Course 1: Natural Language Processing with Classification and Vector Spaces

Bayes' Rule

Conditional probabilities

- P(B | A) Probability of B, given A.
- P(A | B) Given an element from set A, the probability that it belongs to set B.

Example:

"Happy weekend": P

"This is a good day": P

"It's a good weather": P

"I'm not happy": N

P(Positive | "happy") = P(Positive ∩ "happy") / P("happy") = 1/2 = 0.5

P("happy" | Positive) = P(Positive ∩ "happy") / P(Positive) = 1/3 = 0.333

Bayes' Rule : P(X | Y) = P(Y | X) * P(X) / P(Y)

Q/ Suppose that in your dataset, 25% of the positive tweets contain the word 'happy'. You also know that a total of 13% of the tweets in your dataset contain the word 'happy', and that 40% of the total number of tweets are positive. You observe the tweet: ''happy to learn NLP'. What is the probability that this tweet is positive?

A/ P(X | Y) = 0.25 * 0.4/0.13

Naive Bayes

Positive tweets

I am happy because I am learning NLP

I am happy, not sad.

Negative tweets

I am sad, I am not learning NLP

I am sad, not happy

word	Pos	Neg
I	3	3
am	3	3
happy	2	1
because	1	0
learning	1	1
NLP	1	1
sad	1	2
not	1	2
N_{class}	13	12

Tweet: I am happy today; I am learning.

$$\prod_{i=1}^{m} \frac{P(w_i|pos)}{P(w_i|neg)} = \frac{0.14}{0.10} = 1.4 \quad > 1$$

$$\cancel{\frac{0.20}{0.20}} * \cancel{\frac{0.20}{0.20}} * \boxed{\frac{0.14}{0.10}} * \cancel{\frac{0.20}{0.20}} * \cancel{\frac{0.20}{0.20}} * \cancel{\frac{0.10}{0.10}}$$

word	Pos	Neg
I	0.20	0.20
am	0.20	0.20
happy	0.14	0.10
because	0.10	0.05
learning	0.10	0.10
NLP	0.10	0.10
sad	0.10	0.15
not	0.10	0.15

Laplacian Smoothing: Adding 1 to the numerator to fix the 0 problem.

Log Likelihood: The product of many small numbers can cause numerical underflow problem, so we add log to fix it.

Applications of Naive Bayes

- Author identification
- Spam filtering
- Information retrieval
- Word disambiguation

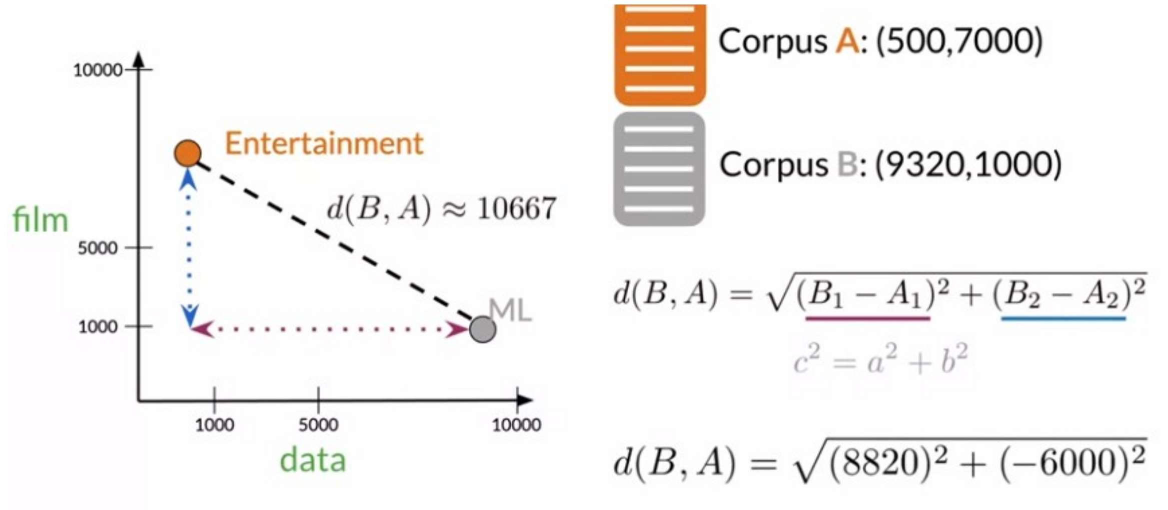
Vector Space Models

You can get vector spaces by two different designs:

- **word by word:** counting the co-occurrence of words with certain distance.
- **word by document:** the co-occurrence of words in the document's corpora.

Two ways to calculate the similarity between 2 vectors:

- **Euclidean Distance:** The length of the straight line that's connects two vectors.

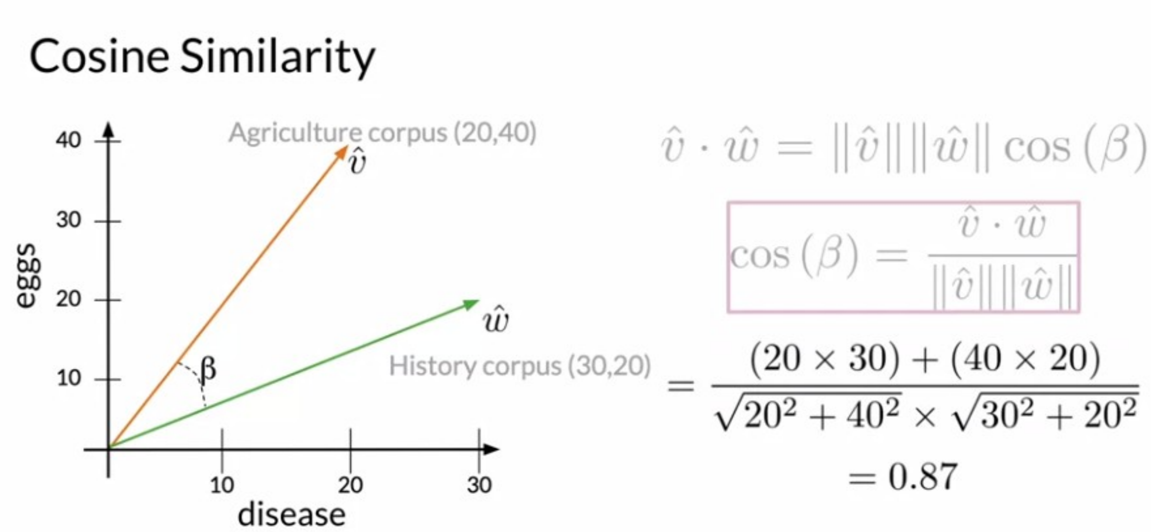


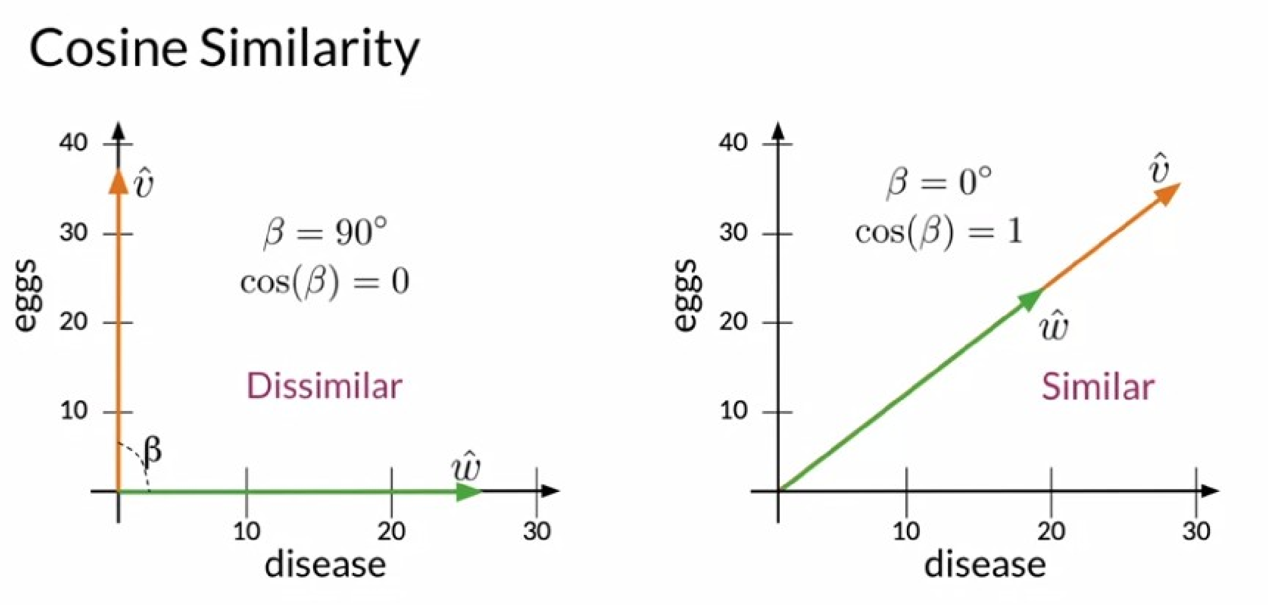
- **Cosine similarity:** Applying cos function on the angle between 2 vectors.

Previous definitions:

vector norm(*//v//*): The square root of the sum of its elements squared.

Dot product(*v.w*): The sum of the products between their elements in each dimension of the vector space.



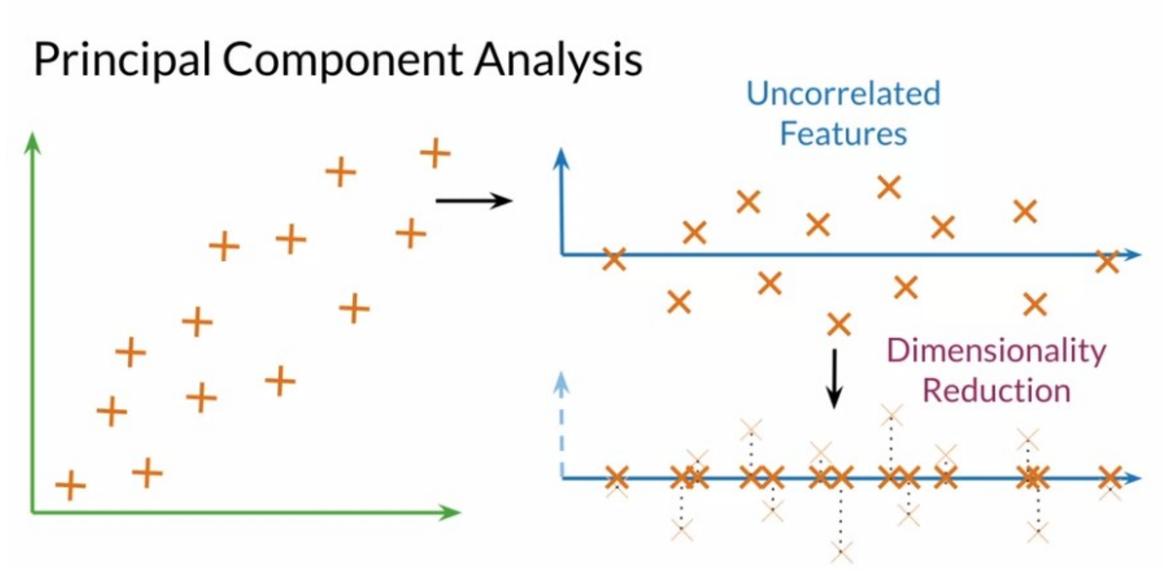


PCA(Principal Component Analysis): An algorithm used for dimensionality reduction by finding uncorrelated features by 3 steps.

1. Mean normalize data
2. Co-variance matrix
3. SVD(Singular Value Decomposition) returns 3 matrices

Eigenvectors: Uncorrelated features.

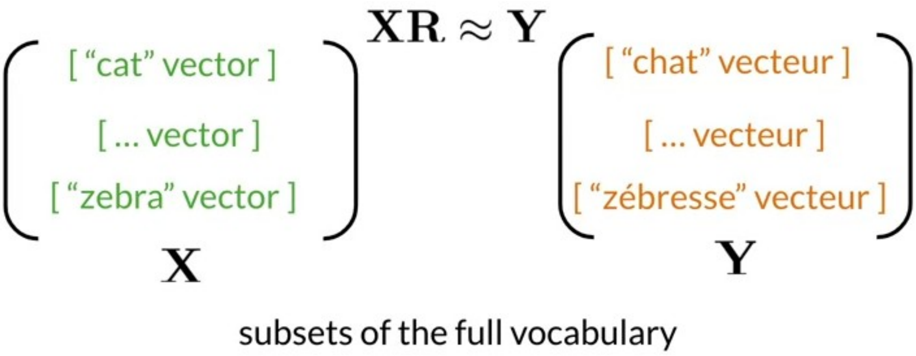
Eigenvalues: The retained features.



Machine Translation

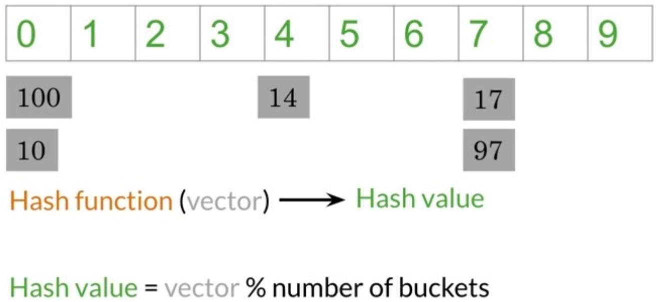
Document to vector
- Documents can be represented as vectors with the same dimension as words by adding the word vectors in the documents.

Transforming word vectors
In order to translate from a language a word vectors are X to another language a word vectors are Y we want to build a matrix R using gradient descent.



K-nearest neighbors
- To translate from X to Y using the R matrix, you may find that XR doesn't correspond to any specific vector in Y.
- KNN can search for the K nearest neighbors from the computed vector XR.
- Thus searching in the whole space can be slow, using a **hash tables** can minimize your search space.

Hash tables and hash functions
- A simple hash function :
  Hash Value = vector % number of buckets



Locality Sensitive Hashing
- Separate the space using hyperplanes