

# Text Classification & Naive Bayes

Chapter 13 - IIR



SAPIENZA  
UNIVERSITÀ DI ROMA

Fabrizio Silvestri

# Quick Intro



# A text classification task: Email spam filtering

From: '''' <takworl1d@hotmail.com>

Subject: real estate is the only way... gem oalvgkay

Anyone can buy real estate with no money down

Stop paying rent TODAY !

There is no need to spend hundreds or even thousands for similar courses

I am 22 years old and I have already purchased 6 properties using the

methods outlined in this truly INCREDIBLE ebook.

Change your life NOW !

=====

Click Below to order:

<http://www.wholesaledaily.com/sales/nmd.htm>

=====

- How would you proceed to decide whether this text is spam or ham?



# Text Classification: a (more) formal definition

- Given:
  - A **document space**  $X$ 
    - Documents are represented in this space – typically some type of high-dimensional space.
  - A fixed set of **classes**  $C = \{c_1, c_2, \dots, c_J\}$ 
    - The classes are human-defined for the needs of an application (e.g., spam vs. nonspam).
  - A **training set**  $D$  of labeled documents.
    - Each labeled document  $(d, c) \in X \times C$

Using a learning method or **learning algorithm**, we then wish to learn a **classifier**  $f$  that maps documents to classes:

$$f : X \rightarrow C$$

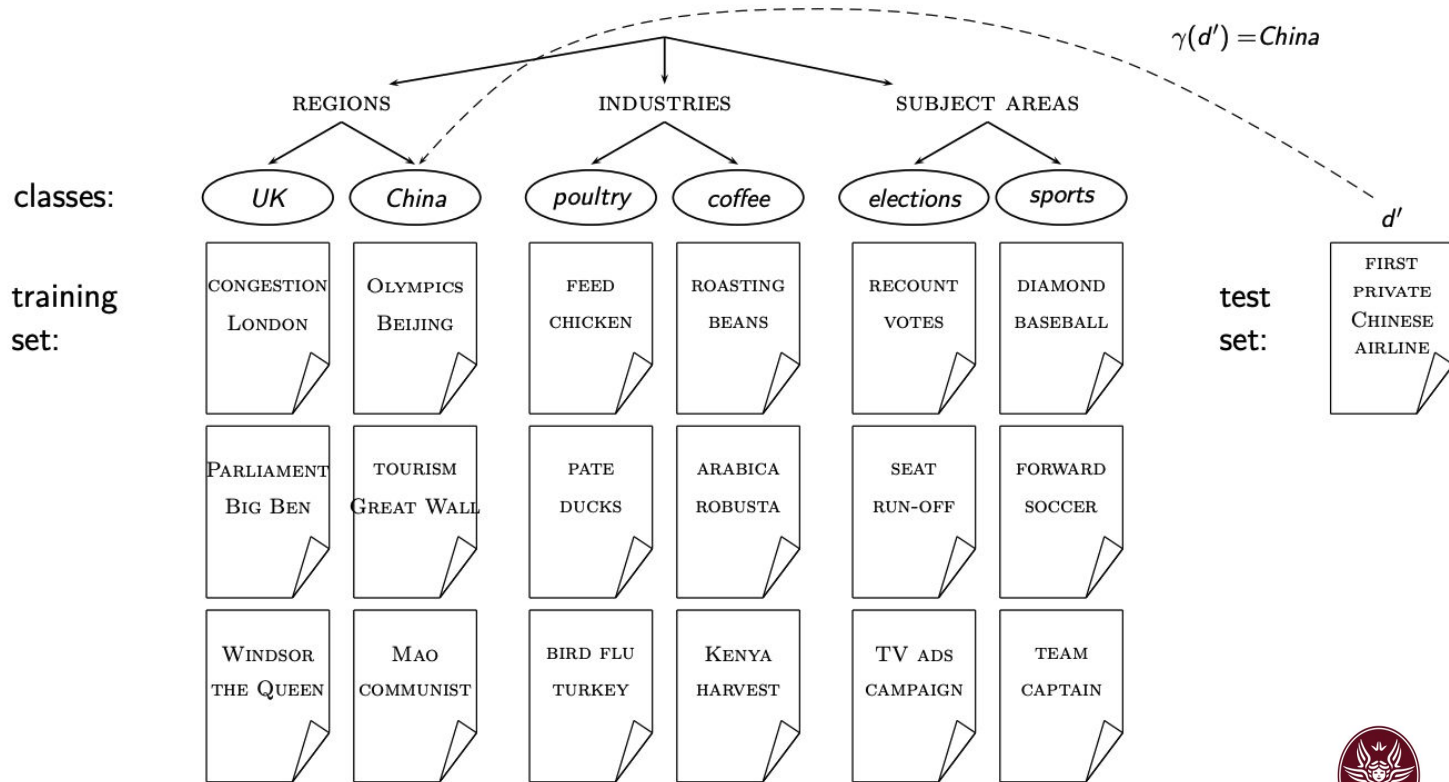


# Text Classification: Inference

- Given a representation for a document  $d \in X$ , determine the most appropriate class using the learnt function  $f$
- In other words  $C = f(d)$



# An example: Topic Classification



# An example: Sentiment Analysis

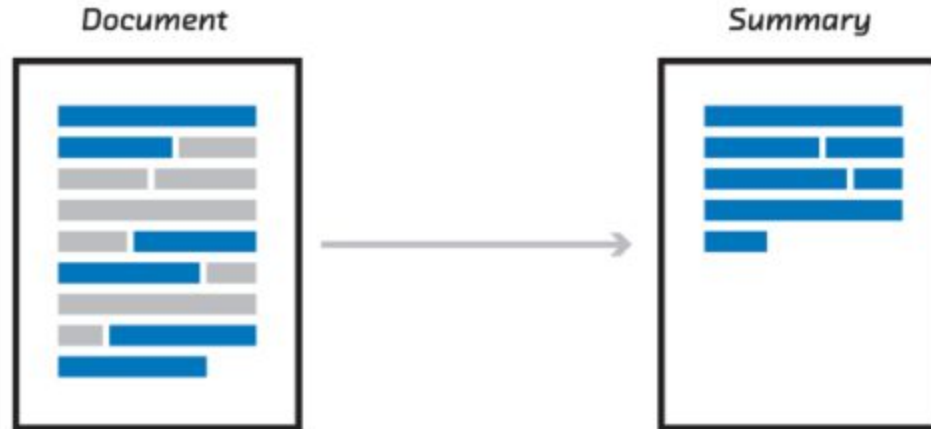


# An example: Misinformation Classification





# An example: Automatic Summarization



# An example: Language Detection



The image is a screenshot of a Google search interface. At the top left is the Google logo. To its right is a search bar containing the text "language 'ciao'". To the right of the search bar are icons for a close button (X), voice search (microphone), and a search icon (magnifying glass). Below the search bar is a horizontal menu with links: "All" (with a magnifying glass icon), "Images" (with a picture icon), "Videos" (with a play button icon), "News" (with a newspaper icon), "Maps" (with a location pin icon), and "More" (with a three-dot icon). To the right of these links are "Settings" and "Tools". Below the menu, it says "About 20,700,000 results (0.58 seconds)". The main heading is "Italian". Below this is a paragraph: "Ciao (/ˈtʃaʊ/; **Italian** pronunciation: [ˈtʃaːo]) is an informal salutation in the **Italian language** that is used for both "hello" and "goodbye". Originally from the **Venetian language**, it has entered the vocabulary of **English** and of many **other languages** around the world." Below the paragraph is a URL: "https://en.wikipedia.org › wiki › Ciao". At the bottom is a link: "Ciao - Wikipedia".

Google

language "ciao"

× |  

 All  Images  Videos  News  Maps  More Settings Tools

About 20,700,000 results (0.58 seconds)

## Italian

Ciao (/ˈtʃaʊ/; **Italian** pronunciation: [ˈtʃaːo]) is an informal salutation in the **Italian language** that is used for both "hello" and "goodbye". Originally from the **Venetian language**, it has entered the vocabulary of **English** and of many **other languages** around the world.

[https://en.wikipedia.org › wiki › Ciao](https://en.wikipedia.org/wiki/Ciao)

[Ciao - Wikipedia](#)




# An example: Question Detection

Google

how tall is peppa pig

AI Images News Videos Shopping More Settings Tools

About 450,000,000 results (0.60 seconds)



**7 feet, 1 inch**

Peppa's height is **7 feet, 1 inch**.

[https://peppafanon.fandom.com/wiki/Peppa\\_Pig\\_\(character\)](https://peppafanon.fandom.com/wiki/Peppa_Pig_(character))

[Peppa Pig \(character\) | Peppa Pig Fanon Wiki | Fandom](#)

About featured snippets Feedback

People also ask

- Is Peppa Pig 7 ft tall?
- How tall is Peppa Pig in real life?
- How tall is George the Pig?
- How did Peppa Pig die?


Feedback



# An example: Vertical Selection

mineral water

About 417,000,000 results (1.07 seconds)



**Mineral water** is water from a mineral spring that contains various **minerals**, such as salts and sulfur compounds. **Mineral water** may usually be still or sparkling (carbonated/effervescent) according to the presence or absence of added gases.

[https://en.wikipedia.org/wiki/Mineral\\_water](https://en.wikipedia.org/wiki/Mineral_water)  
**Mineral water - Wikipedia**

People also ask

Is it good to drink mineral water? ▾

What is the best mineral water to drink? ▾

Can you make your own mineral water? ▾

Is mineral water the same as distilled water? ▾

[Feedback](#)

<https://www.healthline.com/nutrition/mineral-water-...>  
**Does Mineral Water Have Health Benefits? - Healthline**

Sep 4, 2019 — As its name suggests, mineral water can contain high amounts of minerals and other naturally occurring compounds, including magnesium, ...

What it is · [Benefits](#) · [Bottom line](#) · [Mineral](#)

Find results on

Indiamart  
[Mineral Water - Natural Miner...](#)

Kapacz  
[Compare types of bottled water](#)

Nature  
[Influence of a](#)

VS.

Google

johnson lindenstrauss

About 79,600 results (0.44 seconds)

[https://en.wikipedia.org/wiki/Johnson-Lindenstrauss\\_lemma](https://en.wikipedia.org/wiki/Johnson-Lindenstrauss_lemma) · **Wikipedia**

In mathematics, the **Johnson-Lindenstrauss lemma** is a result named after William B. Johnson and Joram Lindenstrauss concerning low-distortion embeddings ...

[Lemma](#) · [Alternate statement](#) · [Speeding up the JL...](#) · [Tensorized Random...](#)

<https://home.ttic.edu/~gregory/courses/lectures> · PDF  
**Random Projections 1 The Johnson-Lindenstrauss ... - TTIC**

1 The Johnson-Lindenstrauss lemma. Theorem 1.1. (Johnson-Lindenstrauss) Let  $\epsilon \in (0, 1/2)$ . Let  $Q \subset \mathbb{R}^d$  be a set of  $n$  points and  $k = 20 \log n \cdot \epsilon^{-2}$ . There.

<https://cs.stanford.edu/Lectures/lecture1> · PDF  
**The Johnson-Lindenstrauss Lemma - Stanford Computer ...**

Sep 23, 2009 — The Johnson-Lindenstrauss Lemma states that any  $n$  points in high dimensional euclidian space can be mapped onto  $k$  dimensions where  $k \geq O(\log n/\epsilon^2)$  without distorting the euclidian distance between any two points. more than a factor of  $1 \pm \epsilon$ . [1]

<https://www.cantorsparadise.com/the-johnson-lindenstrauss-lemma>  
**The Johnson-Lindenstrauss Lemma. Why you don't always ...**

Apr 7, 2020 — Johnson (1944-) and Joram Lindenstrauss (1936–2012). Informally, the lemma says that, given  $N$  points with  $N$  coordinates each (i.e., these ...

<http://cseweb.ucsd.edu/~dakane/derandomizedJL> · PDF  
**Almost Optimal Explicit Johnson-Lindenstrauss ... - UCSD CSE**

by DM Kane · Cited by 57 — Abstract. The Johnson-Lindenstrauss lemma is a fundamental result in probability with several ap- plications in the design and analysis of algorithms.



# Some Text Classification Methods



# Rule Based

- E.g., Google Alerts is rule-based classification.
- There are IDE-type development environments for writing very complex rules efficiently. (e.g., Verity)
- Often: Boolean combinations (as in Google Alerts)
- Accuracy is very high if a rule has been carefully refined over time by a subject expert.
- Building and maintaining rule-based classification systems is cumbersome and expensive.



# Statistical Modeling

- This was our definition of the classification problem – text classification as a learning problem
  - (i) Supervised learning of a classification function  $\mathbf{f}$  and
  - (ii) application of  $\mathbf{f}$  to classifying new documents
- We will look at two methods for doing this: Naive Bayes and SVMs
- No free lunch: requires hand-classified training data
  - But this manual classification can be done by non-experts.



# Naïve Bayes





# What is a Naïve Bayes Classifier

- The Naive Bayes classifier is a probabilistic classifier.
- We compute the probability of a document  $d$  being in a class  $c$  as follows:

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c)$$

- $n_d$  is the length of the document. (number of tokens)
- $P(t_k | c)$  is the conditional probability of term  $t_k$  occurring in a document of class  $c$
- $P(t_k | c)$  as a measure of **how much evidence**  $t_k$  contributes that  $c$  is the correct class.
- $P(c)$  is the prior probability of  $c$ .
- If a document's terms do not provide clear evidence for one class vs. another, we choose the  $c$  with highest  $P(c)$ .



# Maximum A Posteriori (MAP) Classifier

- Our goal in Naive Bayes classification is to find the “best” class.
- The best class is the most likely or Maximum A Posteriori (MAP) class  $c_{\text{map}}$ :

$$c_{\text{map}} = \arg \max_{c \in \mathbb{C}} \hat{P}(c|d) = \arg \max_{c \in \mathbb{C}} \hat{P}(c) \prod_{1 \leq k \leq n_d} \hat{P}(t_k|c)$$



## Actually...

- Multiplying lots of small probabilities can result in floating point underflow.
- Since  $\log(xy) = \log(x) + \log(y)$ , we can sum log probabilities instead of multiplying probabilities.
- Since log is a monotonic function, the class with the highest score does not change.
- So what we usually compute in practice is:

$$c_{\text{map}} = \arg \max_{c \in \mathbb{C}} [\log \hat{P}(c) + \sum_{1 \leq k \leq n_d} \log \hat{P}(t_k | c)]$$



# Summarizing: Naïve Bayes Classifier

$$c_{\text{map}} = \arg \max_{c \in \mathbb{C}} [\log \hat{P}(c) + \sum_{1 \leq k \leq n_d} \log \hat{P}(t_k | c)]$$

- Each conditional parameter  $\log \hat{P}(t_k | c)$  is a weight that
- indicates how good an indicator  $t_k$  is for  $c$ .
- The prior  $\log \hat{P}(c)$  is a weight that indicates the relative frequency of  $c$ .
- The sum of log prior and term weights is then a measure of
- how much evidence there is for the document being in the class.
- We select the class with the most evidence.



# Parameter Estimation: MLE

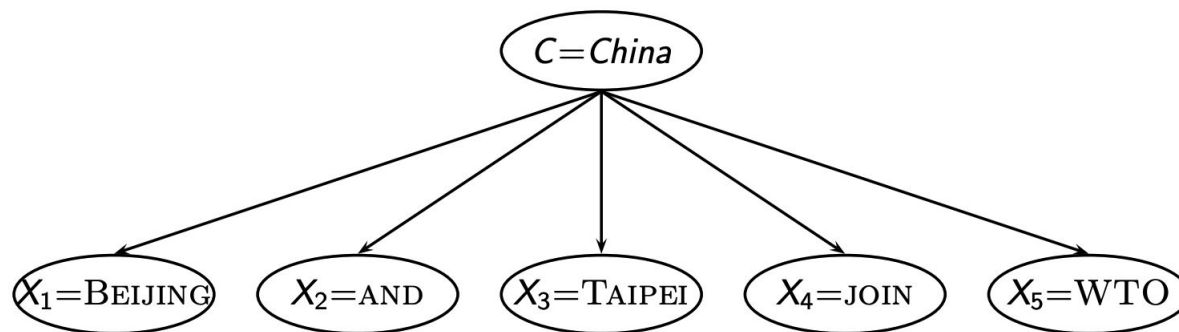
- Estimate parameters  $P(c)$  and  $P(t_k | c)$  from train data: How?
- Prior:  $P(c) = N_c / N$ 
  - $N_c$  : number of docs in class  $c$ ;  $N$ : total number of docs
- Conditional probabilities:

$$\hat{P}(t|c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}}$$

- $T_{ct}$  is the number of tokens of  $t$  in training documents from class  $c$  (includes multiple occurrences)
- We've made a **Naive Bayes independence assumption** here:  
 $P(t_k | c) = P(t_k | c)$ , independent of position.



# How to deal with zeros?



$$P(China|d) \propto P(China) \cdot P(BEIJING|China) \cdot P(AND|China) \\ \cdot P(TAIPEI|China) \cdot P(JOIN|China) \cdot P(WTO|China)$$

We will get  $P(China|d) = 0$  for any document that contains WTO!

...er occurs in class China in the train set:

$$\hat{P}(WTO|China) = \frac{T_{China,WTO}}{\sum_{t' \in V} T_{China,t'}} = \frac{0}{\sum_{t' \in V} T_{China,t'}} = 0$$



# Add Smoothing

- Before:

$$\hat{P}(t|c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}}$$

- Now: Add one to each count to avoid zeros:

$$\hat{P}(t|c) = \frac{T_{ct} + 1}{\sum_{t' \in V} (T_{ct'} + 1)} = \frac{T_{ct} + 1}{(\sum_{t' \in V} T_{ct'}) + B}$$

- B is the number of bins – in this case the number of different words or the size of the vocabulary  $|V| = M$



# Naïve Bayes: Summary

- Estimate parameters from the training corpus using add-one smoothing
- For a new document, for each class, compute sum of (i) log of prior and (ii) logs of conditional probabilities of the terms
- Assign the document to the class with the largest score





# Naïve Bayes: Training

TRAINMULTINOMIALNB( $\mathbb{C}, \mathbb{D}$ )

```
1   $V \leftarrow \text{EXTRACTVOCABULARY}(\mathbb{D})$ 
2   $N \leftarrow \text{COUNTDOCS}(\mathbb{D})$ 
3  for each  $c \in \mathbb{C}$ 
4  do  $N_c \leftarrow \text{COUNTDOCSINCLASS}(\mathbb{D}, c)$ 
5       $\text{prior}[c] \leftarrow N_c / N$ 
6       $\text{text}_c \leftarrow \text{CONCATENATETEXTOFALLDOCSINCLASS}(\mathbb{D}, c)$ 
7      for each  $t \in V$ 
8      do  $T_{ct} \leftarrow \text{COUNTTOKENSOFTERM}(\text{text}_c, t)$ 
9      for each  $t \in V$ 
10     do  $\text{condprob}[t][c] \leftarrow \frac{T_{ct}+1}{\sum_{t'} (T_{ct'}+1)}$ 
11 return  $V, \text{prior}, \text{condprob}$ 
```



## Naïve Bayes: Testing

```
APPLYMULTINOMIALNB( $\mathbb{C}$ ,  $V$ , prior, condprob,  $d$ )  
1   $W \leftarrow \text{EXTRACTTOKENSFROMDOC}(V, d)$   
2  for each  $c \in \mathbb{C}$   
3  do  $\text{score}[c] \leftarrow \log \text{prior}[c]$   
4    for each  $t \in W$   
5    do  $\text{score}[c] + = \log \text{condprob}[t][c]$   
6  return  $\arg \max_{c \in \mathbb{C}} \text{score}[c]$ 
```



# Exercise: Parameter Estimation

	docID	words in document	in $c = \textit{China}$ ?
training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
test set	5	Chinese Chinese Chinese Tokyo Japan	?

$$\hat{P}(c) = \frac{N_c}{N}$$

$$\hat{P}(t|c) = \frac{T_{ct} + 1}{\sum_{t' \in V} (T_{ct'} + 1)} = \frac{T_{ct} + 1}{(\sum_{t' \in V} T_{ct'}) + B}$$

( $B$  is the number of bins – in this case the number of different words or the size of the vocabulary  $|V| = M$ )

$$c_{\text{map}} = \arg \max_{c \in \mathbb{C}} [\hat{P}(c) \cdot \prod_{1 \leq k \leq n_d} \hat{P}(t_k|c)]$$



# Solution

Priors:  $\hat{P}(c) = 3/4$  and  $\hat{P}(\bar{c}) = 1/4$  Conditional probabilities:

$$\hat{P}(\text{CHINESE}|c) = (5 + 1)/(8 + 6) = 6/14 = 3/7$$

$$\hat{P}(\text{TOKYO}|c) = \hat{P}(\text{JAPAN}|c) = (0 + 1)/(8 + 6) = 1/14$$

$$\hat{P}(\text{CHINESE}|\bar{c}) = (1 + 1)/(3 + 6) = 2/9$$

$$\hat{P}(\text{TOKYO}|\bar{c}) = \hat{P}(\text{JAPAN}|\bar{c}) = (1 + 1)/(3 + 6) = 2/9$$

The denominators are  $(8 + 6)$  and  $(3 + 6)$  because the lengths of  $text_c$  and  $text_{\bar{c}}$  are 8 and 3, respectively, and because the constant  $B$  is 6 as the vocabulary consists of six terms.



## Solution

$$\hat{P}(c|d_5) \propto 3/4 \cdot (3/7)^3 \cdot 1/14 \cdot 1/14 \approx 0.0003$$

$$\hat{P}(\bar{c}|d_5) \propto 1/4 \cdot (2/9)^3 \cdot 2/9 \cdot 2/9 \approx 0.0001$$

Thus, the classifier assigns the test document to  $c = \textit{China}$ . The reason for this classification decision is that the three occurrences of the positive indicator CHINESE in  $d_5$  outweigh the occurrences of the two negative indicators JAPAN and TOKYO.



# Implementing NB

- [NB from scratch](#)
- [Using scikit-learn](#)



# Naïve Bayes: Computational Complexity

mode	time complexity
training	$\Theta( \mathbb{D} L_{\text{ave}} +  \mathbf{C}  \mathbf{V} )$
testing	$\Theta(L_a +  \mathbf{C} M_a) = \Theta( \mathbf{C} M_a)$

- $L_{\text{ave}}$ : average length of a training doc,  $L_a$ : length of the test doc,  $M_a$ : number of distinct terms in the test doc,  $\mathbf{D}$ : training set,  $\mathbf{V}$ : vocabulary,  $\mathbf{C}$ : set of classes
- $\Theta(|\mathbf{D}|L_{\text{ave}})$  is the time it takes to compute all counts.
- $\Theta(|\mathbf{C}||\mathbf{V}|)$  is the time it takes to compute the parameters from the counts.
- Generally:  $|\mathbf{C}||\mathbf{V}| < |\mathbf{D}|L_{\text{ave}}$
- Test time is also linear (in the length of the test document).
- Thus: **Naive Bayes is linear** in the size of the training set
- (training) and the test document (testing). **This is optimal.**



# Theoretical Analysis of Naïve Bayes





# Properties of NB

- Now we want to gain a better understanding of the properties of Naive Bayes.
- We will formally derive the classification rule . . .
- . . . and make our assumptions explicit.



# Derivation of NB

- We want to find the class that is most likely given the document:

$$c_{\text{map}} = \arg \max_{c \in \mathbb{C}} P(c|d)$$

- Apply Bayes rule  $P(c|d) = (1/P(d)) * P(d|c)P(c)$

$$c_{\text{map}} = \arg \max_{c \in \mathbb{C}} \frac{P(d|c)P(c)}{P(d)}$$

- Drop denominator since  $P(d)$  is the same for all classes:

$$c_{\text{map}} = \arg \max_{c \in \mathbb{C}} P(d|c)P(c)$$



# Data sparsity

$$\begin{aligned}c_{\text{map}} &= \arg \max_{c \in \mathbb{C}} P(d|c)P(c) \\ &= \arg \max_{c \in \mathbb{C}} P(\langle t_1, \dots, t_k, \dots, t_{n_d} \rangle | c)P(c)\end{aligned}$$

- There are too many parameters  $P(\langle t_1, \dots, t_k, \dots, t_{n_d} \rangle | c)$ , one for each unique combination of a class and a sequence of words.
- We would need a very, very large number of training examples to estimate that many parameters.
- This is the problem of **data sparseness**.



# NB: conditional independence assumption

- To reduce the number of parameters to a manageable size, we make the Naive Bayes conditional independence assumption:

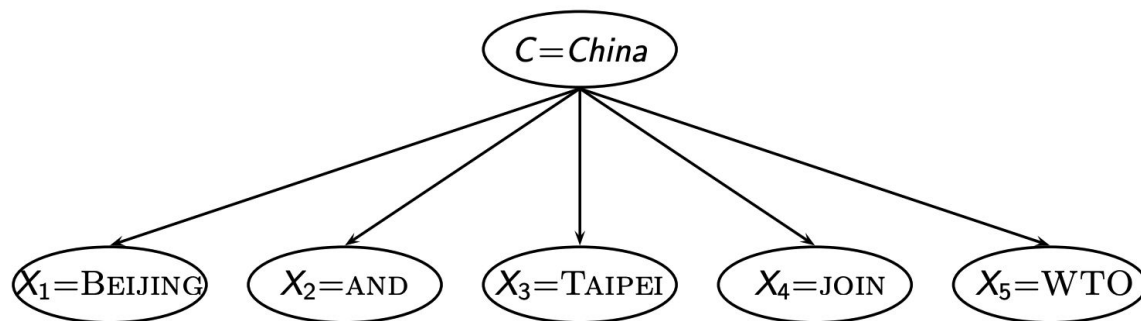
$$P(d|c) = P(\langle t_1, \dots, t_{n_d} \rangle | c) = \prod_{1 \leq k \leq n_d} P(X_k = t_k | c)$$

- We assume that the probability of observing the conjunction of attributes is equal to the product of the individual probabilities  $P(X_k = t_k | c)$ . Recall from earlier the estimates for these conditional probabilities:

$$\hat{P}(t|c) = \frac{T_{ct}+1}{(\sum_{t' \in V} T_{ct'})+B}$$



# NB as a Generative Model



$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c)$$

- Generate a class with probability  $P(c)$
- Generate each of the words (in their respective positions), conditional on the class, but independent of each other, with probability  $P(t_k|c)$
- To classify docs, we “reengineer” this process and find the class that is most likely to have generated the doc.

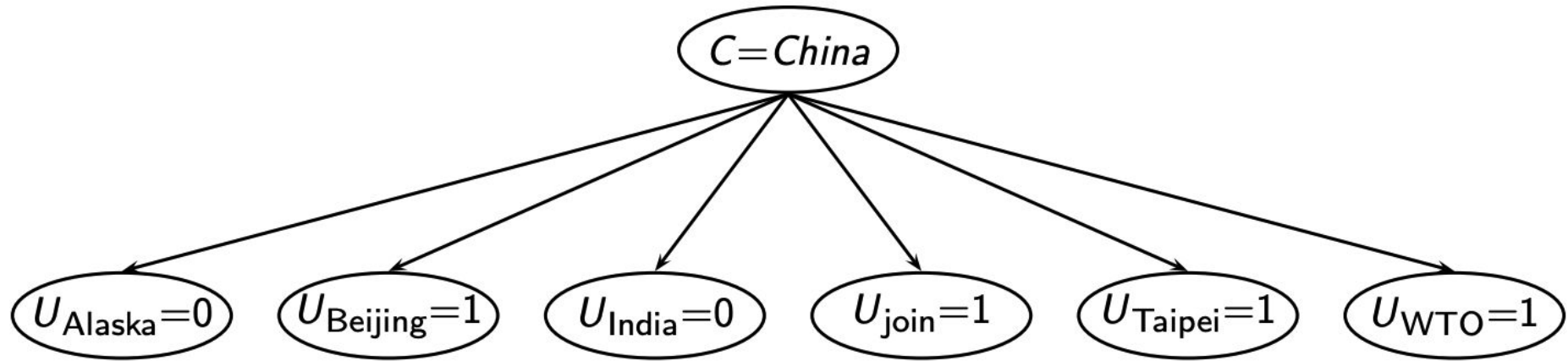


# Naïve Bayes: The Second Independence Assumption

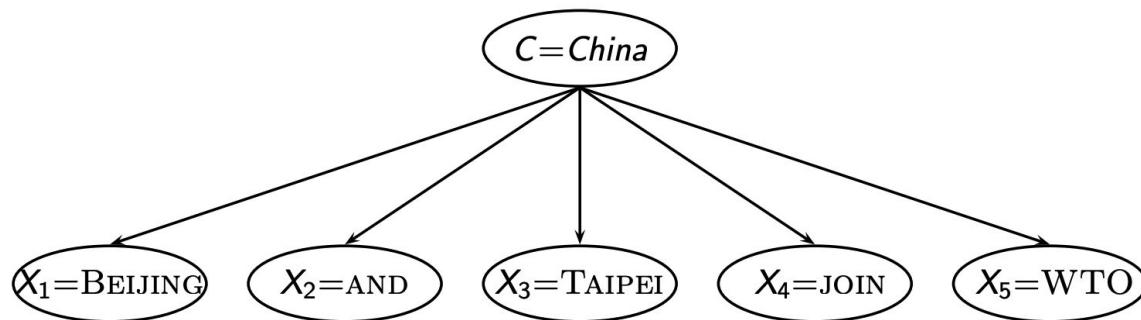
- $\hat{P}(X_{k_1} = t|c) = \hat{P}(X_{k_2} = t|c)$
- For example, for a document in the class UK, the probability of generating queen in the first position of the document is the same as generating it in the last position.
- The two independence assumptions amount to the bag of words model.



# Bernoulli NB



# Gaussian NB



$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c)$$

- Each  $P(t_k|c) \sim N(m, s)$ 
  - Normally distributed with parameter  $m$  and  $s$  estimated on the training set





# Violations of the Independence Assumptions

- Conditional independence:
  - $P(\langle t_1, \dots, t_{n_d} \rangle | c) = \prod_{1 \leq k \leq n_d} P(X_k = t_k | c)$
- Positional independence:
  - $\hat{P}(X_{k_1} = t | c) = \hat{P}(X_{k_2} = t | c)$
- The independence assumptions do not really hold of documents written in natural language.
- Exercise
  - Examples for why conditional independence assumption is not really true?
  - Examples for why positional independence assumption is not really true?
- How can Naive Bayes work if it makes such inappropriate assumptions?



# Why does NB work?

- Naive Bayes can work well even though conditional independence assumptions are badly violated.
- Example:

	$c_1$	$c_2$	class selected
true probability $P(c d)$	0.6	0.4	$c_1$
$\hat{P}(c) \prod_{1 \leq k \leq n_d} \hat{P}(t_k c)$	0.00099	0.00001	
NB estimate $\hat{P}(c d)$	0.99	0.01	$c_1$

- Double counting of evidence causes underestimation (0.01) and overestimation (0.99).
- Classification is about predicting the correct class and not about accurately estimating probabilities.
- Naive Bayes is terrible for correct estimation . . .
  - but it often performs well at accurate prediction (choosing the correct class).



# NB is it so Naive?

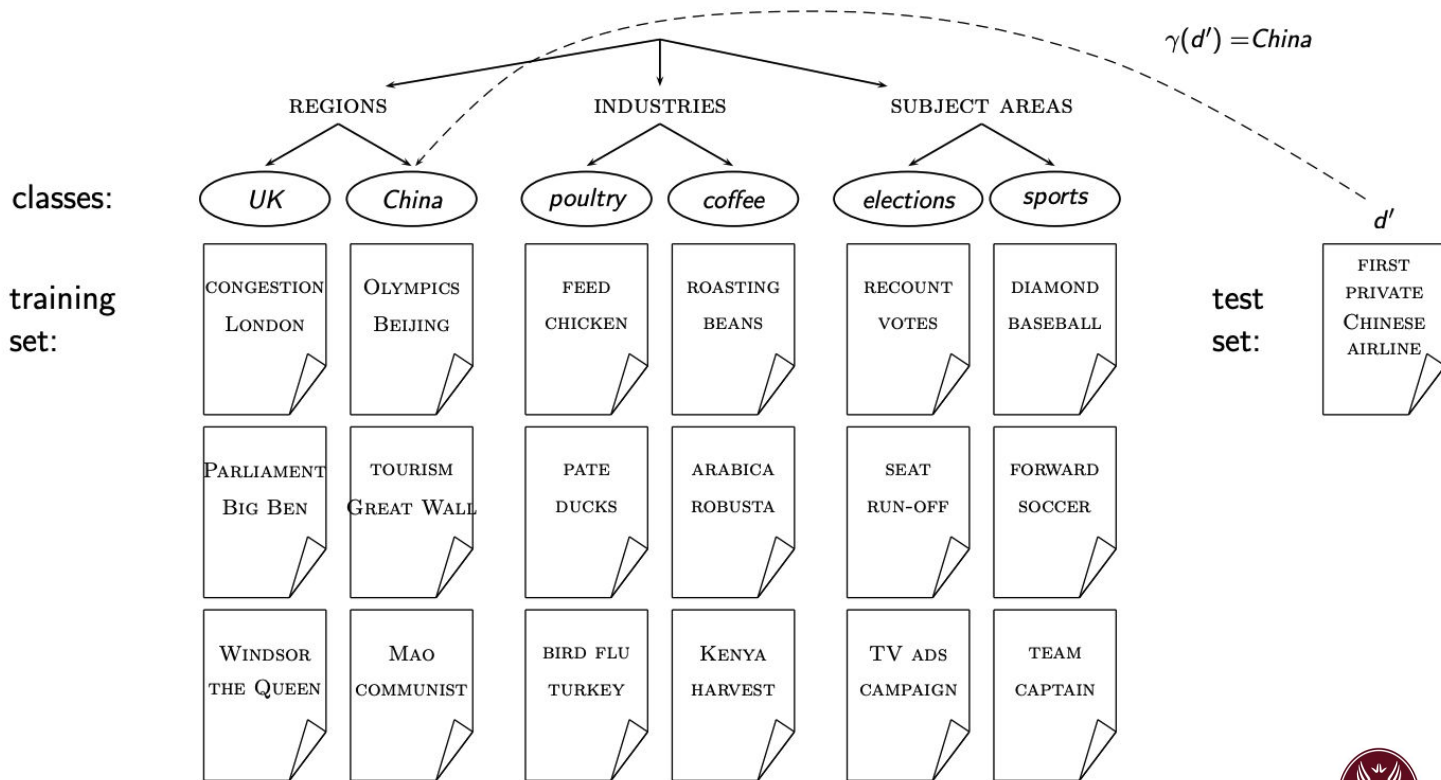
- Naive Bayes has won some bakeoffs (e.g., KDD-CUP 97)
- More robust to nonrelevant features than some more complex learning methods
- More robust to concept drift (changing of definition of class over time) than some more complex learning methods
- Better than methods like decision trees when we have **many equally important features**
- A good dependable baseline for text classification (but not the best)
- Optimal if independence assumptions hold (never true for text, but true for some domains)
- Very fast
- Low storage requirements



# Evaluation



# Reuters Collection



# The Reuters Collection

symbol	statistic	value
<i>N</i>	documents	800,000
<i>L</i>	avg. # word tokens per document	200
<i>M</i>	word types	400,000

type of class	number	examples
region	366	UK, China
industry	870	poultry, coffee
subject area	126	elections, sports



# An example of a Reuters Document

May 4, 2021  
2:17 PM CEST

India

## 'Last resort': Desperate for oxygen, Indian hospitals go to court

4 minute read

Aditya Kalra

[f](#) [t](#) [l](#) [e](#)



Patients suffering from the coronavirus disease (COVID-19) get treatment at the casualty ward in Lok Nayak Jai Prakash

A court in India's capital New Delhi has become the last hope for many hospitals [struggling to get oxygen for COVID-19 patients](#) as supplies run dangerously short while government officials bicker over who is responsible.



# Evaluating Classification

- Evaluation must be done on test data that are independent of the training data, i.e., training and test sets are disjoint.
- It's easy to get good performance on a test set that was available to the learner during training (e.g., just memorize the test set).
- Measures: Precision, recall,  $F_1$ , classification accuracy

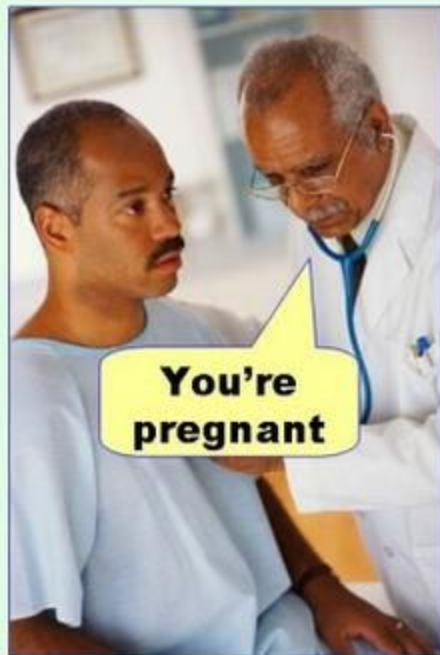




# The Conf

Actu

**Type I error**  
(false positive)



**Type II error**  
(false negative)



# $F_1$ Measure

- $F_1$  allows us to trade off precision against recall.

$$F_1 = \frac{1}{\frac{1}{2}\frac{1}{P} + \frac{1}{2}\frac{1}{R}} = \frac{2PR}{P + R}$$

- This is the harmonic mean of  $P$  and  $R$ :  $\frac{1}{F} = \frac{1}{2}(\frac{1}{P} + \frac{1}{R})$



# Averaging: Micro vs. Macro

- We now have an evaluation measure ( $F_1$ ) for one class.
- But we also want a single number that measures the aggregate performance over all classes in the collection.
- Macroaveraging
  - Compute  $F_1$  for each of the C classes
  - Average these C numbers
- Microaveraging
  - Compute TP, FP, FN for each of the C classes
  - Sum these C numbers (e.g., all TP to get aggregate TP)
  - Compute F1 for aggregate TP, FP, FN



# Takeaway Messages

- Text classification: definition & relevance to information retrieval
- Naive Bayes: simple baseline text classifier
- Theory: derivation of Naive Bayes classification rule & analysis
- Evaluation of text classification: how do we know it worked / didn't work?

