CS4025: Naïve Bayesian Classifier for Text

Chenghua Lin
Computing Science

What you should know

- Why it is called naïve?
  - The independence assumption
- How to build a naïve Bayes classifier
  - What happen in training time
  - What happen in testing time
- How to deal with unseen features
  - Smoothing

Overview

- We have learnt -- Bayes rule
- Today we learn about NB:
  - How to turn Bayes rule into a classifier, i.e., <u>a naïve Bayes classifier</u>
  - A supervised probabilistic model of the observed data
  - Can be used to predict the class label of new/unseen data

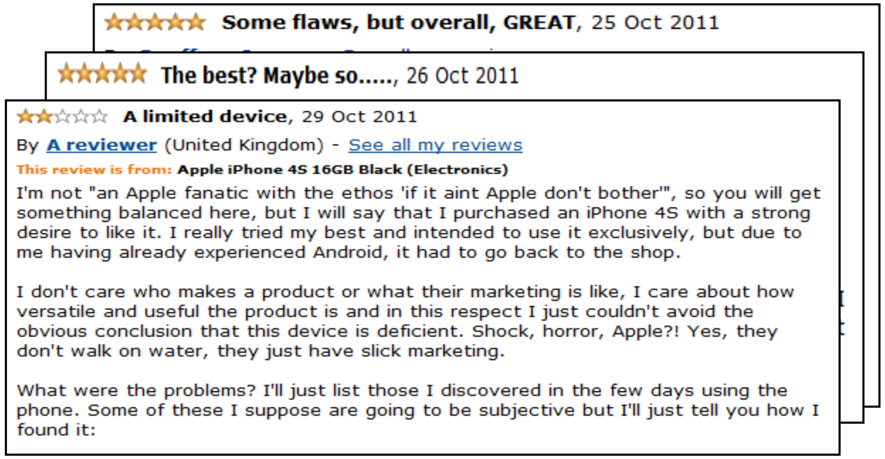
Supervised Classification

- **Supervised learning**: the machine learning task of inferring a function from labeled training data

- Given:
  - **Target**: a fixed set of **classes**: Y = {y_1, y_2,…, y_n}, e.g. {sports, politics, …, music}
  - **Training data**: a collection of data objects *X with known classes Y, i.e.* (X, Y) = {(x_1,y_1), (x_2, y_2)… ($x_{n,}$,y_n)}. E.g {(doc1, sports), (doc2, sports), (doc3, music) …}.
  - **Testing data**: a description of an unseen instance, D_{new} *e.g. a new document **without** class label information*

- Goal:
  - Predict the category/class of D_{new}: $y(x) \in Y$, where $y(x)$ is a *classification function, aka trained model***,** whose domain is *X* and whose range is *Y*.

Supervised Classifier



★★★★★ **Some flaws, but overall, GREAT**, 25 Oct 2011

★★★★★ **The best? Maybe so.....**, 26 Oct 2011

★★☆☆☆ **A limited device**, 29 Oct 2011

By **A reviewer** (United Kingdom) - See all my reviews

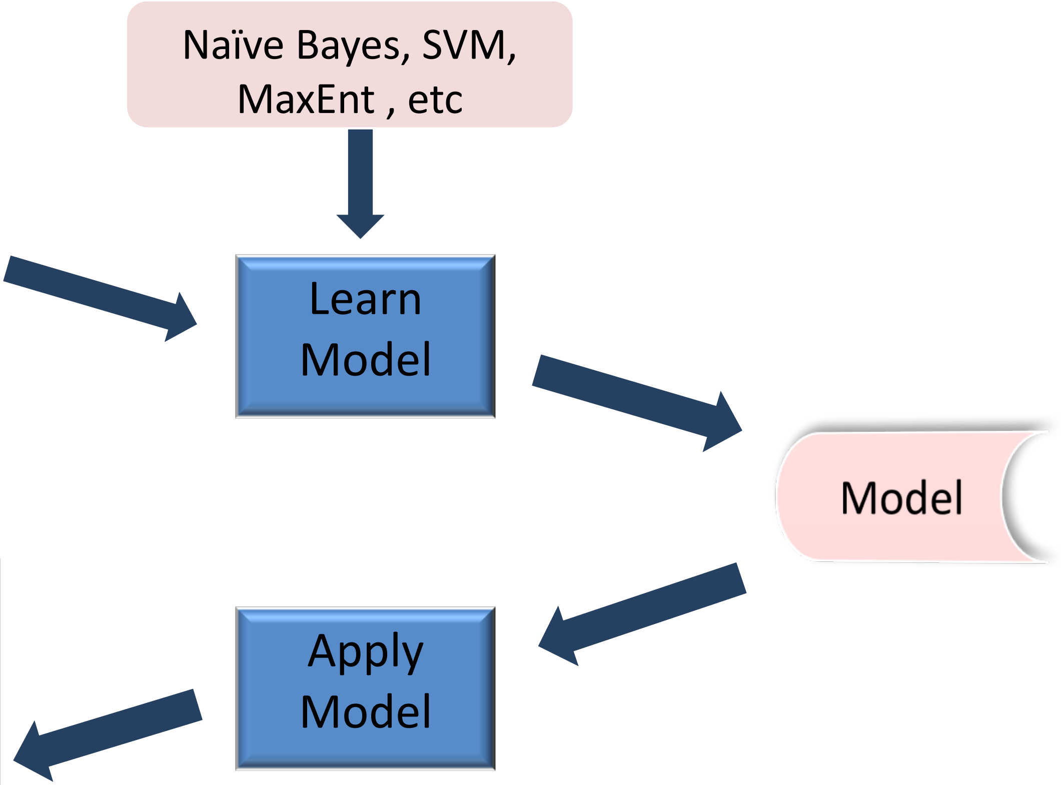
This review is from: **Apple iPhone 4S 16GB Black (Electronics)**

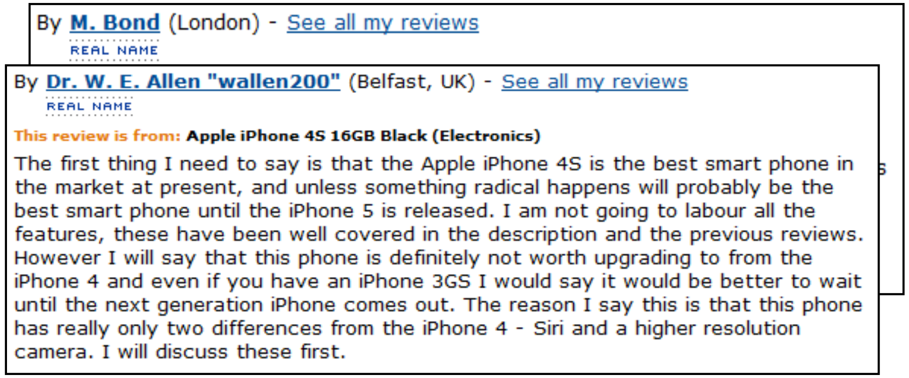
I'm not "an Apple fanatic with the ethos 'if it aint Apple don't bother'", so you will get something balanced here, but I will say that I purchased an iPhone 4S with a strong desire to like it. I really tried my best and intended to use it exclusively, but due to me having already experienced Android, it had to go back to the shop.

I don't care who makes a product or what their marketing is like, I care about how versatile and useful the product is and in this respect I just couldn't avoid the obvious conclusion that this device is deficient. Shock, horror, Apple?! Yes, they don't walk on water, they just have slick marketing.

What were the problems? I'll just list those I discovered in the few days using the phone. Some of these I suppose are going to be subjective but I'll just tell you how I found it:

Training set





By **M. Bond** (London) - See all my reviews
REAL NAME

By **Dr. W. E. Allen "wallen200"** (Belfast, UK) - See all my reviews
REAL NAME

This review is from: **Apple iPhone 4S 16GB Black (Electronics)**

The first thing I need to say is that the Apple iPhone 4S is the best smart phone in the market at present, and unless something radical happens will probably be the best smart phone until the iPhone 5 is released. I am not going to labour all the features, these have been well covered in the description and the previous reviews. However I will say that this phone is definitely not worth upgrading to from the iPhone 4 and even if you have an iPhone 3GS I would say it would be better to wait until the next generation iPhone comes out. The reason I say this is that this phone has really only two differences from the iPhone 4 - Siri and a higher resolution camera. I will discuss these first.

Test set

- Rely on syntactic or co-occurrence patterns in large text corpora

What can you do with classification?

Applications:

- Topic classification
  - Given a news article, predict the topic of the article, e.g., *finance* vs. *sports*
- Spam Classification
  - Given an email, predict whether it is a spam or not
- Medical Diagnosis
  - Given a list of symptoms, predict whether a patient has cancer or not
- Weather
  - Based on temperature, humidity, etc… predict if it will rain tomorrow

Supervised Learning for Classification

- Many commercial systems (partly) rely on classification techniques (Google AdSense and Trends, Yahoo!, …)
  - Naive Bayes (simple, robust method)
  - Decision trees (intuitive, powerful)
  - Support-vector machines (new, more powerful)
  - plus many other methods …
- No free lunch: requires hand-classified training data
- But data can be built up (and refined) by amateurs, e.g., Amazon Mechanical Turk
- Note that many commercial systems use a mixture of methods

The Bayes Rule

Prior

Likelihood

$$\boxed{p(Y \mid X_1,\ldots,X_n) = \frac{P(X_1,\ldots,X_n \mid Y)P(Y)}{P(X_1,\ldots,X_n)}}$$

Posterior

Normalization Constant

$P(Y)$: Prior belief (probability of hypothesis Y before seeing any data)

$P(X_1,\ldots,X_n \mid Y)$: Likelihood (probability of the data if the hypothesis Y is true)

$P(X_1,\ldots,X_n)$: Data evidence (marginal probability of data)

$P(Y \mid X_1,\ldots,X_n)$: Posterior (probability of hypothesis Y after having seen the data)

Bayes Classifiers

Task: Given a **trained Bayes classifier**, predict a new instance *D* based on a tuple of attribute values into one of the classes $y_j \in Y$

P('sports' |"The football match of the year ….")

$$D = \langle x_1, x_2, \ldots, x_n \rangle$$

Apply Bayes rule!

$$y_D = \underset{y_j \in Y}{\operatorname{argmax}} P(y_j \mid x_1, x_2, \ldots, x_n)$$

$$= \underset{y_j \in Y}{\operatorname{argmax}} \frac{P(x_1, x_2, \ldots, x_n \mid y_j) P(y_j)}{P(x_1, x_2, \ldots, x_n)}$$

$$\propto \underset{y_j \in Y}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid y_j) P(y_j)$$

Can be learned from Training data!

argmax: return the argument value for which the probability expression takes the maximum value

Bayes Classifiers

- $P(y_j)$
  - The probability of class label y_j, e.g. Prob(politics)
  - Can be estimated from the frequency of classes in the training examples.
- $P(x_1, x_2, \ldots, x_n | y_j)$
  - The probability of generating observed instances/data given a class label y_j.
  - For instance, given a class label '*sports*', what is the probability of observing document *d*,

    e.g. Prob("The football match of the year ….."| 'sports')
  - Could only be estimated if a very, very **large** number of training examples was available.
  - Why??

Issues with the Bayes Model

- The issue with **explicitly modeling** $P(x_1,x_2,\ldots,x_n|y_j)$
  - Usually way too many parameters parameters
  - We’ ll run out of space
  - We’ ll run out of time, because …

O(|X|^n•|Y|)

$$
\begin{aligned}
&P(x_1,x_2,\ldots,x_n \mid y_j)\\
&= P(x_1 \mid y_j)P(x_2,\ldots,x_n \mid y_j,x_1)\\
&= P(x_1 \mid y_j)P(x_2 \mid y_j,x_1)P(x_3,\ldots,x_n \mid y_j,x_1,x_2)\\
&= P(x_1 \mid y_j)P(x_2 \mid y_j,x_1)\ldots.P(x_n \mid y_j,x_1,x_2,\ldots,x_n)
\end{aligned}
$$

The Independence Assumption

- Assume A and B are Boolean Random variables. Then

"A and B are independent"

if and only if

$$P(A|B) = P(A)$$

"A and B are independent" is often notated as

$$A \perp B$$

Naïve Bayes Model

- The problem with **explicitly modeling** P(X_1,…,X_n|Y) is that there are usually way too many parameters:

$$P(x_1, x_2, \ldots, x_n \mid y_j)$$

$$= P(x_1 \mid y_j) P(x_2, \ldots, x_n \mid y_j, x_1)$$

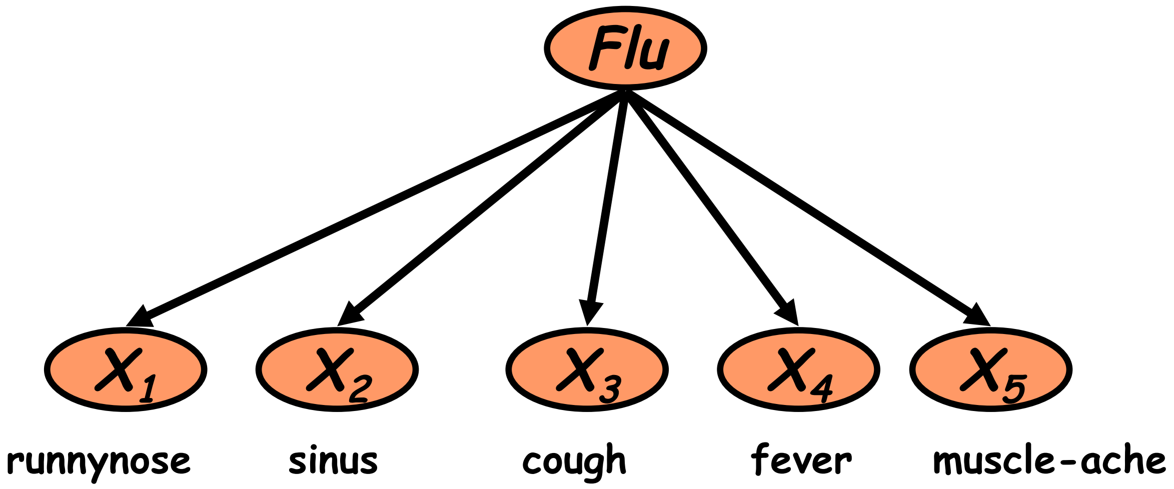
$$= P(x_1 \mid y_j) P(x_2 \mid y_j, x_1) P(x_3, \ldots, x_n \mid y_j, x_1, x_2)$$

$$= P(x_1 \mid y_j) P(x_2 \mid y_j, x_1) \ldots . P(x_n \mid y_j, x_1, x_2, \ldots, x_n)$$

- **Solution**: assume that all features are independent **given the class label Y**, yielding the naïve Bayes version

$$P(x_1, x_2, \ldots, x_n \mid y_j) = \prod_{i=1}^{n} P(x_i \mid y_j)$$

The Independence Assumption



- Features (term presence) are *independent* of each other given the class:

$$P(X_1,\ldots,X_5 \mid Y) = P(X_1 \mid Y) \bullet P(X_2 \mid Y) \bullet \cdots \bullet P(X_5 \mid Y)$$

NB Model Parameters

- For the Naive Bayes classifier, we need to “learn” two functions:
  - the likelihood
  - the prior

$$P(Y|X_1,\ldots,X_n) = \frac{\overbrace{P(X_1,\ldots,X_n|Y)}^{\text{Likelihood}}\overbrace{P(Y)}^{\text{Prior}}}{\underbrace{P(X_1,\ldots,X_n)}_{\text{Normalization Constant}}}$$

How to estimate these parameters??? We will see later on

Multinomial Naïve Bayes Training

Learning Algorithm for Text Classification

- From training corpus, extract *Vocabulary*
- Calculate $P(y_j)$ *each* y_j *in Y*

$$P(y_j) = \frac{|\, docs_j \,|}{|\, \text{total \# documents} \,|}$$

- Calculate $P(x_k \mid y_j)$ terms
  - for each word x_k in the vocabulary
  - n_k : number of occurrences of x_k in a subset of documents for which the target class is y_j
  - n : total number of word tokens in a subset of documents for which the target class is y_j

$$P(x_k \mid y_j) = \frac{n_k}{n}$$

Smoothing

Note: the vocabulary is derived from the entire training corpus for all possible labels.

- This means that some words may only appear in some particular classes → $n_k = 0$

$$P(x_k \mid y_j) = \frac{n_k}{n}$$

- Calculate $P(x_k \mid y_j)$ terms with **add one smoothing**
  - n_k : number of occurrences of x_k in a subset of documents for which the target class is y_j
  - n : total number of word tokens in a subset of documents for which the target class is y_j

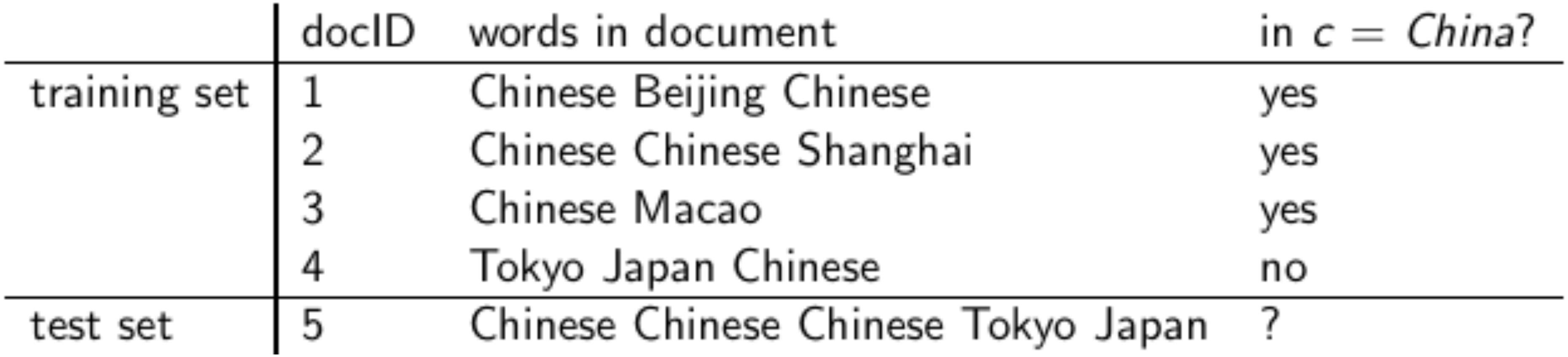
$$P(x_k \mid y_j) = \frac{n_k + 1}{n + |Vocabulary|}$$

Naïve Bayes: Classifying

- For all word positions in the testing document *d* which contain tokens found in *Vocabulary*
- Return y_d, where

$$y_d = \operatorname*{argmax}_{y_\mathrm{j} \in Y} P(y_j) \prod_{i \in positions} P(x_i \mid y_j)$$

Exercise



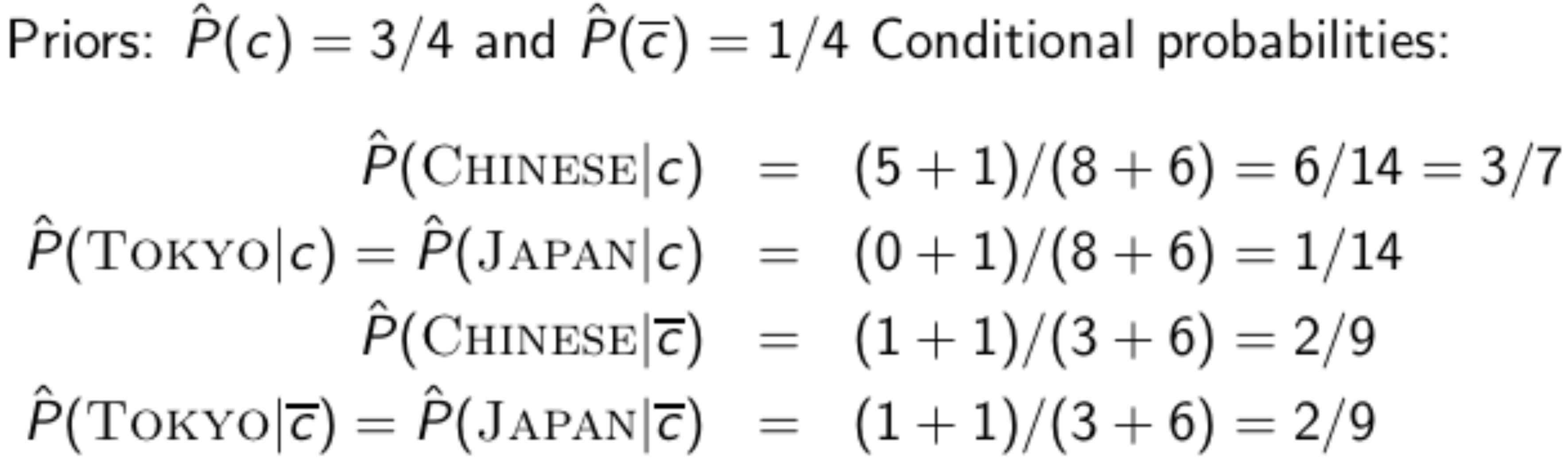
	docID	words in document	in $c = China$?
training set	1	Chinese Beijing Chinese	yes
	2	Chinese Chinese Shanghai	yes
	3	Chinese Macao	yes
	4	Tokyo Japan Chinese	no
test set	5	Chinese Chinese Chinese Tokyo Japan	?

- Estimate model parameters of a Naive Bayes classifier
- Predict the label for the test document

$$P(y_j) = \frac{|docs_j|}{|\text{total \# documents}|} \qquad P(x_k \mid y_j) = \frac{n_k + 1}{n + |Vocabulary|}$$

Example: Training Phase

Model parameter estimation

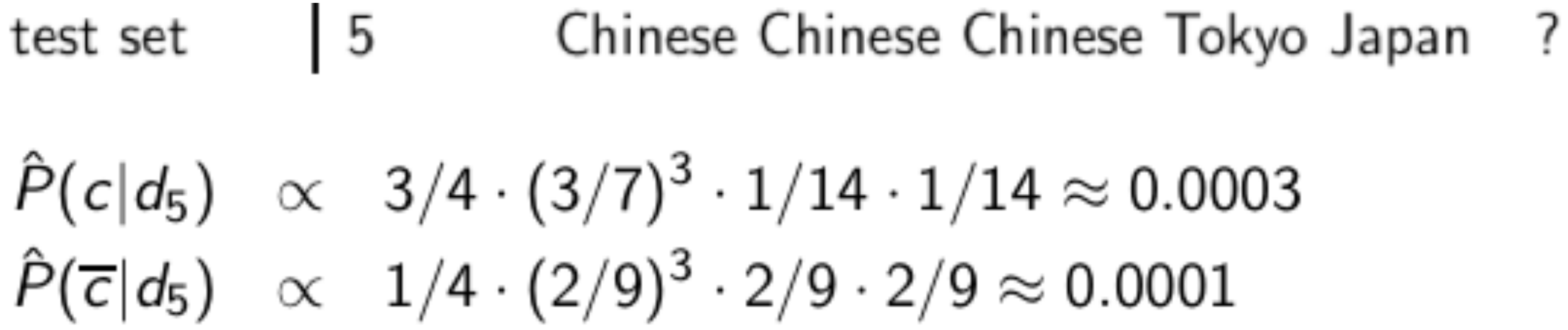


Priors: $\hat{P}(c) = 3/4$ and $\hat{P}(\overline{c}) = 1/4$ Conditional probabilities:

$$\begin{aligned}
\hat{P}(\textsc{Chinese}|c) &= (5+1)/(8+6) = 6/14 = 3/7\\
\hat{P}(\textsc{Tokyo}|c) = \hat{P}(\textsc{Japan}|c) &= (0+1)/(8+6) = 1/14\\
\hat{P}(\textsc{Chinese}|\overline{c}) &= (1+1)/(3+6) = 2/9\\
\hat{P}(\textsc{Tokyo}|\overline{c}) = \hat{P}(\textsc{Japan}|\overline{c}) &= (1+1)/(3+6) = 2/9
\end{aligned}$$

Example: Testing Phase

Classification



test set	5	Chinese Chinese Chinese Tokyo Japan	?

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

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

Thus, the classifier assigns the test document to *c* = *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.

What you should know

- Why it is called naïve?
  - The independence assumption
- How to build a naïve Bayes classifier
  - What happen in training time
  - What happen in testing time
- How to deal with unseen features in training example
  - Smoothing

Conclusions

- Naïve Bayes is:
  - Really easy to implement and often works well
  - Often a good first thing to try
- Actually, the Naïve Bayes assumption is almost never true
- Still… Naïve Bayes often performs surprisingly well even when its assumption does not hold