

# Text Classification and Naïve Bayes

# The Task of Text Classification



# Is this spam?

**Subject:** Important notice!

**From:** Stanford University <newsforum@stanford.edu>

**Date:** October 28, 2011 12:34:16 PM PDT

**To:** undisclosed-recipients;;

---

**Greats News!**

You can now access the latest news by using the link below to login to Stanford University News Forum.

<http://www.123contactform.com/contact-form-StanfordNew1-236335.html>

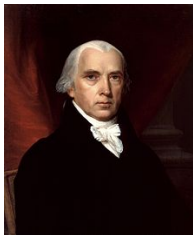
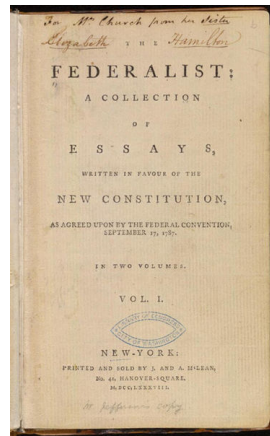
Click on the above link to login for more information about this new exciting forum. You can also copy the above link to your browser bar and login for more information about the new services.

© Stanford University. All Rights Reserved.

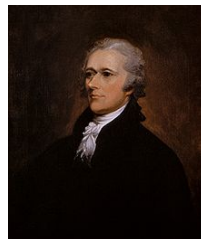


# Who wrote which Federalist papers?

- 1787-8: anonymous essays try to convince New York to ratify U.S Constitution: Jay, Madison, Hamilton.
- Authorship of 12 of the letters in dispute
- 1963: solved by Mosteller and Wallace using Bayesian methods



James Madison



Alexander Hamilton



# Male or female author?

女：用較多代名詞

男：用較多事實、The

1. By 1925 present-day Vietnam was divided into three parts under French colonial rule. **The** southern region embracing Saigon and the Mekong delta was **the** colony of Cochin-China; **the** central area with its imperial capital at Hue was the protectorate of Annam...
2. Clara never failed to be astonished by the extraordinary felicity of **her** own name. **She** found it hard to trust herself to the mercy of fate, which had managed over the years to convert **her** greatest shame into one of her greatest assets...



# Positive or negative movie review?

分析評論在商場上是很重要的技術



- unbelievably **disappointing**



- Full of zany characters and richly applied satire, and some great plot twists



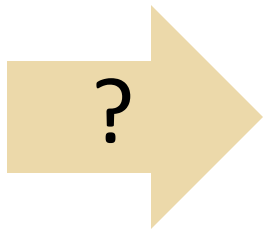
- this is the **greatest** screwball comedy ever filmed



- It was **pathetic**. The **worst** part about it was the boxing scenes.

# MeSH Subject Category Hierarchy

# MeSH Subject Category Hierarchy



- Antagonists and Inhibitors
- Blood Supply
- Chemistry
- Drug Therapy
- Embryology
- Epidemiology
- ...



# Text Classification

自動為文章分類

- Assigning subject categories, topics, or genres
- Spam detection 找出垃圾信件
- Authorship identification 判斷一篇文章的作者是誰
- Age/gender identification 判斷作者的年紀與性別
- Language Identification 判斷每個語言的文法結構異同
- Sentiment analysis 文字的情緒、觀點分析
- ...



# Text Classification: definition

- *Input:*
  - a document  $d$
  - a fixed set of classes  $C = \{c_1, c_2, \dots, c_J\}$
- *Output:* a predicted class  $c \in C$





# Classification Methods:

## Hand-coded rules

- Rules based on combinations of words or other features
  - spam: black-list-address OR ("dollars" AND "have been selected")
- Accuracy can be high
  - If rules carefully refined by expert
- But building and maintaining these rules is expensive

找垃圾信最直覺的做法：列出寄件人黑名單、列出垃圾信最可能出現的字，再一一去比對  
但這樣雖然很準確卻很慢，可以透過機器學習建立一個classifier(下一頁)



# Classification Methods: Supervised Machine Learning

- *Input:*
  - a document  $d$
  - a fixed set of classes  $C = \{c_1, c_2, \dots, c_J\}$
  - A training set of  $m$  hand-labeled documents  $(d_1, c_1), \dots, (d_m, c_m)$
- *Output:*
  - a learned classifier  $\gamma: d \rightarrow c$



# Classification Methods: Supervised Machine Learning

- Any kind of classifier
  - Naïve Bayes
  - Logistic regression
  - Support-vector machines
  - k-Nearest Neighbors
- ...

# Text Classification and Naïve Bayes

# The Task of Text Classification

# Text Classification and Naïve Bayes

# Naïve Bayes (I)



# Naïve Bayes Intuition

- Simple (“naïve”) classification method based on Bayes rule
- Relies on very simple representation of document
  - **Bag of words**

最直觀的做法：先定義每個分類出現的關鍵字，再去搜尋文本中包含哪些分類的關鍵字即可



# The bag of words representation

Y (

I love this movie! It's sweet,  
but with satirical humor. The  
dialogue is great and the  
adventure scenes are fun... It  
manages to be whimsical and  
romantic while laughing at the  
conventions of the fairy tale  
genre. I would recommend it to  
just about anyone. I've seen  
it several times, and I'm  
always happy to see it again  
whenever I have a friend who  
hasn't seen it yet.

) = C

建立一個方法  
輸入文章內容  
輸出他的分類  
例如：影評是  
正評還是負評





# The bag of words representation

Y (

I **love** this movie! It's **sweet**, but with **satirical** humor. The dialogue is **great** and the adventure scenes are **fun**... It manages to be **whimsical** and **romantic** while **laughing** at the conventions of the fairy tale genre. I would **recommend** it to just about anyone. I've seen it **several** times, and I'm always **happy** to see it **again** whenever I have a friend who hasn't seen it yet.

) = C

只要抓出關鍵字去判斷即可，沒必要每個字都去判斷，很浪費時間







# The bag of words representation: using a subset of words

Y (

```
x love xxxxxxxxxxxxxxxxxxxx sweet
xxxxxxxx satirical xxxxxxxxxxxx
xxxxxxxxxxxx great xxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxx fun xxxx
xxxxxxxxxxxxxxxx whimsical xxxx
romantic xxxx laughing
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxx recommend xxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx several xxxxxxxxxxxxxxxxxxxxxxxx
xxxxx happy xxxxxxxxxxxx again
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

)

= C







# The bag of words representation

$Y($

|           |     |
|-----------|-----|
| great     | 2   |
| love      | 2   |
| recommend | 1   |
| laugh     | 1   |
| happy     | 1   |
| ...       | ... |

$) = C$



# Bag of words for document classification

Test document

parser  
language  
label  
translation  
...

Machine Learning

learning  
training  
algorithm  
shrinkage  
network...

NLP

parser  
tag  
training  
translation  
language...

Garbage Collection

garbage  
collection  
memory  
optimization  
region...

Planning

planning  
temporal  
reasoning  
plan  
language...

GUI

...

?

# Text Classification and Naïve Bayes

# Naïve Bayes (I)

# Text Classification and Naïve Bayes

# Formalizing the Naïve Bayes Classifier



# Bayes' Rule Applied to Documents and Classes

- For a document *d* and a class *c*

$$P(c | d) = \frac{P(d | c)P(c)}{P(d)}$$

要找出輸入文章所對應到的分類



# Naïve Bayes Classifier (I)

$$C_{MAP} = \operatorname{argmax}_{c \in C} P(c | d)$$

MAP is “maximum a posteriori” = most likely class

某文本d最可能被歸類的分類

就是對於每個分類c做 $P(c|d)$ 的最大值

可以用貝氏法簡化式子

$P(d)$ 是常數可以消去

$$= \operatorname{argmax}_{c \in C} \frac{P(d | c)P(c)}{P(d)}$$

Bayes Rule

$$= \operatorname{argmax}_{c \in C} \underbrace{P(d | c)}_{\text{Likelihood.}} \underbrace{P(c)}_{\text{Prior}}$$

Dropping the denominator



## Naïve Bayes Classifier (II)

$$C_{MAP} = \operatorname{argmax}_{c \in C} P(d \mid c)P(c)$$

$$= \operatorname{argmax}_{c \in C} P(x_1, x_2, \dots, x_n \mid c)P(c)$$

Document  $d$   
represented as  
features  
 $x_1 \dots x_n$

$d$ 的定義即為 $d$ 所有的單字 $x_1 \sim x_n$ 的發生機率交集





# Naïve Bayes Classifier (IV)

$$C_{MAP} = \operatorname{argmax}_{c \in C} P(x_1, x_2, \dots, x_n | c) P(c)$$

如果有 $n$ 個feature，就要處理 $X^n * C$ 個參數

$O(|X|^n \cdot |C|)$  parameters

Could only be estimated if a very, very large number of training examples was available.

How often does this class occur?

We can just count the relative frequencies in a corpus



# Multinomial Naïve Bayes Independence Assumptions

計算likelihood  $P(x_1, x_2, \dots, x_n | c)$

簡化：假設單字出現的位置不影響字意、且所有單字出現的機率都是獨立的

這是不合理的簡化，但可以用來先了解貝氏機率的概念

- **Bag of Words assumption**: Assume position doesn't matter
- **Conditional Independence**: Assume the feature probabilities  $P(x_i | c_j)$  are independent given the class  $c$ .

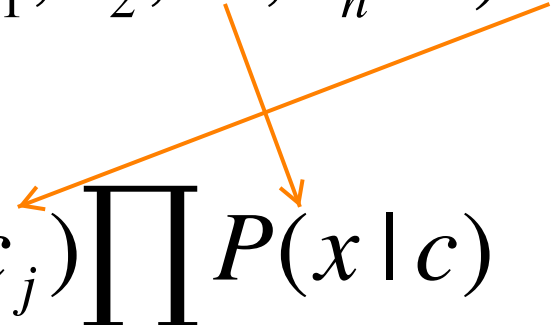
假設所有單字的出現是獨立事件，才能這樣簡化

$$P(x_1, \dots, x_n | c) = P(x_1 | c) \cdot P(x_2 | c) \cdot P(x_3 | c) \cdot \dots \cdot P(x_n | c)$$



# Multinomial Naïve Bayes Classifier

$$c_{MAP} = \operatorname{argmax}_{c \in C} P(x_1, x_2, \dots, x_n | c) P(c)$$

$$c_{NB} = \operatorname{argmax}_{c \in C} P(c_j) \prod_{x \in X} P(x | c)$$


因為獨立，所以能改寫成連乘



# Applying Multinomial Naive Bayes Classifiers to Text Classification

positions  $\leftarrow$  all word positions in test document

考慮每個單字 $x_j$ 出現的位置

走訪整篇文章，並記算每一個位置的每一個字

$$c_{NB} = \operatorname{argmax}_{c_j \in C} P(c_j) \prod_{i \in \text{positions}} P(x_i | c_j)$$

Class 1:  $P(c_1) \prod P(x_i | c_1)$

Class 2:  $P(c_2) \prod P(x_i | c_2)$

找出算出來值最大的Class，就是答案

...

Class N:  $P(c_N) \prod P(x_i | c_N)$

# Text Classification and Naïve Bayes

# Formalizing the Naïve Bayes Classifier

# Text Classification and Naïve Bayes

# Naïve Bayes: Learning



# Learning the Multinomial Naïve Bayes Model

- First attempt: maximum likelihood estimates
  - simply use the frequencies in the data

Prior

$$\hat{P}(c_j) = \frac{\text{doccount}(C = c_j)}{N_{doc}}$$

符合Cj的文本 / 所有文本  
= 任意文本符合Cj的機率

Likelihood

$$\hat{P}(w_i | c_j) = \frac{\text{count}(w_i, c_j)}{\sum_{w \in V} \text{count}(w, c_j)}$$

給定的單字集合 $w_i$ 存在Cj的數量 / Cj包含的所有單字 $w$   
= Cj存在 $w_i$ 的機率



# Parameter estimation

$$\hat{P}(w_i | c_j) = \frac{\text{count}(w_i, c_j)}{\sum_{w \in V} \text{count}(w, c_j)}$$

fraction of times word  $w_i$  appears  
among all words in documents of topic  $c_j$

- Create **mega-document** for topic  $j$  by concatenating all docs in this topic
  - Use frequency of  $w$  in mega-document





# Problem with Maximum Likelihood

- What if we have seen no training documents with the word *fantastic* and classified in the topic **positive** (*thumbs-up*)?

$$\hat{P}(\text{"fantastic"} \mid \text{positive}) = \frac{\text{count}(\text{"fantastic"}, \text{positive})}{\sum_{w \in V} \text{count}(w, \text{positive})} = 0$$

- Zero probabilities cannot be conditioned away, no matter the other evidence! 如果某個單字未被訓練到某個class，那這個class被選上的機率恆等於零  
這是不對的，所以要透過+1來避免這件事情發生

$$c_{MAP} = \operatorname{argmax}_c \hat{P}(c) \prod_i \hat{P}(x_i \mid c)$$



# Laplace (add-1) smoothing for Naïve Bayes

$$\begin{aligned}\hat{P}(w_i | c) &= \frac{\text{count}(w_i, c) + 1}{\sum_{w \in V} (\text{count}(w, c) + 1)} \\ &= \frac{\text{count}(w_i, c) + 1}{\left( \sum_{w \in V} \text{count}(w, c) \right) + |V|}\end{aligned}$$



# Multinomial Naïve Bayes: Learning

參考p.44

$n_k = \text{count}(w_i, c) =$  給定的所有單字出現在c的數目

$n = \sum \text{count}(w, c) =$  class c所有單字數(可重複)

Vocabulary = 所有class的所有字數(不重複)

- From training corpus, extract *Vocabulary*

—隨機文本為 $C_j$ 的機率

- Calculate  $P(c_j)$  terms

- For each  $c_j$  in  $C$  do

$\text{docs}_j \leftarrow$  all docs with class  $= c_j$

$$P(c_j) \leftarrow \frac{|\text{docs}_j|}{|\text{total \# documents}|}$$

- Calculate  $P(w_k | c_j)$  terms

在 $C_j$ 的所有文本集合

- $\text{Text}_j \leftarrow$  single doc containing all  $\text{docs}_j$

- For each word  $w_k$  in *Vocabulary*

$n_k \leftarrow$  # of occurrences of  $w_k$  in  $\text{Text}_j$

$$P(w_k | c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha | \text{Vocabulary} |}$$

輸入的單字 $w_k$ 出現在 $\text{Text}_j$ 的數量

做平滑化

# Text Classification and Naïve Bayes

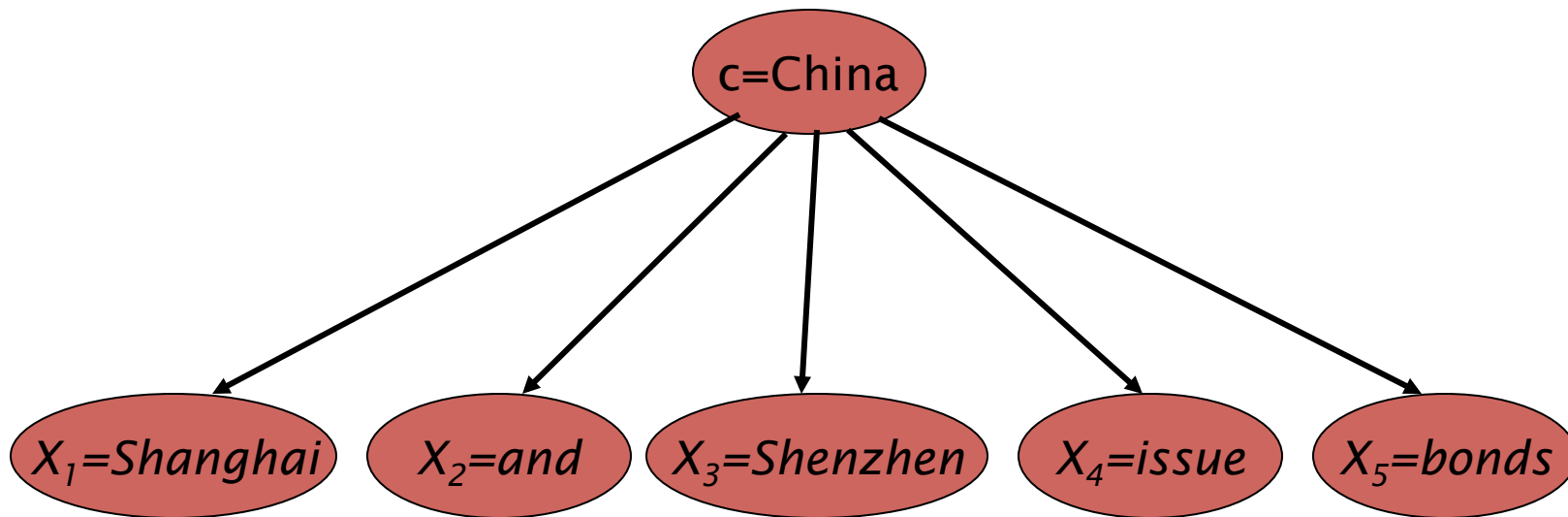
# Naïve Bayes: Learning

# Text Classification and Naïve Bayes

# Naïve Bayes: Relationship to Language Modeling



# Generative Model for Multinomial Naïve Bayes





# Naïve Bayes and Language Modeling

- Naïve bayes classifiers can use any sort of feature
  - URL, email address, dictionaries, network features
- But if, as in the previous slides
  - We use only word features
  - we use all of the words in the text (not a subset)
- Then
  - Naïve bayes has an important similarity to language modeling.



# Each class = a unigram language model

- Assigning each word:  $P(\text{word} \mid c)$
- Assigning each sentence:  $P(s \mid c) = \prod P(\text{word} \mid c)$

一個句子符合class  $c$ 的機率

= 該句子中所有文字符合class  $c$ 的機率相乘

Class  $pos$

|      |      |          |             |             |            |             |
|------|------|----------|-------------|-------------|------------|-------------|
| 0.1  | I    | <u>I</u> | <u>love</u> | <u>this</u> | <u>fun</u> | <u>film</u> |
| 0.1  | love |          |             |             |            |             |
| 0.01 | this | 0.1      | 0.1         | .05         | 0.01       | 0.1         |
| 0.05 | fun  |          |             |             |            |             |
| 0.1  | film |          |             |             |            |             |

$$P(s \mid pos) = 0.0000005$$





# Naïve Bayes as a Language Model

- Which class assigns the higher probability to  $s$ ?

## Model pos

|      |      |
|------|------|
| 0.1  | I    |
| 0.1  | love |
| 0.01 | this |
| 0.05 | fun  |
| 0.1  | film |

## Model neg

|       |      |
|-------|------|
| 0.2   | I    |
| 0.001 | love |
| 0.01  | this |
| 0.005 | fun  |
| 0.1   | film |

| <u>I</u> | <u>love</u> | <u>this</u> | <u>fun</u> | <u>film</u> |
|----------|-------------|-------------|------------|-------------|
| 0.1      | 0.1         | 0.01        | 0.05       | 0.1         |
| 0.2      | 0.001       | 0.01        | 0.005      | 0.1         |

$$P(s|\text{pos}) > P(s|\text{neg})$$

# Text Classification and Naïve Bayes

# Naïve Bayes: Relationship to Language Modeling

# Text Classification and Naïve Bayes

# Multinomial Naïve Bayes: A Worked Example



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

$$\hat{P}(w|c) = \frac{\text{count}(w,c) + 1}{\text{count}(c) + |V|}$$

|          | Doc | Words                               | Class |
|----------|-----|-------------------------------------|-------|
| Training | 1   | Chinese Beijing Chinese             | c     |
|          | 2   | Chinese Chinese Shanghai            | c     |
|          | 3   | Chinese Macao                       | c     |
|          | 4   | Tokyo Japan Chinese                 | j     |
| Test     | 5   | Chinese Chinese Chinese Tokyo Japan | ?     |

**Priors:**

$$P(c) = \frac{3}{4}$$

$$P(j) = \frac{1}{4}$$

某字在某Class的出現次數 + 1

某Class的總字數 + V

V是整個文本出現的唯一字總數

**Choosing a class:**

$$P(c|d5) \propto \frac{3}{4} * \left(\frac{3}{7}\right)^3 * \frac{1}{14} * \frac{1}{14}$$

$$\approx 0.0003$$

**Conditional Probabilities:**

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

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

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

$$P(\text{Chinese}|j) = (1+1) / (3+6) = 2/9$$

$$P(\text{Tokyo}|j) = (1+1) / (3+6) = 2/9$$

$$P(\text{Japan}|j) = (1+1) / (3+6) = 2/9$$

$$P(j|d5) \propto \frac{1}{4} * \left(\frac{2}{9}\right)^3 * \frac{2}{9} * \frac{2}{9}$$

$$\approx 0.0001$$



# Naïve Bayes in Spam Filtering

- SpamAssassin Features: 垃圾信包含的特徵
  - Mentions Generic Viagra
  - Online Pharmacy
  - Mentions millions of (dollar) ((dollar) NN,NNN,NNN.NN)
  - Phrase: impress ... girl
  - From: starts with many numbers
  - Subject is all capitals
  - HTML has a low ratio of text to image area
  - One hundred percent guaranteed
  - Claims you can be removed from the list
  - 'Prestigious Non-Accredited Universities'
  - [http://spamassassin.apache.org/tests\\_3\\_3\\_x.html](http://spamassassin.apache.org/tests_3_3_x.html)



# Summary: Naive Bayes is Not So Naive

- Very Fast, low storage requirements
- Robust to Irrelevant Features

Irrelevant Features cancel each other without affecting results

- Very good in domains with many equally important features

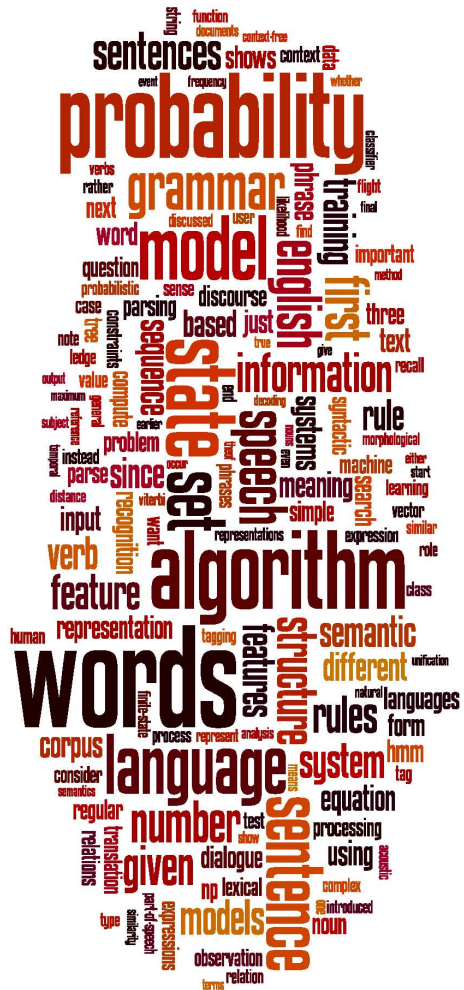
Decision Trees suffer from *fragmentation* in such cases – especially if little data

- Optimal if the independence assumptions hold: If assumed independence is correct, then it is the Bayes Optimal Classifier for problem
- A good dependable baseline for text classification
  - **But we will see other classifiers that give better accuracy**



# Text Classification and Naïve Bayes

# Multinomial Naïve Bayes: A Worked Example



# Text Classification and Naïve Bayes

# Precision, Recall, and the F measure





# The 2-by-2 contingency table

事實(True / False)

判斷

Pos / Neg

|              | correct | not correct |
|--------------|---------|-------------|
| selected     | tp      | fp          |
| not selected | fn      | tn          |

tp(truth positive):選到正確的東西

fp(false positive):選到錯誤的東西

fn(false negative):沒選到正確的東西

tn(truth negative):沒選到錯誤的東西

我們要盡量提高tp+tn，降低fp+fn



# Precision and recall

- **Precision:** % of selected items that are correct 選到的是不是正確的  
**Recall:** % of correct items that are selected 正確的是不是被選到了

$$\text{Precision} = \text{tp} / (\text{tp} + \text{fp})$$

$$\text{Recall} = \text{tp} / (\text{tp} + \text{fn})$$

分開來用這兩個指標才能精確地說明系統的準確度

如果單看  $(\text{tp} + \text{tn}) / (\text{tp} + \text{fp} + \text{fn} + \text{tn})$ ，會不客觀

通常 Precision and Recall is trade-off，所以要綜合這兩個指標去評價一個系統的好壞（下頁）

|              | correct | not correct |
|--------------|---------|-------------|
| selected     | tp      | fp          |
| not selected | fn      | tn          |



# A combined measure: F

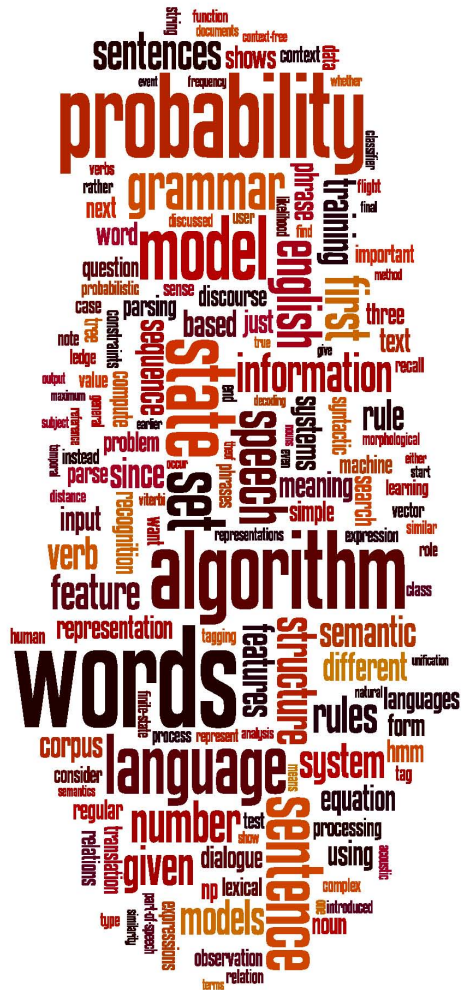
- A combined measure that assesses the P/R tradeoff is F measure (weighted harmonic mean):

$$F = \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}} = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

$\alpha$ 是使用者對於重要性的分配  
如果覺得Precision重要  
那就提高 $\alpha$   
反之則降低

- The harmonic mean is a very conservative average; see IIR § 8.3
- People usually use balanced F1 measure
  - i.e., with  $\beta = 1$  (that is,  $\alpha = \frac{1}{2}$ ):

$$F = 2PR/(P+R)$$



# Text Classification and Naïve Bayes

# Precision, Recall, and the F measure

# Text Classification and Naïve Bayes

# Text Classification: Evaluation



# More Than Two Classes: Sets of binary classifiers

- Dealing with **any-of** or **multivalue** classification
  - A document can belong to 0, 1, or >1 classes.
- For each class  $c \in C$ 
  - Build a classifier  $\gamma_c$  to distinguish  $c$  from all other classes  $c' \in C$
- Given test doc  $d$ ,
  - Evaluate it for membership in each class using each  $\gamma_c$
  - $d$  belongs to **any** class for which  $\gamma_c$  returns true



# More Than Two Classes: Sets of binary classifiers

- One-of or multinomial classification
  - Classes are mutually exclusive: each document in exactly one class
- For each class  $c \in C$ 
  - Build a classifier  $\gamma_c$  to distinguish  $c$  from all other classes  $c' \in C$
- Given test doc  $d$ ,
  - Evaluate it for membership in each class using each  $\gamma_c$
  - $d$  belongs to the one class with maximum score



# Evaluation:

## Classic Reuters-21578 Data Set

- Most (over)used data set, 21,578 docs (each 90 types, 200 tokens)
- 9603 training, 3299 test articles (ModApte/Lewis split)
- 118 categories
  - An article can be in more than one category
  - Learn 118 binary category distinctions
- Average document (with at least one category) has 1.24 classes
- Only about 10 out of 118 **categories** are large

Common categories  
(#train, #test)

- |                            |                       |
|----------------------------|-----------------------|
| • Earn (2877, 1087)        | • Trade (369, 119)    |
| • Acquisitions (1650, 179) | • Interest (347, 131) |
| • Money-fx (538, 179)      | • Ship (197, 89)      |
| • Grain (433, 149)         | • Wheat (212, 71)     |
| • Crude (389, 189)         | • Corn (182, 56)      |





# Reuters Text Categorization data set (Reuters-21578) document

<REUTERS TOPICS="YES" LEWISSPLIT="TRAIN" CGISPLIT="TRAINING-SET" OLDID="12981"  
NEWID="798">

<DATE> 2-MAR-1987 16:51:43.42</DATE>

<TOPICS><D>livestock</D><D>hog</D></TOPICS>

<TITLE>AMERICAN PORK CONGRESS KICKS OFF TOMORROW</TITLE>

<DATELINE> CHICAGO, March 2 - </DATELINE><BODY>The American Pork Congress kicks off tomorrow, March 3, in Indianapolis with 160 of the nations pork producers from 44 member states determining industry positions on a number of issues, according to the National Pork Producers Council, NPPC.

Delegates to the three day Congress will be considering 26 resolutions concerning various issues, including the future direction of farm policy and the tax law as it applies to the agriculture sector. The delegates will also debate whether to endorse concepts of a national PRV (pseudorabies virus) control and eradication program, the NPPC said.

A large trade show, in conjunction with the congress, will feature the latest in technology in all areas of the industry, the NPPC added. Reuter

<sup>57</sup>  
&#3;</BODY></TEXT></REUTERS>



# Confusion matrix c

- For each pair of classes  $\langle c_1, c_2 \rangle$  how many documents from  $c_1$  were incorrectly assigned to  $c_2$ ? 對角線的是正確的，其他都是錯判的
- $c_{3,2}$ : 90 wheat documents incorrectly assigned to poultry

Precision v

Recall →

| Docs in test set | Assigned UK | Assigned poultry | Assigned wheat | Assigned coffee | Assigned interest | Assigned trade |
|------------------|-------------|------------------|----------------|-----------------|-------------------|----------------|
| True UK          | 95          | 1                | 13             | 0               | 1                 | 0              |
| True poultry     | 0           | 1                | 0              | 0               | 0                 | 0              |
| True wheat       | 10          | 90               | 0              | 1               | 0                 | 0              |
| True coffee      | 0           | 0                | 0              | 34              | 3                 | 7              |
| True interest    | -           | 1                | 2              | 13              | 26                | 5              |
| True trade       | 0           | 0                | 2              | 14              | 5                 | 10             |



# Per class evaluation measures

**Recall:** row

$$\begin{aligned} \text{Recall(UK)} \\ &= 95 / 110 \\ &= 0.864 \end{aligned}$$

Fraction of docs in class  $i$  classified correctly:

$$\frac{c_{ii}}{\sum_j c_{ij}} \quad \frac{tp}{tp + fn}$$

**Precision:** column

$$\begin{aligned} \text{Precision(UK)} \\ &= 95 / 105 \\ &= 0.905 \end{aligned}$$

Fraction of docs assigned class  $i$  that are actually about class  $i$ :

$$\frac{c_{ii}}{\sum_j c_{ji}} \quad \frac{tp}{tp + fp}$$

Accuracy

$$\begin{aligned} &= 95 + 1 + 0 + 34 + 26 + 10 / \text{ALL} \\ &= 166 / 334 = 0.497 \end{aligned}$$

**Accuracy:** (1 - error rate)

Fraction of docs classified correctly:

$$\frac{\sum_i c_{ii}}{\sum_j \sum_i c_{ij}} \quad \frac{tp}{tp + fp + fn + tn}$$



# Micro- vs. Macro-Averaging

- If we have more than one class, how do we combine multiple performance measures into one quantity?
- **Macroaveraging**: Compute performance for each class, then average. 先算出每一個Class的P/R/F  
最後在做平均
- **Microaveraging**: Collect decisions for all classes, compute contingency table, evaluate.  
直接算出所有Class的tp, fp, fn, tn，一次算所有Class的平均



# Micro- vs. Macro-Averaging: Example

Class 1

|                 | Truth:<br>yes | Truth:<br>no |
|-----------------|---------------|--------------|
| Classifier: yes | 10            | 10           |
| Classifier: no  | 10            | 970          |

Class 2

|                 | Truth:<br>yes | Truth:<br>no |
|-----------------|---------------|--------------|
| Classifier: yes | 90            | 10           |
| Classifier: no  | 10            | 890          |

Micro Ave. Table

|                 | Truth:<br>yes | Truth:<br>no |
|-----------------|---------------|--------------|
| Classifier: yes | 100           | 20           |
| Classifier: no  | 20            | 1860         |

- Macroaveraged precision:  $(0.5 + 0.9)/2 = 0.7$  ->每個class視為平等
- Microaveraged precision:  $100/120 = .83$  ->樣本大的class會被放大檢視
- Microaveraged score is dominated by score on common classes



# Development Test Sets and Cross-validation

訓練出來的test set，會用P/R/F/Acc去評量好壞

Training set

Development Test Set

Test Set

- Metric: P/R/F1 or Accuracy
- Unseen test set
  - avoid overfitting ('tuning to the test set')
  - more conservative estimate of performance
- Cross-validation over multiple splits
  - Handle sampling errors from different datasets
  - Pool results over each split
  - Compute pooled dev set performance

Training Set Dev Test

Training Set Dev Test

Dev Test Training Set

Test Set

# Text Classification and Naïve Bayes

# Text Classification: Evaluation

# Text Classification and Naïve Bayes

# Text Classification: Practical Issues





# The Real World

- Gee, I'm building a text classifier for real, now!
- What should I do?



# No training data? Manually written rules

If (wheat or grain) and not (whole or bread) then  
Categorize as grain

- Need careful crafting
  - Human tuning on development data
  - Time-consuming: 2 days per class



# Very little data?

- Use Naïve Bayes
  - Naïve Bayes is a “high-bias” algorithm (Ng and Jordan 2002 NIPS)
- Get more labeled data
  - Find clever ways to get humans to label data for you
- Try semi-supervised training methods:
  - Bootstrapping, EM over unlabeled documents, ...



# A reasonable amount of data?

- Perfect for all the clever classifiers
  - SVM
  - Regularized Logistic Regression
- You can even use user-interpretable decision trees
  - Users like to hack
  - Management likes quick fixes



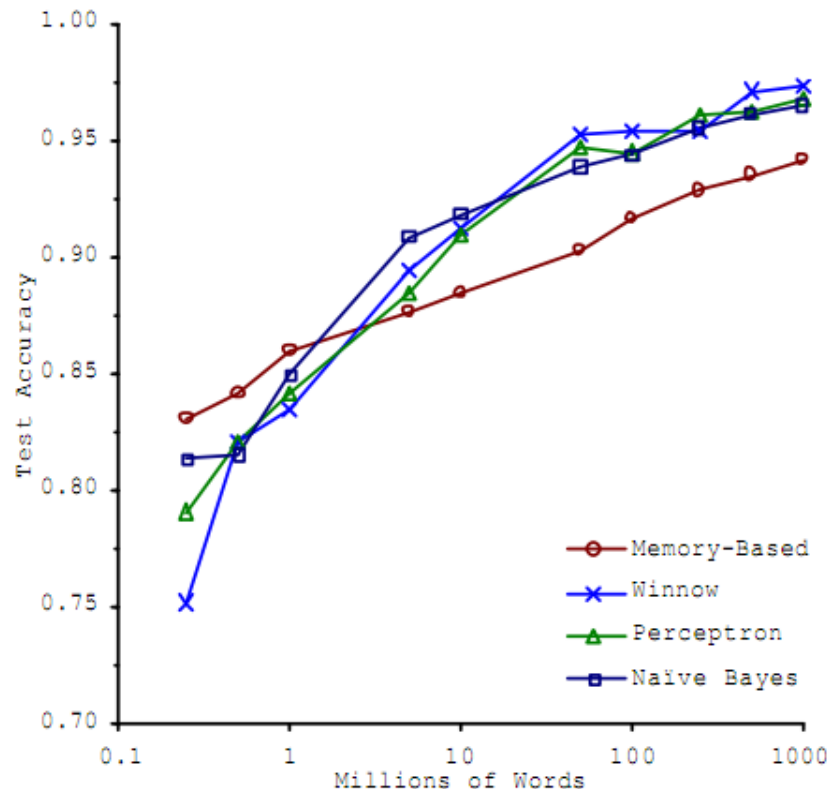
# A huge amount of data?

- Can achieve high accuracy!
- At a cost:
  - SVMs (train time) or kNN (test time) can be too slow
  - Regularized logistic regression can be somewhat better
- So Naïve Bayes can come back into its own again!



# Accuracy as a function of data size

- With enough data
  - Classifier may not matter



Brill and Banko on spelling correction



## Real-world systems generally combine:

- Automatic classification
- Manual review of uncertain/difficult/"new" cases



# Underflow Prevention: log space

- Multiplying lots of probabilities can result in floating-point underflow.
- Since  $\log(xy) = \log(x) + \log(y)$ 
  - Better to sum logs of probabilities instead of multiplying probabilities.
- Class with highest un-normalized log probability score is still most probable.

$$c_{NB} = \operatorname{argmax}_{c_j \in C} \log P(c_j) + \sum_{i \in \text{positions}} \log P(x_i | c_j)$$

- Model is now just max of sum of weights 把連乘改為log連加





# How to tweak performance

- Domain-specific features and weights: *very* important in real performance
- Sometimes need to collapse terms:
  - Part numbers, chemical formulas, ...
  - But stemming generally doesn't help
- Upweighting: Counting a word as if it occurred twice:
  - title words (Cohen & Singer 1996)
  - first sentence of each paragraph (Murata, 1999)
  - In sentences that contain title words (Ko *et al*, 2002)

# Text Classification and Naïve Bayes

# Text Classification: Practical Issues