Information Retrieval & Social Web

CS 525/DS 595

Worcester Polytechnic Institute
Department of Computer Science
Instructor: Prof. Kyumin Lee

Previous Class…

Hubs & Authorities

Root set and base set

- Do a regular web search first
- Call the search result the root set
- Find all pages that are linked to or link to pages in the root set
- Call first larger set the base set
- Finally, compute hubs and authorities for the base set (which we'll view as a small web graph)

Previous Class…

Unranked Evaluation
→ Precision, Recall,
Accuracy and F-
measure

Previous Class…

Unranked Evaluation
→ Precision, Recall, Accuracy and F-measure

Ranked Evaluation
→ MAP and NDCG

Today

- Text Classification: Definition and Overview
- Vector Space Classification
  - Rocchio
  - kNN

Formal definition of Text Classification: 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:

$$\Upsilon : X \to C$$

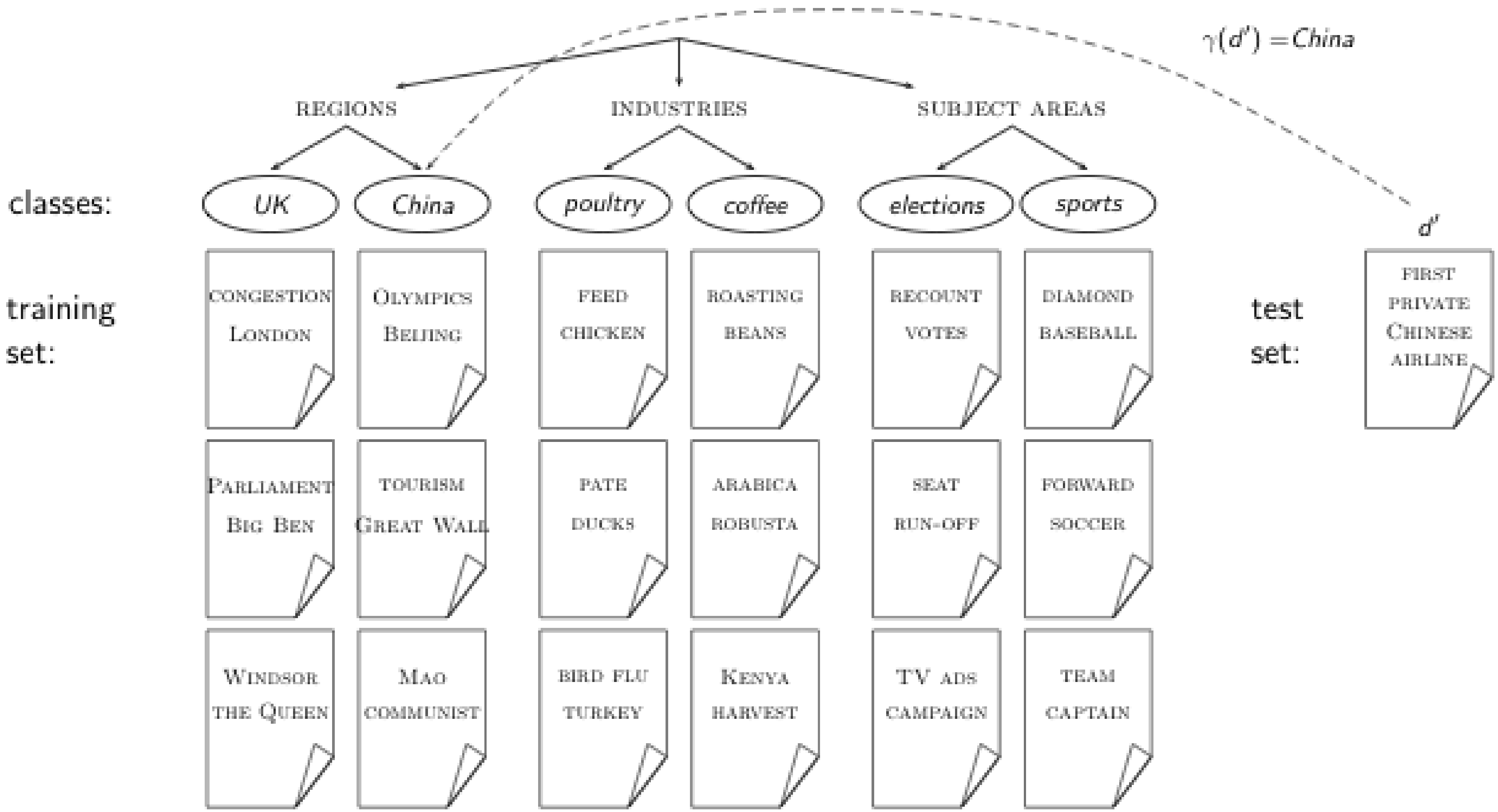
Formal definition of Text Classification : Application/Testing

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



Exercise

- Find examples of uses of text classification in information retrieval

Examples of how search engines use classification

- Language identification (classes: English vs. French etc.)
- The automatic detection of spam pages (spam vs. nonspam)
- The automatic detection of sexually explicit content (sexually explicit vs. not)
- Topic-specific or *vertical* search – restrict search to a "vertical" like "related to health" (relevant to vertical vs. not)
- Standing queries (e.g., Google Alerts)
- Sentiment detection: is a movie or product review positive or negative (positive vs. negative)

How can we classify?

Any classification method?

Classification methods: 1. Manual

- Manual classification was used by the original Yahoo! Directory. Also: Looksmart, about.com, PubMed
- Very accurate if job is done by experts
- Consistent when the problem size and team is small
- Scaling manual classification is difficult and expensive.
- → We need automatic methods for classification.

Classification methods: 2. Rule-based

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

A Verity topic (a complex classification rule)

comment line	`# Beginning of art topic definition`
top-level topic	`art ACCRUE`
topic definition modifiers	`/author = "fsmith"`
	`/date = "30-Dec-01"`
	`/annotation = "Topic created by fsmith"`
subtopictopic	`* 0.70 performing-arts ACCRUE`
evidencetopic	`** 0.50 WORD`
topic definition modifier	`/wordtext = ballet`
evidencetopic	`** 0.50 STEM`
topic definition modifier	`/wordtext = dance`
evidencetopic	`** 0.50 WORD`
topic definition modifier	`/wordtext = opera`
evidencetopic	`** 0.30 WORD`
topic definition modifier	`/wordtext = symphony`
subtopic	`* 0.70 visual-arts ACCRUE`
	`** 0.50 WORD`
	`/wordtext = painting`
	`** 0.50 WORD`
	`/wordtext = sculpture`
subtopic	`* 0.70 film ACCRUE`
	`** 0.50 STEM`
	`/wordtext = film`
subtopic	`** 0.50 motion-picture PHRASE`
	`*** 1.00 WORD`
	`/wordtext = motion`
	`*** 1.00 WORD`
	`/wordtext = picture`
	`** 0.50 STEM`
	`/wordtext = movie`
subtopic	`* 0.50 video ACCRUE`
	`** 0.50 STEM`
	`/wordtext = video`
	`** 0.50 STEM`
	`/wordtext = vcr`
	`# End of art topic`

[Verity was bought by Autonomy, which was bought by HP ...]

Classification methods: 3. Statistical/Probabilistic

- This was our definition of the classification problem – text classification as a learning problem
- (i) Supervised learning of a the classification function Υ and

  (ii) its application to classifying new documents
- We will look at a couple of methods for doing this:Rocchio, kNN, Naive Bayes
- No free lunch: requires hand-classified training data
- But this manual classification can be done by non-experts.

Vector Space Classification

Recall: Vector Space Representation

- Each document is a vector, one component for each term.
- Terms are axes.
- High-dimensional vector space: 100,000s dimensions
- Normalize vectors (documents) to unit length.
- How can we do classification in this space?

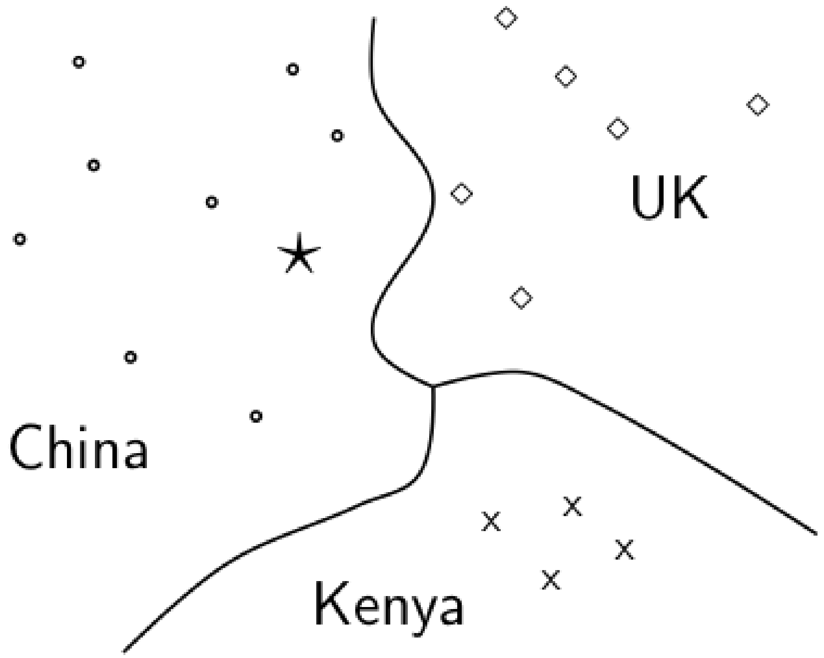
Vector Space Classification

- As before, the training set is a set of documents, each labeled with its class (e.g., topic)
- In vector space classification, this set corresponds to a labeled set of points (or, equivalently, vectors) in the vector space

- Premise 1: Documents in the same class form a contiguous region
- Premise 2: Documents from different classes don't overlap.

- We define lines, surfaces, hypersurfaces to divide regions.

Classes in the vector space



Should the document ⋆ be assigned to China, UK or Kenya?
Find separators between the classes
Based on these separators:
⋆ should be assigned to China
How do we find separators that do a good job at classifying new documents like ⋆? – Main topic of today

Rocchio

Rocchio classification: Basic idea

- Compute a centroid for each class
  - The centroid is the average of all documents in the class.
- Assign each test document to the class of its closest centroid.

Recall definition of centroid

$$\vec{\mu}(c) = \frac{1}{|D_c|} \sum_{d \in D_c} \vec{v}(d)$$

where D_c is the set of all documents that belong to class c and $\vec{v}(d)$ is the vector space representation of *d*.

Rocchio algorithm

$\textsc{TrainRocchio}(\mathbb{C}, \mathbb{D})$

for each $c_j \in \mathbb{C}$

do $D_j \leftarrow \{d : \langle d, c_j \rangle \in \mathbb{D}\}$

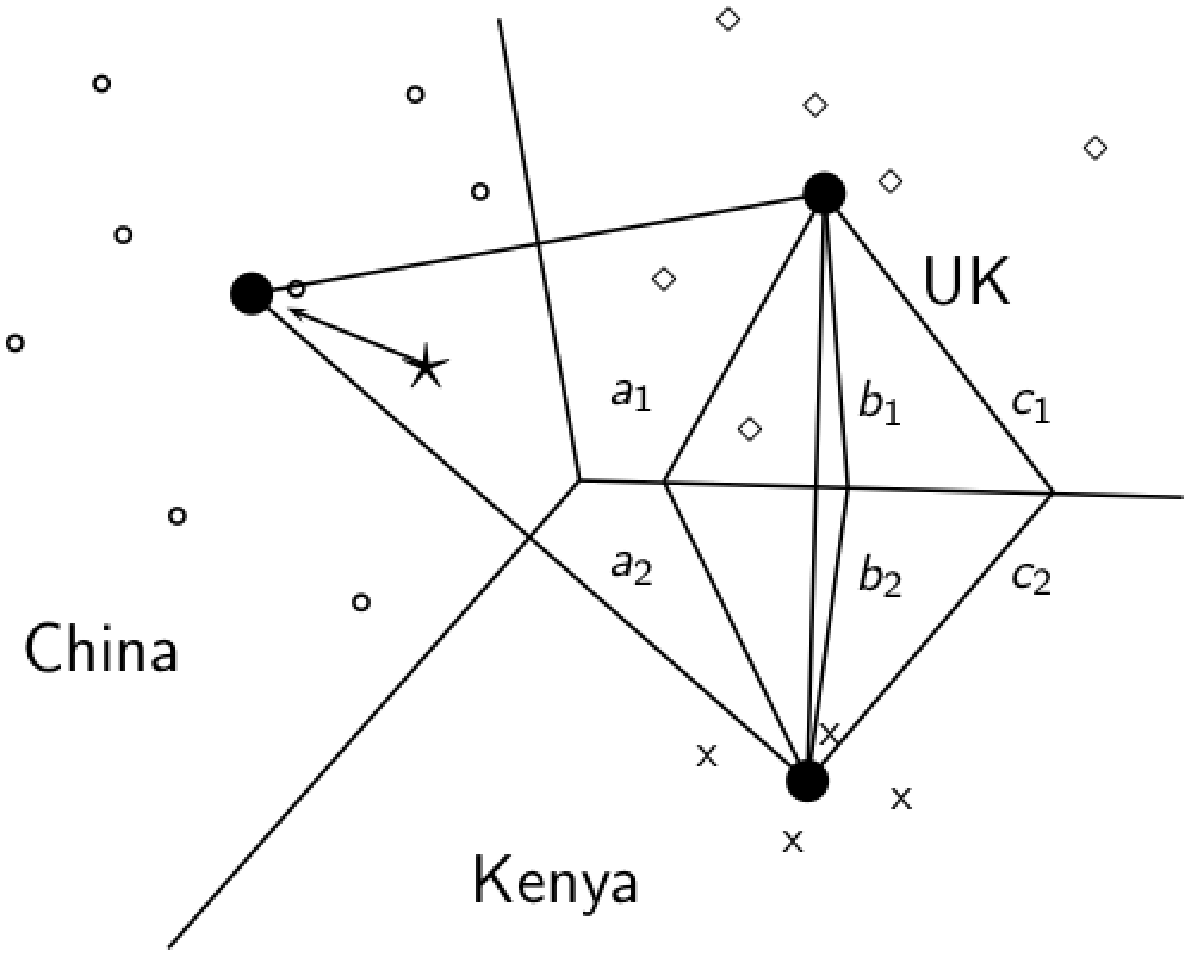
$\quad \vec{\mu}_j \leftarrow \frac{1}{|D_j|} \sum_{d \in D_j} \vec{v}(d)$

return $\{\vec{\mu}_1, \ldots, \vec{\mu}_J\}$

$\textsc{ApplyRocchio}(\{\vec{\mu}_1, \ldots, \vec{\mu}_J\}, d)$

return $\arg\min_j |\vec{\mu}_j - \vec{v}(d)|$

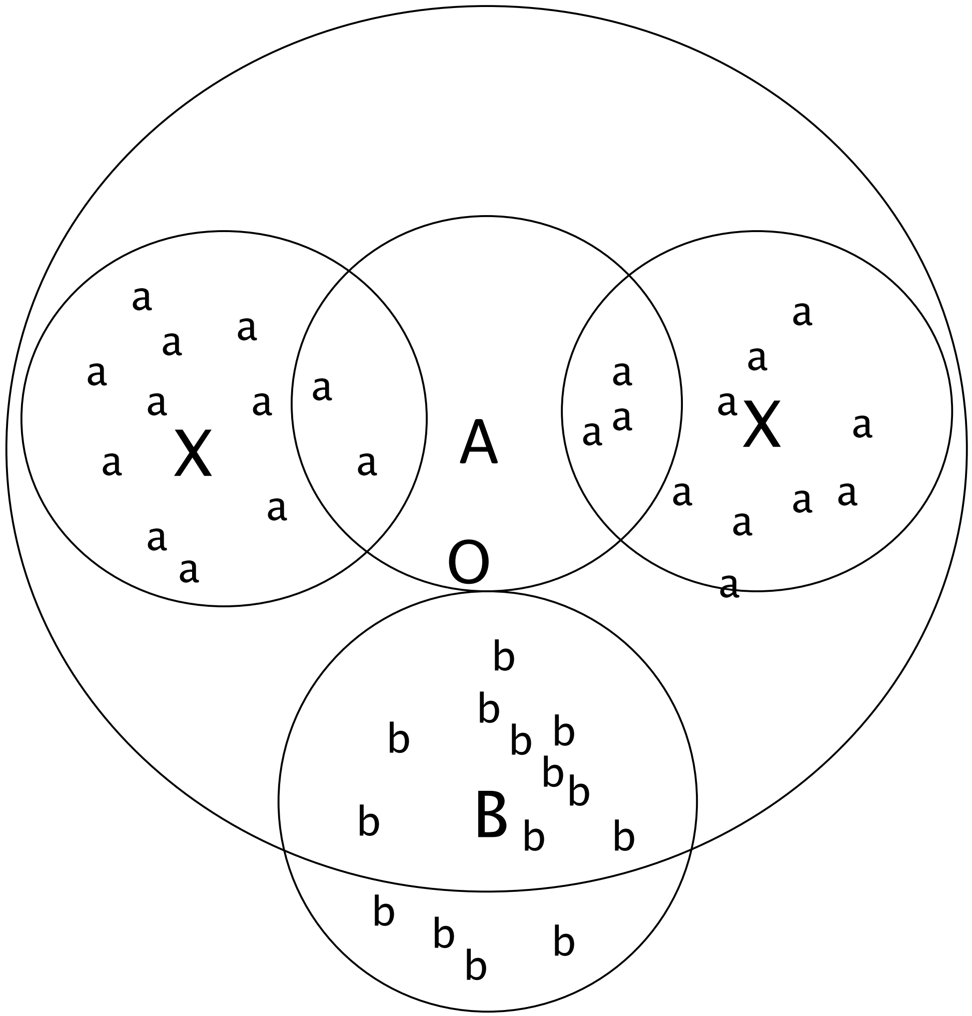
Rocchio illustrated : a1 = a2, b1 = b2, c1 = c2



Rocchio properties

- Rocchio forms a simple representation for each class: the centroid
  - We can interpret the centroid as the prototype of the class.
- Classification is based on similarity to / distance from centroid/prototype.
- Does not guarantee that classifications are consistent with the training data!

Rocchio cannot handle nonconvex, multimodal classes



Exercise: Why is Rocchio not expected to do well for the classification task a vs. b here?

- A is centroid of the a's, B is centroid of the b's.
- The point o is closer to A than to B.
- But o is a better fit for the b class.
- A is a multimodal class with two prototypes.
- But in Rocchio we only have one prototype.

kNN (k Nearest Neighbors)

kNN classification

- kNN classification is another vector space classification method.
- It also is very simple and easy to implement.
- kNN is more accurate (in most cases) than Naive Bayes and Rocchio.
- 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.

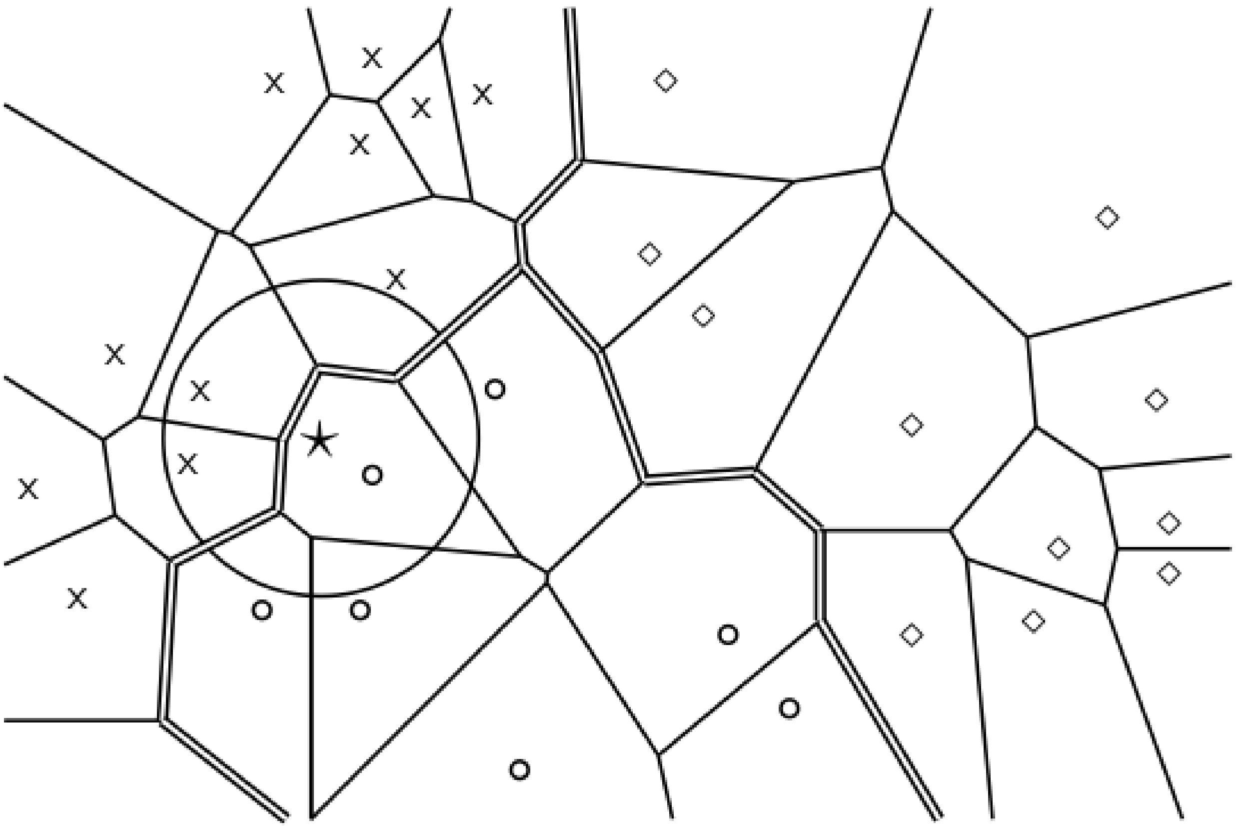
kNN classification

- kNN = *k* nearest neighbors
- kNN classification rule for *k* = 1 (1NN): Assign each test document to the class of its nearest neighbor in the training set.
- 1NN is not very robust – one document can be mislabeled or atypical.
- kNN classification rule for *k* > 1 (kNN): Assign each test document to the majority class of its *k* nearest neighbors in the training set.
- Rationale of kNN: contiguity hypothesis
  - We expect a test document d to have the same label as the training documents located in the local region surrounding *d*.

Probabilistic kNN

- Probabilistic version of kNN: $P(c|d)$ = fraction of k neighbors of d that are in c
- kNN classification rule for probabilistic kNN: Assign d to class c with highest $P(c|d)$

Probabilistic kNN



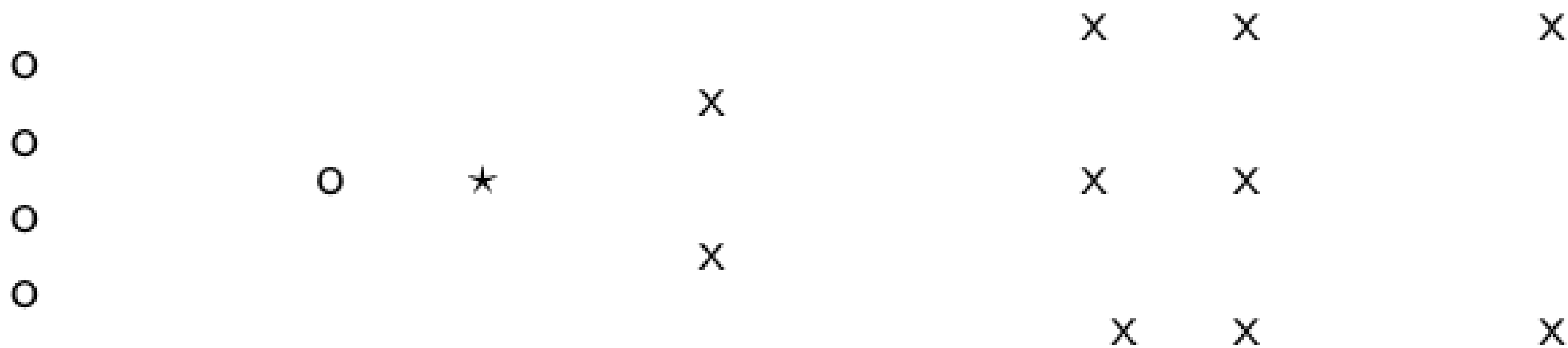
1NN, 3NN
classification
decision
for star?

kNN algorithm

$\textsc{Train-kNN}(\mathbb{C}, \mathbb{D})$
1 $\mathbb{D}' \leftarrow \textsc{Preprocess}(\mathbb{D})$
2 $k \leftarrow \textsc{Select-k}(\mathbb{C}, \mathbb{D}')$
3 **return** $\mathbb{D}', k$

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

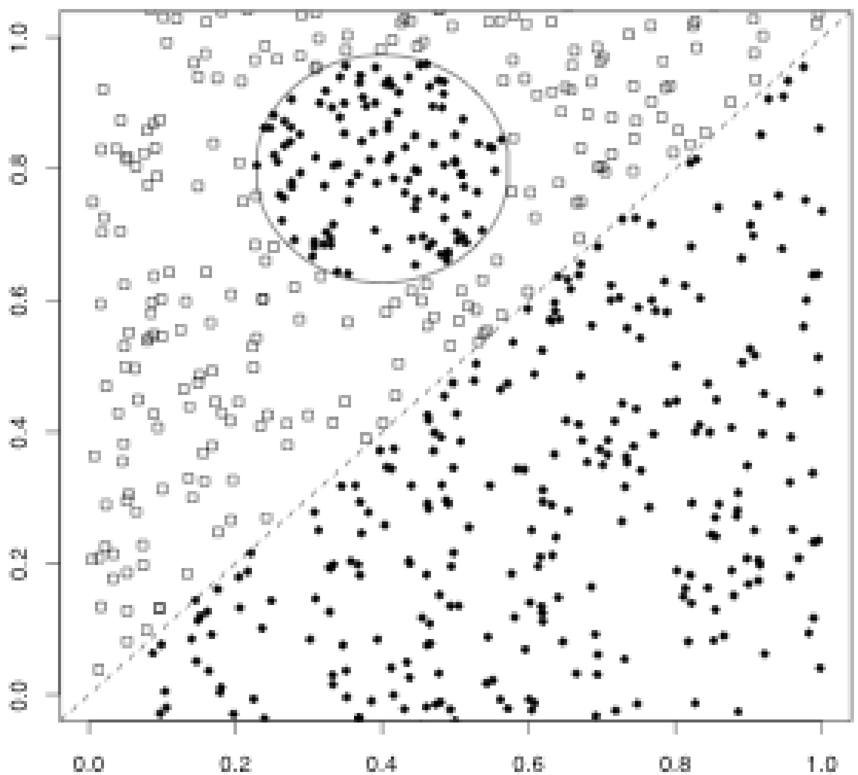
Exercise



How is star classified by:

(i) 1-NN (ii) 3-NN (iii) 9-NN (iv) 15-NN (v) Rocchio?

A nonlinear problem (Rocchio vs kNN)



- Linear classifier like Rocchio does badly on this task.
- kNN will do well (assuming enough training data)

kNN: Discussion

- 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.

- But kNN can be very inaccurate if training set is small.

Naive Bayes Classifier

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:

Prior

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

Posterior

- $P(c)$ is the prior probability of 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.
- If a document's terms do not provide clear evidence for one class vs. another, we choose the c with highest $P(c)$ probability.

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}:

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

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:

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

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:

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

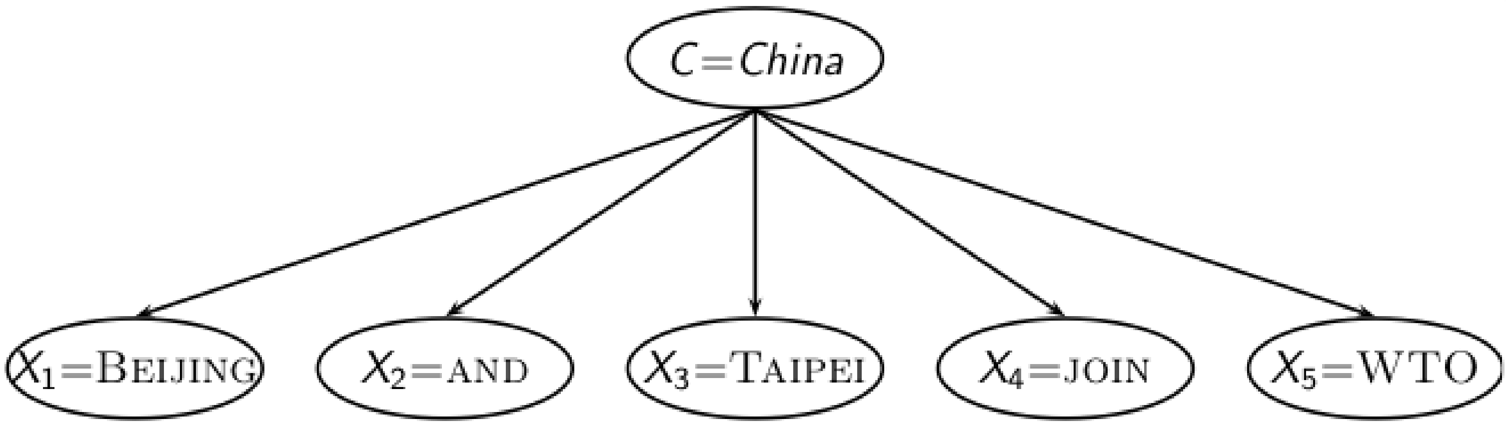
- N_c : number of docs in class *c*; *N*: total number of docs
- Conditional probabilities:

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

- 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:

$$\hat{P}(t_{k_1}|c) = \hat{P}(t_{k_2}|c)$$

The problem with maximum likelihood estimates: Zeros



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

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

The problem with maximum likelihood estimates: Zeros (cont)

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

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

- → We will get P(China|d) = 0 for any document that contains WTO!
- Zero probabilities cannot be conditioned away.

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 different words (in this case the size of the vocabulary: $|V| = M$)

To avoid zeros: Add-one smoothing

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

Naive Bayes: Training

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

$V \leftarrow \textsc{ExtractVocabulary}(\mathbb{D})$
$N \leftarrow \textsc{CountDocs}(\mathbb{D})$
for each $c \in \mathbb{C}$
do $N_c \leftarrow \textsc{CountDocsInClass}(\mathbb{D}, c)$
    $prior[c] \leftarrow N_c / N$
    $text_c \leftarrow \textsc{ConcatenateTextOfAllDocsInClass}(\mathbb{D}, c)$
    for each $t \in V$
    do $T_{ct} \leftarrow \textsc{CountTokensOfTerm}(text_c, t)$
    for each $t \in V$
    do $condprob[t][c] \leftarrow \frac{T_{ct}+1}{\sum_{t'}(T_{ct'}+1)}$
return $V, prior, condprob$

Naive Bayes: Testing

```
ApplyMultinomialNB(ℂ, V, prior, condprob, d)
W ← ExtractTokensFromDoc(V, d)
for each c ∈ ℂ
do score[c] ← log prior[c]
    for each t ∈ W
    do score[c] += log condprob[t][c]
return arg max_{c∈ℂ} score[c]
```

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	Japan Chinese Chinese Chinese Tokyo	?

- What do we need?
  - Class priors: P(c), P(not c)
  - Conditional probabilities: P(t|c), P(t|not c)

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	Japan Chinese Chinese Chinese Tokyo	?

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

$$\hat{P}(c) = \boxed{} \text{ and } \hat{P}(\overline{c}) = \boxed{}$$

$$\begin{aligned}
\hat{P}(\text{Chinese}|c) &= \\
\hat{P}(\text{Tokyo}|c) = \hat{P}(\text{Japan}|c) &= \\
\hat{P}(\text{Chinese}|\overline{c}) &= \\
\hat{P}(\text{Tokyo}|\overline{c}) = \hat{P}(\text{Japan}|\overline{c}) &=
\end{aligned}$$

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	Japan Chinese Chinese Chinese Tokyo	?

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

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

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

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	Japan Chinese Chinese Chinese Tokyo	?

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

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

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

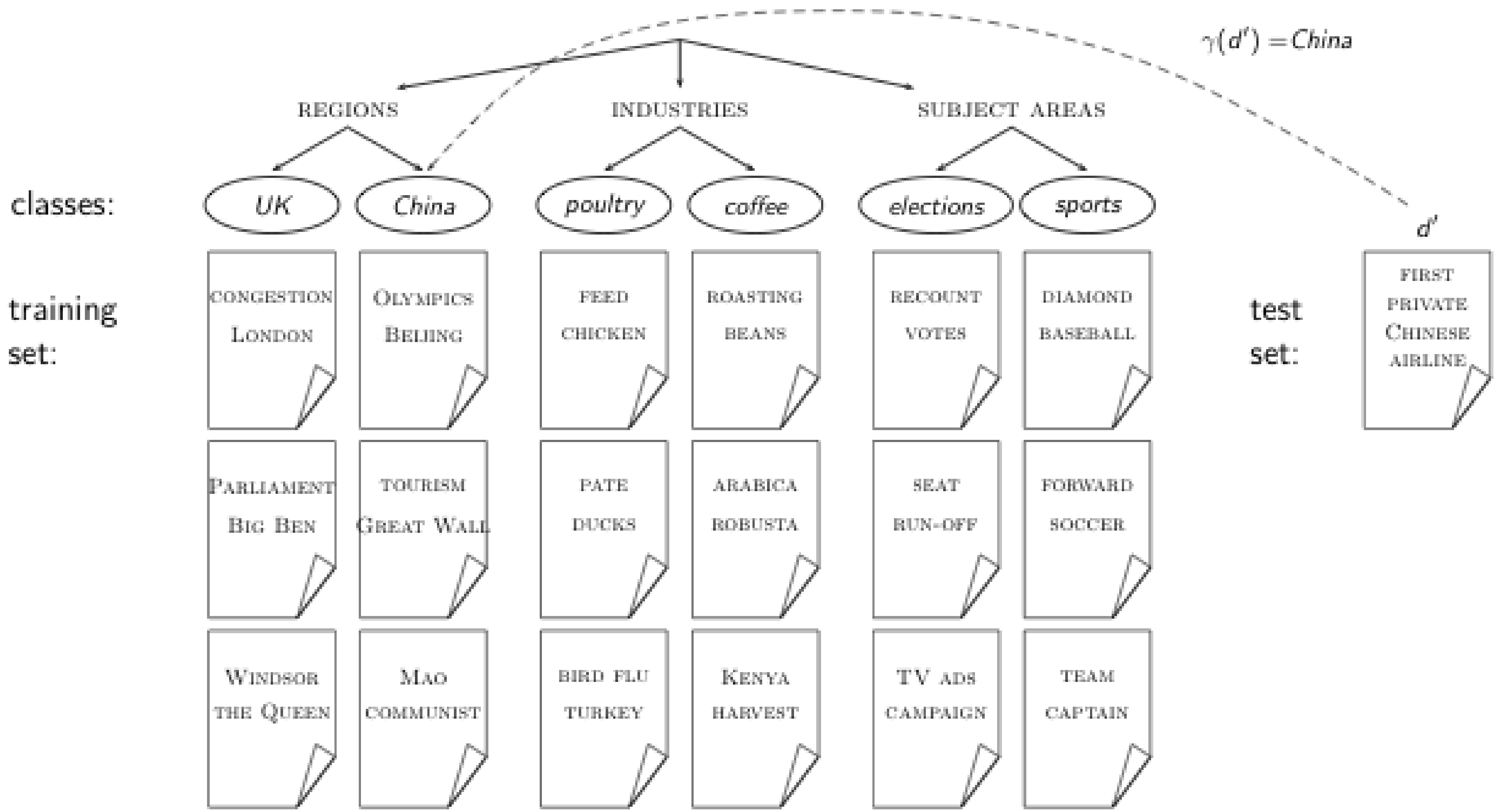
Thus, the classifier assigns the test document to c = *China*.

Example for Rocchio Classification

	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	Japan Chinese Chinese Chinese Tokyo	?

Evaluating a Classifier

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 Enoug

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)

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

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

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

Averaging: Micro vs. Macro

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

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

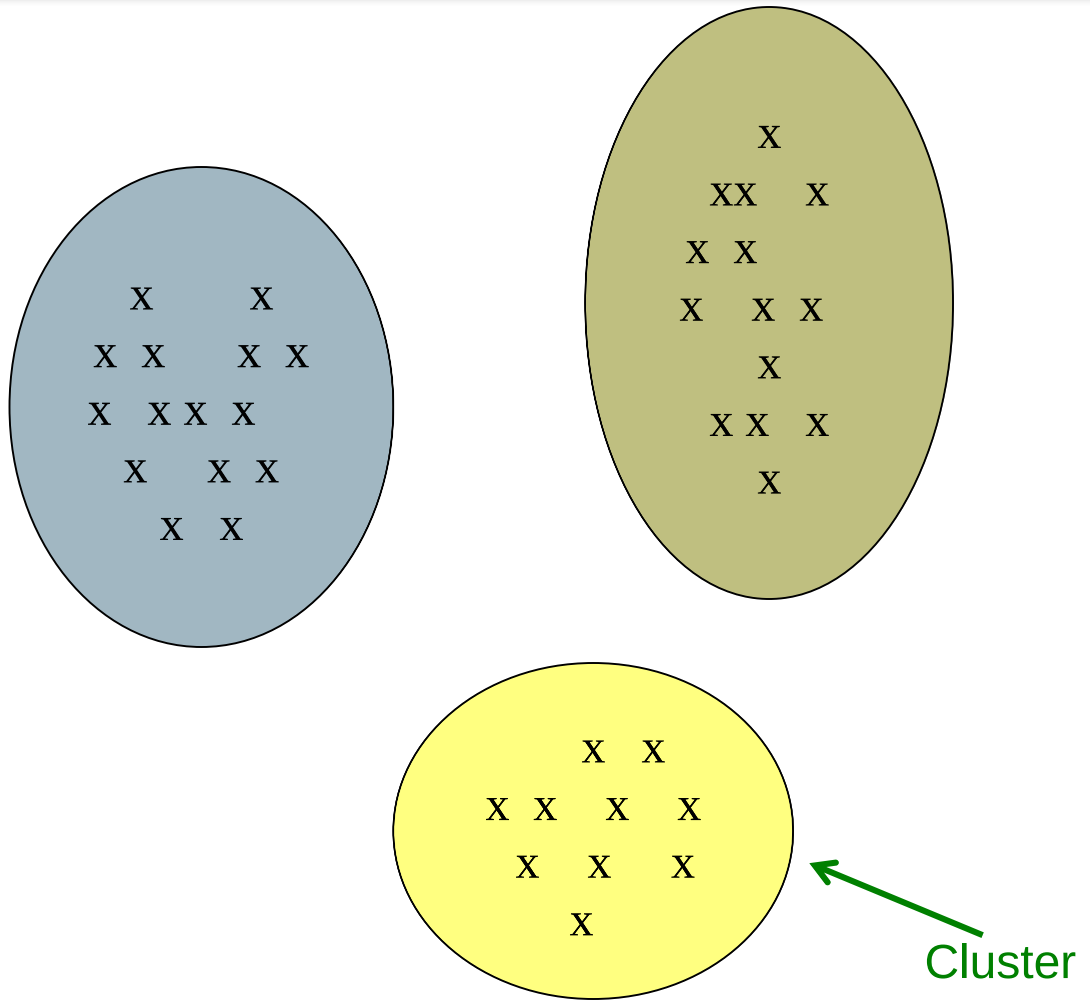
Evaluation measure: F_1 Naive Bayes does pretty well, but some methods beat it consistently (e.g., SVM).

Clustering

What is Clustering?

- **Clustering** is the process of grouping a set of documents into clusters of similar documents.
  - Documents within a cluster should be similar.
  - Documents from different clusters should be dissimilar.
- Clustering is the most common form of *unsupervised learning.*
  - Unsupervised learning = learning from raw data, as opposed to supervised data where a classification of examples is given
- A common and important task that finds many applications in IR and other places

Example Clusters



Clustering is Hard!

Why is it hard?

- Clustering in two dimensions looks easy
- Clustering small amounts of data looks easy
- And in most cases, looks are not deceiving

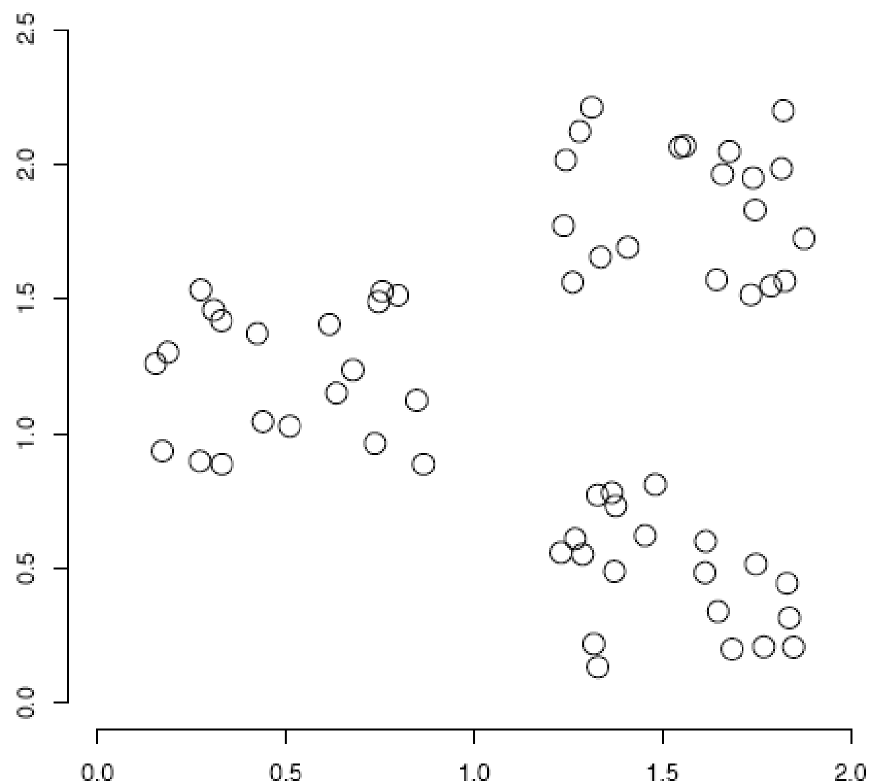
- Many applications involve not 2, but 10 or 10,000 dimensions
- **High-dimensional spaces look different:** Almost all pairs of points are at about the same distance

- Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

Clustering: Application Examples

- Biology: taxonomy of living things: kingdom, phylum, class, order, family, genus and species
- Information retrieval: document clustering
- Land use: Identification of areas of similar land use in an earth observation database
- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults
- Climate: understanding earth climate, find patterns of atmospheric and ocean
- Economic Science: market research

Data set with clear cluster structure



How would you design an algorithm for finding these three clusters?

Example: Clustering Songs

- **Intuitively**: Music divides into categories, and customers prefer a few categories
  - But what are categories really?
- Represent a song by a set of customers who downloaded it
- Similar songs have similar sets of downloaders, and vice-versa

- **Goal: Find clusters of similar songs**

Challenge

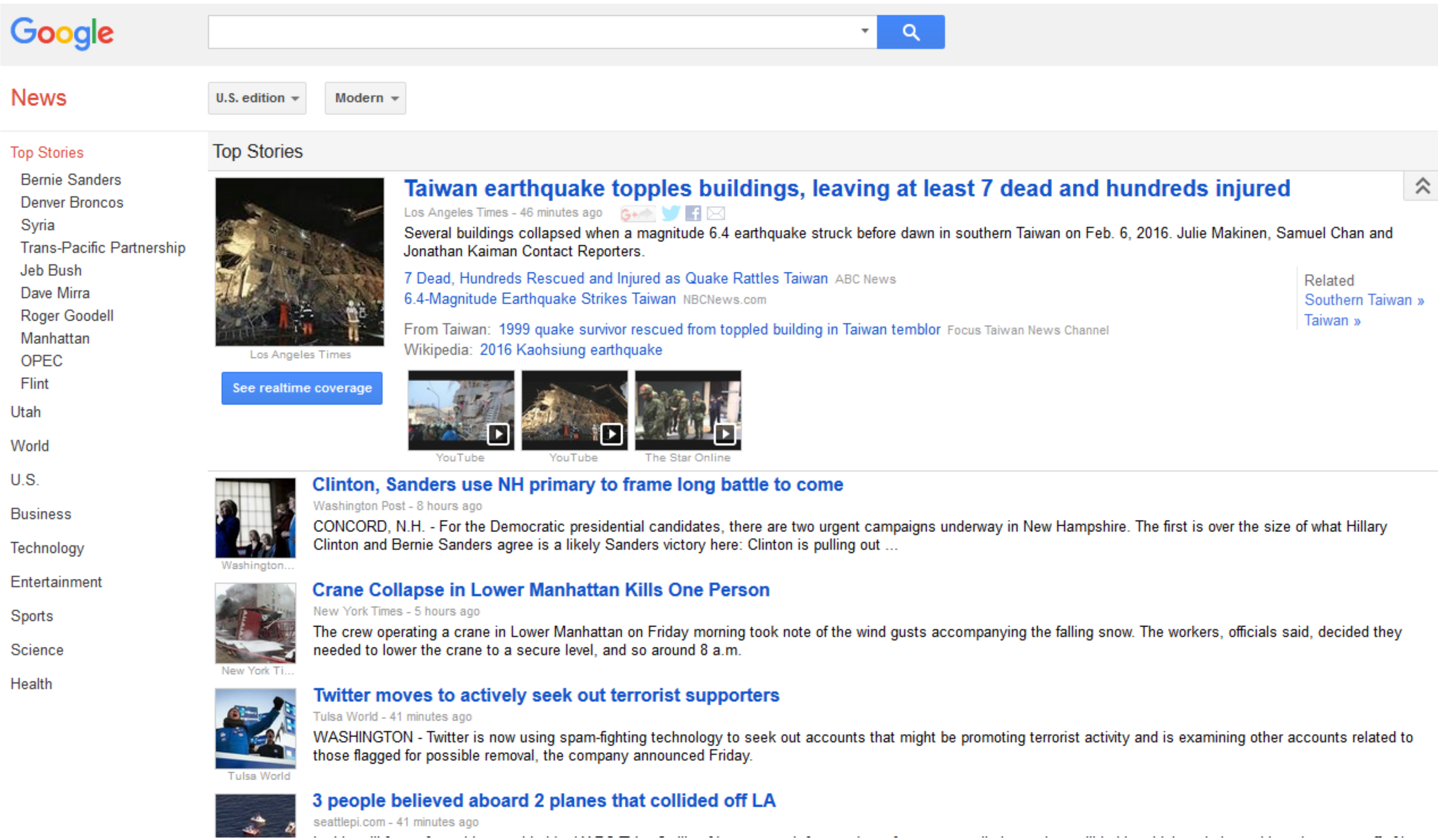
- To cluster songs:
  - How do we define the problem?
  - How do we tackle it?
- Hint: Represent a song by a set of customers who downloaded it

Clustering in IR

Applications of clustering in IR

- **Whole corpus analysis/navigation**
  - **Better user interface: search without typing**
- For improving recall in search applications
  - Better search results
- For better navigation of search results
  - Effective "user recall" will be higher
- For speeding up vector space retrieval
  - Cluster-based retrieval gives faster search

Google News: automatic clustering gives an effective news presentation metaphor



Applications of clustering in IR

- Whole corpus analysis/navigation
  - Better user interface: search without typing
- **For improving recall in search applications**
  - **Better search results**
- For better navigation of search results
  - Effective "user recall" will be higher
- For speeding up vector space retrieval
  - Cluster-based retrieval gives faster search

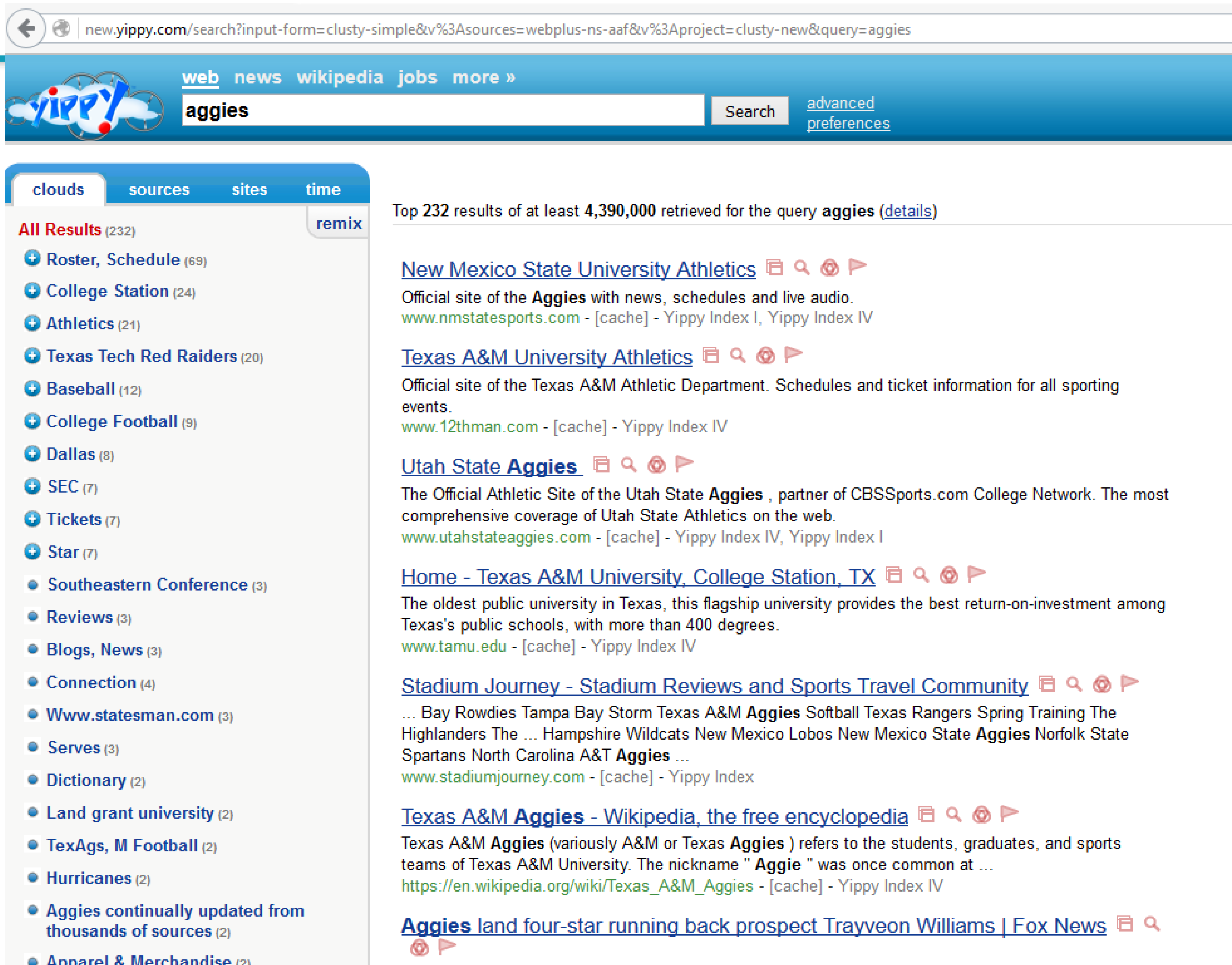
For improving search recall

- *Cluster hypothesis* - Documents in the same cluster behave similarly with respect to relevance to information needs
- Therefore, to improve search recall:
  - Cluster docs in corpus a priori
  - When a query matches a doc *D*, also return other docs in the cluster containing *D*
- Hope if we do this: The query "car" will also return docs containing *automobile*
  - Because clustering grouped together docs containing *car* with those containing *automobile.*

Applications of clustering in IR

- Whole corpus analysis/navigation
  - Better user interface: search without typing
- For improving recall in search applications
  - Better search results
- **For better navigation of search results**
  - **Effective "user recall" will be higher**
- For speeding up vector space retrieval
  - Cluster-based retrieval gives faster search

Yippy.com



new.yippy.com/search?input-form=clusty-simple&v%3Asources=webplus-ns-aaf&v%3Aproject=clusty-new&query=aggies
web news wikipedia jobs more »
aggies
Search
advanced
preferences
clouds sources sites time
remix
All Results (232)
Roster, Schedule (69)
College Station (24)
Athletics (21)
Texas Tech Red Raiders (20)
Baseball (12)
College Football (9)
Dallas (8)
SEC (7)
Tickets (7)
Star (7)
Southeastern Conference (3)
Reviews (3)
Blogs, News (3)
Connection (4)
Www.statesman.com (3)
Serves (3)
Dictionary (2)
Land grant university (2)
TexAgs, M Football (2)
Hurricanes (2)
Aggies continually updated from thousands of sources (2)
Apparel & Merchandise (2)
Top 232 results of at least 4,390,000 retrieved for the query aggies (details)
New Mexico State University Athletics
Official site of the Aggies with news, schedules and live audio.
www.nmstatesports.com - [cache] - Yippy Index I, Yippy Index IV
Texas A&M University Athletics
Official site of the Texas A&M Athletic Department. Schedules and ticket information for all sporting events.
www.12thman.com - [cache] - Yippy Index IV
Utah State Aggies
The Official Athletic Site of the Utah State Aggies , partner of CBSSports.com College Network. The most comprehensive coverage of Utah State Athletics on the web.
www.utahstateaggies.com - [cache] - Yippy Index IV, Yippy Index I
Home - Texas A&M University, College Station, TX
The oldest public university in Texas, this flagship university provides the best return-on-investment among Texas's public schools, with more than 400 degrees.
www.tamu.edu - [cache] - Yippy Index IV
Stadium Journey - Stadium Reviews and Sports Travel Community
... Bay Rowdies Tampa Bay Storm Texas A&M Aggies Softball Texas Rangers Spring Training The Highlanders The ... Hampshire Wildcats New Mexico Lobos New Mexico State Aggies Norfolk State Spartans North Carolina A&T Aggies ...
www.stadiumjourney.com - [cache] - Yippy Index
Texas A&M Aggies - Wikipedia, the free encyclopedia
Texas A&M Aggies (variously A&M or Texas Aggies) refers to the students, graduates, and sports teams of Texas A&M University. The nickname " Aggie " was once common at ...
https://en.wikipedia.org/wiki/Texas_A&M_Aggies - [cache] - Yippy Index IV
Aggies land four-star running back prospect Trayveon Williams | Fox News

Applications of clustering in IR

- Whole corpus analysis/navigation
  - Better user interface: search without typing
- For improving recall in search applications
  - Better search results
- For better navigation of search results
  - Effective "user recall" will be higher
- **For speeding up vector space retrieval**
  - **Cluster-based retrieval gives faster search**

Issues for clustering

- Representation for clustering
  - Document representation
    - Vector space? Normalization?
  - Need a notion of similarity/distance
- How many clusters?
  - Fixed a priori?
  - Completely data driven?
    - Avoid “trivial” clusters - too large or small
      - If a cluster's too large, then for navigation purposes you've wasted an extra user click without whittling down the set of documents much.

Flat vs. Hierarchical Clustering

- Flat algorithms
  - Usually start with a random (partial) partitioning
  - Refine it iteratively
    - Main algorithm: K-means
- Hierarchical algorithms
  - Create a hierarchy
  - Bottom-up, agglomerative
  - Top-down, divisive