Lecture 7: Text Classification and Naive Bayes

Information Retrieval
Computer Science Tripos Part II

Simone Teufel

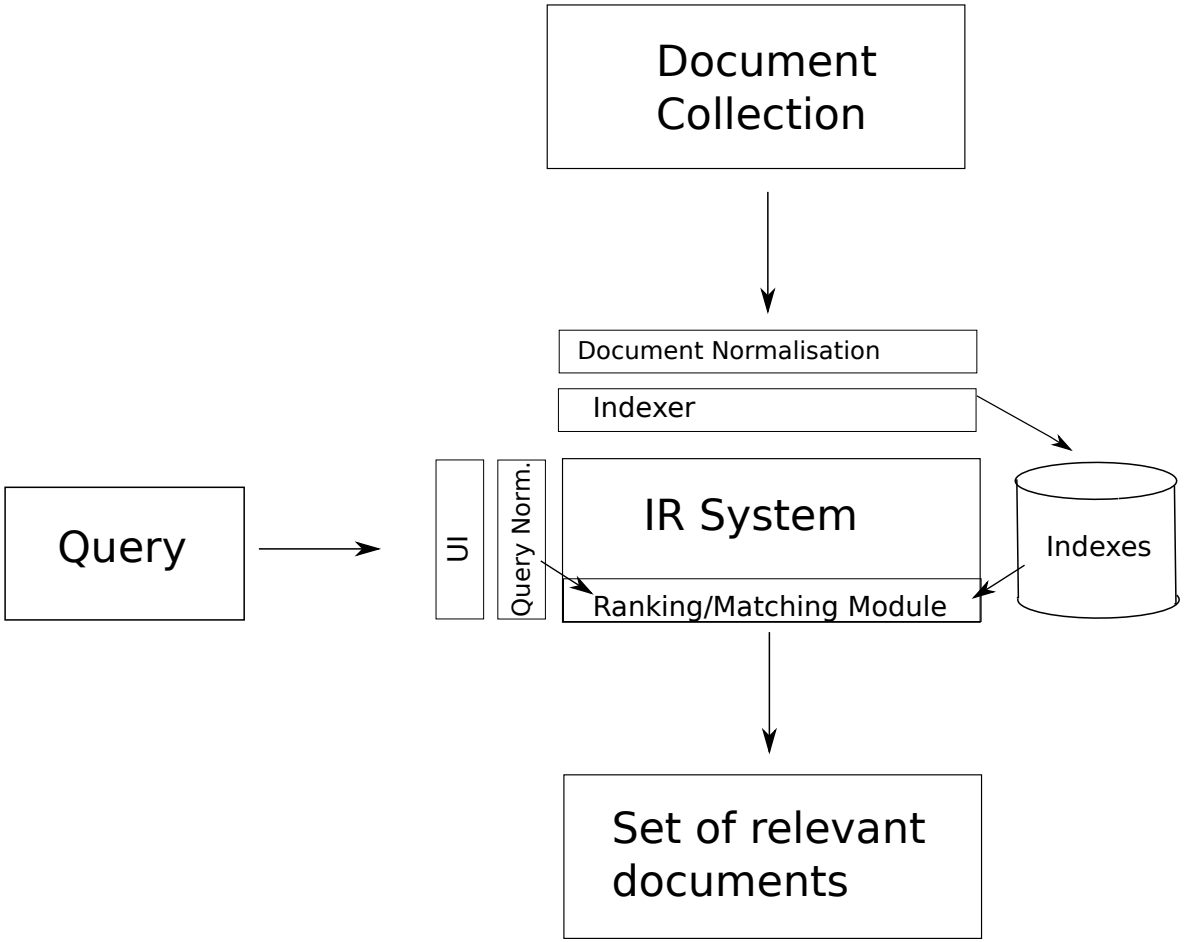
Natural Language and Information Processing (NLIP) Group

UNIVERSITY OF CAMBRIDGE

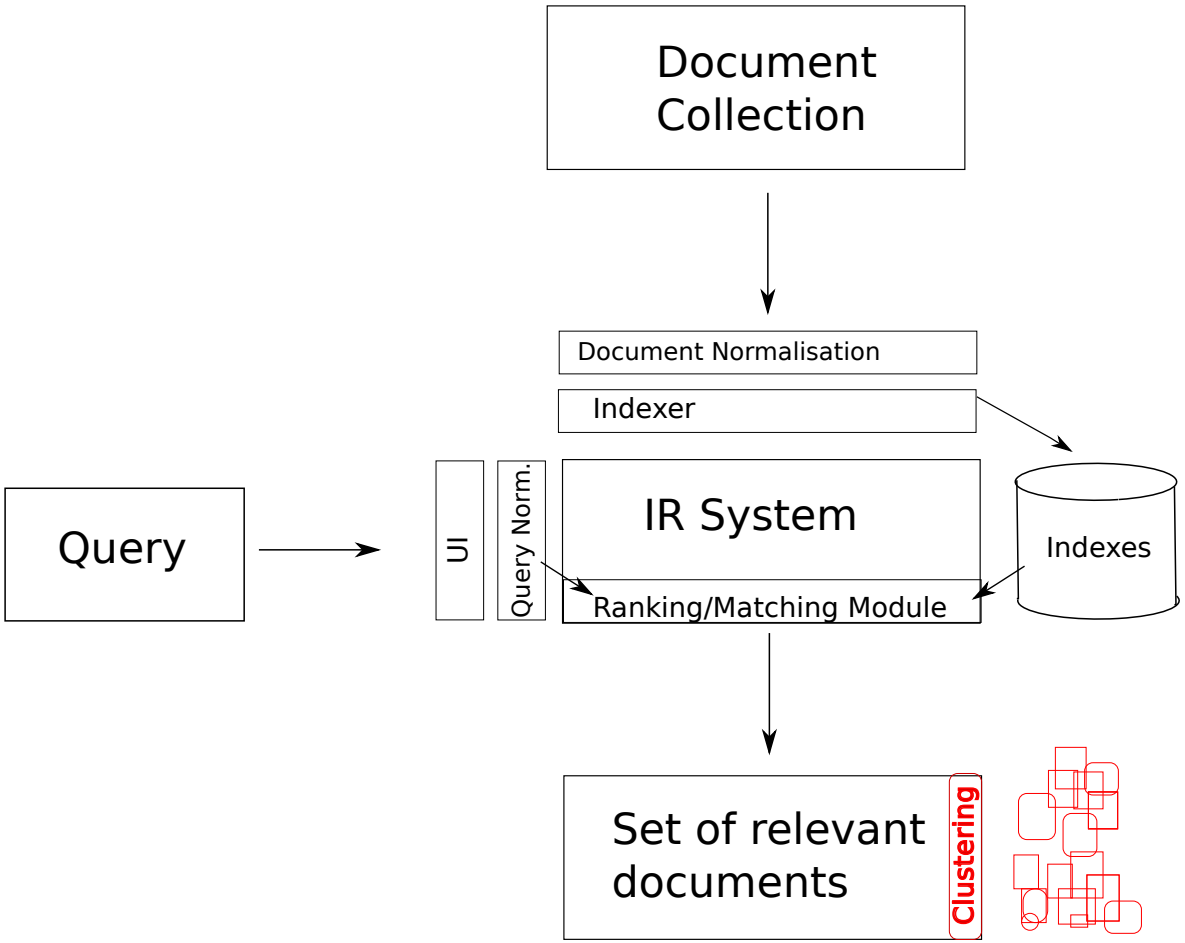
Simone.Teufel@cl.cam.ac.uk

Lent 2014

IR System Components

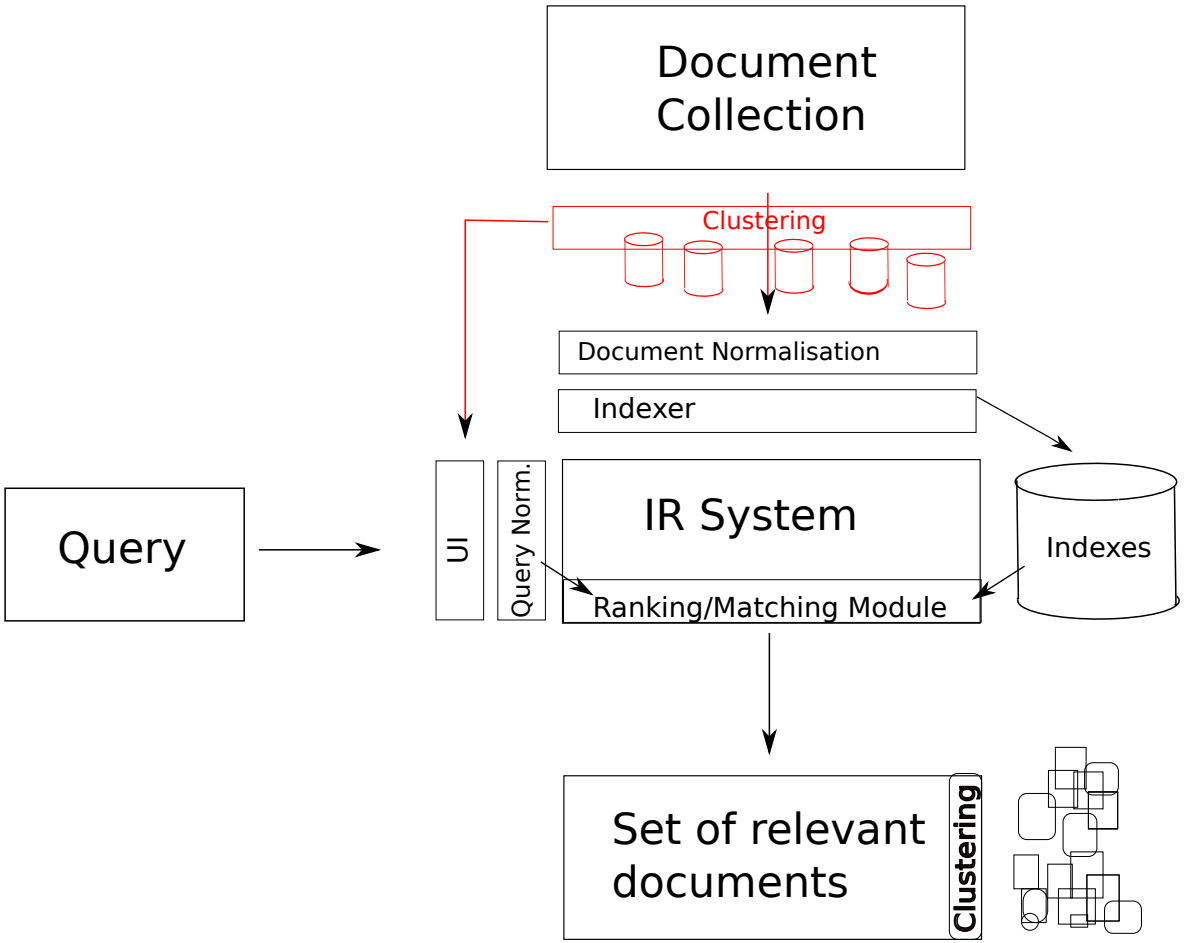


IR System Components



Still: clustering your output documents

IR System Components



Today: classifying your input documents

Overview

Overview

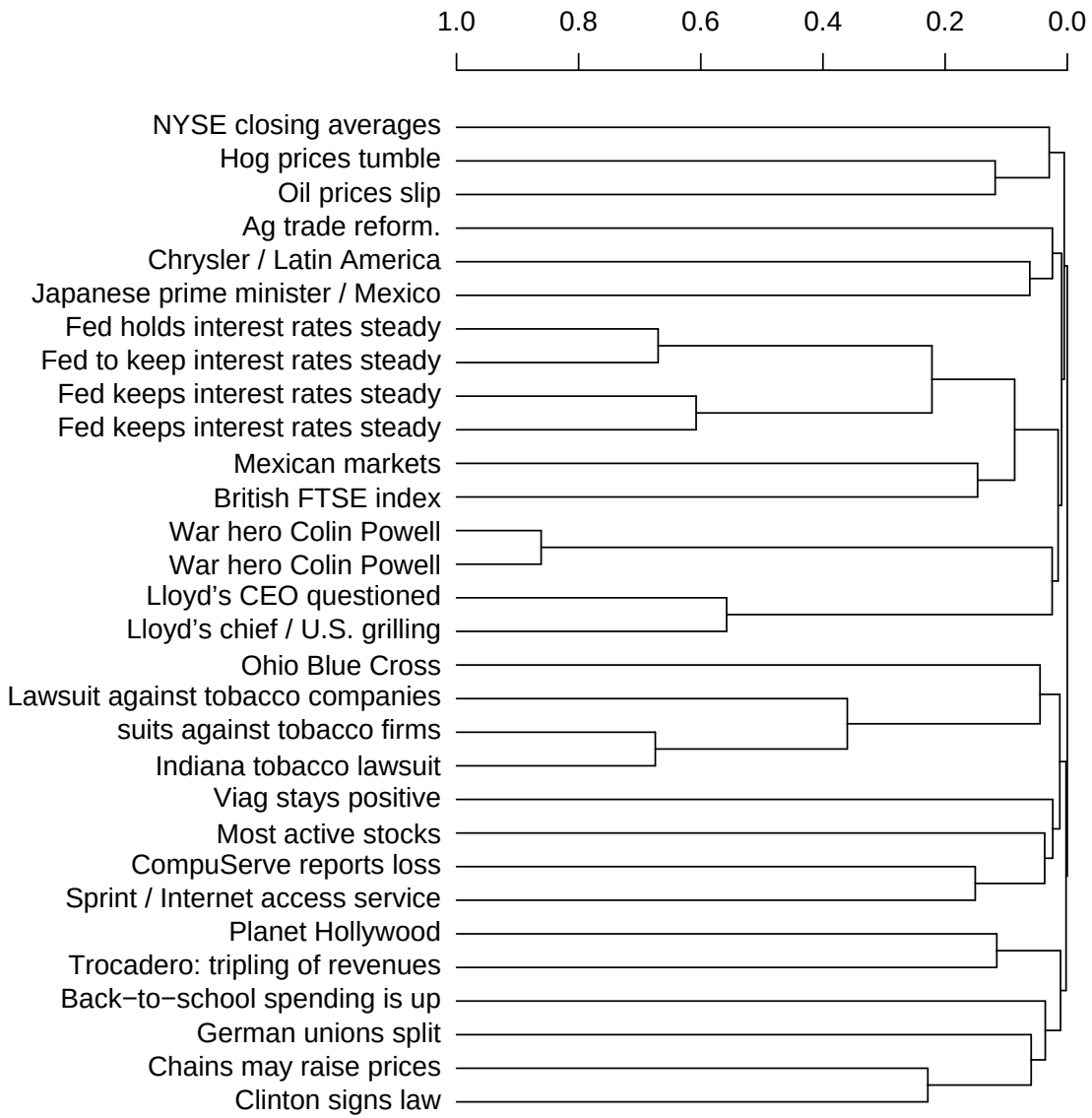
Hierarchical clustering

- Imagine we now want to create a hierarchy in the form of a binary tree.
- Assumes a similarity measure for determining the similarity of two clusters.
- Up to now, our similarity measures were for documents.
- We will look at different cluster similarity measures.
- Main algorithm: HAC (hierarchical agglomerative clustering)

HAC: Basic algorithm

- Start with each document in a separate cluster
- Then repeatedly merge the two clusters that are most similar
- Until there is only one cluster.
- The history of merging is a hierarchy in the form of a binary tree.
- The standard way of depicting this history is a dendrogram.

A dendrogram



Term–document matrix to document–document matrix

Log frequency weighting and cosine normalisation

SaS	PaP	WH
0.789	0.832	0.524
0.515	0.555	0.465
0.335	0.000	0.405
0.000	0.000	0.588

SaS	P(SaS,SaS)	P(PaP,SaS)
PaP	P(SaS,PaP)	P(PaP,PaP)
WH	P(SaS,WH)	P(PaP,WH)
	SaS	PaP

SaS	1	.94	.79
PaP	.94	1	.69
WH	.79	.69	1
	SaS	PaP	WH

- Applying the proximity metric to all pairs of documents. . .
- creates the document-document matrix, which reports similarities/distances between objects (documents)
- The diagonal is trivial (identity)
- As proximity measures are symmetric, the matrix is a triangle

Hierarchical clustering: agglomerative (BottomUp, greedy)

Given: a set $X = x_1, ...x_n$ of objects;
Given: a function $sim : \mathcal{P}(X) \times \mathcal{P}(X) \rightarrow \mathcal{R}$

for i:= 1 **to** n **do**
    $c_i := x_i$
C :=c_1, ... c_n
j := n+1
while C $>$ 1 **do**
    $(c_{n_1}, c_{n_2}) := max_{(c_u, c_v) \in C \times C} sim(c_u, c_v)$
    $c_j := c_{n_1} \cup c_{n_2}$
    C := C $\{ c_{n_1}, c_{n_2} \} \cup c_j$
    j:=j+1
end

Similarity function $sim : \mathcal{P}(X) \times \mathcal{P}(X) \rightarrow \mathcal{R}$ measures similarity between clusters, not objects

Computational complexity of the basic algorithm

- First, we compute the similarity of all $N \times N$ pairs of documents.
- Then, in each of N iterations:
  - We scan the $O(N \times N)$ similarities to find the maximum similarity.
  - We merge the two clusters with maximum similarity.
  - We compute the similarity of the new cluster with all other (surviving) clusters.
- There are $O(N)$ iterations, each performing a $O(N \times N)$ "scan" operation.
- Overall complexity is $O(N^3)$.
- Depending on the similarity function, a more efficient algorithm is possible.

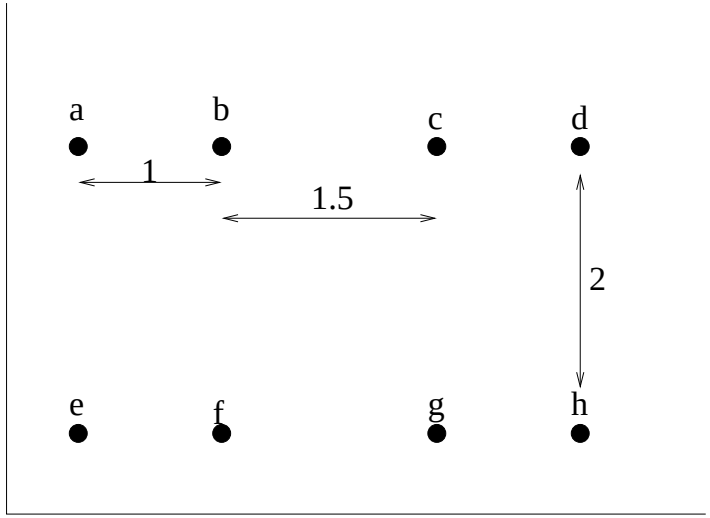
Hierarchical clustering: similarity functions

Similarity between two clusters c_k and c_j (with similarity measure s) can be interpreted in different ways:

- Single Link Function: Similarity of two most similar members
$$sim(c_u, c_v) = max_{x \in c_u, y \in c_k} s(x, y)$$
- Complete Link Function: Similarity of two least similar members
$$sim(c_u, c_v) = min_{x \in c_u, y \in c_k} s(x, y)$$
- Group Average Function: Avg. similarity of each pair of group members
$$sim(c_u, c_v) = avg_{x \in c_u, y \in c_k} s(x, y)$$

Example: hierarchical clustering; similarity functions

Cluster 8 objects a-h; Euclidean distances (2D) shown in diagram



b	1						
c	2.5	1.5					
d	3.5	2.5	1				
e	2	$\sqrt{5}$	$\sqrt{10.25}$	$\sqrt{16.25}$			
f	$\sqrt{5}$	2	$\sqrt{6.25}$	$\sqrt{10.25}$	1		
g	$\sqrt{10.25}$	$\sqrt{6.25}$	2	$\sqrt{5}$	2.5	1.5	
h	$\sqrt{16.25}$	$\sqrt{10.25}$	$\sqrt{5}$	2	3.5	2.5	1
	a	b	c	d	e	f	g

Single Link is $O(n^2)$

b	1						
c	2.5	1.5					
d	3.5	2.5	1				
e	2	$\sqrt{5}$	$\sqrt{10.25}$	$\sqrt{16.25}$			
f	$\sqrt{5}$	2	$\sqrt{6.25}$	$\sqrt{10.25}$	1		
g	$\sqrt{10.25}$	$\sqrt{6.25}$	2	$\sqrt{5}$	2.5	1.5	
h	$\sqrt{16.25}$	$\sqrt{10.25}$	$\sqrt{5}$	2	3.5	2.5	1
	a	b	c	d	e	f	g

c	2.5	1.5					
d	3.5	2.5	1				
e	2	$\sqrt{5}$	$\sqrt{10.25}$	$\sqrt{16.25}$			
f	$\sqrt{5}$	2	$\sqrt{6.25}$	$\sqrt{10.25}$	1		
g	$\sqrt{10.25}$	$\sqrt{6.25}$	2	$\sqrt{5}$	2.5	1.5	
h	$\sqrt{16.25}$	$\sqrt{10.25}$	$\sqrt{5}$	2	3.5	2.5	1
	a+b		c	d	e	f	g
c	~~2.5~~	1.5					
d	~~3.5~~	2.5	1				
e	2	~~$\sqrt{5}$~~	$\sqrt{10.25}$	$\sqrt{16.25}$			
f	~~$\sqrt{5}$~~	2	$\sqrt{6.25}$	$\sqrt{10.25}$	1		

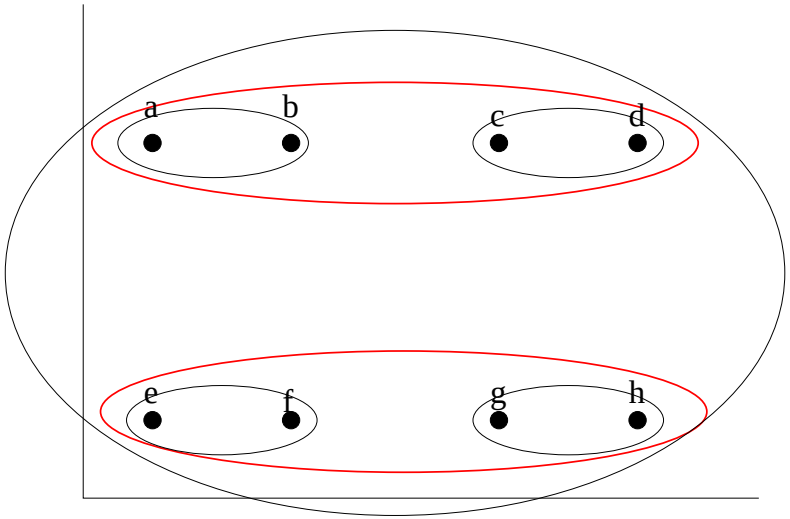
Single Link

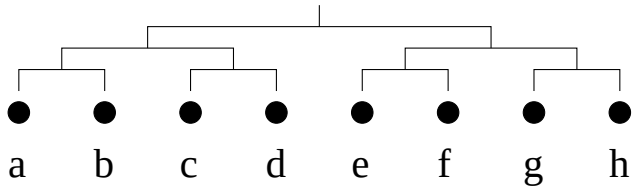
After Step 4 (a–b, c–d, e–f, g–h merged):

c–d	1.5		
e–f	2	$\sqrt{6.25}$	
g–h	$\sqrt{6.25}$	2	1.5
	a–b	c–d	e–f

"min-min" at each step

Clustering Result under Single Link





Complete Link

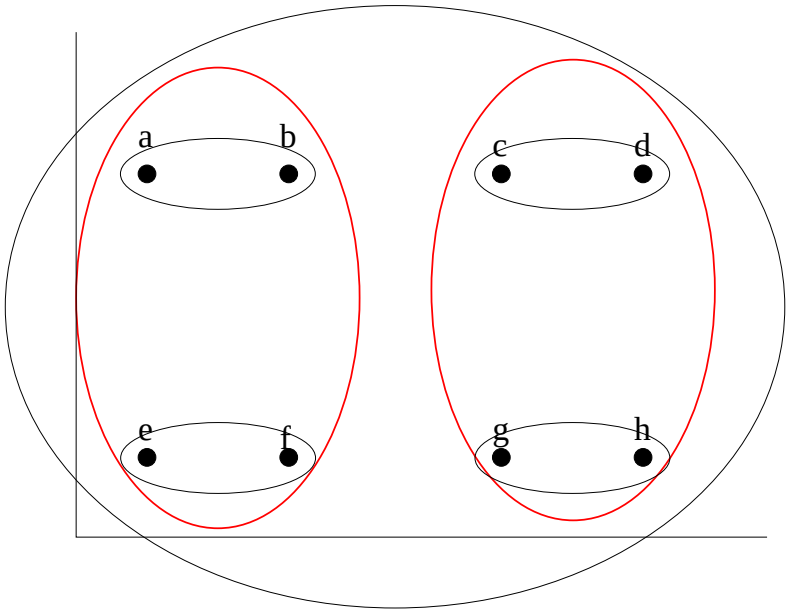
b	1						
c	2.5	1.5					
d	3.5	2.5	1				
e	2	$\sqrt{5}$	$\sqrt{10.25}$	$\sqrt{16.25}$			
f	$\sqrt{5}$	2	$\sqrt{6.25}$	$\sqrt{10.25}$	1		
g	$\sqrt{10.25}$	$\sqrt{6.25}$	2	$\sqrt{5}$	2.5	1.5	
h	$\sqrt{16.25}$	$\sqrt{10.25}$	$\sqrt{5}$	2	3.5	2.5	1
	a	b	c	d	e	f	g

After step 4 (a–b, c–d, e–f, g–h merged):

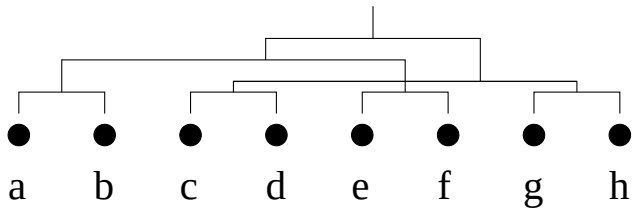
c–d	2.5	1.5				
	3.5	2.5				
e–f	2	$\sqrt{5}$	$\sqrt{10.25}$	$\sqrt{16.25}$		
	$\sqrt{5}$	2	$\sqrt{6.25}$	$\sqrt{10.25}$		
g–h	$\sqrt{10.25}$	$\sqrt{6.25}$	2	$\sqrt{5}$	2.5	1.5
	$\sqrt{16.25}$	$\sqrt{10.25}$	$\sqrt{5}$	2	3.5	2.5
	a–b		c–d		e–f	

"max-min" at each step → ab/ef and cd/gh merges next

Clustering result under complete link



Complete Link is $O(n^3)$



Example: gene expression data

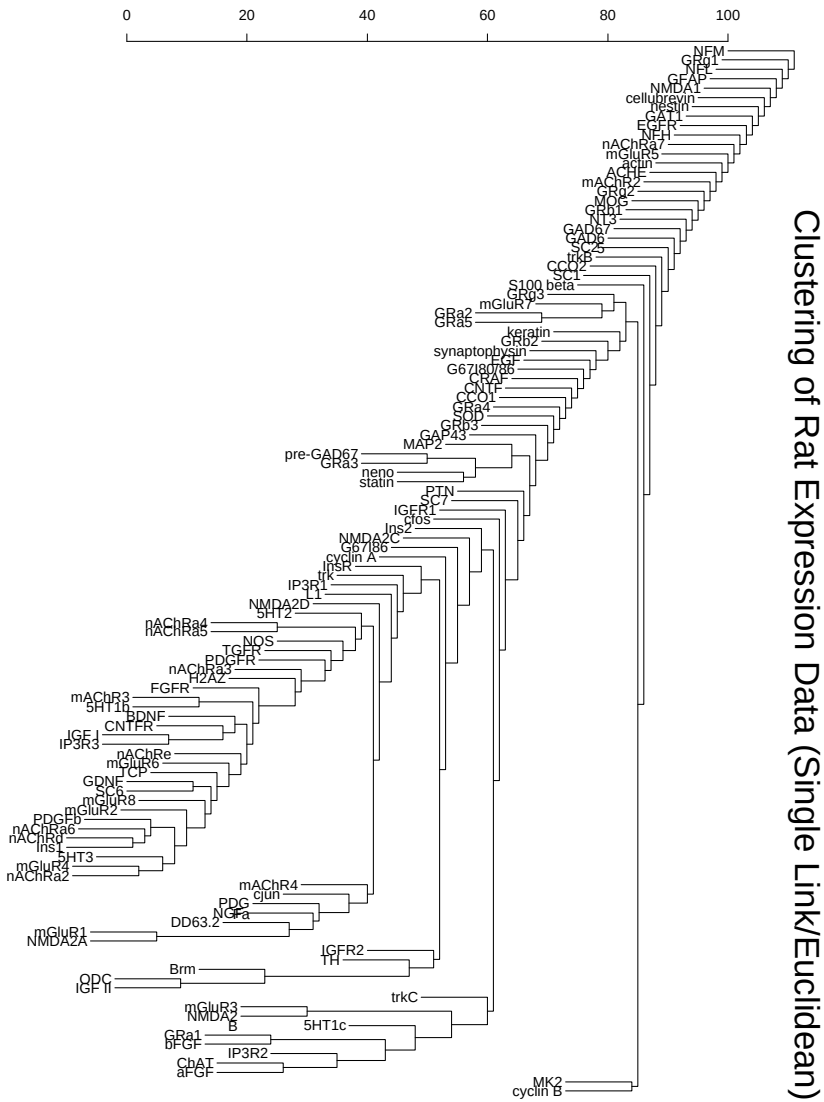
- An example from biology: cluster genes by function
- Survey 112 rat genes which are suspected to participate in development of CNS
- Take 9 data points: 5 embryonic (E11, E13, E15, E18, E21), 3 postnatal (P0, P7, P14) and one adult
- Measure expression of gene (how much mRNA in cell?)
- These measures are normalised logs; for our purposes, we can consider them as weights
- Cluster analysis determines which genes operate at the same time

Rat CNS gene expression data (excerpt)

gene	genbank locus	E11	E13	E15	E18	E21	P0	P7	P14	A
keratin	RNKER19	1.703	0.349	0.523	0.408	0.683	0.461	0.32	0.081	0
cellubrevin	s63830	5.759	4.41	1.195	2.134	2.306	2.539	3.892	3.953	2.72
nestin	RATNESTIN	2.537	3.279	5.202	2.807	1.5	1.12	0.532	0.514	0.443
MAP2	RATMAP2	0.04	0.514	1.553	1.654	1.66	1.491	1.436	1.585	1.894
GAP43	RATGAP43	0.874	1.494	1.677	1.937	2.322	2.296	1.86	1.873	2.396
L1	S55536	0.062	0.162	0.51	0.929	0.966	0.867	0.493	0.401	0.384
NFL	RATNFL	0.485	5.598	6.717	9.843	9.78	13.466	14.921	7.862	4.484
NFM	RATNFM	0.571	3.373	5.155	4.092	4.542	7.03	6.682	13.591	27.692
NFH	RATNFHPEP	0.166	0.141	0.545	1.141	1.553	1.667	1.929	4.058	3.859
synaptophysin	RNSYN	0.205	0.636	1.571	1.476	1.948	2.005	2.381	2.191	1.757
neno	RATENONS	0.27	0.704	1.419	1.469	1.861	1.556	1.639	1.586	1.512
S100 beta	RATS100B	0.052	0.011	0.491	1.303	1.487	1.357	1.438	2.275	2.169
GFAP	RNU03700	0	0	0	0.292	2.705	3.731	8.705	7.453	6.547
MOG	RATMOG	0	0	0	0	0.012	0.385	1.462	2.08	1.816
GAD65	RATGAD65	0.353	1.117	2.539	3.808	3.212	2.792	2.671	2.327	2.351
pre-GAD67	RATGAD67	0.073	0.18	1.171	1.436	1.443	1.383	1.164	1.003	0.985
GAD67	RATGAD67	0.297	0.307	1.066	2.796	3.572	3.182	2.604	2.307	2.079
G67I80/86	RATGAD67	0.767	1.38	2.35	1.88	1.332	1.002	0.668	0.567	0.304
G67I86	RATGAD67	0.071	0.204	0.641	0.764	0.406	0.202	0.052	0.022	0
GAT1	RATGABAT	0.839	1.071	5.687	3.864	4.786	4.701	4.879	4.601	4.679
ChAT	(*)	0	0.022	0.369	0.322	0.663	0.597	0.795	1.015	1.424
ACHE	S50879	0.174	0.425	1.63	2.724	3.279	3.519	4.21	3.885	3.95
ODC	RATODC	1.843	2.003	1.803	1.618	1.569	1.565	1.394	1.314	1.11
TH	RATTOHA	0.633	1.225	1.007	0.801	0.654	0.691	0.23	0.287	0
NOS	RRBNOS	0.051	0.141	0.675	0.63	0.86	0.926	0.792	0.646	0.448
GRa1	(#)	0.454	0.626	0.802	0.972	1.021	1.182	1.297	1.469	1.511

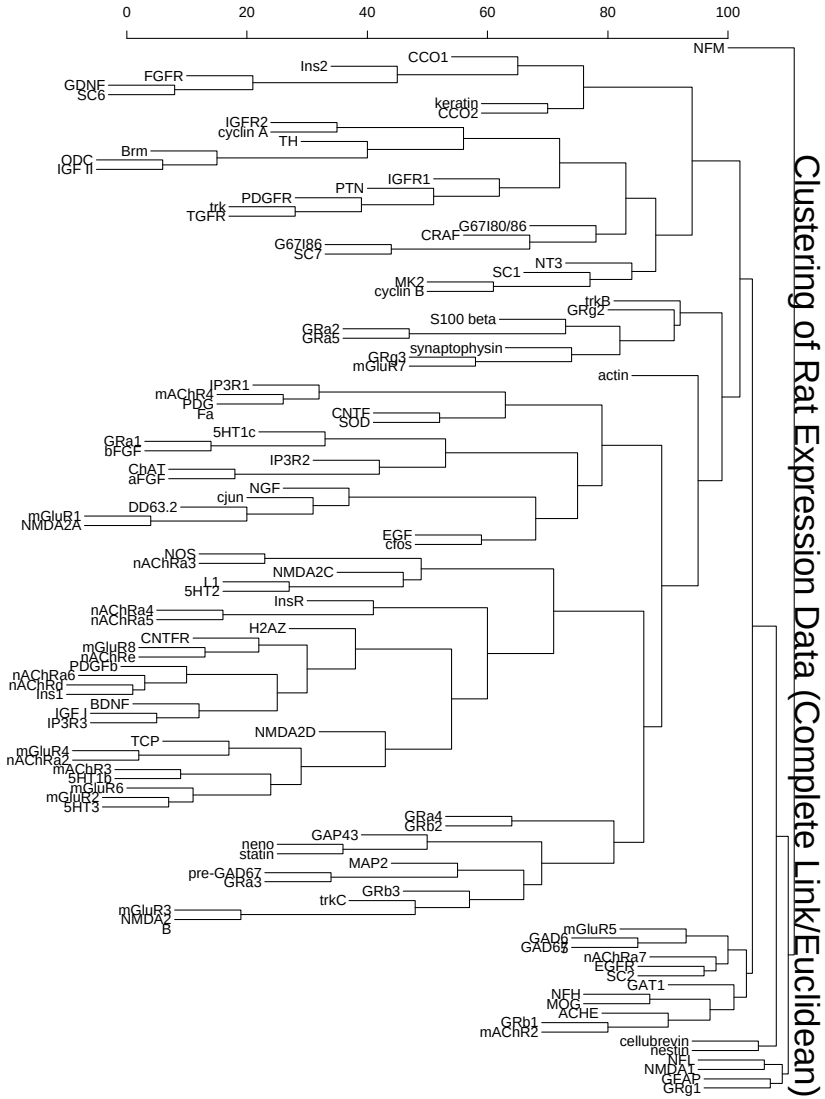
...

Rat CNS gene clustering – single link

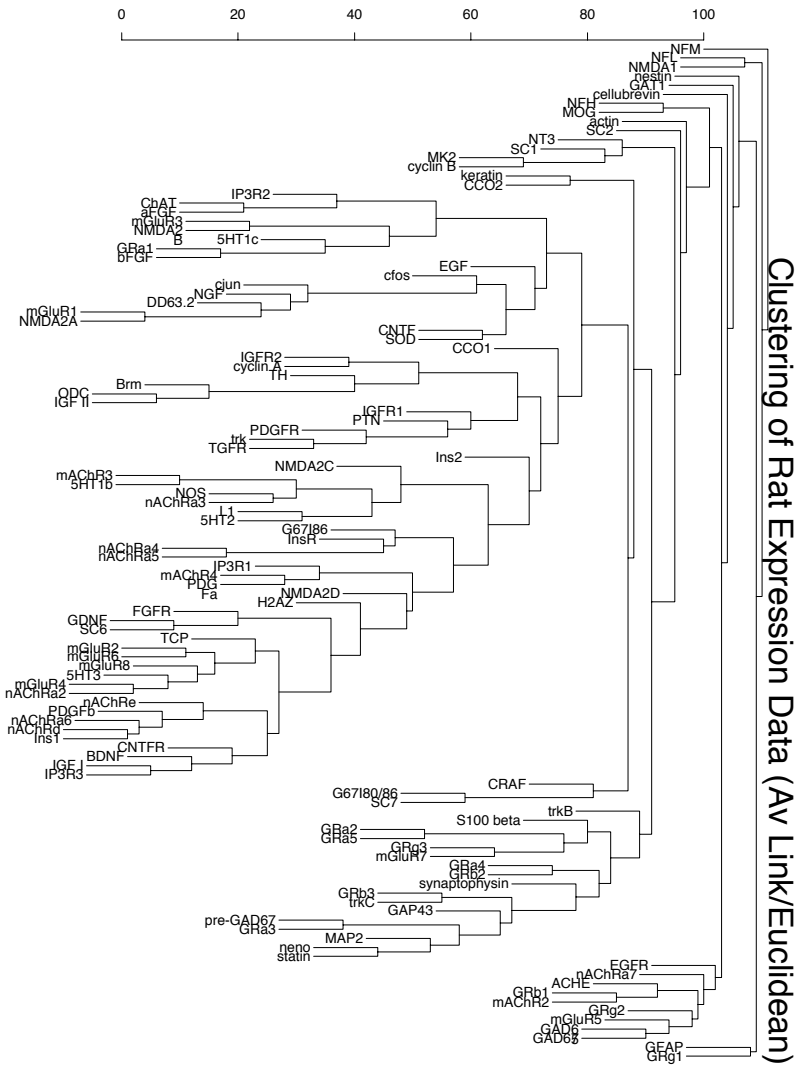


Clustering of Rat Expression Data (Single Link/Euclidean)

Rat CNS gene clustering – complete link



Rat CNS gene clustering – group average link



Flat or hierarchical clustering?

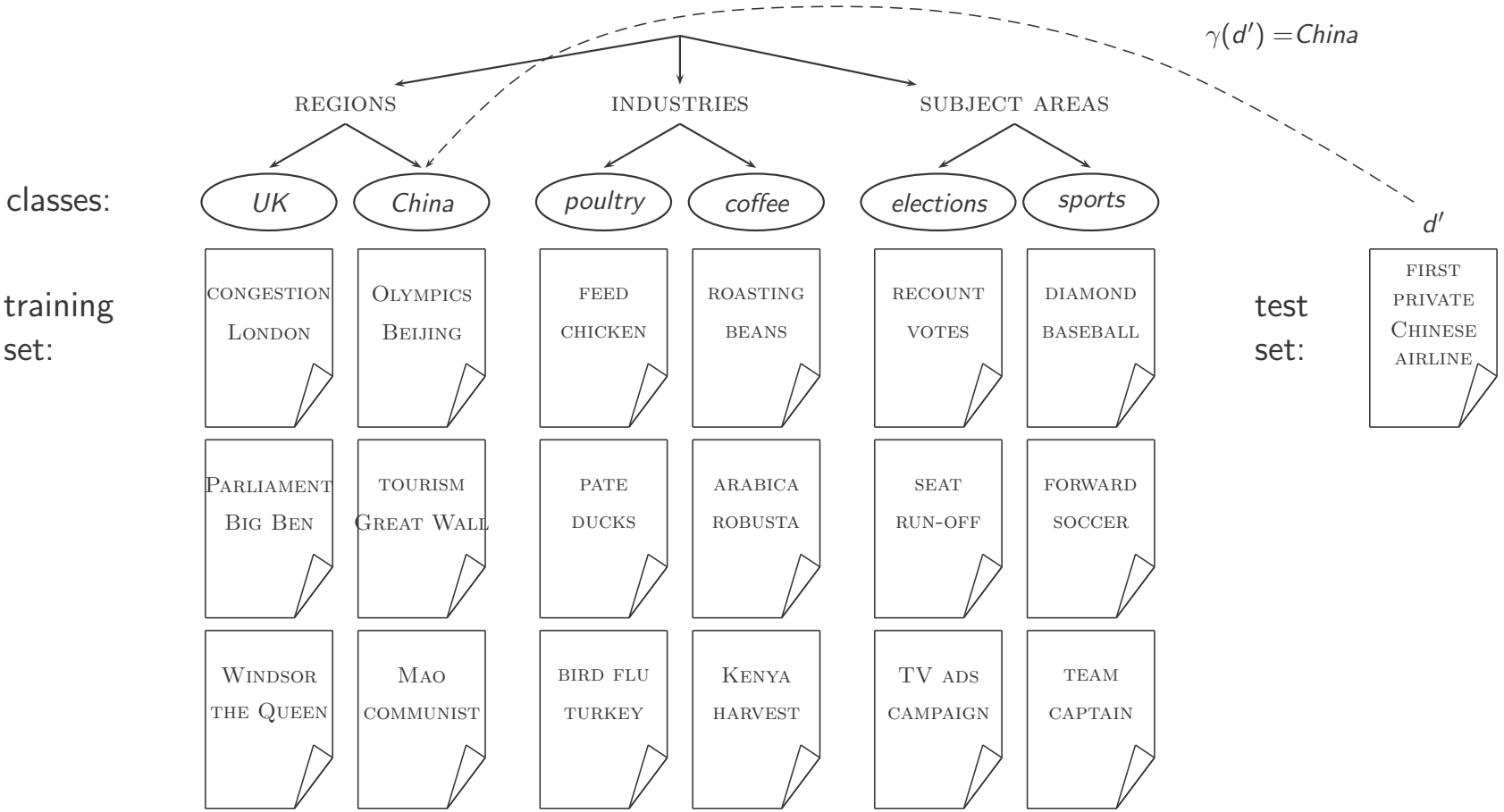
- When a hierarchical structure is desired: hierarchical algorithm
- For high efficiency, use flat clustering
- For deterministic results, use HAC
- Humans are bad at interpreting hierarchical clusterings (unless cleverly visualised)
- HAC also can be applied if K cannot be predetermined (can start without knowing K)

Overview

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
=================================================
```

Topic classification



Statistical/probabilistic classification methods

- Possible to do classification manually or by rules
- But: problems with scaling, expense
- Text classification can be seen as a learning problem
- (i) Supervised learning of a the classification function γ and (ii) application of γ to classifying new documents
- We will look at one of the simplest methods for doing this: Naive Bayes
- No free lunch: requires hand-classified training data
- But this manual classification can be done by non-experts.

Examples of how search engines use classification

- Language identification (classes: English vs. French etc.)
- Automatic detection of spam pages (spam vs. nonspam)
- Sentiment detection: is a movie or product review positive or negative (positive vs. negative)
- Topic-specific or *vertical* search – restrict search to a "vertical" like "related to health" (relevant to vertical vs. not)

Formal definition of TC: Training

Given:

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

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

$$\gamma : \mathbb{X} \to \mathbb{C}$$

Formal definition of TC: Application/Testing

Given: a description $d \in \mathbb{X}$ of a document

Determine: $\gamma(d) \in \mathbb{C}$, that is, the class most appropriate for d

The Naive 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 class

- 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_{map}:

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

Taking the log

- 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:

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

Naive Bayes classifier

- Classification rule:

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

- Simple interpretation:
  - 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 take 1: Maximum likelihood

- Estimate parameters $\hat{P}(c)$ and $\hat{P}(t_k|c)$ from train data: How?
- Prior:

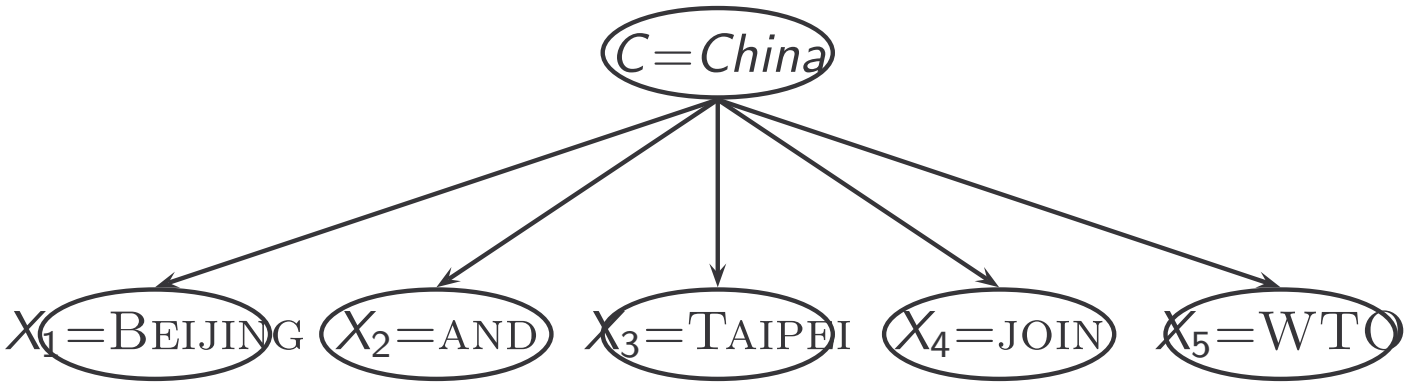
$$\hat{P}(c) = \frac{N_c}{N}$$

- 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: $\hat{P}(t_{k_1}|c) = \hat{P}(t_{k_2}|c)$, independent of positions k_1, k_2

The problem with maximum likelihood estimates: Zeros



$$P(China|d) \propto P(China) \cdot P(\text{Beijing}|China) \cdot P(\text{and}|China) \cdot P(\text{Taipei}|China) \cdot P(\text{join}|China) \cdot P(\text{WTO}|China)$$

- If WTO never occurs in class China in the train set:

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

The problem with maximum likelihood estimates: Zeros

- If there are no occurrences of WTO in documents in class China, we get a zero estimate:

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

- $\rightarrow$ We will get $P(\mathit{China}|d) = 0$ for any document that contains WTO!

To avoid zeros: Add-one 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 $|V| = M$

Example

	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 parameters of Naive Bayes classifier
- Classify test document

$|text_c| = 8$
$|text_{\overline{c}}| = 3$
B=6 (number of tokens)

Example: Parameter estimates

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

Conditional probabilities:

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

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

Example: Classification

$$\hat{P}(c|d_5) \propto 3/4 \cdot (3/7)^3 \cdot 1/14 \cdot 1/14 \approx 0.0003$$
$$\hat{P}(\overline{c}|d_5) \propto 1/4 \cdot (2/9)^3 \cdot 2/9 \cdot 2/9 \approx 0.0001$$

Thus, the classifier assigns the test document to $c = \textit{China}$. 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.

Time complexity of Naive Bayes

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

- L_{ave}: average length of a training doc, L_a: length of the test doc, M_a: number of distinct terms in the test doc, $\mathbb{D}$: training set, V: vocabulary, $\mathbb{C}$: set of classes
- $\Theta(|\mathbb{D}|L_{ave})$ is the time it takes to compute all counts. Note that $|\mathbb{D}|L_{ave}$ is T, the size of our collection.
- $\Theta(|\mathbb{C}||V|)$ is the time it takes to compute the conditional probabilities from the counts.
- Generally: $|\mathbb{C}||V| < |\mathbb{D}|L_{ave}$
- 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.

Naive Bayes is not so naive

- Multinomial model violates two independence assumptions and yet...
- Naive Bayes has won some competitions (e.g., KDD-CUP 97; prediction of most likely donors for a charity)
- 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

Not covered

- Derivation of NB formula
- Evaluation of text classification

Summary: clustering and classification

- Clustering is unsupervised learning
- Partitional clustering
  - Provides less information but is more efficient (best: $O(kn)$)
  - K-means
    - Complexity $O(knmi)$
    - Guaranteed to converge, non-optimal, dependence on initial seeds
    - Minimize avg square within-cluster difference
  - Hierarchical clustering
    - Best algorithms $O(n^2)$ complexity
    - Single-link vs. complete-link (vs. group-average)
  - Hierarchical and non-hierarchical clustering fulfills different needs (e.g. visualisation vs. navigation)
- Text classification is supervised learning
- Naive Bayes: simple baseline text classifier

Reading

- MRS chapters 13.1-13.4 for text classification
- Have a try – Weka: A data mining software package that includes an implementation of Naive Bayes