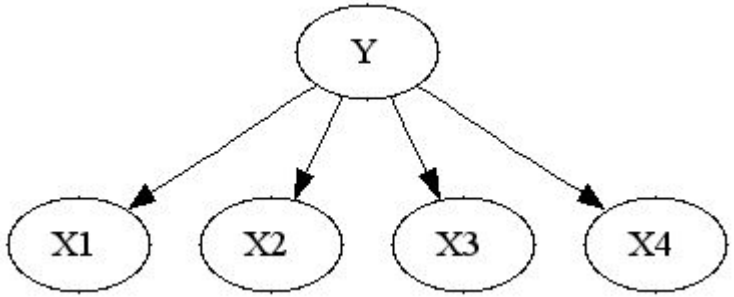
Naive Bayes

for text classification

DSI, jf.omhover

Credits: drawing upon Pinar Colak, Miles Erickson, Mari Pierce-Quinonez



Naive Bayes

for text classification

DSI, jf.omhover

Credits: drawing upon Pinar Colak, Miles Erickson, Mari Pierce-Quinonez

OBJECTIVES

- Explain why Naive Bayes is Naive
- Describe when it is best to use Naive Bayes
- Explain why we need Laplace Smoothing
- Implement a Naive Bayes Algorithm

Naive Bayes in a nutshell

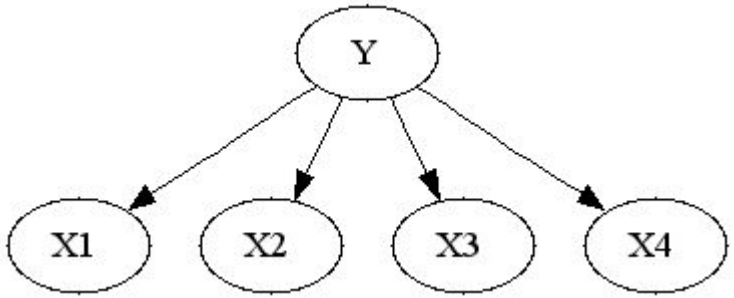
With

- w: word
- y: the **class** of a document
  (0 for neg, 1 for pos)

Given a dataset of labelled documents.

Probability $P(y \mid w_1, w_2, \ldots, w_p) = P(y)\frac{\prod_i P(w_i|y)}{\prod_i P(w_i)}$

Hint: use bayes rule ^^;



Keywords: Naive Bayes, Bayes rule, spam classification
Recommended reading: Naive Bayes explained

When to use Naive Bayes?

- n << p (# of features)
- n somewhat small or
- n quite large
- streams of input data (online learning)
- not bounded by memory (usually)
- Multi-class

Bayes + Naive

Bayes:

$$P(y|w_1 \dots w_p) = \frac{P(w_1 \dots w_p|y)P(y)}{p(w_1 \dots w_p)}$$

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

Naive: we assume words are independent

$$P(w_1 \dots w_p|y) = P(w_1|y) \dots P(w_p|y)$$

$$P(\, y \mid w_1, w_2, \dots, w_p \,) = P(y) \frac{\prod_i P(w_i|y)}{\prod_i P(w_i)}$$

That! We know!

Laplace smoothing

$$P(y \mid w_1, w_2, \ldots, w_p) = P(y)\frac{\prod_i P(w_i|y)}{\prod_i P(w_i)}$$

$$P(w|y) = \frac{n_{w,y}}{N_y}$$

$n_{w,y}$ Number of words w in documents of class y

N_y Number of all words in documents of class y

$|V|$ Size of the vocabulary

$$Laplace\ P(w|y) = \frac{n_{w,y}+\alpha}{N_y+\alpha|V|}$$

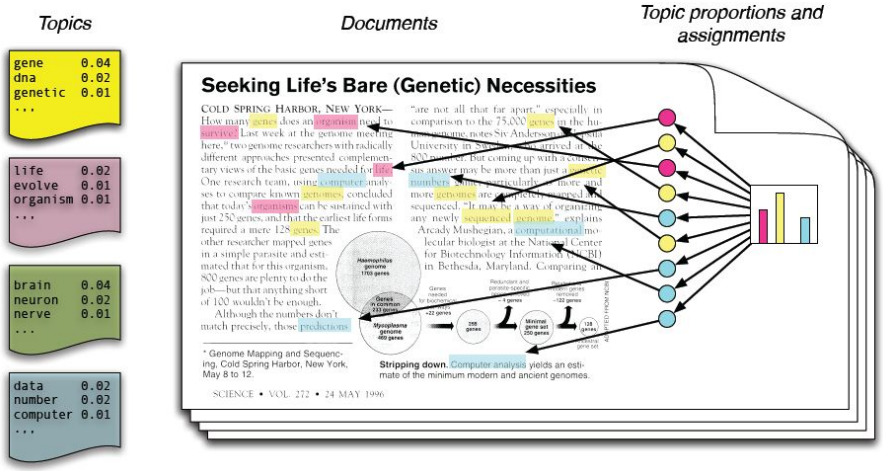
LDA : Latent Dirichlet Allocation (clustering)

With

- w: word
- d: document
- D: the corpus of docs
- k: a given number of topics

A **topic** is a distribution over words.
Each **document** is a mixture of corpus-wide topics.
Each **word** is drawn from these topics.

Find the k distributions that would likely “generate” every document d in D.



Keywords: LDA, Latent Semantic Analysis
Recommended reading: [Topic modeling for Humanists, a guided tour]

Word2Vec (dimensionality reduction)

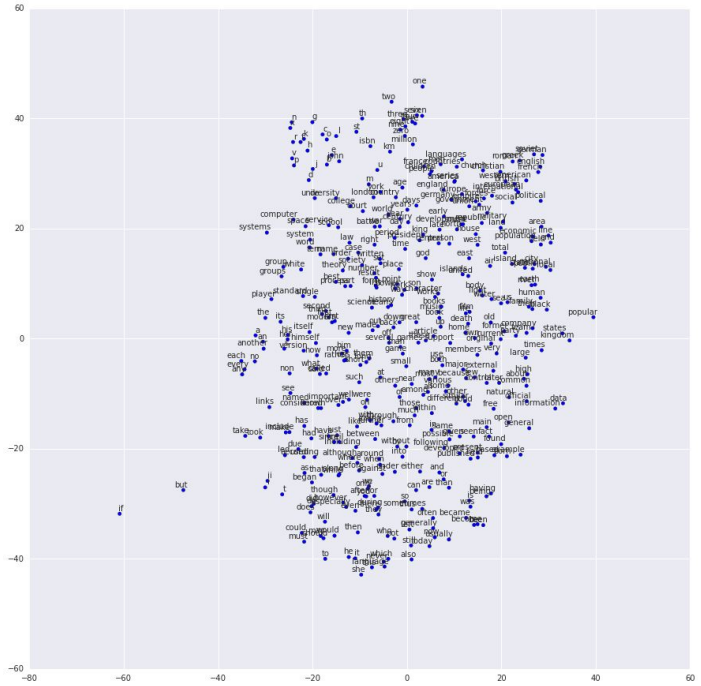
With

- w: word
- **c:context** (as a window surrounding w)
- D the corpus of docs

Model the **probability** $P(D = 1|w, c)$
that w in context c has been observed in D

Train on **positive examples** (from the observed corpus) and **negative examples** (made up)

Somehow analogous to the **matrix factorization** for finding latent features for words.



Keywords: Word2Vec, Skip-Gram with Negative Sampling (SGNS), Pointwise Mutual Information
Recommended reading: Neural Word Embeddings as Implicit Matrix Factorization. Omer Levy and Yoav Goldberg. NIPS 2014. [pdf]