5 \geq 0.5 \\ \vdots \\ pred_m \geq 0.5 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ \vdots \\ pred_m \end{bmatrix}$$

Week 2

Training Naïve Bayes



Training Naïve Bayes



freq(w, class)

word	Pos	Neg
happi	2	1
because	1	0
learn	1	1
NLP	1	1
sad	1	2
not	1	2
N_{class}	7	7

Training Naïve Bayes

freq(w, class)

word	Pos	Neg
happi	2	1
because	1	0
learn	1	1
NLP	1	1
sad	1	2
not	1	2
N_{class}	7	7

Step 3: $P(\mathrm{w}|\mathrm{class})$

$V_{\mathrm{class}} = 6$

$$\frac{\mathrm{freq}(\mathrm{w}, \mathrm{class}) + 1}{N_{\mathrm{class}} + V_{\mathrm{class}}}$$

$$\lambda(w) = log\frac{P(\mathrm{w}|\mathrm{pos})}{P(\mathrm{w}|\mathrm{neg})}$$

Step 4: Get lambda

word	Pos	Neg	λ
happy	0.23	0.15	0.43
because	0.15	0.07	0.6
learning	0.08	0.08	0
NLP	0.08	0.08	0
sad	0.08	0.17	-0.75
not	0.08	0.17	-0.75

1.00

Training Naïve Bayes

Step 5: Get the log prior

D_{pos} = Number of positive tweets
D_{neg} = Number of negative tweets

$$\mathrm{logprior} = log\frac{D_{pos}}{D_{neg}}$$

If dataset is balanced, $D_{pos} = D_{neg}$ and logprior = 0.

Training Naïve Bayes

1.00

Summary

0. Get or annotate a dataset with positive and negative tweets
1. Preprocess the tweets: process_tweet(tweet) → [w_1, w_2, w_3, ...]
2. Compute freq(w, class)
3. Get P(w | pos), P(w | neg)
4. Get λ(w)
5. Compute logprior = log(P(pos) / P(neg))

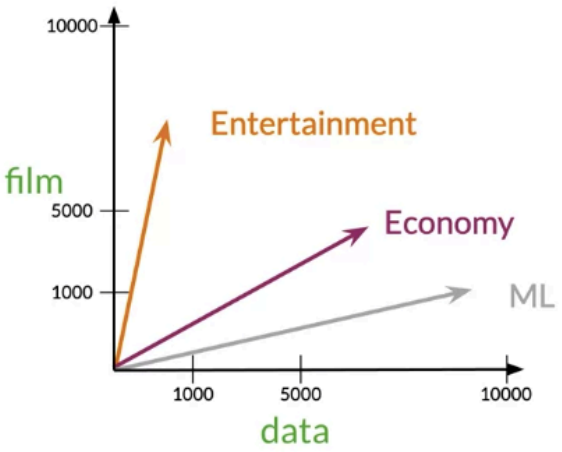
1.50

Summary

- Independence: Not true in NLP
- Relative frequency of classes affect the model

Week 3

Vector Space



	Entertainment	Economy	ML
data	500	6620	9320
film	7000	4000	1000

Measures of "similarity:"
Angle
Distance

Euclidean distance for n-dimensional vectors

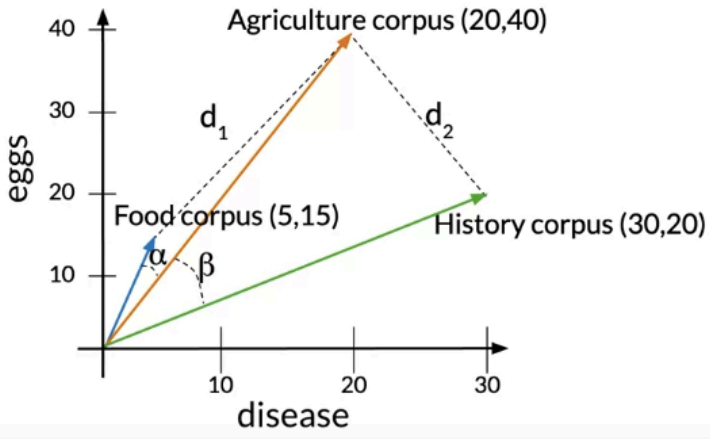
	data	$\vec{w}$ boba	$\vec{v}$ ice-cream
AI	6	0	1
drinks	0	4	6
food	0	6	8

$$= \sqrt{(1-0)^2 + (6-4)^2 + (8-6)^2}$$

$$= \sqrt{1+4+4} = \sqrt{9} = 3$$

$$d(\vec{v}, \vec{w}) = \sqrt{\sum_{i=1}^{n} (v_i - w_i)^2} \longrightarrow \text{Norm of } (\vec{v} - \vec{w})$$

Euclidean distance vs Cosine similarity



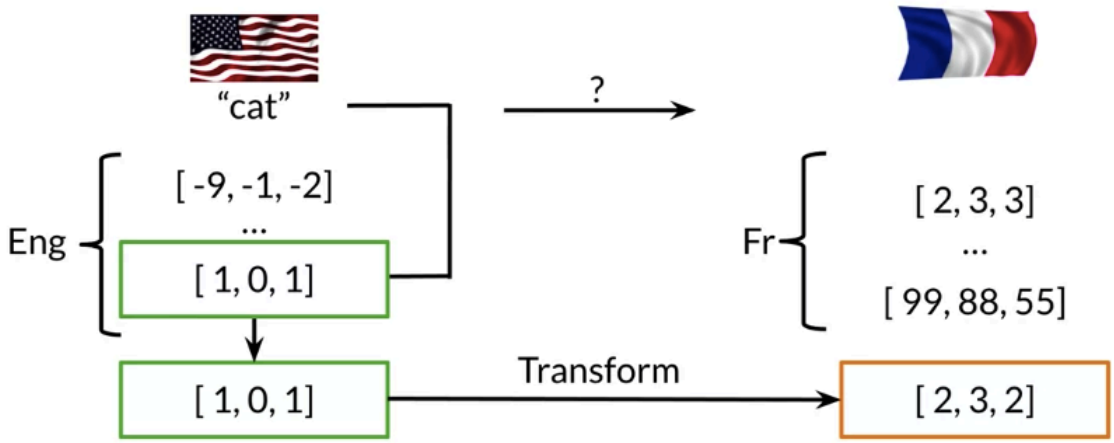
Euclidean distance: $d_2 < d_1$

Angles comparison: $\beta > \alpha$

The cosine of the angle between the vectors

Week 4

Overview of Translation



Align word vectors

$$\begin{pmatrix} \text{["cat" vector]} \\ \text{[... vector]} \\ \text{["zebra" vector]} \end{pmatrix} \mathbf{X} \mathbf{R} \approx \begin{pmatrix} \text{["chat" vecteur]} \\ \text{[... vecteur]} \\ \text{["zébresse" vecteur]} \end{pmatrix} \mathbf{Y}$$

Solving for R

initialize R

in a loop:

$$Loss = \| \mathbf{XR} - \mathbf{Y} \|_F$$

$$g = \frac{d}{dR} Loss$$ gradient

$$R = R - \alpha g$$ update

Frobenius norm

$$\| \mathbf{XR} - \mathbf{Y} \|_F$$

$$\mathbf{A} = \begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix}$$

$$\|\mathbf{A}_F\| = \sqrt{2^2 + 2^2 + 2^2 + 2^2}$$

$$\|\mathbf{A}_F\| = 4$$

$$\|\mathbf{A}\|_F \equiv \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2}$$

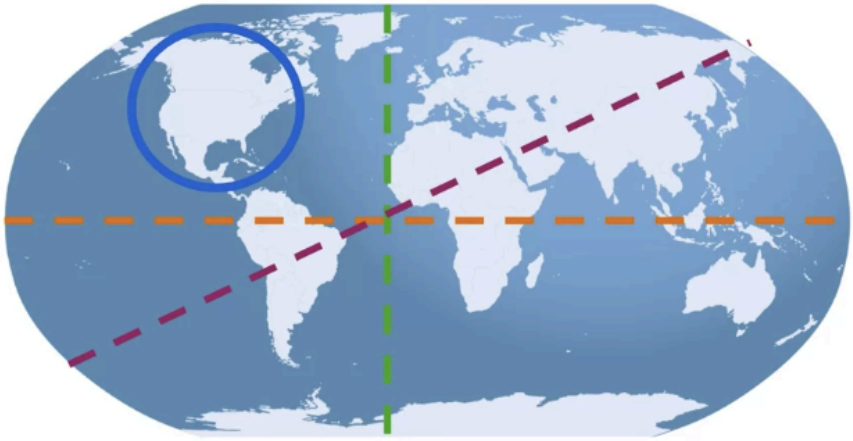
Gradient

$$Loss = \|\mathbf{XR} - \mathbf{Y}\|_F^2$$

$$g = \frac{d}{dR} Loss = \frac{2}{m} \left(\mathbf{X}^T (\mathbf{XR} - \mathbf{Y})\right)$$

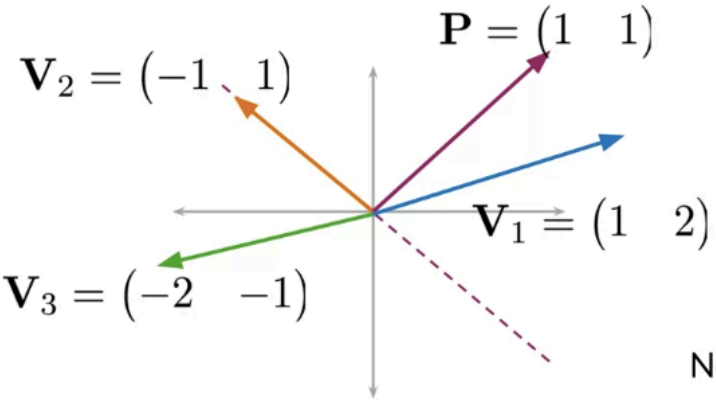
Implement in the assignment!

Nearest neighbors



Hash tables!

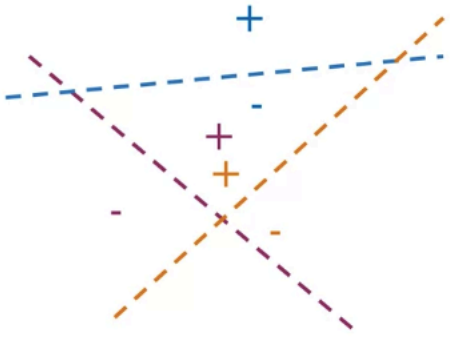
Which side of the plane?



$$\mathbf{P}\mathbf{V}_1^T = 3$$
$$\mathbf{P}\mathbf{V}_2^T = 0$$
$$\mathbf{P}\mathbf{V}_3^T = -3$$

Notice the signs?

Multiple planes, single hash value?



$$\mathbf{P}_1\mathbf{v}^T = 3, sign_1 = +1, h_1 = 1$$

$$\mathbf{P}_2\mathbf{v}^T = 5, sign_2 = +1, h_2 = 1$$

$$\mathbf{P}_3\mathbf{v}^T = -2, sign_3 = -1, h_3 = 0$$

$$hash = 2^0 \times h_1 + 2^1 \times h_2 + 2^2 \times h_3$$
$$= 1 \times 1 + 2 \times 1 + 4 \times 0$$
$$= 3$$