Lecture 3 : Classification (1)

楊立偉教授

wyang@ntu.edu.tw

本投影片修改自Introduction to Information Retrieval一書之投影片
Ch 13~14

近年來機器學習的進展 (1)

- 監督式學習 Supervised learning
  - 演算法及模型、計算複雜度及規模、巨量資料

	Top 10 Data Mining Algorithm (IEEE ICDM 2006)
1	C4.5 and Beyond 決策樹及規則
2	k-Means 資料分群
3	Support Vector Machines (SVM) 自動分類
4	Apriori 關聯分析
5	Expectation Maximization (EM) 最大概似估計
6	PageRank 連結分析
7	AdaBoost 適應增強學習
8	k-Nearest Neighbor (kNN) 自動分類
9	Naive Bayes (NB) 自動分類
10	Classification and Regression Trees (CART) 決策樹

近年來機器學習的進展 (2)

Supervised learning 訓練到自動、依樣畫葫蘆、到相互對抗

Unsupervised learning 有目標，進行嘗試、給予獎勵或懲罰

Self-supervised learning 從資料本身學習

What Machine Learning Can Do

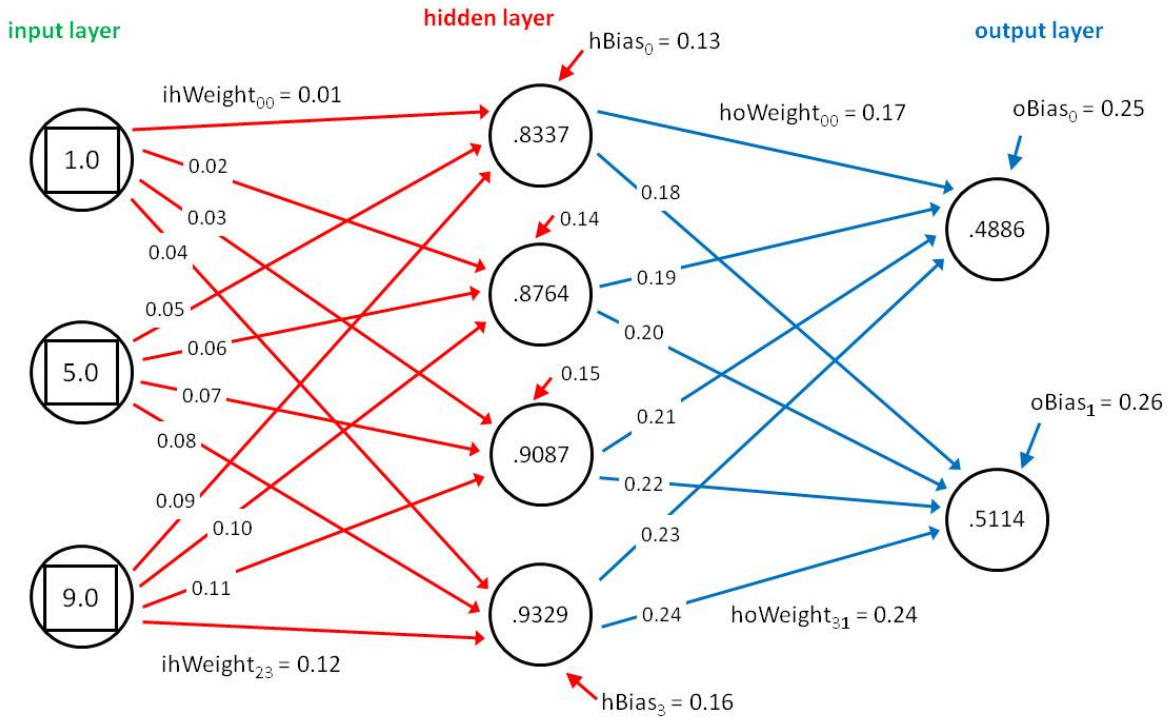
A simple way to think about supervised learning.

INPUT A		RESPONSE B	APPLICATION
Picture	臉部辨識	Are there human faces? (0 or 1)	Photo tagging
Loan application	貸款核准	Will they repay the loan? (0 or 1)	Loan approvals
Ad plus user information	精準廣告	Will user click on ad? (0 or 1)	Targeted online ads
Audio clip	語音辨識	Transcript of audio clip	Speech recognition
English sentence	機器翻譯	French sentence	Language translation
Sensors from hard disk, plane engine, etc.		Is it about to fail?	Preventive maintenance
Car camera and other sensors	自動駕駛	Position of other cars	Self-driving cars

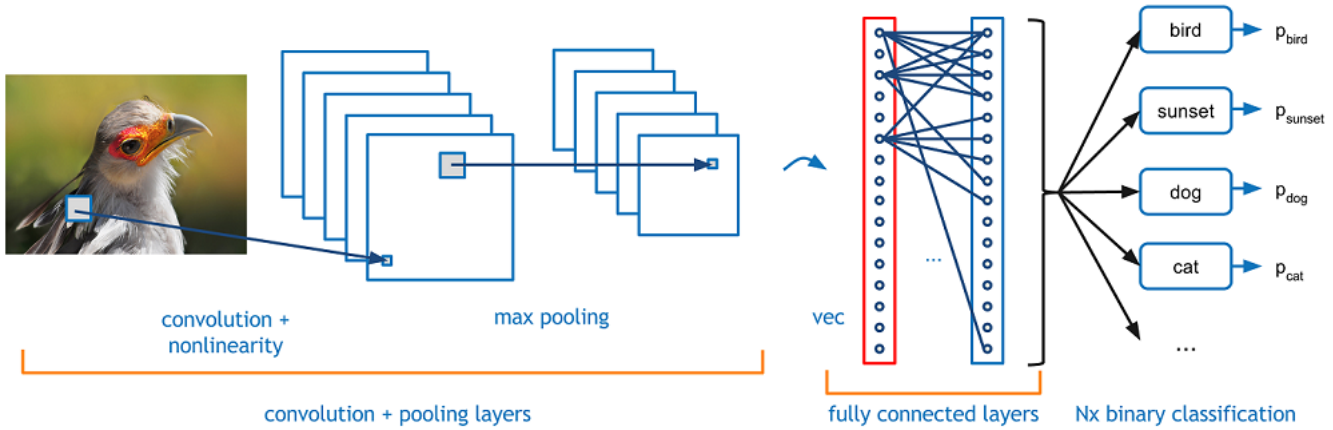
SOURCE ANDREW NG "What Artificial Intelligence Can and Can't Do Right Now" by Andrew Ng, Harvard Business Review, 2016 Nov.

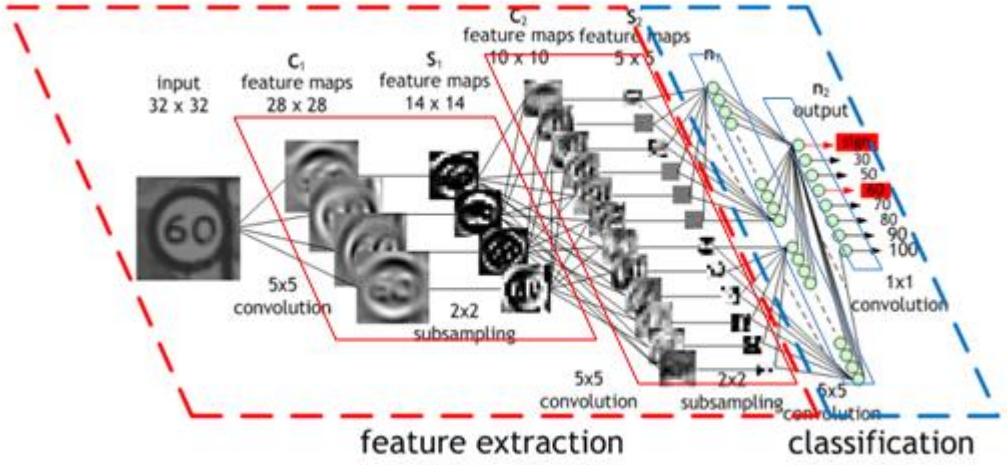
近年來機器學習的進展 (3)

- Neural Networks (NN) to Deep Learning (DL)
  - Backpropagation and Gradient Descent algorithm



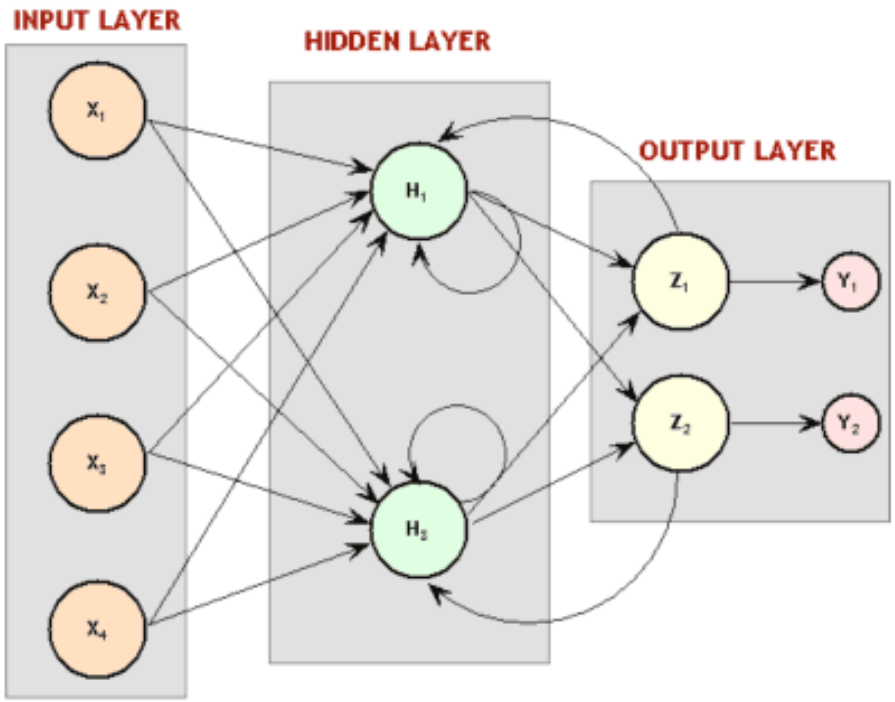
- Convolution Neural Networks (CNN)





Ref: Imagenet classification with deep convolutional neural networks 2010

- Recurrent Neural Networks (RNN)



Ref: Recurrent neural network based language model. 2010

- Long Short-Term Memory (LSTM) RNN

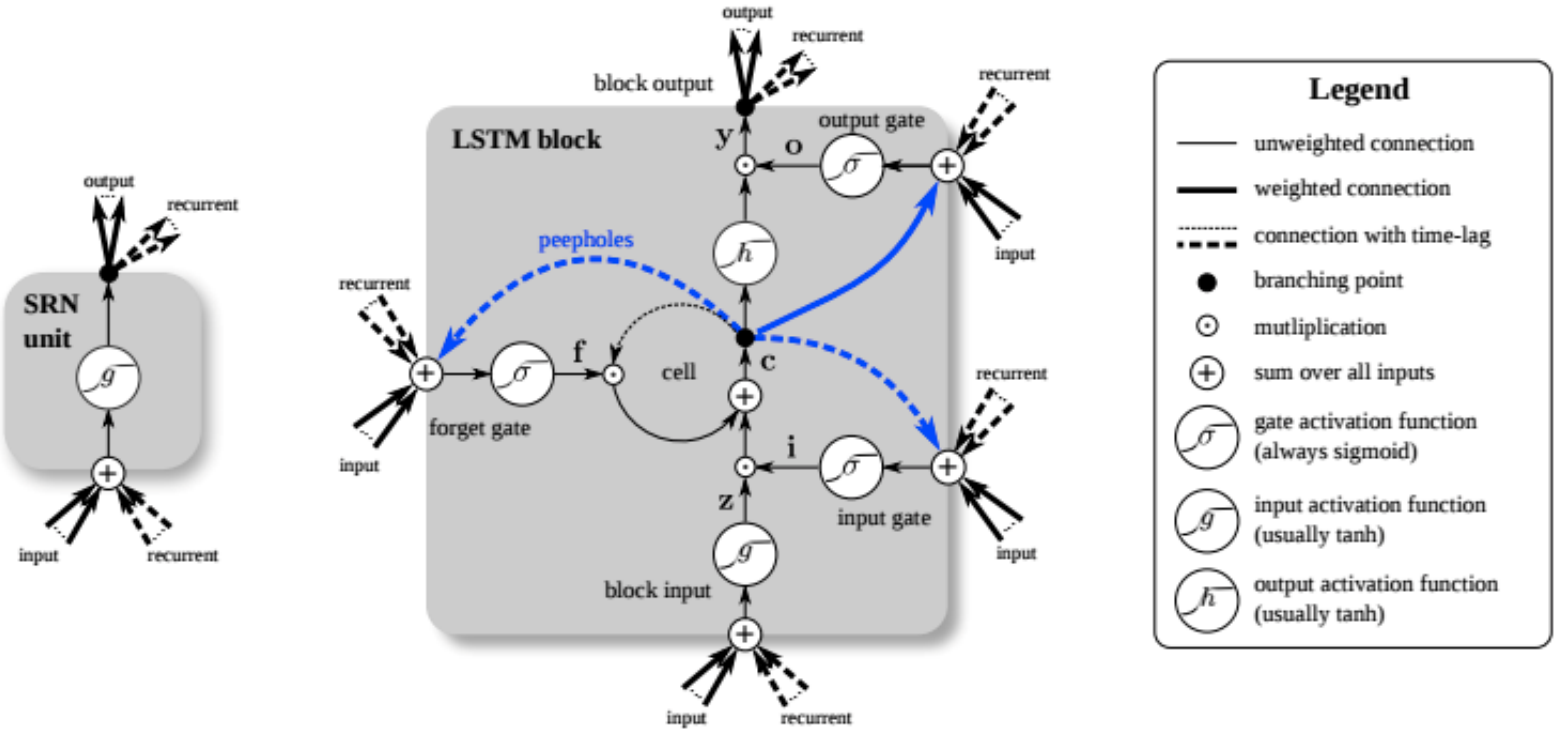


Figure 1. Detailed schematic of the Simple Recurrent Network (SRN) unit (left) and a Long Short-Term Memory block (right) as used in the hidden layers of a recurrent neural network.

Ref: http://193.6.4.39/~czap/letoltes/IS14/IS2014/PDF/AUTHOR/IS141304.PDF, 2014

Self-attention Transformer and BERT (1)

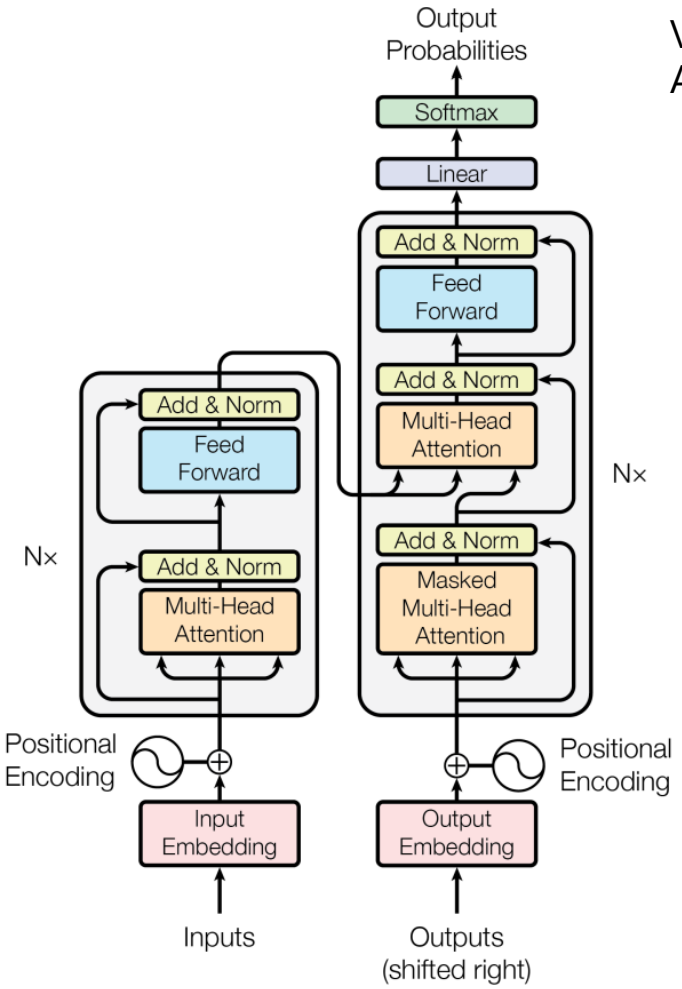
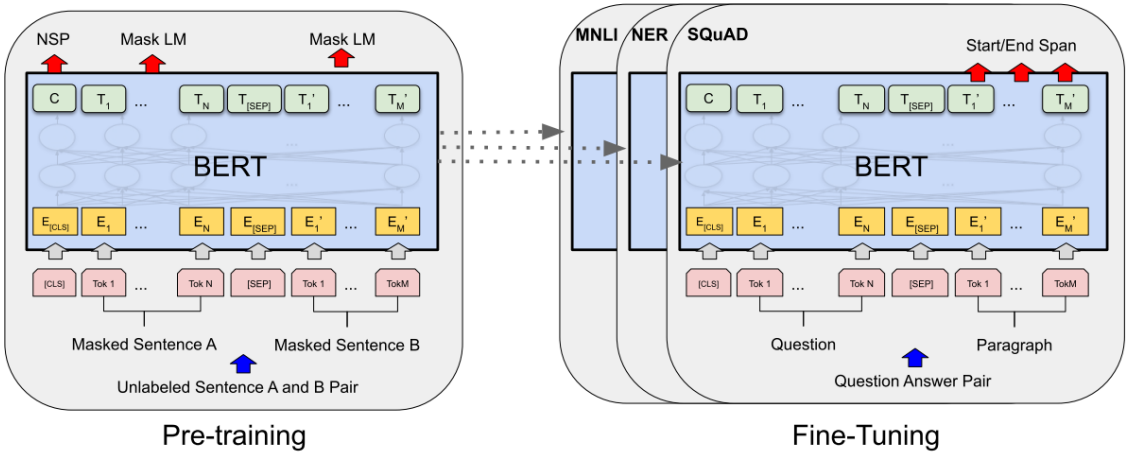


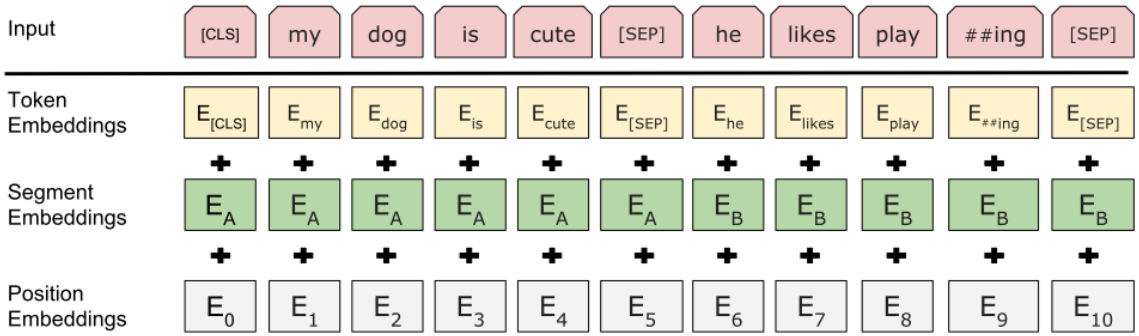
Figure 1: The Transformer - model architecture.

Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.

Self-attention Transformer and BERT (2)

Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).





Text Classification

A text classification task: Email spam filtering

```
From: ''' <takworlld@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 write a program that would automatically detect and delete this type of message?

Formal definition of TC: Training

Given:

- A document space X
  - Documents are represented in this space – typically some type of high-dimensional space.
- A fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$
  - The classes are human-defined for the needs of an application (e.g., relevant vs. nonrelevant).
- A training set D of labeled documents with each labeled document $\langle d, c \rangle \in X \times C$

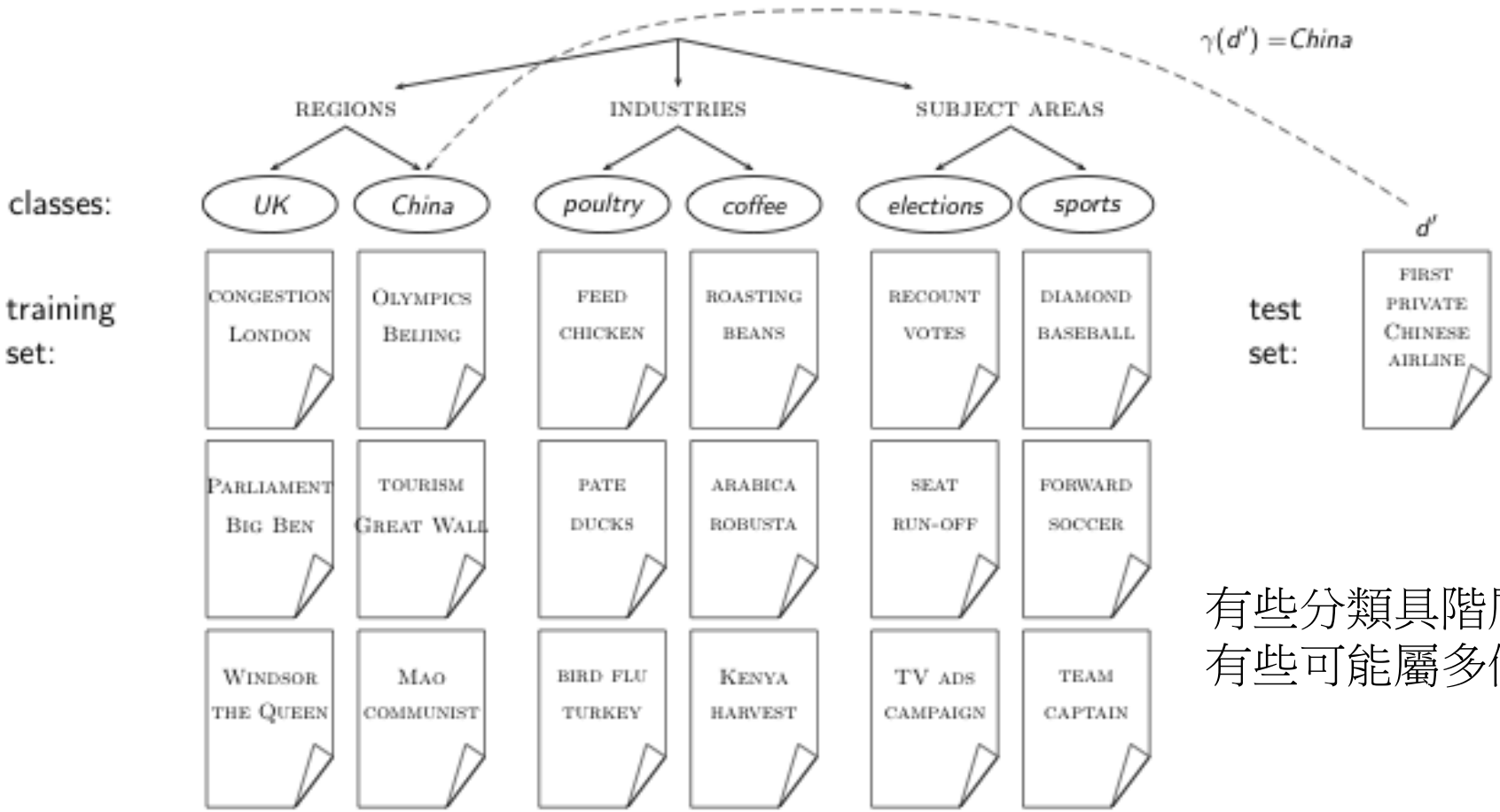
Using a learning method or learning algorithm, we then wish to learn a classifier Υ that maps documents to classes:

$$\Upsilon : X \rightarrow C$$

Formal definition of TC: Application/Testing

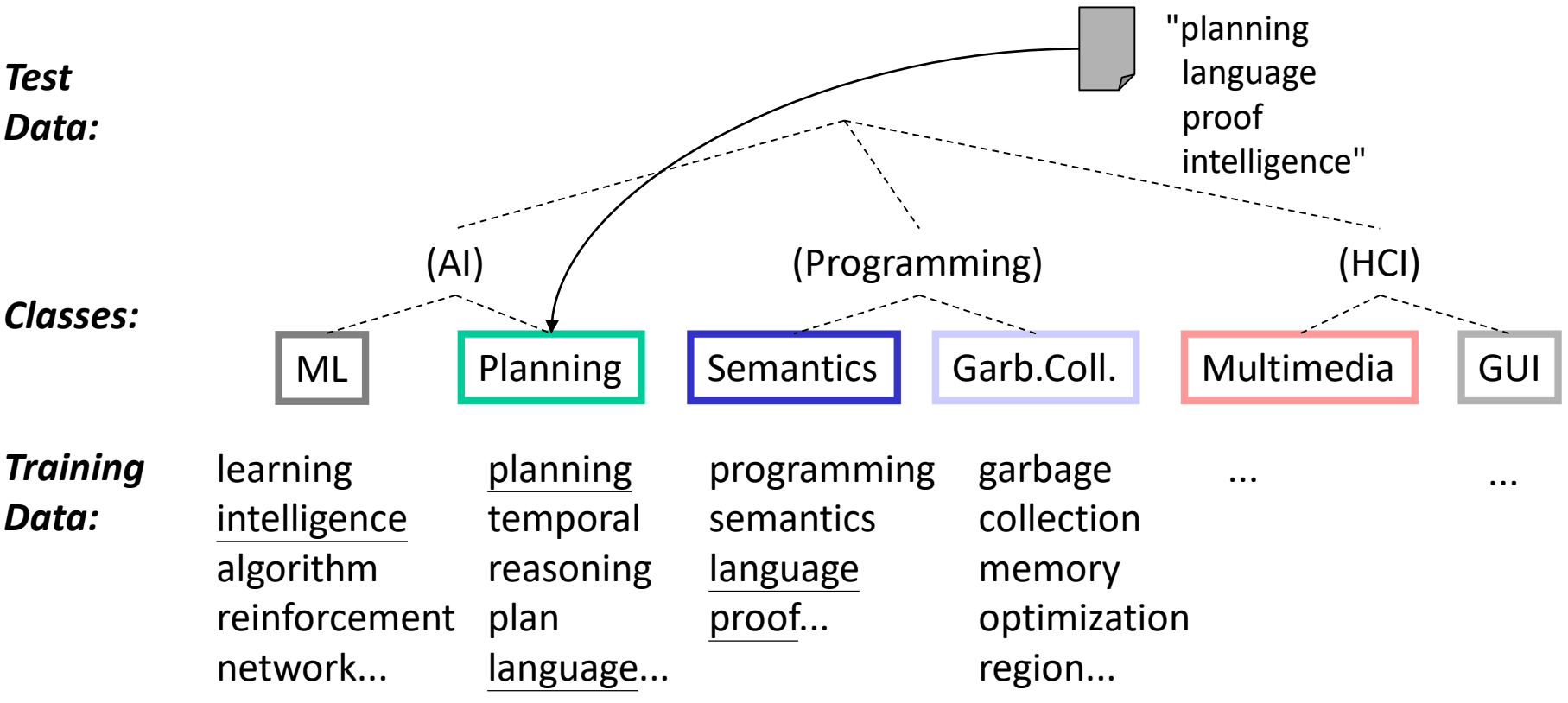
Given: a description $d \in X$ of a document Determine: $\Upsilon(d) \in C$, that is, the class that is most appropriate for d

Topic classification



有些分類具階層性
有些可能屬多個分類

Example


Test Data:
"planning language proof intelligence"
Classes:
(AI)
(Programming)
(HCI)
ML
Planning
Semantics
Garb.Coll.
Multimedia
GUI
Training Data:
learning intelligence algorithm reinforcement network...
planning temporal reasoning plan language...
programming semantics language proof...
garbage collection memory optimization region...
...
...

More examples of TC Applications

Assign labels to each document : Labeling 貼標籤 (歸類)

- Labels are most often topics such as Yahoo-categories 主題

  e.g., "finance," "sports," "news>world>asia>business"
- Labels may be language or genres 語言或型式

  e.g., “English" “Chinese" “French“

  e.g., "editorials" "movie-reviews" "news“
- Labels may be sentiments 情緒

  e.g., “like”, “hate”, “neutral”
- Labels may be domain-specific binary 是否屬於某領域

  e.g., "interesting-to-me" : "not-interesting-to-me”

  e.g., “spam” : “not-spam”

  e.g., “contains adult language” :“doesn’t”

More examples of TC Applications

- 新聞自動分類：將每日新聞自動歸類重新編排
- 專利自動分類：將申請的專利給予適當的類別以利分派審查
- 1999案件自動分類：將民眾抱怨快速分派給正確的單位做處理`
- 用文件分類預測股價走勢：出些某些文件，之後七日內該股票之之平均收盤價走高或走低（看漲文件及看跌文件）

Classification Methods (1)

- Manual classification 人工分類
  - Used by Yahoo, ODP, PubMed
  - Very accurate when job is done by experts

    靠領域與分類專家，所以很準
  - Consistent when the problem size and team is small

    當資料量大，用人工判斷會有主觀不一致的問題
  - Difficult and expensive to scale

    need automatic methods for classification

註 : ODP - Open Directory Project 開放分類目錄計劃

Classification Methods (2)

- Rule-based systems 規則式分類
  - Google Alerts is an example of rule-based classification
  - Assign category if document contains a given boolean combination of words
    - 使用布林條件，例如 (文化創意 | 文創) → 歸於文化類
  - Accuracy is often very high if a rule has been carefully refined over time by a subject expert
  - Building and maintaining these rules is cumbersome and expensive
    - 例如 (文化創意 | 文創 | 電影 | 工藝 | 藝術....) → 歸於文化類
      需要很多的列舉與排除

Classification Methods (3)

- Statistical/Probabilistic systems 統計機率式
- Text classification as a learning problem
  - (i) Supervised learning of a the classification function ϒ and
  - (ii) its application to classifying new documents
- Examples
  - Naive Bayes (simple, common method)
  - k-Nearest Neighbors (simple, powerful)
  - Support-vector machines (new, more powerful)
- No free lunch: requires hand-classified training data
  - But data can be built up (and refined) by non-experts

Naïve Bayes

Overview

- The Naive Bayes classifier is a probabilistic classifier.
  基於機率學的貝氏定理
- Build a *generative model* that approximates how data is produced
- Uses *prior* probability (事前機率; 先天機率) of each category given no information about an item.
- Categorization produces a *posterior* probability (事後機率; 條件機率) distribution over the possible categories given a description of an item.

訣竅：將條件機率展開成可以計算的機率，然後計算之

Posterior probability 條件機率

- 條件機率

	年齡		
性別	20歲	非20歲	和
男	14	6	20
女	21	9	30
和	35	15	50

A：20歲 B：女性

已知學生為20歲中女性之機率

P(B|A)=21/35=0.6

或利用公式 $P(B|A) = P(A \cap B) / P(A) = 0.42/0.7 = 0.6$

Bayes' Rule

$$P(C, X) = P(C \mid X)P(X) = P(X \mid C)P(C)$$

$$\boxed{P(C \mid X) = \frac{P(X \mid C)P(C)}{P(X)}}$$

Naive Bayes Classifiers (1)

- Task: Classify a new instance *D* based on a tuple of attribute values into one of the classes $c_j \in C$
- Naive Bayes classification is to find the "best" class (the most likely or maximum a posteriori (MAP) class C_{map})

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

$$c_{MAP} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j \mid x_1, x_2, \ldots, x_n)$$

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

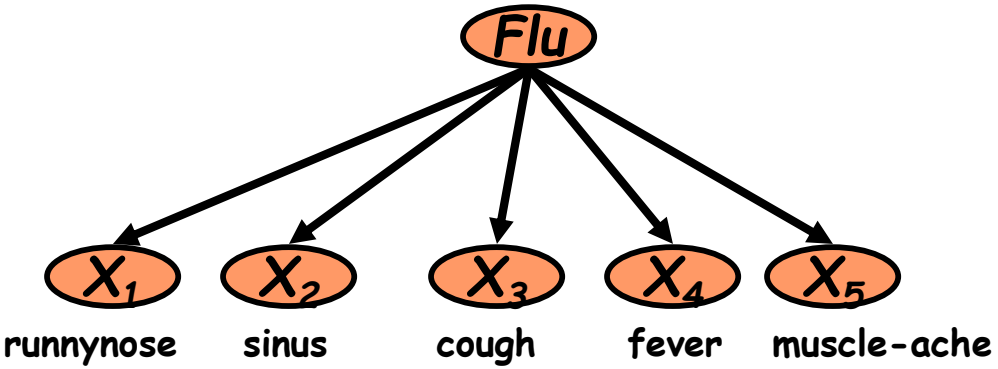
as *P(D)* is constant

$$= \underset{c_j \in C}{\operatorname{argmax}} P(x_1, x_2, \ldots, x_n \mid c_j) P(c_j)$$

Naive Bayes Classifiers (2)

- $P(c_j)$
  - Can be estimated from the frequency of classes in the training examples. 可以由訓練資料中計算而得
- $P(x_1,x_2,\ldots,x_n|c_j)$
  - Could be estimated if a very large number of training examples was available.
  - applying Naïve Bayes Conditional Independence Assumption

Naïve Bayes Assumption

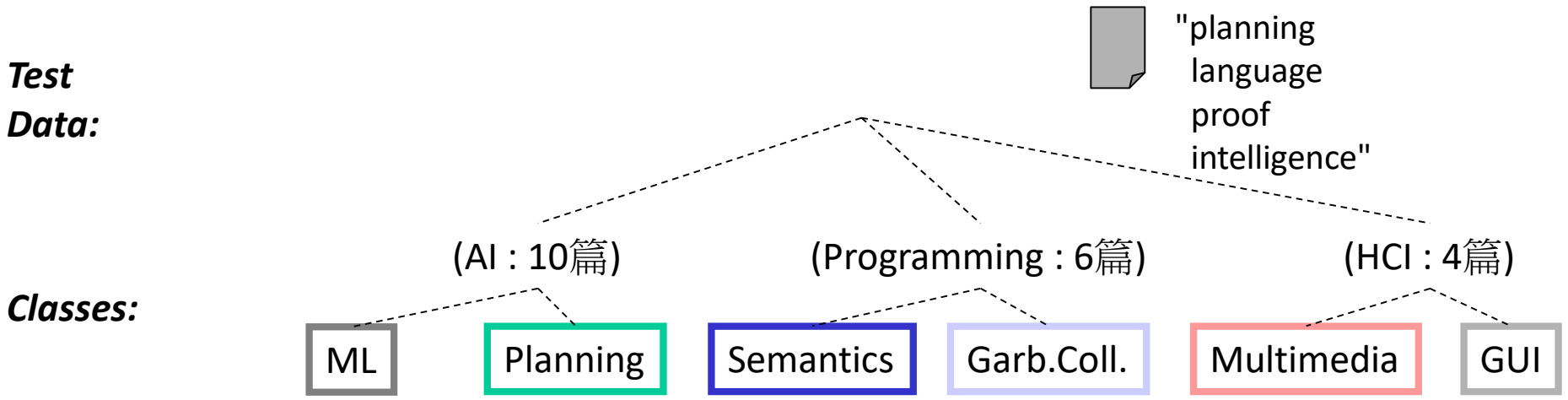


- Conditional Independence Assumption:

  Assume that the probability of observing the conjunction of attributes is equal to the product of the individual probabilities P(xi|cj).

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

Exercise



共8+4+3+2+7+6+2+3+7+6=48　共4+5+8+3+2+4+8+5+3=42

Exercise (cont.)

- $P(c_j)$
  - P(AI)=10/20, P(Programming)=6/20
- $P(x_1,x_2,\ldots,x_n|c_j)$
  - P(intelligence|AI)=4/48, P(planning|AI)=6/48, P(language|AI)=6/48,
  - P(language|Programming)=8/42, P(proof|Programming)=3/42

若不做smoothing 並以相加(不取log)取代相乘的近似算法

AI: 10/20*(4/48+6/48+6/48)=0.167 ←歸類為AI類

Programming: 6/20*(8/42+3/42)=0.078

Naïve Bayes: Learning

- From training corpus, extract *Vocabulary* 先取出所有可能的詞
- Calculate required $P(c_j)$ and $P(x_k \mid c_j)$ 能算的先算
  - For each c_j in C do
    - $docs_j \leftarrow$ subset of documents for which the target class is c_j

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

  - Concatenate all $docs_j$ into a single document $Text_j$
    - for each word x_k in *Vocabulary*

      $n_k \leftarrow$ number of occurrences of x_k in $Text_j$

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

調整項：Smoothing to avoid over-fitting (avoid zero)

Naïve Bayes: Classifying

- positions ← all word positions in current document which contain tokens found in *Vocabulary*
- Return c_{NB}, where

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}} P(c_j) \prod_{i \in positions} P(x_i \mid c_j)$$

取機率最大的類別　　有出現的詞, 其機率相乘

Smoothing to avoid over-fitting

- If a document contains a term x which never appears in the category c, the p(x|c) will always be zero, and the product will be zero, too.
- to add one smoothing to avoid zeros :

Before $$P(x_k \mid c_j) \leftarrow \frac{n_k}{n}$$

After $$P(x_k \mid c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha \mid Vocabulary \mid}$$

Naive Bayes: Training

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

1 $V \leftarrow$ ExtractVocabulary($\mathbb{D}$)
2 $N \leftarrow$ CountDocs($\mathbb{D}$)
3 **for each** $c \in \mathbb{C}$
4 **do** $N_c \leftarrow$ CountDocsInClass($\mathbb{D}, c$)
5 $prior[c] \leftarrow N_c/N$
6 $text_c \leftarrow$ ConcatenateTextOfAllDocsInClass($\mathbb{D}, c$)
7 **for each** $t \in V$
8 **do** $T_{ct} \leftarrow$ CountTokensOfTerm($text_c, t$)
9 **for each** $t \in V$
10 **do** $condprob[t][c] \leftarrow \frac{T_{ct}+1}{\sum_{t'}(T_{ct'}+1)}$
11 **return** $V, prior, condprob$

Naive Bayes: Testing

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

由相乘積轉成log相加

Naïve Bayes : discussion

Violation of NB Assumptions

- Conditional independence
  - 是否可以假設兩詞的出現為獨立事件？
  - 與 VSM 的問題類似：向量空間之兩兩詞間是否為正交？
- Conclusion
  - Naive Bayes can work well even though conditional independence assumptions are badly violated
  - Because classification is about predicting the correct class and not about accurately estimating probabilities.

NB with Feature Selection (1)

- Text collections have a large number of features
  - 10,000 – 1,000,000 unique words … and more

Feature (文件或類別的特徵) 若是選得好

- Reduces training time
  - Training time for some methods is quadratic or worse in the number of features
- Can improve generalization (performance)
  - Eliminates noise features
  - Avoids over-fitting

NB with Feature Selection (2)

- 2 ideas beyond TF-IDF 兩種評估好壞的指標
  - Hypothesis testing statistics:
    - Are we confident that the value of one categorical variable is associated with the value of another
    - Chi-square test 卡方檢定
  - Information theory:
    - How much information does the value of one categorical variable give you about the value of another
    - Mutual information

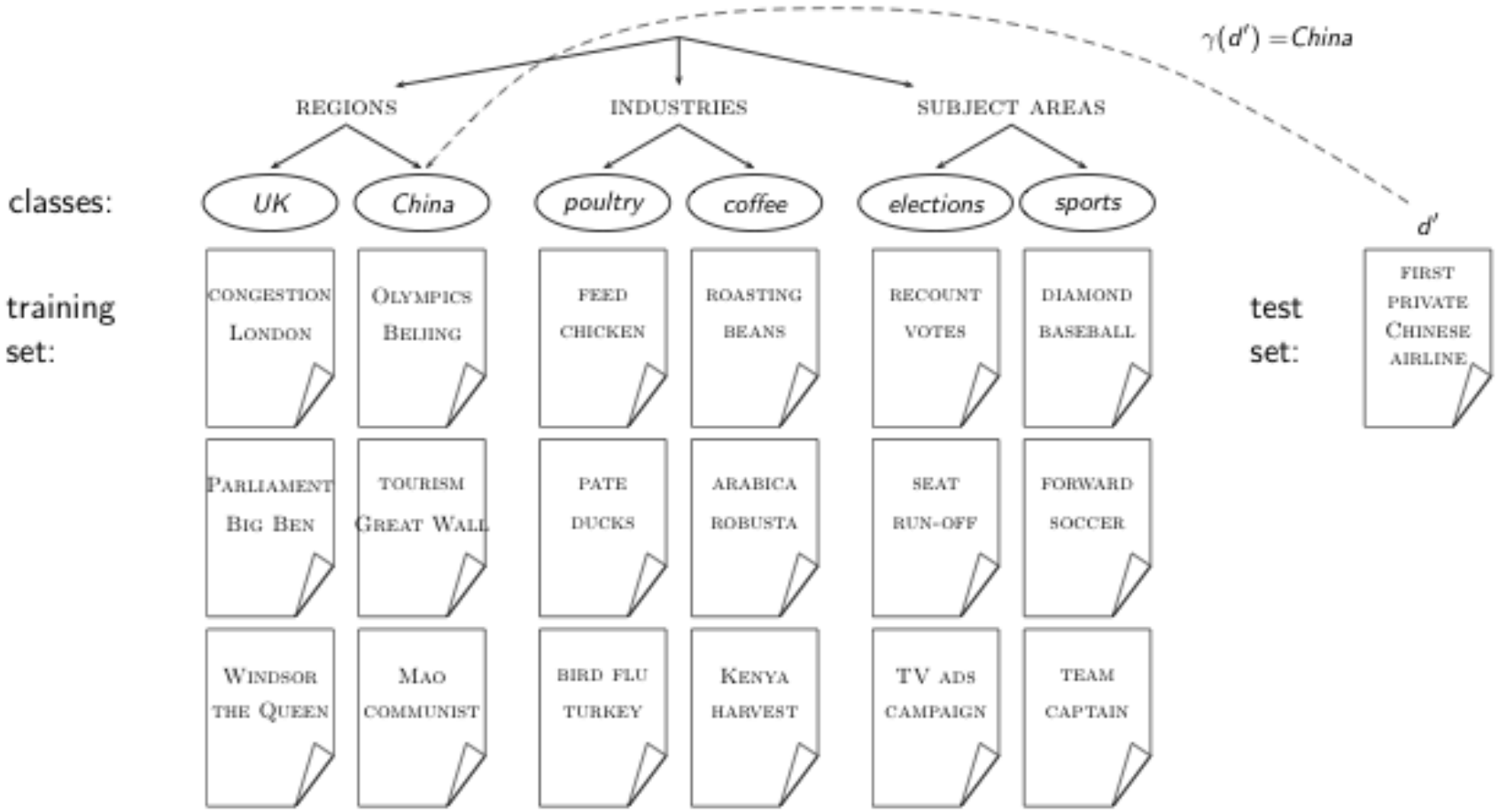
- They're similar, but χ^2 measures confidence in association, (based on available statistics), while MI measures extent of association (assuming perfect knowledge of probabilities)

Naive Bayes is not so naive

- 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 : Learning with one pass over the data; testing linear in the number of attributes, and document collection size
- Low storage requirements

Naïve Bayes : evaluation

Evaluation on Reuters



Example: The Reuters collection

symbol	statistic	value
N	documents	800,000
L	avg. # word tokens per document	200
M	word types	400,000
	avg. # bytes per word token (incl. spaces/punct.)	6
	avg. # bytes per word token (without spaces/punct.)	4.5
	avg. # bytes per word type	7.5
	non-positional postings	100,000,000

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

A Reuters document

REUTERS

You are here: Home > News > Science > Article

Go to a Section: U.S. International Business Markets Politics Entertainment Technology Sports Oddly Enou

Extreme conditions create rare Antarctic clouds

Tue Aug 1, 2006 3:20am ET

Email This Article | Print This Article | Reprints

[-] Text [+]

SYDNEY (Reuters) - Rare, mother-of-pearl colored clouds caused by extreme weather conditions above Antarctica are a possible indication of global warming, Australian scientists said on Tuesday.

Known as nacreous clouds, the spectacular formations showing delicate wisps of colors were photographed in the sky over an Australian

Evaluating classification

- Evaluation must be done on test data that are independent of the training data (usually a disjoint set of instances).
- 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

Precision *P* and recall *R*

	in the class	not in the class
predicted to be in the class	true positives (TP)	false positives (FP)
predicted to not be in the class	false negatives (FN)	true negatives (TN)

↑ also known as confusion matrix

$$P = TP / (TP + FP)$$

$$R = TP / (TP + FN)$$

參考 Terminology

真陽性 (TP, true positive)
- 正確的肯定
- 又稱：命中 (hit)

真陰性 (TN, true negative)
- 正確的否定
- 又稱：正確拒絕 (correct rejection)

偽陽性 (FP, false positive)
- 錯誤的肯定，
- 又稱：假警報 (false alarm)，第一型錯誤

偽陰性 (FN, false negative)
- 錯誤的否定
- 又稱：未命中 (miss)，第二型錯誤

真陽性率 (TPR, true positive rate)
- 又稱：命中率 (hit rate)、敏感度(sensitivity)
- TPR = TP / P = TP / (TP+FN)
- 即為 Recall

偽陽性率(FPR, false positive rate)
- 又稱：錯誤命中率，假警報率 (false alarm rate)
- FPR = FP / N = FP / (FP + TN)

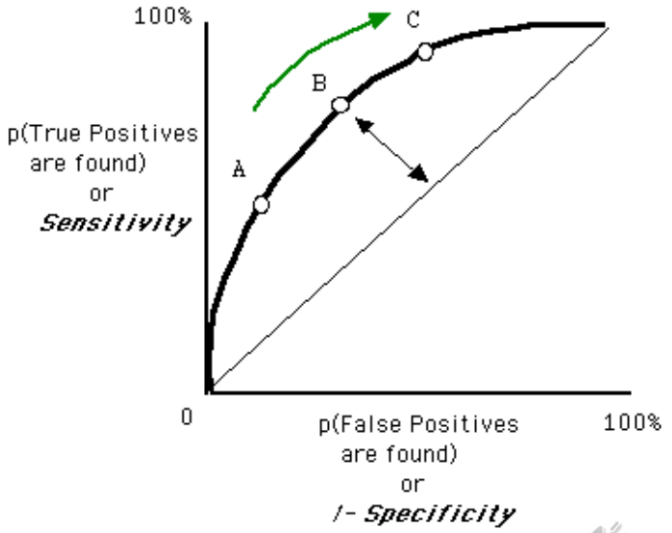
準確度 (ACC, accuracy)
- ACC = (TP + TN) / (P + N)
- 即：(真陽性+真陰性) / 總樣本數

真陰性率 (TNR)
- 又稱：特異度 (SPC, specificity)
- SPC = TN / N = TN / (FP + TN) = 1 - FPR

Ref : Wikipedia, http://zh.wikipedia.org/wiki/ROC曲線

參考 ROC Curve and AUC

- 對一個二元分類器，給予不同閥值參數，依 X=FPR, Y=TPR 所繪製出的曲線，稱為 receiver operating characteristic curve
  - 閥值最嚴時，沒有樣本被預測為陽性，FPR及TPR均為0，即 (0,0)
  - 閥值最鬆時，全部樣本被預測為陽性，FPR及TPR均為1，即 (1,1)
  - 對角線為隨機分類，對角線以上表示勝過隨機分類；對角線以下表示略於隨機分類
- ROC Curve下的面積稱AUC
  - Area under the Curve of ROC
  - 因為是1x1方格，AUC面積必在0~1之間
  - AUC值越大的分類器，正確率越高



Ref : Wikipedia, http://zh.wikipedia.org/wiki/ROC曲線

A combined measure: *F*

- 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*: $\frac{1}{F} = \frac{1}{2}\left(\frac{1}{P} + \frac{1}{R}\right)$

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 F_1 for aggregate TP, FP, FN

Exercise

	Class A	Class B
# of records	800	200
accuracy %	80%	60%
macro average accuracy %	(80%+60%)/2=70%	
micro average accuracy %	(800*80%+200*60%)/(800+200)=76%	

若類別A及B的資料筆數相同，則macro avg.等於micro avg.

micro avg.可視為依類別大小加權後的平均數

Naive Bayes vs. other methods

(a)

	NB	Rocchio	kNN	SVM
micro-avg-L (90 classes)	80	85	86	89
macro-avg (90 classes)	47	59	60	60

(b)

	NB	Rocchio	kNN	trees	SVM
earn	96	93	97	98	98
acq	88	65	92	90	94
money-fx	57	47	78	66	75
grain	79	68	82	85	95
crude	80	70	86	85	89
trade	64	65	77	73	76
interest	65	63	74	67	78
ship	85	49	79	74	86
wheat	70	69	77	93	92
corn	65	48	78	92	90
micro-avg (top 10)	82	65	82	88	92
micro-avg-D (118 classes)	75	62	n/a	n/a	87

Evaluation measure: F_1 Naive Bayes does pretty well, but some methods beat it consistently (e.g., SVM).

Alternative measurement

- Classification accuracy : c/n

  n : the total number of test instances

  c : the number of test instances correctly classified by the system.

- 當各類別大小差異大時，Classification accuracy易受影響
  - Ex. Class A有800筆、Class B有200筆，則全部猜A的accuracy至少有80%
  - 此時仍應檢視各類別的Precision及Recall

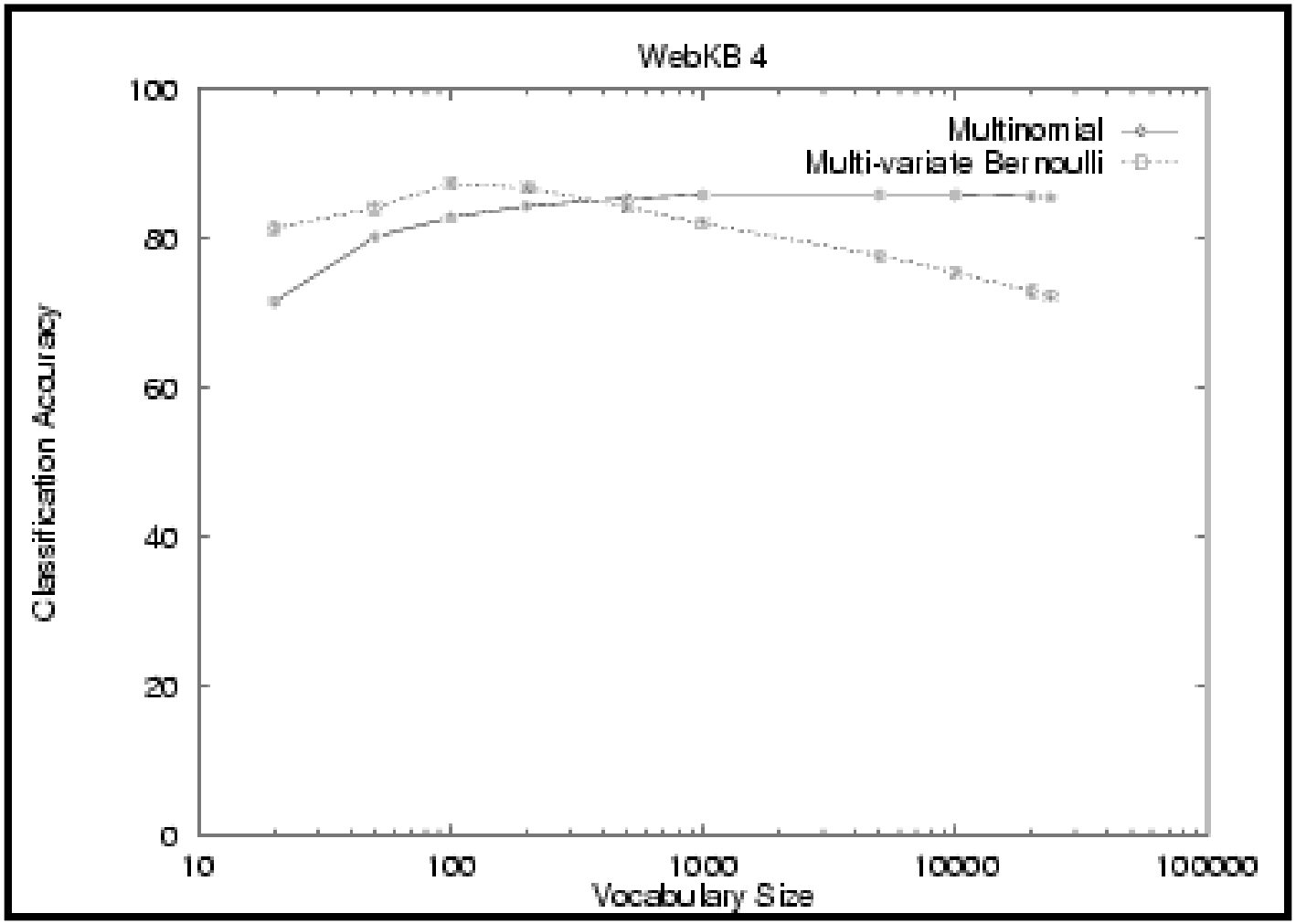
Case study : WebKB Experiment

- Classify webpages from CS departments into:
  - student, faculty, course,project
- Train on ~5,000 hand-labeled web pages
  - Cornell, Washington, U.Texas, Wisconsin
- Crawl and classify a new site (CMU)
  - for testing

- Results:

	Student	Faculty	Person	Project	Course	Departmt
Extracted	180	66	246	99	28	1
Correct	130	28	194	72	25	1
Accuracy:	72%	42%	79%	73%	89%	100%

NB Model Comparison



Faculty

associate	0.00417
chair	0.00303
member	0.00288
ph	0.00287
director	0.00282
fax	0.00279
journal	0.00271
recent	0.00260
received	0.00258
award	0.00250

Students

resume	0.00516
advisor	0.00456
student	0.00387
working	0.00361
stuff	0.00359
links	0.00355
homepage	0.00345
interests	0.00332
personal	0.00332
favorite	0.00310

Courses

homework	0.00413
syllabus	0.00399
assignments	0.00388
exam	0.00385
grading	0.00381
midterm	0.00374
pm	0.00371
instructor	0.00370
due	0.00364
final	0.00355

Departments

departmental	0.01246
colloquia	0.01076
epartment	0.01045
seminars	0.00997
schedules	0.00879
webmaster	0.00879
events	0.00826
facilities	0.00807
eople	0.00772
postgraduate	0.00764

Research Projects

investigators	0.00256
group	0.00250
members	0.00242
researchers	0.00241
laboratory	0.00238
develop	0.00201
related	0.00200
arpa	0.00187
affiliated	0.00184
project	0.00183

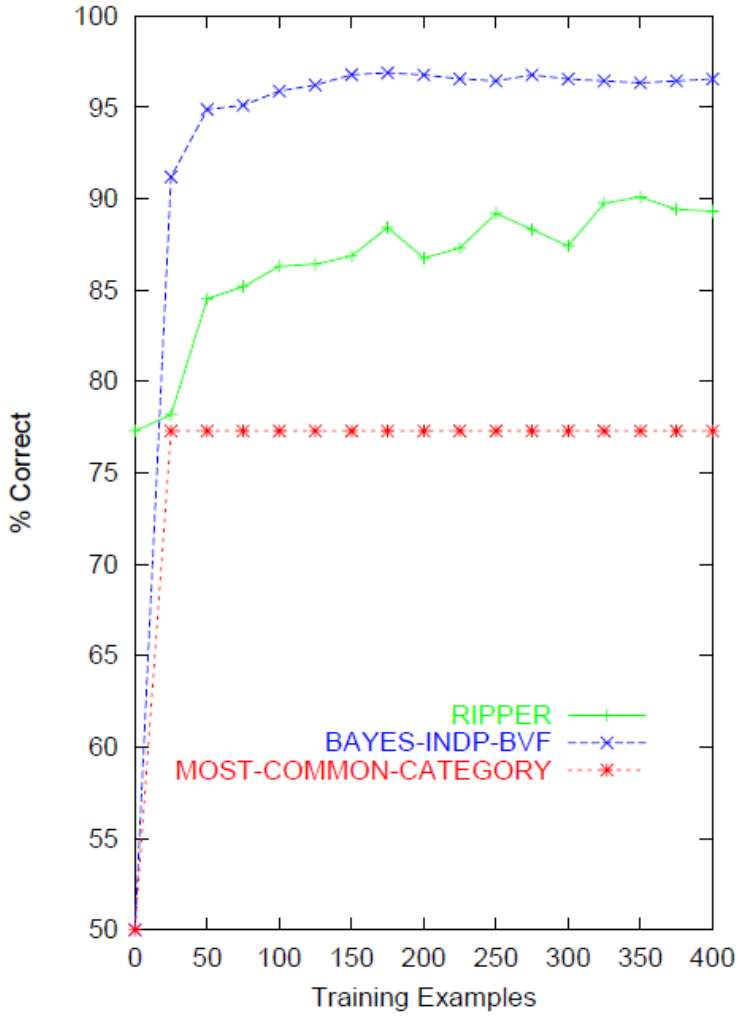
Others

type	0.00164
jan	0.00148
enter	0.00145
random	0.00142
program	0.00136
net	0.00128
time	0.00128
format	0.00124
access	0.00117
begin	0.00116

Case study : Apache SpamAssassin

- Naïve Bayes classifier for spam filtering
  - Widely used in spam filters
    - Classic Naive Bayes superior when appropriately used
    - 有很多衍生版本
- Many email filters use NB classifiers
  - But also many other things: black hole lists, etc.
    同時混用很多其它技巧，如黑名單

Naïve Bayes on spam email



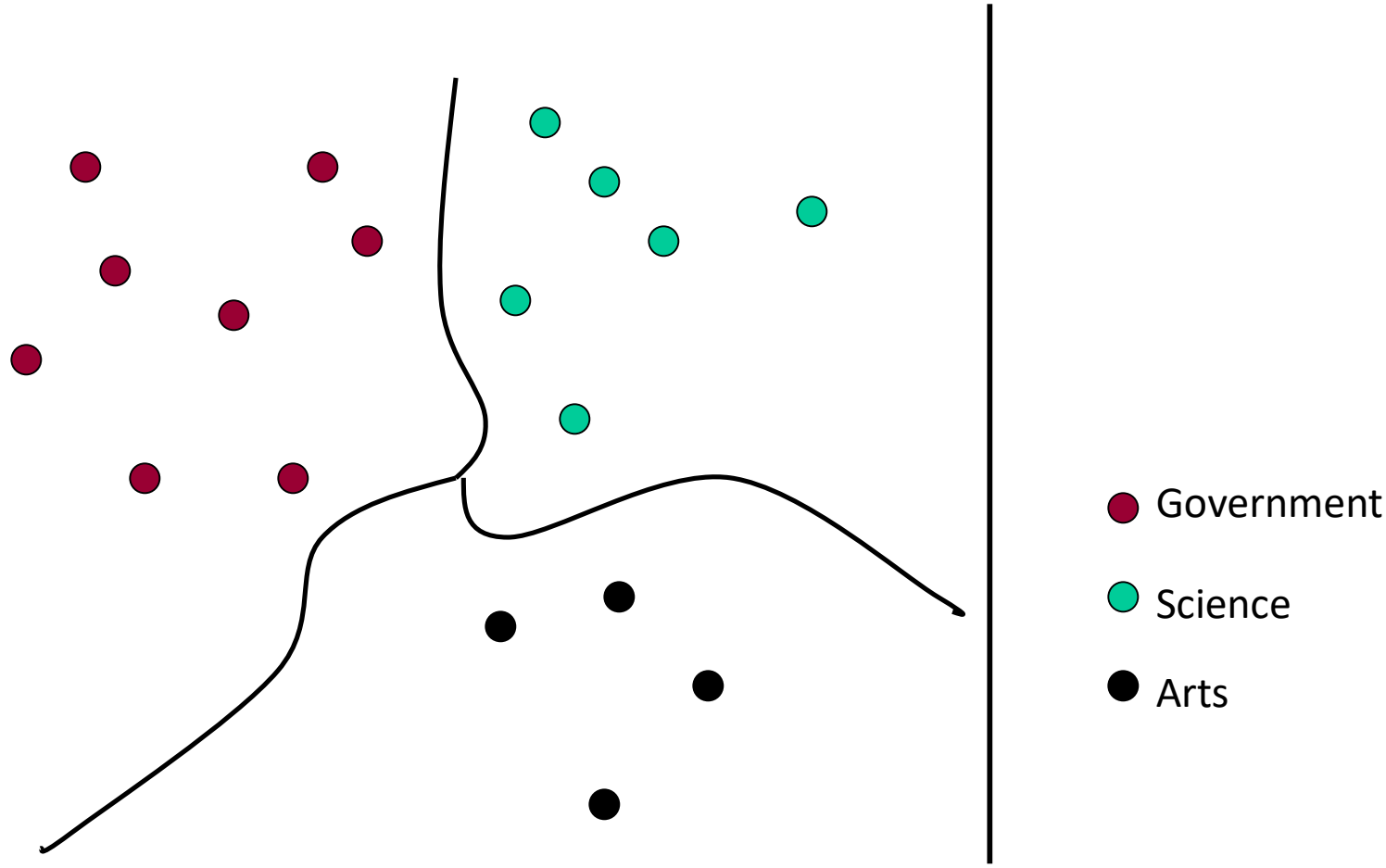
kNN : K Nearest Neighbors

Vector space classification

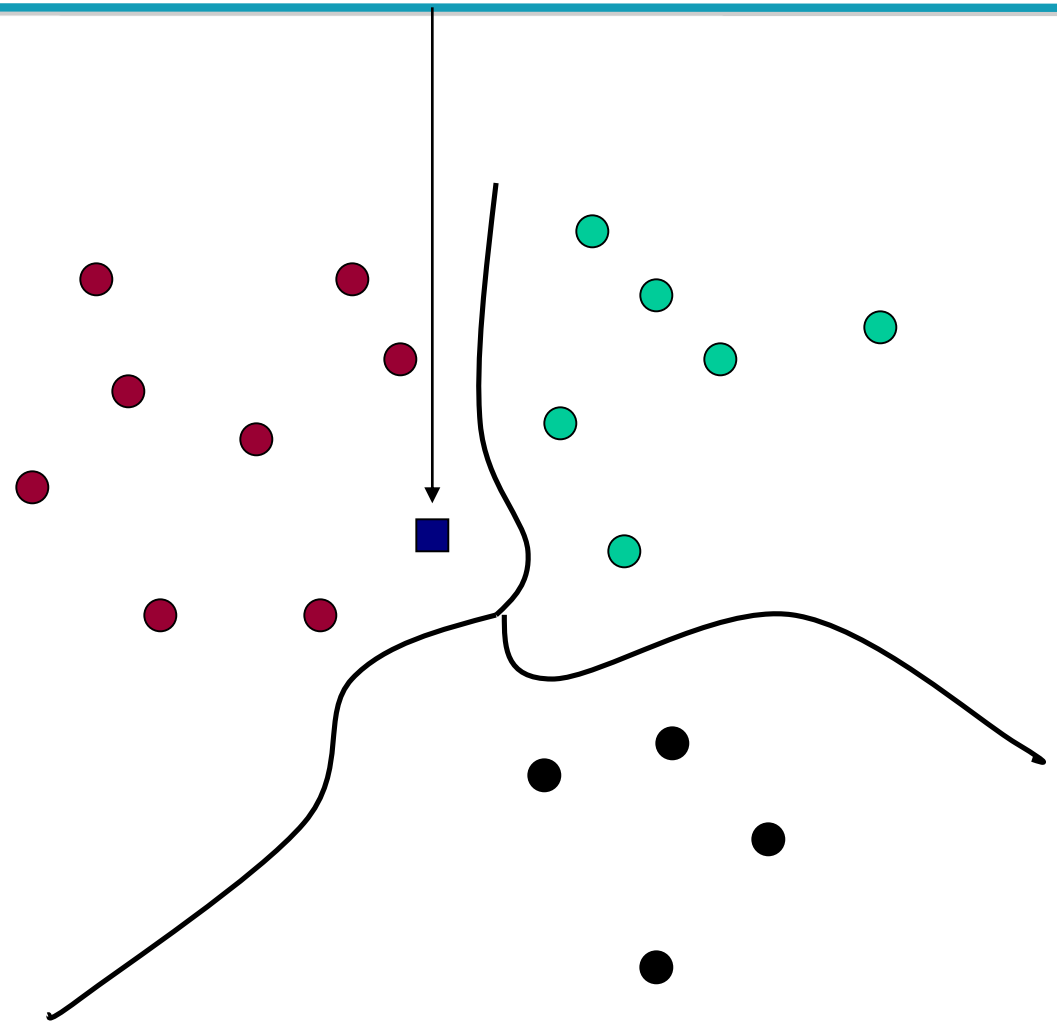
- As before, the training set is a set of documents, each labeled with its class.
- In vector space classification, this set corresponds to a labeled set of points or vectors in the vector space.
- Premise 1: Documents in the same class form a contiguous region. 同類別文件在空間中較相近
- Premise 2: Documents from different classes don't overlap.
- We define lines, surfaces, hypersurfaces to divide regions.

不同的類別間可找出一個分割線或(超)平面

Classes in a Vector Space



Test Document = Government



Similarity hypothesis true in general?

- Government
- Science
- Arts

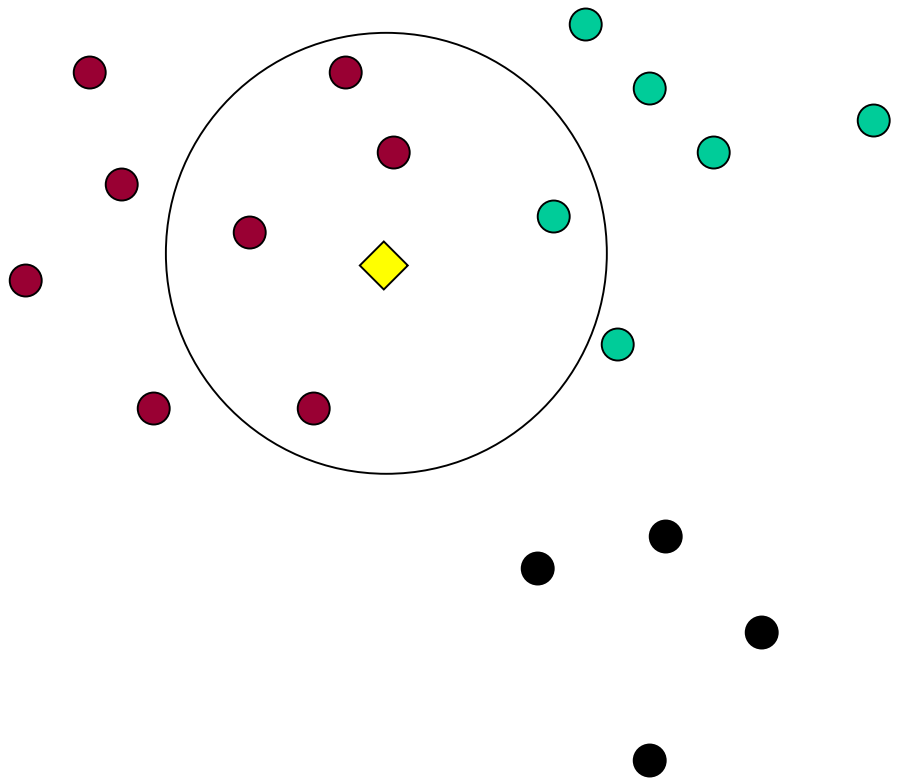
k Nearest Neighbor Classification

- To classify document *d* into class c
- Define *k*-neighborhood N as *k* nearest neighbors of *d*
- Count number of documents i in N that belong to c
- Estimate P(c|*d*) as i/k
- Choose as class argmax_c P(c|*d*) [= majority class]

訣竅：挑最近的 k 個鄰居出來統計 (投票)，

看最多人屬哪一類，自己標成那一類

Example: k=5 (5NN)



P(science| ◆)?

P(Government| ◆)?

- Government
- Science
- Arts

Nearest-Neighbor Learning Algorithm

- Learning is just storing the representations of the training examples in *D*.
- Testing instance *x*:
  - Compute similarity between *x* and all examples in *D*.

    計算文件 x 與其它訓練文件的相似度
  - Assign *x* the category of the most similar example in *D*.

    決定文件 x 的類別
- Does not explicitly compute a generalization or category
- Also called: 此方法又稱為
  - Case-based learning
  - Memory-based learning
  - Lazy learning

kNN algorithm

TRAIN-KNN($\mathbb{C}, \mathbb{D}$)
1 $\mathbb{D}' \leftarrow$ PREPROCESS($\mathbb{D}$)
2 $k \leftarrow$ SELECT-K($\mathbb{C}, \mathbb{D}'$)
3 **return** $\mathbb{D}', k$

APPLY-KNN($\mathbb{D}', k, d$)
1 $S_k \leftarrow$ COMPUTENEARESTNEIGHBORS($\mathbb{D}', k, d$)
2 **for each** $c_j \in \mathbb{C}(\mathbb{D}')$
3 **do** $p_j \leftarrow |S_k \cap c_j|/k$
4 **return** $\arg\max_j p_j$

Time complexity of kNN

- kNN test time proportional to the size of the training set
- The larger the training set, the longer it takes to classify a test document.
- kNN is inefficient for very large training sets.

kNN : discussion

kNN classification

- kNN classification is vector space classification method.
- It also is very simple and easy to implement.
- kNN is more accurate (in most cases) than Naive Bayes and others.
- If you need to get a pretty accurate classifier up and running in a short time . . .
- . . . and you don't care about efficiency that much . . .
- . . . use kNN.

Discussion of kNN (1)

- No training necessary
  - But linear preprocessing of documents is as expensive as training Naive Bayes.
  - We always preprocess the training set, so in reality training time of kNN is linear.
- kNN is very accurate if training set is large.
  - kNN Is Close to Optimal (ref. Cover and Hart, Nearest neighbor pattern classification, 1967)
  - But kNN can be very inaccurate if training set is small.
- kNN scores is hard to convert to probabilities

Discussion of kNN (2)

- Using only the closest example to determine the categorization is subject to errors due to:
  k 若只取一個易受下列影響
  - A single atypical example. 特例
  - Noise (i.e. error) in the category label of a single training example. 同類別中的雜訊
- More robust alternative is to find the k most-similar examples and return the majority category of these k examples. 選 k 個再用投票多數是較穩當的做法
- Value of k is typically odd to avoid ties; 3 and 5 are most common. 通常 k 要取單數, 常見的是3或5個

Nearest Neighbor with Inverted Index

- Finding nearest neighbors requires a linear search through |*D*| documents in collection
- Determining *k* nearest neighbors is the same as determining the *k* best retrievals using the test document as a query to a database of training documents.
  - 查詢結果前 k 名投票即可得
  - 配合使用Inverted Index可提升實作效率

Support Vector Machine (SVM)

Support Vector Machine (1)

- A kind of large-margin classifier
  - From Intensive machine-learning research in the last two decades to improve classifier effectiveness
- Vector space based machine-learning method aiming to find a decision boundary between two classes that is maximally far from any point in the training data
  - possibly discounting some points as outliers or noise

Support Vector Machine (2)

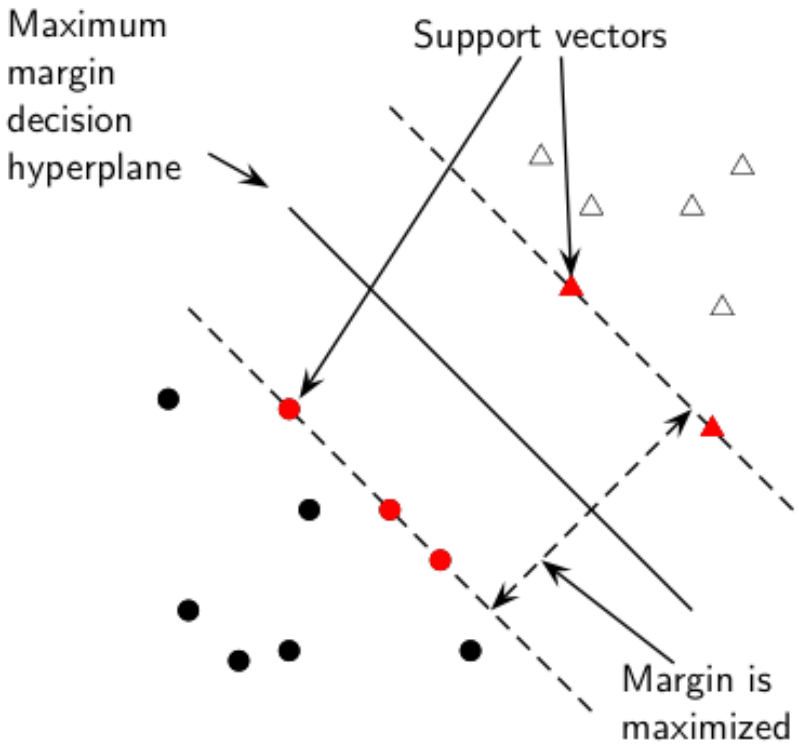
- 2-class training data
- decision boundary

  → **linear separator**
- criterion: being maximally far

  away from any data point

  → determines classifier **margin**
- linear separator position

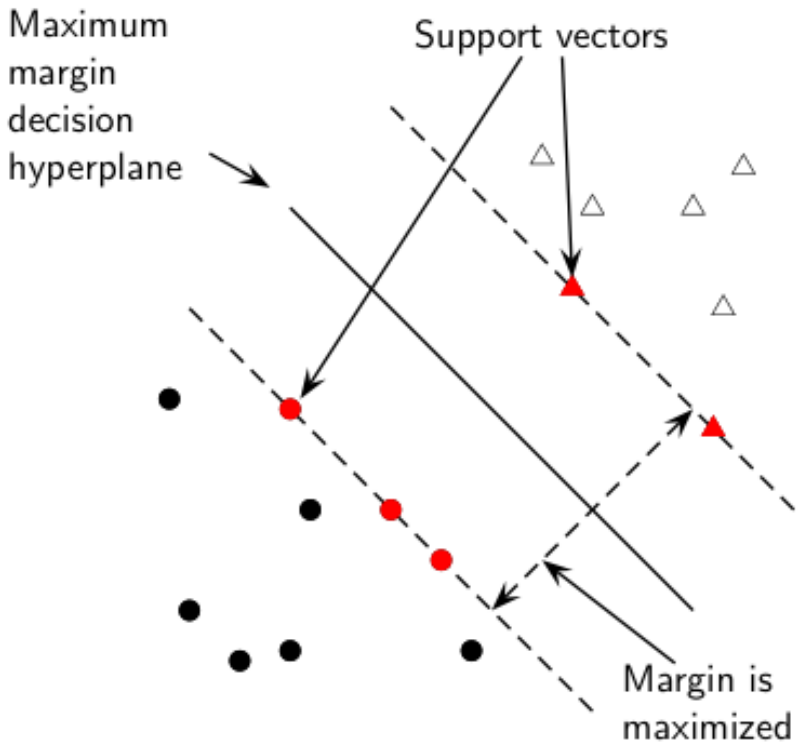
  defined by **support vectors**



Why maximize the margin?

Points near decision surface

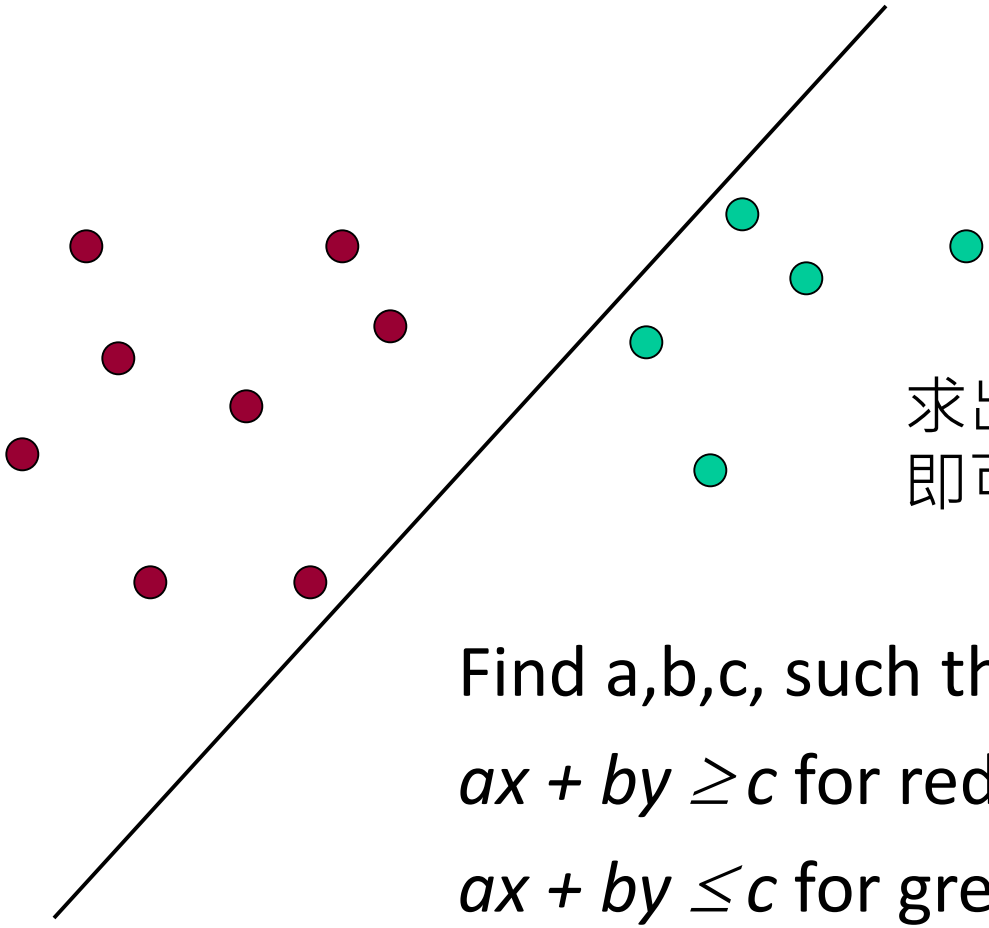
→ uncertain classification decisions



A classifier with a large margin makes certainty classification decisions.

- to reduce errors in measurement or doc. variation

Linear programming



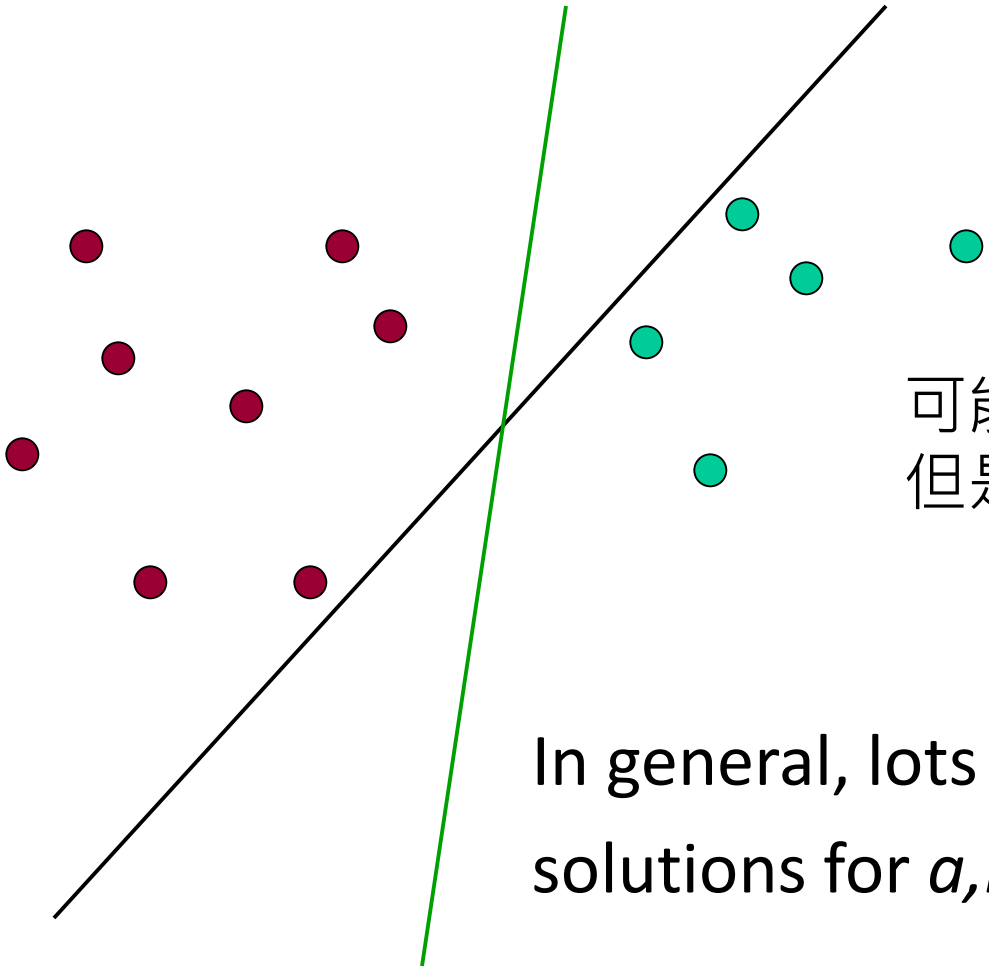
求出多項式的係數
即可決定該平面

Find a,b,c, such that

$ax + by \geq c$ for red points

$ax + by \leq c$ for green points.

Which Hyperplane?

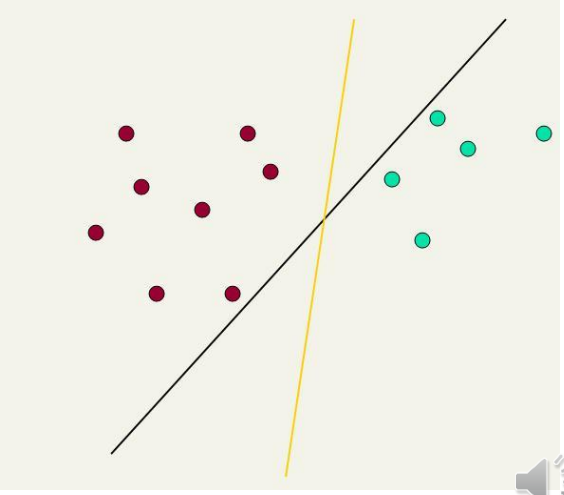


可能有多組解
但是要挑哪個超平面

In general, lots of possible solutions for *a,b,c.*

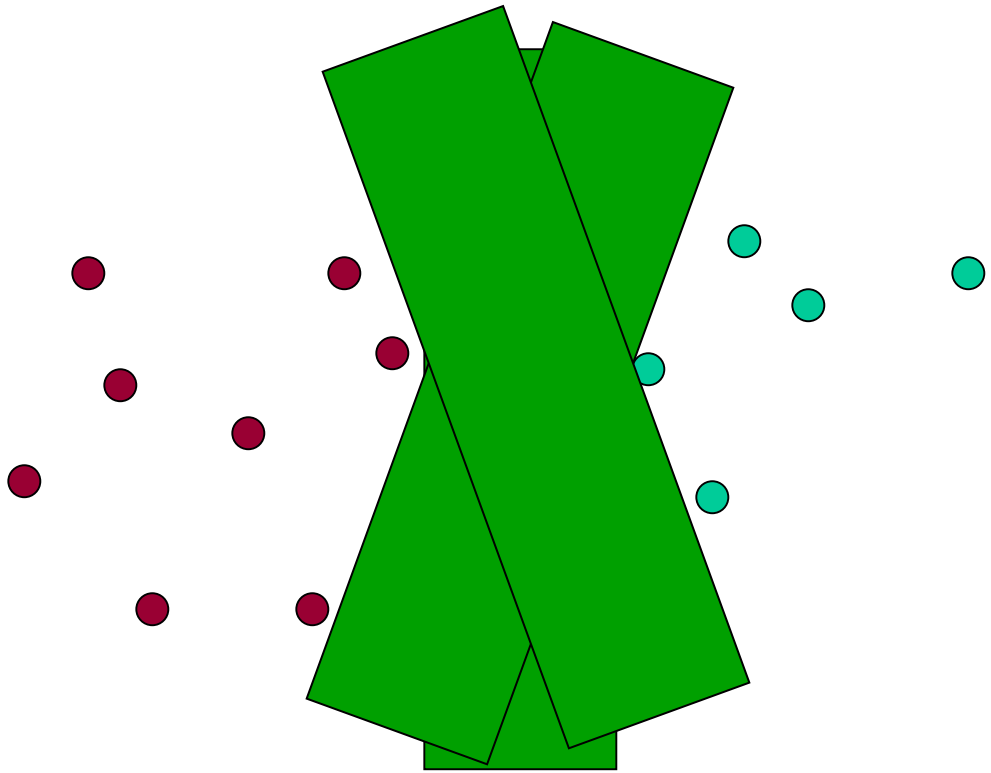
Which Hyperplane?

- Lots of possible solutions for *a,b,c.*
- Some methods find a separating hyperplane, but not the optimal one, while the other methods find an optimal separating hyperplane
- Which points should influence optimality?
  - All points 用全部的點去求最佳解
    - Linear regression
  - Only “difficult points” close to decision boundary 只用邊界附近的困難點
    - Support vector machines (SVM)



Which Hyperplane?

- If you have to place a fat separator between classes, you have less choices, and so the capacity of the model has been decreased



Formalize an SVM with algebra

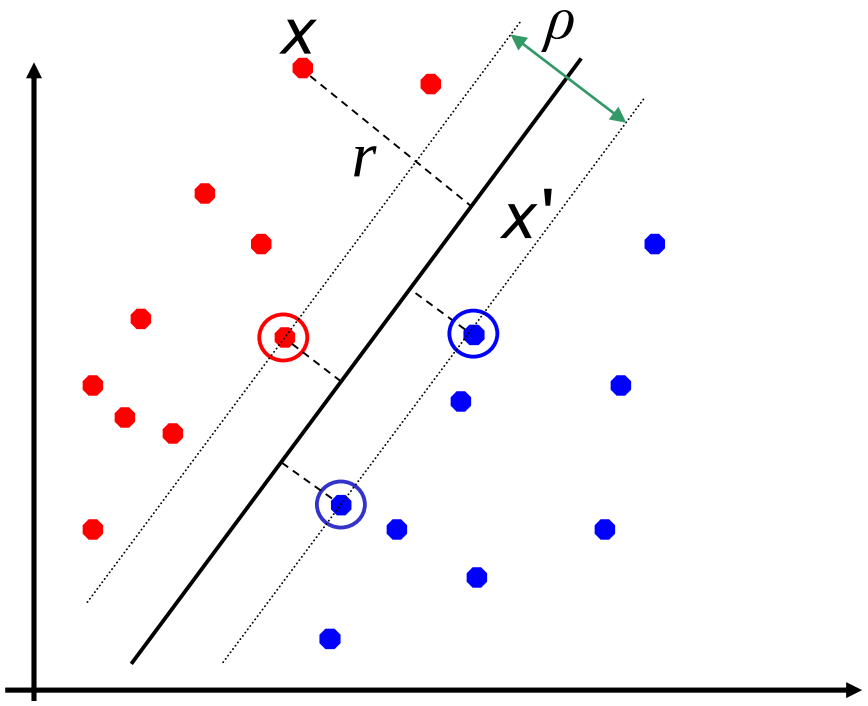
- Hyperplane : an n-dimensional generalization of a plane
- Decision hyperplane :
  - given a normal vector w (weight vector) which is perpendicular to the hyperplane
  - all points x on the hyperplane satisfy $w^T x + b = 0$
  - any point in two training set will individually satisfy

$$w^T x + b = +1$$

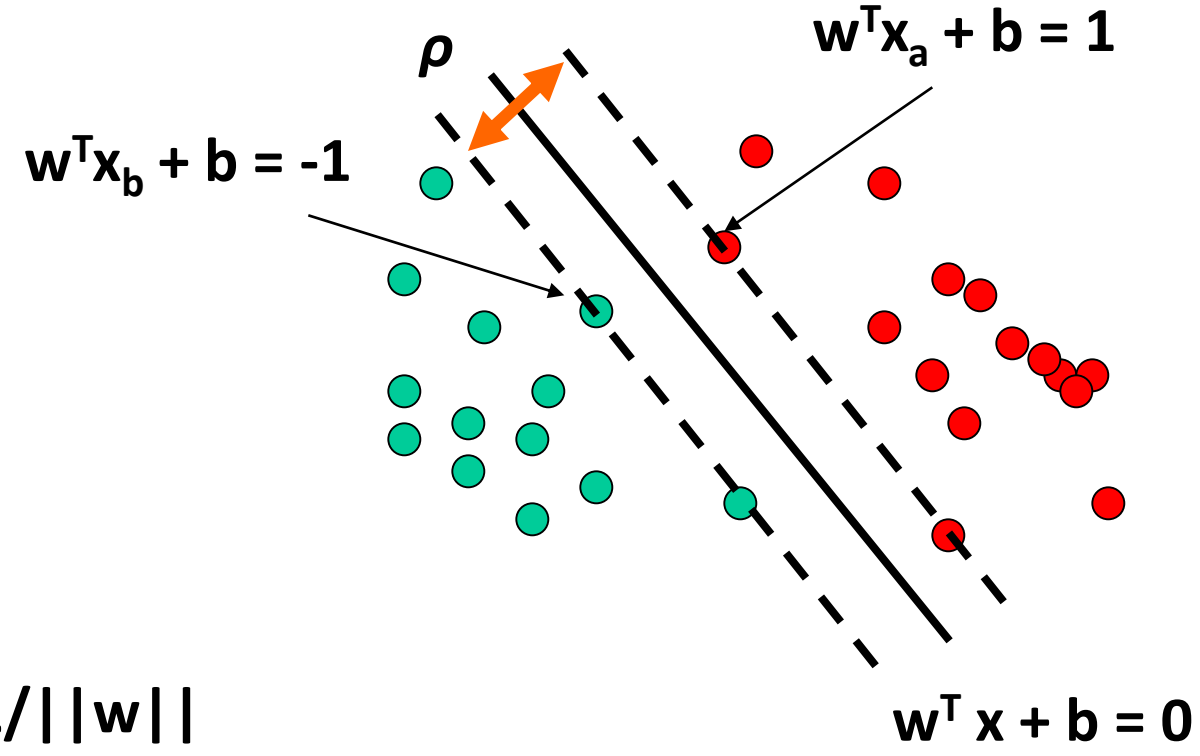
$$w^T x + b = -1$$

Geometric Margin

- Distance from example to the separator is $r = y\dfrac{\mathbf{w}^T\mathbf{x}+b}{\|\mathbf{w}\|}$
- Examples closest to the hyperplane are ***support vectors***.
- ***Margin*** ρ of the separator is the width of separation between support vectors of classes.



Linear Support Vector Machine

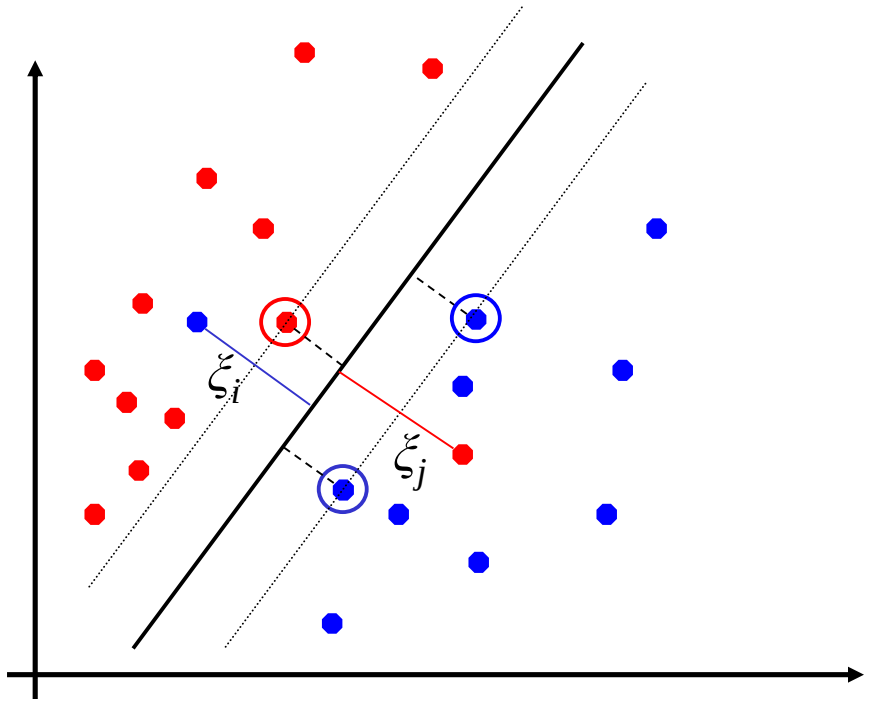


- **Hyperplane**
  $w^T x + b = 0$

- This implies:
  $w^T(x_a - x_b) = 2$
  $\rho = ||x_a - x_b|| = 2/||w||$

Soft Margin Classification

- If the training set is not linearly separable, *slack variables* ξ_i can be added to allow misclassification of difficult or noisy examples.
- Make it allow some errors.



SVM Resources

- SVM Light, Cornell University
  - http://svmlight.joachims.org/
- libSVM, National Taiwan University
  - http://www.csie.ntu.edu.tw/~cjlin/libsvm/
  - Reference
    - http://www.cmlab.csie.ntu.edu.tw/~cyy/learning/tutorials/libsvm.pdf
    - http://ntu.csie.org/~piaip/svm/svm_tutorial.html

- Reference
  - 支持向量機教學文件（中文版）

    http://www.cmlab.csie.ntu.edu.tw/~cyy/learning/tutorials/SVM1.pdf
  - Support Vector Machines 簡介

    http://www.cmlab.csie.ntu.edu.tw/~cyy/learning/tutorials/SVM2.pdf
  - Support Vector Machine 簡介

    http://www.cmlab.csie.ntu.edu.tw/~cyy/learning/tutorials/SVM3.pdf

Classification with SVMs

- Given a new point

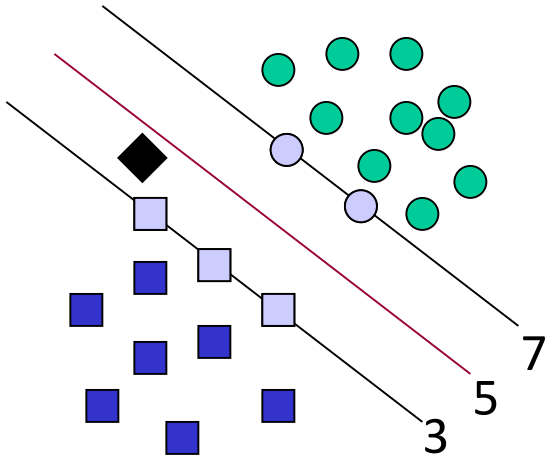
  score its projection onto the hyperplane:
  - compute score: $wx + b$
  - set confidence threshold t.

計算離哪邊較近，並給予門檻值

Score > t: yes

Score < -t: no

Else: don't know



SVM : discussion

Multiclass SVMs

SVMs: inherently two-class classifiers.

- Most common technique in practice: build |C| *one-versus-rest* classifiers (commonly referred to as "*one-versus-all*" or OVA classification), and choose the class which classifies the test data with greatest margin
- Another strategy: build a set of *one-versus-one* classifiers, and choose the class that is selected by the most classifiers.
  - this involves building $|C|(|C| - 1)/2$ classifiers
  - use binary decision tree or majority votes.

Exercise

用二元分類器模擬多分類效果，假設有政治、娛樂、科技三類

- 建立政治-其他、娛樂-其他、科技-其他三個分類器，再取離得最近的當作分類結果；若有n個類別，需要建立n個分類器
- 任二類建立一個分類器，最後採總積分制決定分類結果。例如

  政治-娛樂　　歸類為政治

  政治-科技　　歸類為科技

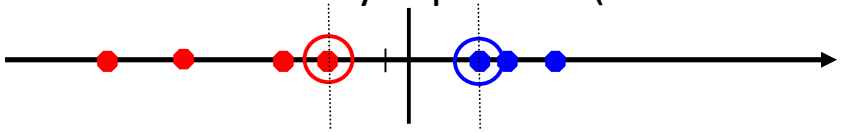
  娛樂-科技　　歸類為科技

  若採積分制，最後結果為科技>政治>娛樂

  若有n個類別，需要建立C_2^n個分類器

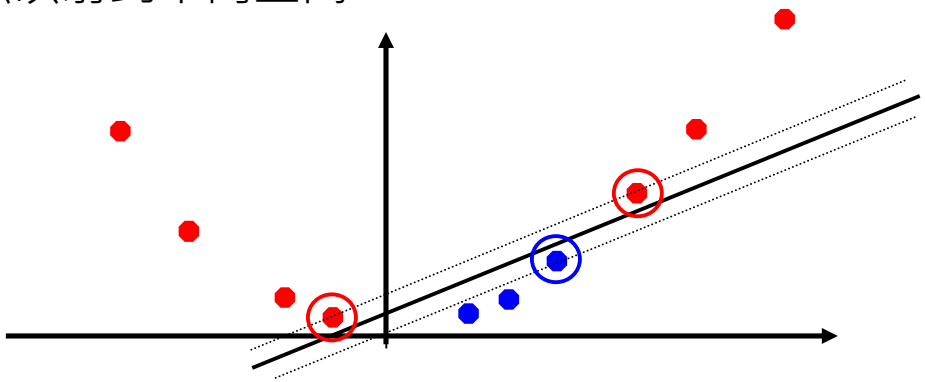
Non-linear SVMs

- Datasets that are linearly separable (with some noise) work out great:



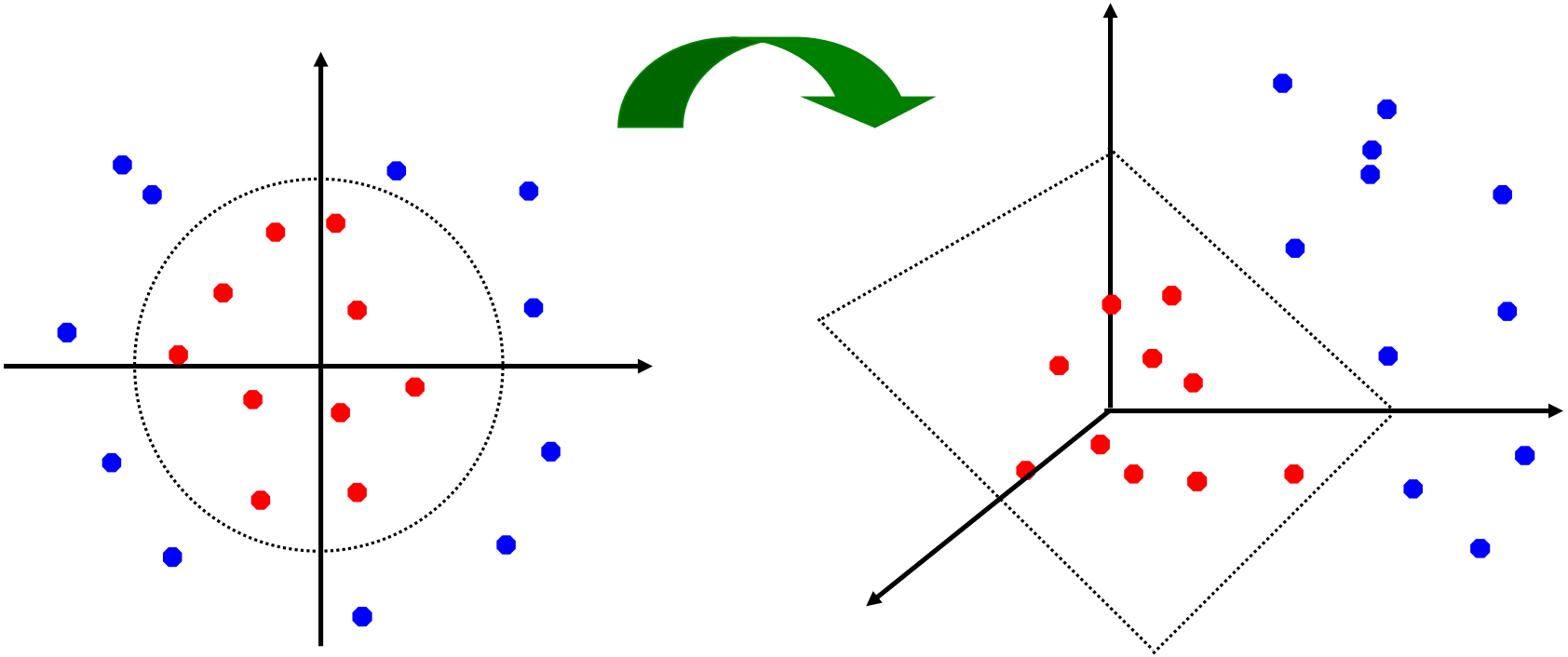
- But what are we going to do if the dataset is just too hard? 分不開怎麼辦

- How about … mapping data to a higher-dimensional space:
  想辦法映射到不同空間



Non-linear SVMs: Feature spaces

- General idea: the original feature space can always be mapped to some higher-dimensional feature space where the training set is separable:

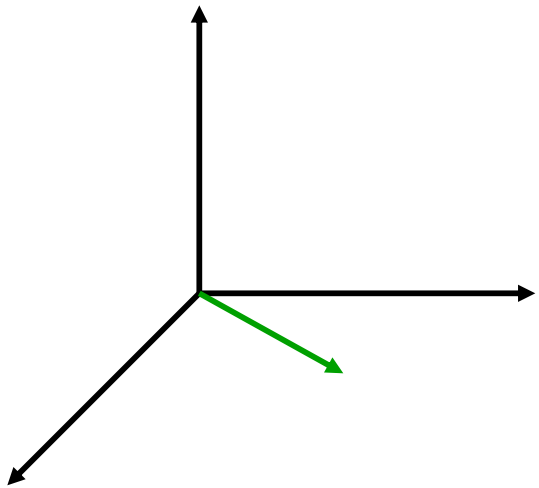


SVM is good for Text Classification

- Documents are zero along almost all axes
- Most document pairs are very far apart (i.e., not strictly orthogonal, but only share very common words and a few scattered others)

  其實文件很多軸上的值是 0 ;

  在空間軸上離很開



- Virtually all document sets are separable, for most any classification. This is why linear classifiers are quite successful in this domain

  文件混在一起的情況不多，

  所以用Linear分類器, 在文件分類上很適合

Issues in Text Classification

(1) What kind of classifier to use

- Document Classification is useful for many commercial applications
- What kind of classifier to use ?
  - How much training data do you have?

    需要準備多少訓練資料
    - None
    - Very little
    - Quite a lot
    - A huge amount and its growing

If you have no labeled training data

- Try hand-written rules solution 人工規則
  - If (Baseball OR Basketball) then categorize as Sport
- In practice, rules get a lot bigger than this 通常規則很多
- With careful crafting (human tuning on development data) performance is high: 但是效果很好
- Amount of work required is huge 但仍大量人工檢查與維護

If you have fairly little data ?

- Naïve Bayes should do well in such circumstances (Ng and Jordan 2002 NIPS) 適合訓練用文件只有很少的時候
- The practical answer is to get more labeled data as soon as you can
  - How to let people be willing to label data for you ?

    可思考如何運用網路上的資料

    Ex. 信件分類、書籤、Social Tagging

If you have a huge amount of data ?

- Great in theory for doing accurate classification
- But expensive methods like SVMs (train time) or kNN (test time) are quite impractical

  運算量大的演算法不合用
- Try Naïve Bayes again.

(2) Large and difficult categories

- Easy for small number of well-separated categories
- Accurate classification over large sets of closely related classes is *inherently difficult*.
  - Ex. Web directories (e.g. the Yahoo! Directory consists of over 200,000 categories or the Open Directory Project)
  - Ex. library classification schemes (Library of Congress)
  - Classifier combination is a useful technique
    - Voting of multiple classifiers
  - Use a hybrid automatic/manual solution

(3) Other techniques

- Try differentially weighting contributions from different document zones:
  - Upweighting title words helps (Cohen & Singer 1996) 提高標題權重
  - Upweighting the first sentence of each paragraph helps (Murata, 1999) 提高每段的第一句權重
  - Upweighting sentences that contain title words helps (Ko et al, 2002) 提高包含標題之句子的權重
  - Summarization as feature selection for text categorization (Kolcz, Prabakarmurthi, and Kolita, CIKM 2001) 先做自動摘要

(4) Problem of Concept drift

- Categories change over time 類別是會隨時間變的
- Example: “president of the united states”
  - 1999: clinton is great feature
  - 2002: clinton is bad feature 已經不是總統了
- One measure of a text classification system is how well it protects against concept drift.
  - 多久需要檢視訓練資料，並重新訓練？

Dumais et al. 1998 : Reuters - Accuracy

(a)

	NB	Rocchio	kNN	SVM
micro-avg-L (90 classes)	80	85	86	89
macro-avg (90 classes)	47	59	60	60

(b)

	NB	Rocchio	kNN	trees	SVM
earn	96	93	97	98	98
acq	88	65	92	90	94
money-fx	57	47	78	66	75
grain	79	68	82	85	95
crude	80	70	86	85	89
trade	64	65	77	73	76
interest	65	63	74	67	78
ship	85	49	79	74	86
wheat	70	69	77	93	92
corn	65	48	78	92	90
micro-avg (top 10)	82	65	82	88	92
micro-avg-D (118 classes)	75	62	n/a	n/a	87

Evaluation measure: F_1 Naive Bayes does pretty well, but some methods beat it consistently (e.g., SVM).

Results for Kernels (Joachims 1998)

					SVM (poly) degree d =					SVM (rbf) width γ =			
	Bayes	Rocchio	C4.5	k-NN	1	2	3	4	5	0.6	0.8	1.0	1.2
earn	95.9	96.1	96.1	97.3	98.2	98.4	**98.5**	98.4	98.3	**98.5**	98.5	98.4	98.3
acq	91.5	92.1	85.3	92.0	92.6	94.6	**95.2**	95.2	95.3	95.0	95.3	95.3	**95.4**
money-fx	62.9	67.6	69.4	78.2	66.9	72.5	75.4	74.9	**76.2**	74.0	75.4	**76.3**	75.9
grain	72.5	79.5	89.1	82.2	91.3	93.1	**92.4**	91.3	89.9	**93.1**	91.9	91.9	90.6
crude	81.0	81.5	75.5	85.7	86.0	87.3	88.6	**88.9**	87.8	**88.9**	89.0	88.9	88.2
trade	50.0	77.4	59.2	77.4	69.2	75.5	76.6	77.3	**77.1**	76.9	78.0	**77.8**	76.8
interest	58.0	72.5	49.1	74.0	69.8	63.3	67.9	73.1	**76.2**	74.4	75.0	**76.2**	76.1
ship	78.7	83.1	80.9	79.2	82.0	85.4	86.0	**86.5**	86.0	**85.4**	86.5	87.6	87.1
wheat	60.6	79.4	85.5	76.6	83.1	84.5	85.2	**85.9**	83.8	**85.2**	85.9	85.9	85.9
corn	47.3	62.2	87.7	77.9	86.0	86.5	85.3	**85.7**	83.9	**85.1**	85.7	85.7	84.5
microavg.	**72.0**	**79.9**	**79.4**	**82.3**	84.2	85.1	85.9	86.2	85.9	86.4	86.5	86.3	86.2
					combined: **86.0**					combined: **86.4**			

Yang & Liu: SVM vs. Other Methods

Table 1: Performance summary of classifiers

method	miR	miP	miF1	maF1	error
SVM	.8120	.9137	.8599	.5251	.00365
KNN	.8339	.8807	.8567	.5242	.00385
LSF	.8507	.8489	.8498	.5008	.00414
NNet	.7842	.8785	.8287	.3765	.00447
NB	.7688	.8245	.7956	.3886	.00544

miR = micro-avg recall; miP = micro-avg prec.;
miF1 = micro-avg F1; maF1 = macro-avg F1.

Text Classification : conclusion

- Choose a approach
  - Do no classification 不分類
  - Do it all manually 人工分類
  - Do it all with an automatic classifier 全自動分類
    - Mistakes have a cost 要挑出錯也要成本
    - Do it with a combination of automatic classification and manual review of uncertain/difficult/“new” cases
- Commonly the last method is most cost efficient and is adopted

Discussions