ΕΠΛ660

Ανάκτηση Πληροφοριών και Μηχανές Αναζήτησης

- Classification and data clustering

Slides by Manning, Raghavan, Schutze

Categorization/Classification

- Given:
  - A description of an instance, $d \in X$
    - X is the *instance language* or *instance space*.
      - Issue: how to represent text documents.
      - Usually some type of high-dimensional space
  - A fixed set of classes:

    $C = \{c_1, c_2, \ldots, c_J\}$
- Determine:
  - The category of d: $\gamma(d) \in C$, where $\gamma(d)$ is a *classification function* whose domain is X and whose range is C.
    - We want to know how to build classification functions ("classifiers").

Supervised Classification

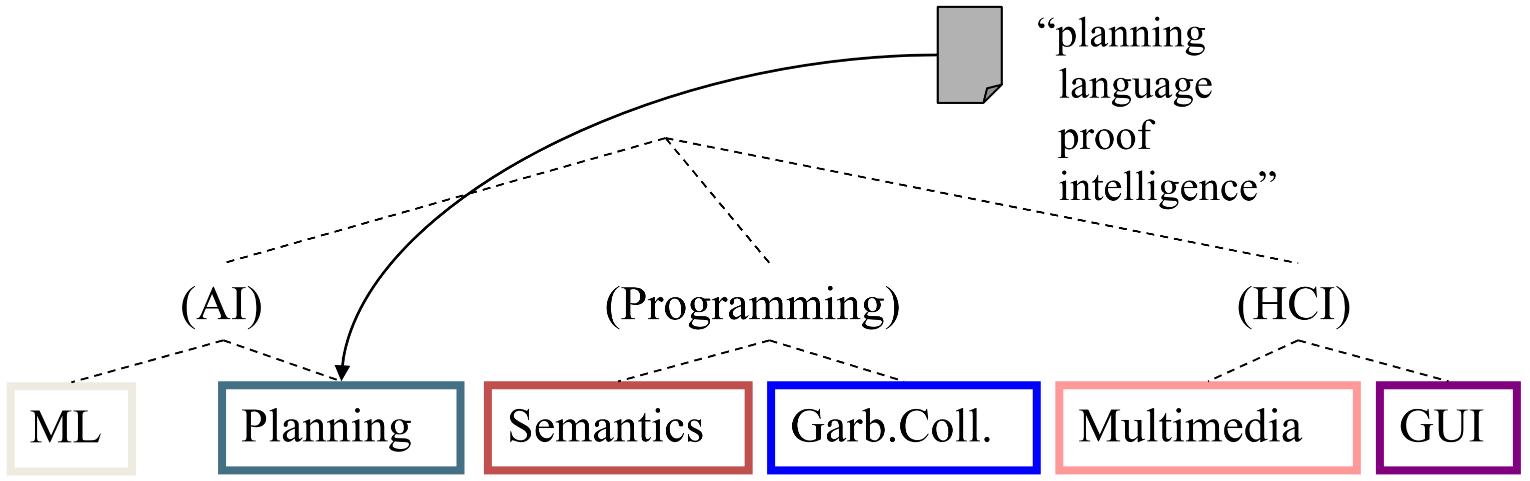
- Given:
  - A description of an instance, $d \in X$
    - X is the *instance language* or *instance space*.
  - A fixed set of classes:

    $C = \{c_1, c_2,\ldots, c_J\}$
  - A training set D of labeled documents with each labeled document $\langle d,c\rangle \in X \times C$
- Determine:
  - A learning method or algorithm which will enable us to learn a classifier $\gamma: X \rightarrow C$
  - For a test document d, we assign it the class $\gamma(d) \in C$

Document Classification

Test Data:

Classes:



Training Data:	learning	planning	programming	garbage	...	...
	intelligence	temporal	semantics	collection		
	algorithm	reasoning	language	memory		
	reinforcement	plan	proof...	optimization		
	network...	language...		region...		

(Note: in real life there is often a hierarchy, not present in the above problem statement; and also, you get papers on ML approaches to Garb. Coll.)

More Text Classification Examples

Many search engine functionalities use classification

- Assigning labels to documents or web-pages:
- Labels are most often topics such as Yahoo-categories
  - *"finance," "sports," "news>world>asia>business"*
- Labels may be genres
  - *"editorials" "movie-reviews" "news”*
- Labels may be opinion on a person/product
  - *“like”, “hate”, “neutral”*
- Labels may be domain-specific
  - *"interesting-to-me" : "not-interesting-to-me”*
  - *“contains adult language” : “doesn’t”*
  - *language identification: English, French, Chinese, …*
  - *search vertical: about Linux versus not*
  - *“link spam” : “not link spam”*

Probabilistic relevance feedback

- Rather than reweighting in a vector space…
- If user has told us some relevant and some irrelevant documents, then we can proceed to build a probabilistic classifier,
    - such as the Naive Bayes model we will look at today:
  - $P(t_k|R) = |\mathbf{D}_{rk}| / |\mathbf{D}_r|$
  - $P(t_k|NR) = |\mathbf{D}_{nrk}| / |\mathbf{D}_{nr}|$
    - t_k is a term; $\mathbf{D}_r$ is the set of known relevant documents; $\mathbf{D}_{rk}$ is the subset that contain t_k; $\mathbf{D}_{nr}$ is the set of known irrelevant documents; $\mathbf{D}_{nrk}$ is the subset that contain t_k.

Recall a few probability basics

- For events *a* and *b:*
- Bayes' Rule

$$p(a,b) = p(a \cap b) = p(a \mid b)\, p(b) = p(b \mid a)\, p(a)$$

$$p(\bar{a} \mid b)\, p(b) = p(b \mid \bar{a})\, p(\bar{a})$$

$$\underset{\text{Posterior}}{p(a \mid b)} = \frac{p(b \mid a)\, p(a)}{p(b)} = \frac{p(b \mid a)\, \overset{\text{Prior}}{p(a)}}{\sum_{x=a,\bar{a}} p(b \mid x)\, p(x)}$$

- Odds:

$$O(a) = \frac{p(a)}{p(\bar{a})} = \frac{p(a)}{1 - p(a)}$$

Bayesian Methods

- Learning and classification methods based on probability theory.
- Bayes theorem plays a critical role in probabilistic learning and classification.
- Builds a *generative model* that approximates how data is produced
- Uses *prior* probability of each category given no information about an item
- Categorization produces a *posterior* probability distribution over the possible categories given a description of an item.

Bayes' Rule for text classification

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

$$P(c,d) = P(c \mid d)P(d) = P(d \mid c)P(c)$$

$$\boxed{P(c \mid d) = \frac{P(d \mid c)P(c)}{P(d)}}$$

Naive Bayes Classifiers

Task: Classify a new instance *d* based on a tuple of attribute values into one of the classes $c_j \in C$

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

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

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

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

MAP is "maximum a posteriori" = most likely class

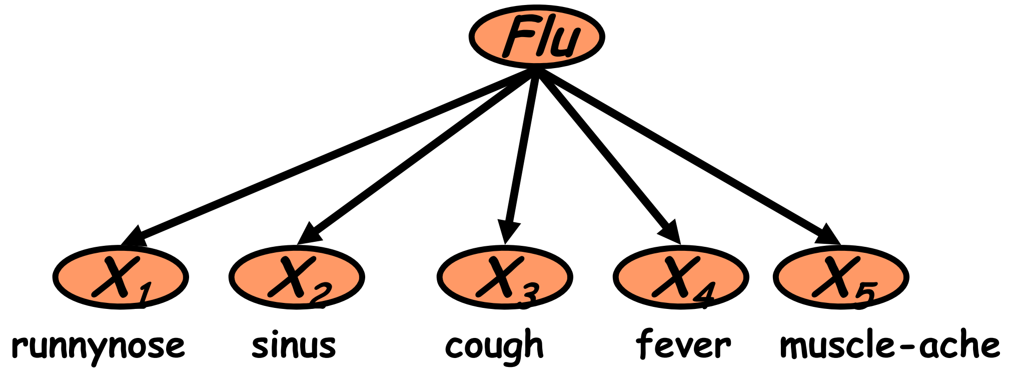
Naïve Bayes Classifier: Naïve Bayes Assumption

- $P(c_j)$
  - Can be estimated from the frequency of classes in the training examples.
- $P(x_1,x_2,\ldots,x_n|c_j)$
  - $O(|X|^n \bullet |C|)$ parameters
  - Could only be estimated if a very, very large number of training examples was available.

Naïve Bayes Conditional Independence Assumption:

- Assume that the probability of observing the conjunction of attributes is equal to the product of the individual probabilities $P(x_i|c_j)$.

The Naïve Bayes Classifier

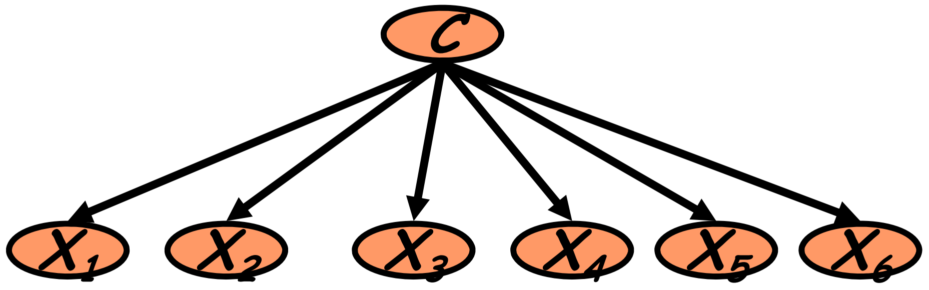


- **Conditional Independence Assumption:** features detect term presence and are independent of each other given the class:

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

- This model is appropriate for binary variables
  - Multivariate Bernoulli model

Learning the Model

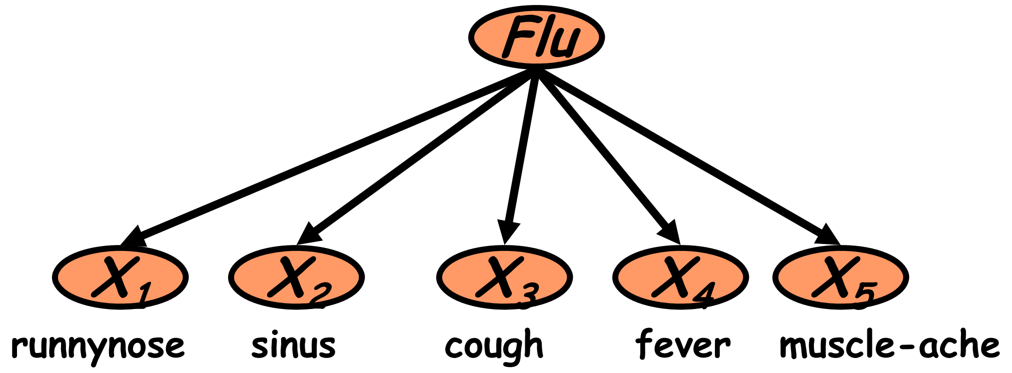


- First attempt: maximum likelihood estimates
  - simply use the frequencies in the data

$$\hat{P}(c_j) = \frac{N(C = c_j)}{N}$$

$$\hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j)}{N(C = c_j)}$$

Problem with Maximum Likelihood



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

- What if we have seen no training documents with the word ***muscle-ache*** and classified in the topic ***Flu***?

$$\hat{P}(X_5 = t \mid C = nf) = \frac{N(X_5 = t, C = nf)}{N(C = nf)} = 0$$

- Zero probabilities cannot be conditioned away, no matter the other evidence!

$$\ell = \arg\max_c \hat{P}(c) \prod_i \hat{P}(x_i \mid c)$$

Smoothing to Avoid Overfitting

$$\hat{P}(x_i \mid c_j) = \frac{N(X_i = x_i, C = c_j) + 1}{N(C = c_j) + k}$$

# of values of X_i

- Somewhat more subtle version

overall fraction in data where $X_i = x_{i,k}$

$$\hat{P}(x_{i,k} \mid c_j) = \frac{N(X_i = x_{i,k}, C = c_j) + m p_{i,k}}{N(C = c_j) + m}$$

extent of "smoothing"

Naive Bayes Classifier

$$c_{NB} = \underset{c_j \in C}{\operatorname{argmax}}[\log P(c_j) + \sum_{i \in positions} \log P(x_i \mid c_j)]$$

- Simple interpretation: Each conditional parameter log $P(x_i|c_j)$ is a weight that indicates how good an indicator x_i is for c_j.
- The prior log $P(c_j)$ is a weight that indicates the relative frequency of c_j.
- The sum 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 for it

Two Naive Bayes Models

- Model 1: Multivariate Bernoulli
  - One feature X_w for each word in dictionary
  - X_w = true in document d if w appears in d
  - Naive Bayes assumption:
    - Given the document's topic, appearance of one word in the document tells us nothing about chances that another word appears
- This is the model used in the binary independence model in classic probabilistic relevance feedback on hand-classified data (Maron in IR was a very early user of NB)

Two Naive Bayes Models

- Model 2: Multinomial = Class conditional unigram
  - One feature X_i for each word pos in document
    - feature's values are all words in dictionary
  - Value of X_i is the word in position i
  - Naïve Bayes assumption:
    - Given the document's topic, word in one position in the document tells us nothing about words in other positions
  - Second assumption:
    - Word appearance does not depend on position

$$P(X_i = w \mid c) = P(X_j = w \mid c)$$

for all positions *i*,*j*, word *w*, and class *c*

    - Just have one multinomial feature predicting all words

Parameter estimation

- Multivariate Bernoulli model:
- $\hat{P}(X_w = t \mid c_j) =$ fraction of documents of topic c_j in which word *w* appears

- Multinomial model:

$\hat{P}(X_i = w \mid c_j) =$ fraction of times in which word *w* appears among all words in documents of topic c_j

  - Can create a mega-document for topic *j* by concatenating all documents in this topic
  - Use frequency of *w* in mega-document

Classification

- Multinomial vs Multivariate Bernoulli?
- Multinomial model is almost always more effective in text applications!
  - See results figures later
- See *IIR* sections 13.2 and 13.3 for worked examples with each model

Exercise

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

- Estimate parameters of Naive Bayes classifier
- Classify test document

Example: Parameter estimates

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

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

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* = *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`.

Feature Selection: Why?

- Text collections have a large number of features
  - 10,000 – 1,000,000 unique words … and more
- May make using a particular classifier feasible
  - Some classifiers can't deal with 100,000 of features
- 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 overfitting

Feature selection: how?

- Two ideas:
  - Hypothesis testing statistics:
    - Are we confident that the value of one categorical variable is associated with the value of another
    - Chi-square test (χ^2)
  - 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)

χ^2 statistic (CHI)

- $\chi 2$ is interested in $(f_o - f_e)^2/f_e$ summed over all table entries: is the observed number what you'd expect given the marginals?

$$\chi^2(j,a) = \sum (O-E)^2/E = (2-.25)^2/.25 + (3-4.75)^2/4.75$$
$$+ (500-502)^2/502 + (9500-9498)^2/9498 = 12.9 \ (p < .001)$$

- The null hypothesis is rejected with confidence .999,
- since 12.9 > 10.83 (the value for .999 confidence).

	Term = jaguar	*Term ≠ jaguar*
Class = auto	2 *(0.25)*	500 *(502)*
Class ≠ auto	3 *(4.75)*	9500 *(9498)*

expected: f_e

observed: f_o

χ^2 statistic (CHI)

There is a simpler formula for 2x2 χ^2:

$$\chi^2(t,c) = \frac{N \times (AD - CB)^2}{(A+C) \times (B+D) \times (A+B) \times (C+D)}$$

$A = \#(t,c)$	$C = \#(\neg t,c)$
$B = \#(t,\neg c)$	$D = \#(\neg t, \neg c)$

$$N = A + B + C + D$$

Feature selection via Mutual Information

- In training set, choose *k* words which best discriminate (give most info on) the categories.
- The Mutual Information between a word, class is:

$$I(w,c) = \sum_{e_w \in \{0,1\}} \sum_{e_c \in \{0,1\}} p(e_w, e_c) \log \frac{p(e_w, e_c)}{p(e_w)p(e_c)}$$

  - For each word *w* and each category *c*

Feature selection via MI (contd.)

- For each category we build a list of *k* most discriminating terms.
- For example (on 20 Newsgroups):
  - ***sci.electronics:*** circuit, voltage, amp, ground, copy, battery, electronics, cooling, …
  - ***rec.autos:*** car, cars, engine, ford, dealer, mustang, oil, collision, autos, tires, toyota, …
- Greedy: does not account for correlations between terms
- Why?

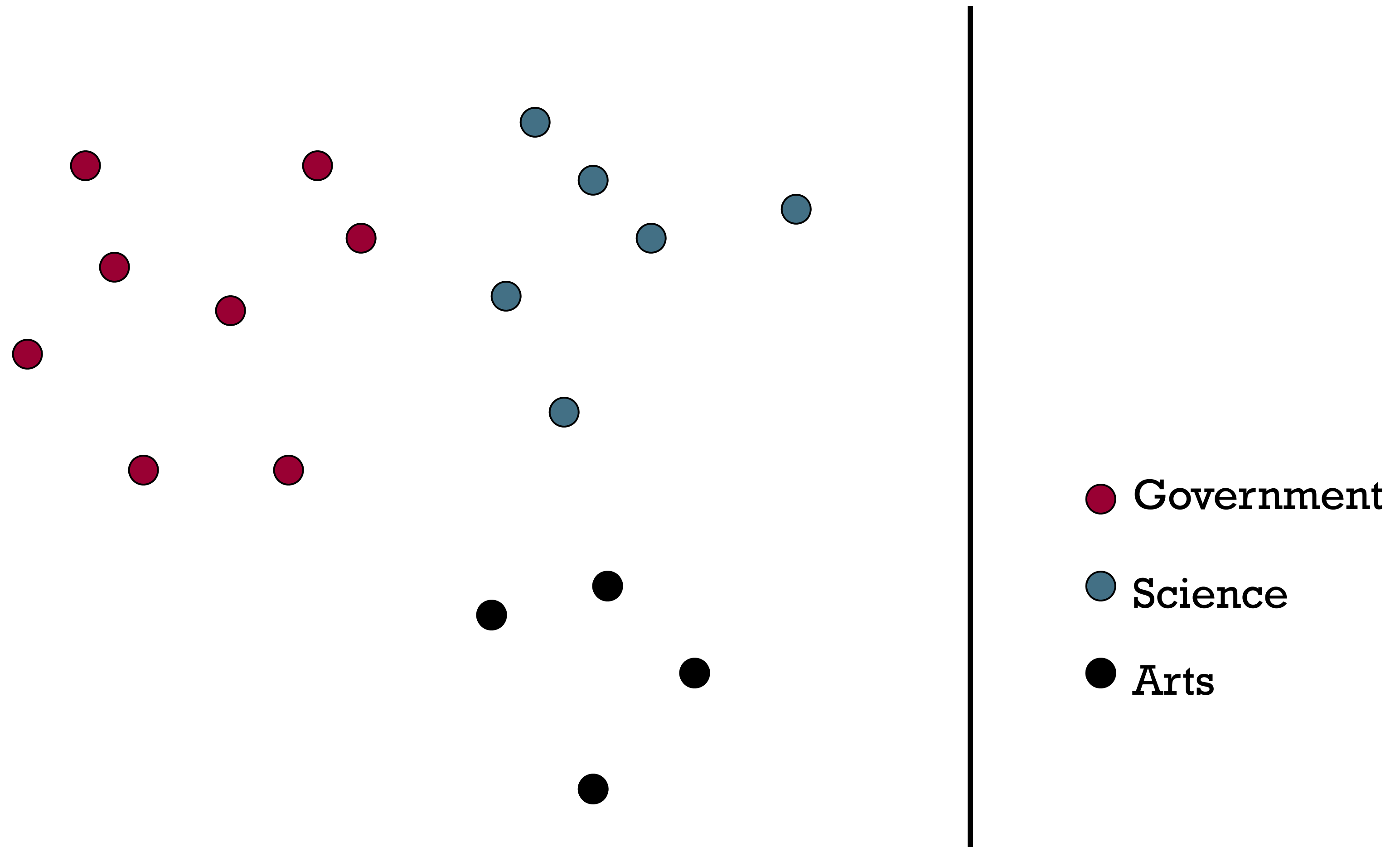
Feature Selection

- Mutual Information
  - Clear information-theoretic interpretation
  - May select rare uninformative terms
- Chi-square
  - Statistical foundation
  - May select very slightly informative frequent terms that are not very useful for classification
- Just use the commonest terms?
  - No particular foundation
  - In practice, this is often 90% as good

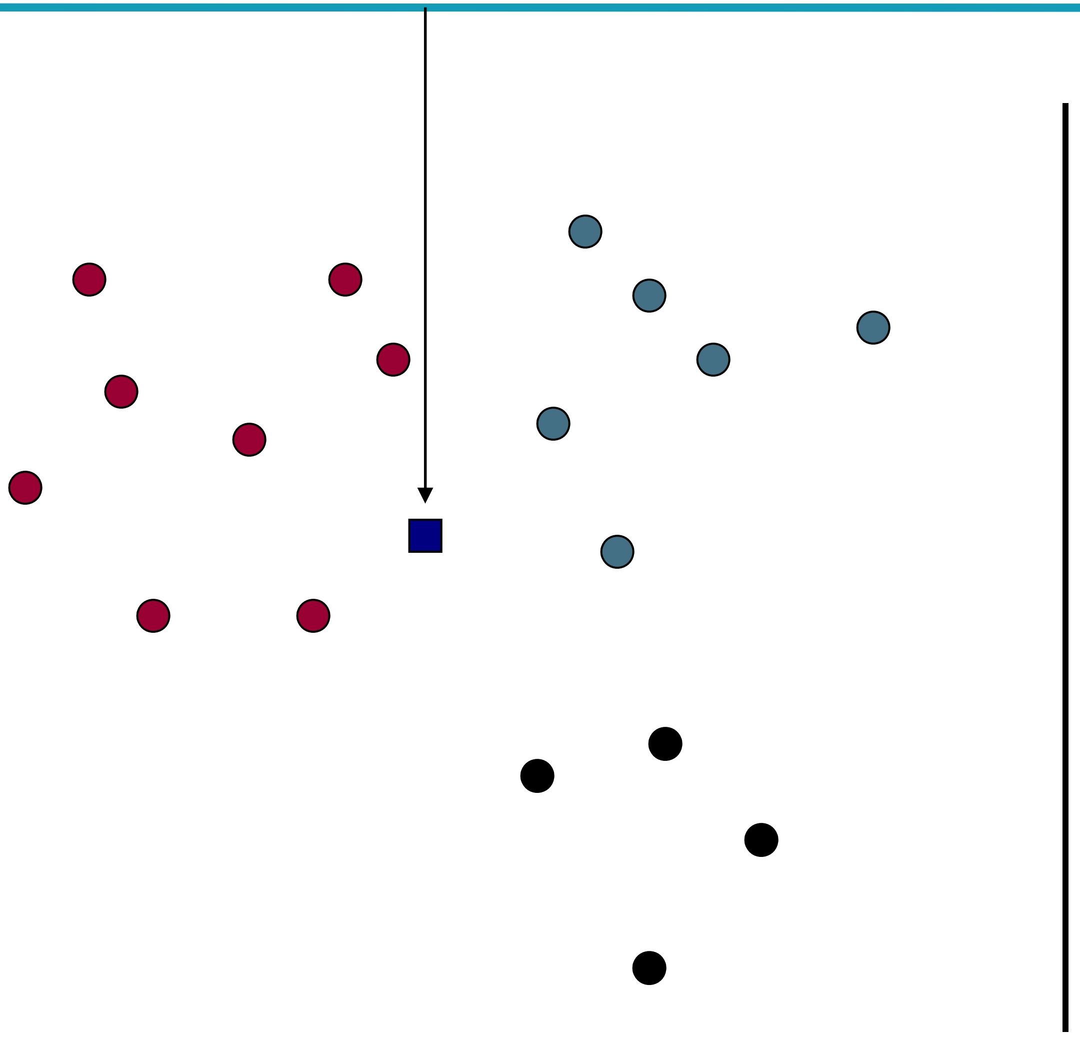
Classification Using Vector Spaces

- 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 of space
- Premise 2: Documents from different classes don't overlap (much)
- We define surfaces to delineate classes in the space

Documents in a Vector Space



Test Document of what class?

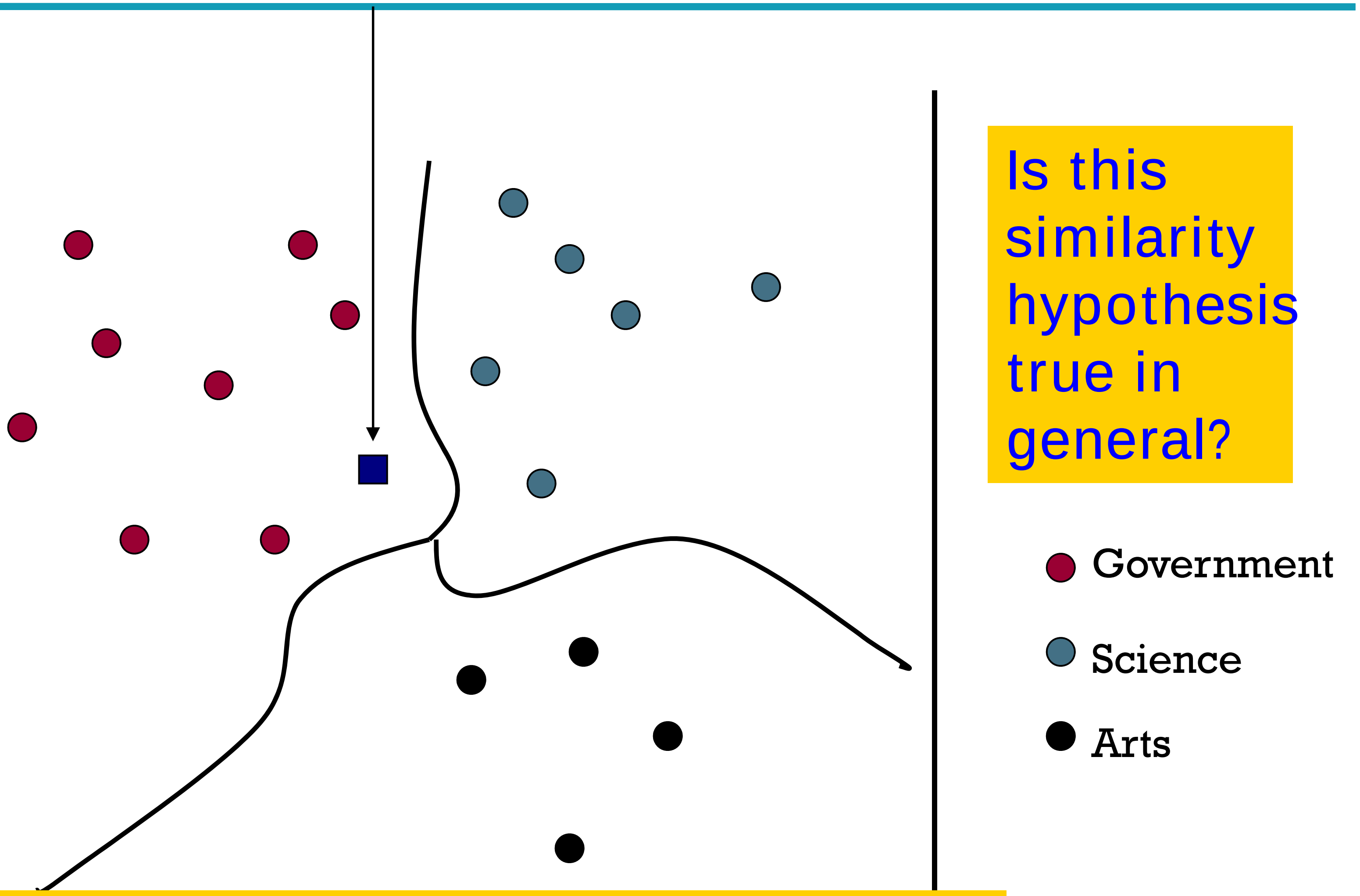


● Government

● Science

● Arts

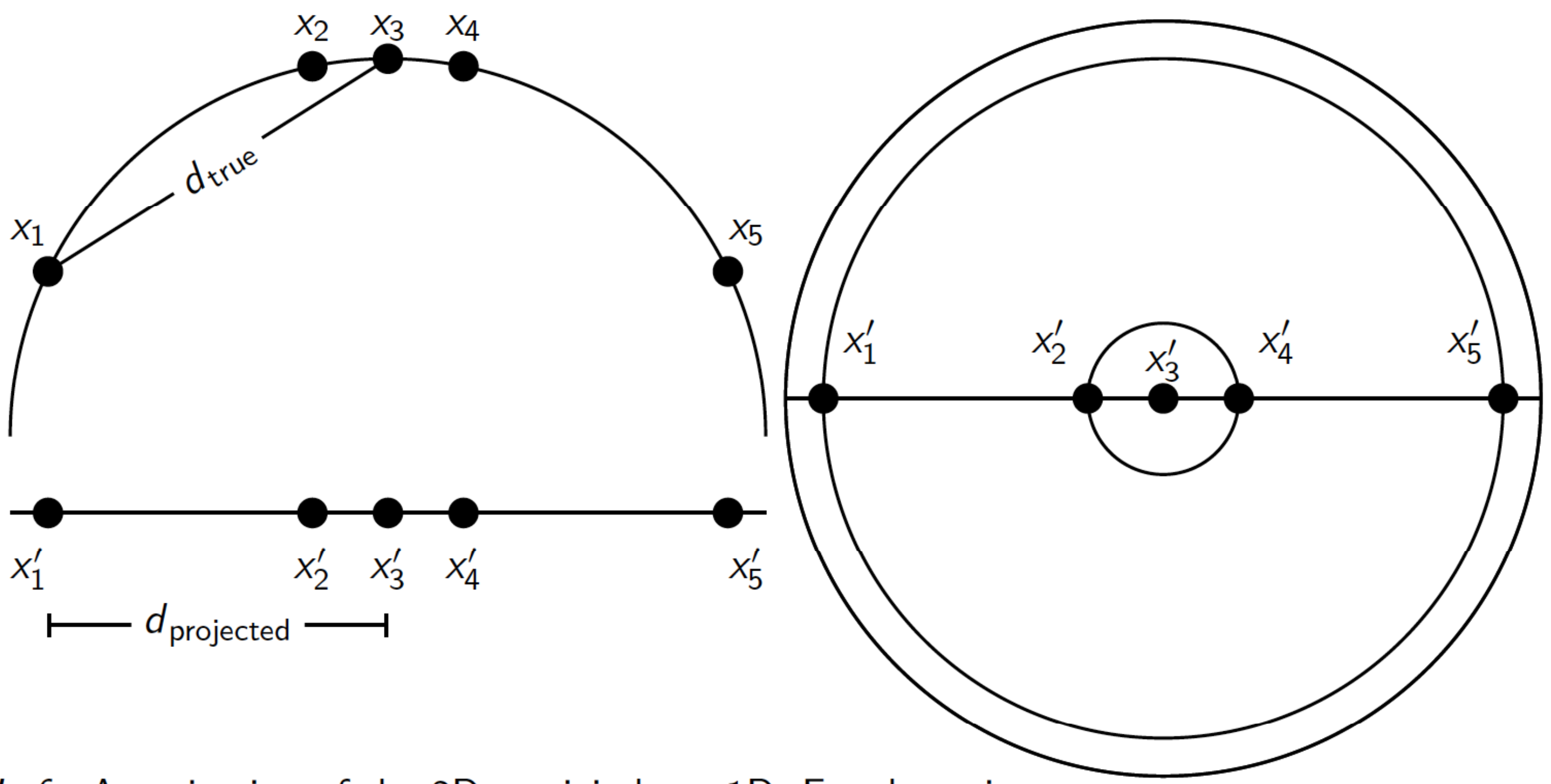
Test Document = Government



Is this similarity hypothesis true in general?

Our main topic today is how to find good separators

Aside: 2D/3D graphs can be misleading


x_2 x_3 x_4
d_{true}
x_1
x_5
x'_1 x'_2 x'_3 x'_4 x'_5
$d_{\text{projected}}$
x'_1 x'_2 x'_3 x'_4 x'_5

Left: A projection of the 2D semicircle to 1D. For the points x_1, x_2, x_3, x_4, x_5 at x coordinates $-0.9, -0.2, 0, 0.2, 0.9$ the distance $|x_2 x_3| \approx 0.201$ only differs by 0.5% from $|x'_2 x'_3| = 0.2$; but $|x_1 x_3|/|x'_1 x'_3| = d_{\text{true}}/d_{\text{projected}} \approx 1.06/0.9 \approx 1.18$ is an example of a large distortion (18%) when projecting a large area. *Right:* The corresponding projection of the 3D hemisphere to 2D.

Using Rocchio for vector space classification

- The principal difference between relevance feedback and text classification:
  - The training set is given as part of the input in text classification.
  - It is interactively created in relevance feedback.

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

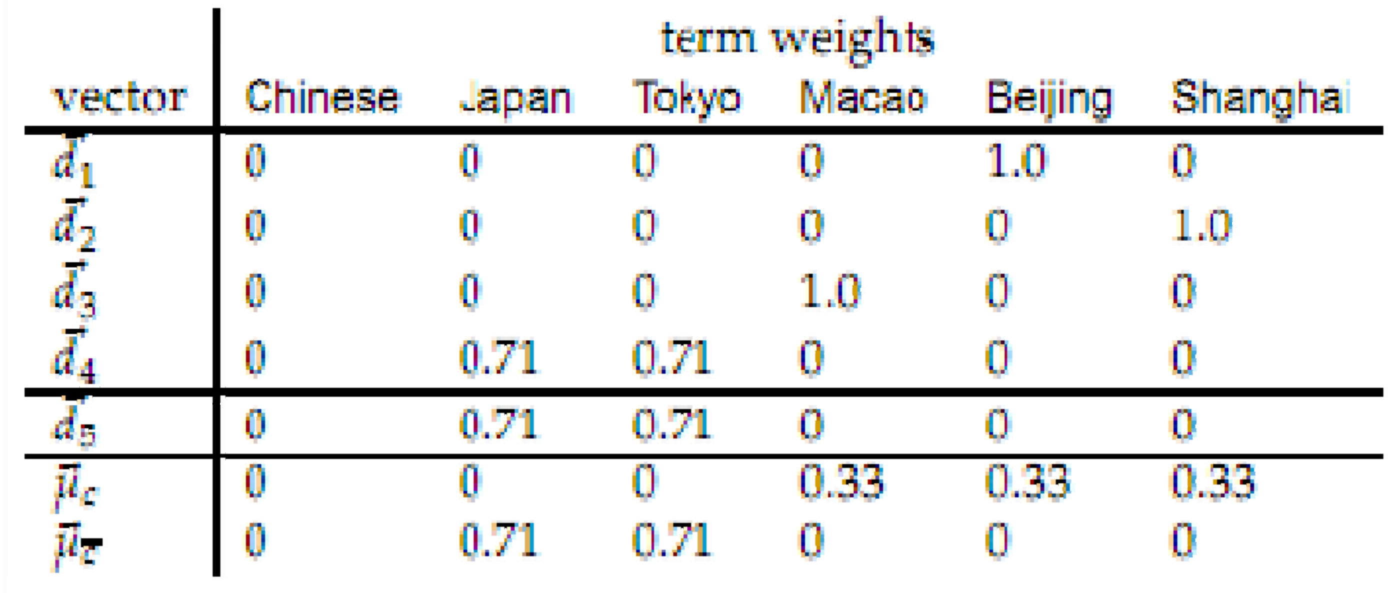
```
TRAINROCCHIO($\mathbb{C}, \mathbb{D}$)
1  for each $c_j \in \mathbb{C}$
2  do $D_j \leftarrow \{d : \langle d, c_j \rangle \in \mathbb{D}\}$
3       $\vec{\mu}_j \leftarrow \frac{1}{|D_j|} \sum_{d \in D_j} \vec{v}(d)$
4  return $\{\vec{\mu}_1, \ldots, \vec{\mu}_J\}$

APPLYROCCHIO($\{\vec{\mu}_1, \ldots, \vec{\mu}_J\}, d$)
1  return $\arg\min_j |\vec{\mu}_j - \vec{v}(d)|$
```

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 Classification: Example

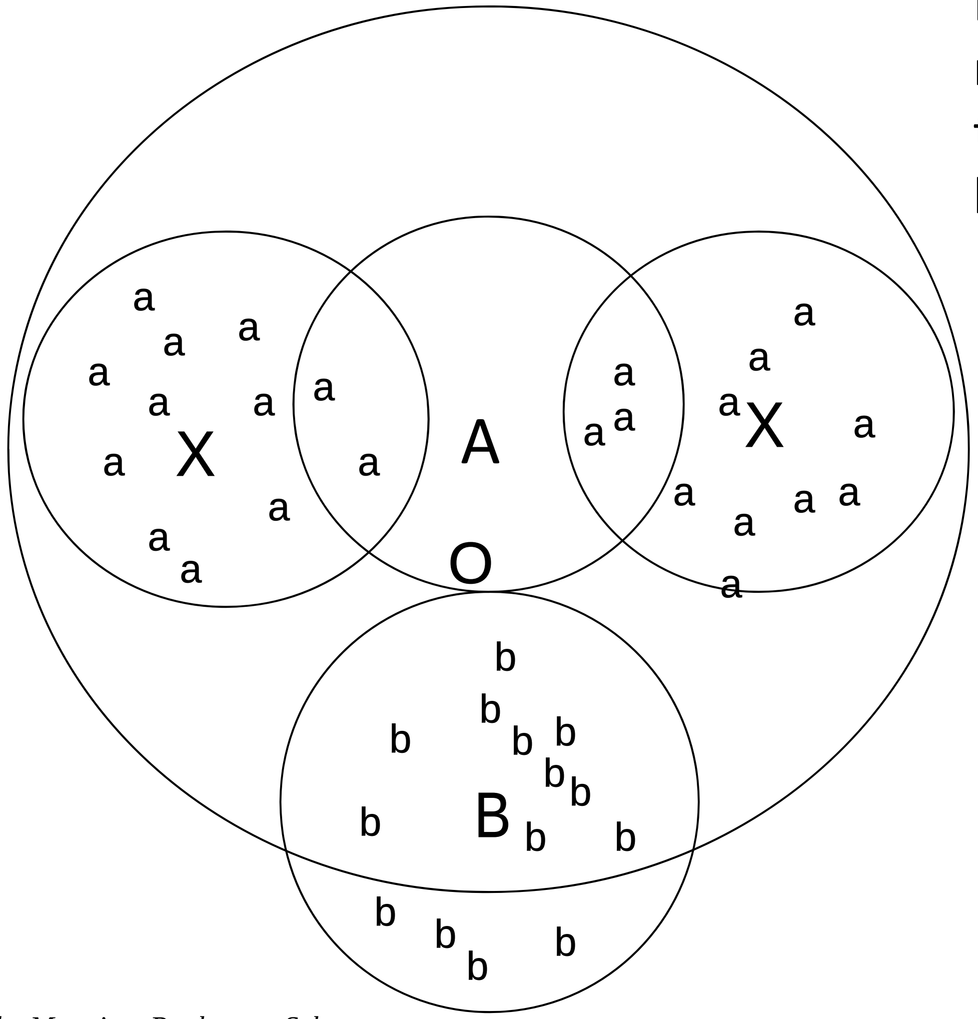


	term weights					
vector	Chinese	Japan	Tokyo	Macao	Beijing	Shanghai
$\vec{d}_1$	0	0	0	0	1.0	0
$\vec{d}_2$	0	0	0	0	0	1.0
$\vec{d}_3$	0	0	0	1.0	0	0
$\vec{d}_4$	0	0.71	0.71	0	0	0
$\vec{d}_5$	0	0.71	0.71	0	0	0
$\vec{\mu}_c$	0	0	0	0.33	0.33	0.33
$\vec{\mu}_{\overline{c}}$	0	0.71	0.71	0	0	0

- The separating hyperplane in this case has the following parameters:

$$\begin{aligned} \vec{w} &\approx (0\ -0.71\ -0.71\ 1/3\ 1/3\ 1/3)^T \\ b &= -1/3 \end{aligned}$$

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.

Relevance feedback

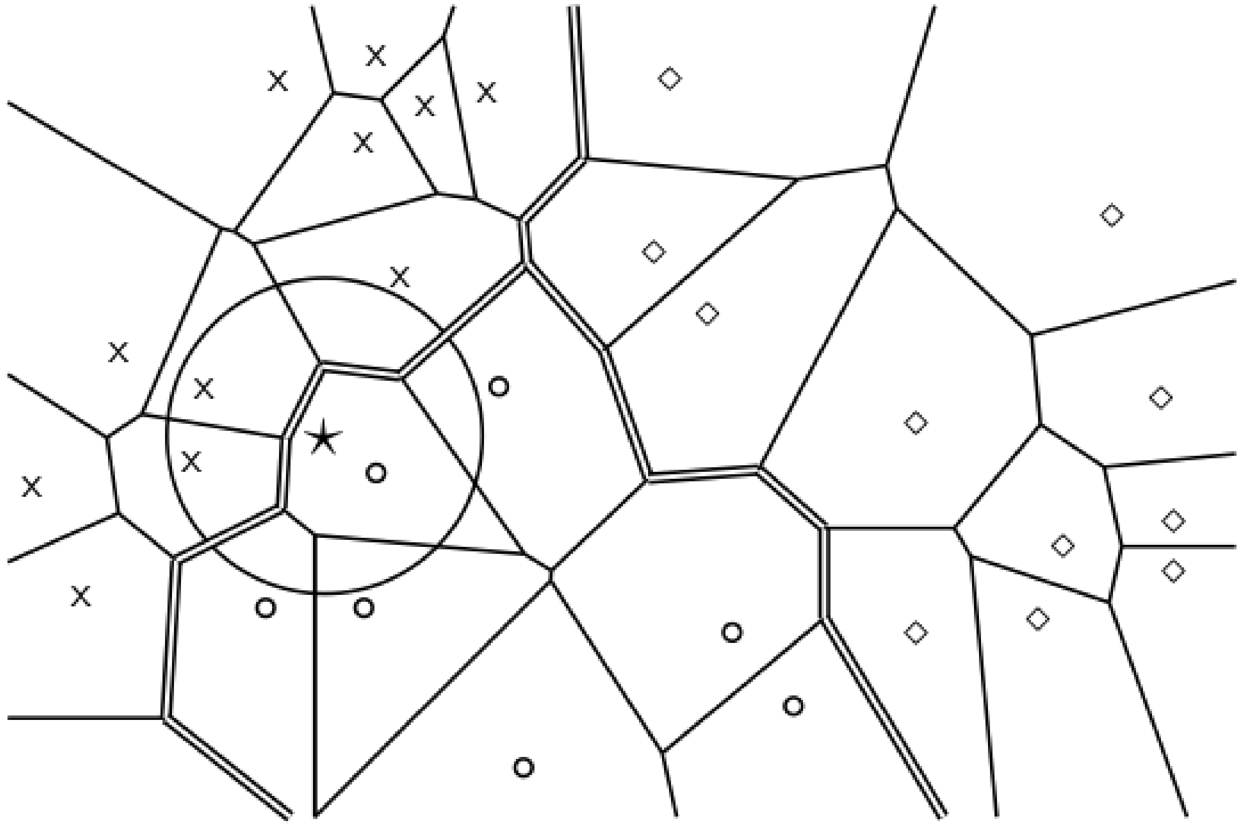
- In relevance feedback, the user marks documents as relevant/nonrelevant.
- Relevant/nonrelevant can be viewed as classes or categories.
- For each document, the user decides which of these two classes is correct.
- The IR system then uses these class assignments to build a better query ("model") of the information need . . .
- . . . and returns better documents.
- Relevance feedback is a form of text classification.

k Nearest Neighbor Classification

- kNN = k Nearest Neighbor

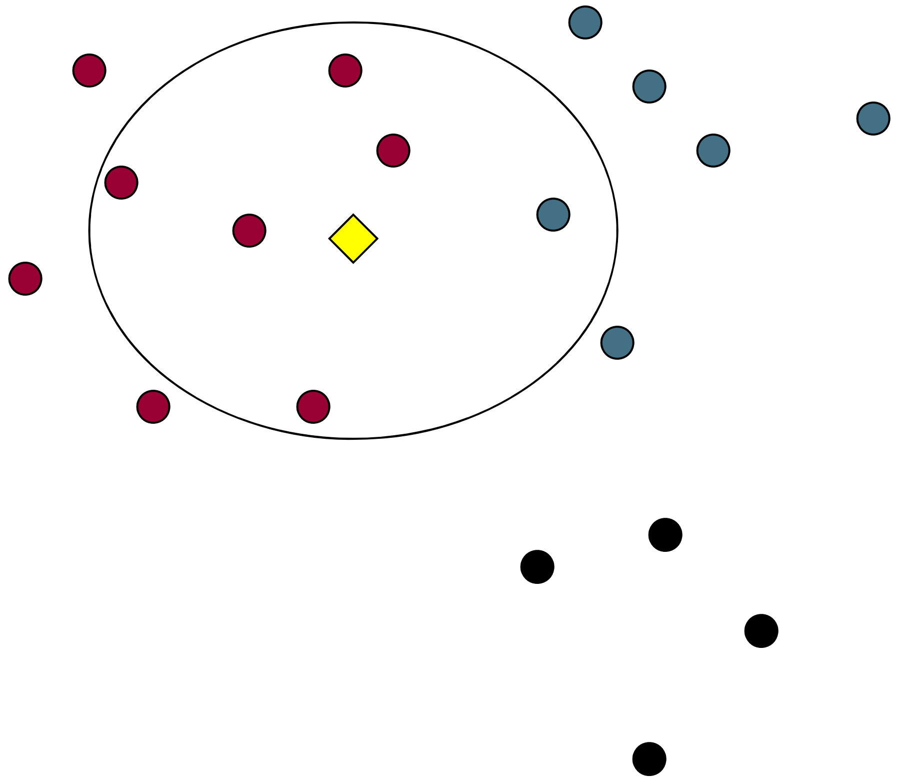
- To classify a document *d* into class c:
- Define *k*-neighborhood N as *k* nearest neighbors of *d*
- Count number of documents i in N that belong to c
- Estimate P(c|*d*) as i/k
- Choose as class argmax$_c$ P(c|*d*) [= majority class]

Probabilistic kNN



1NN, 3NN classification decision for star?

Example: k=6 (6NN)



P(science|◇)?

● Government

● Science

● Arts

Slides by Manning, Raghavan, Schutze

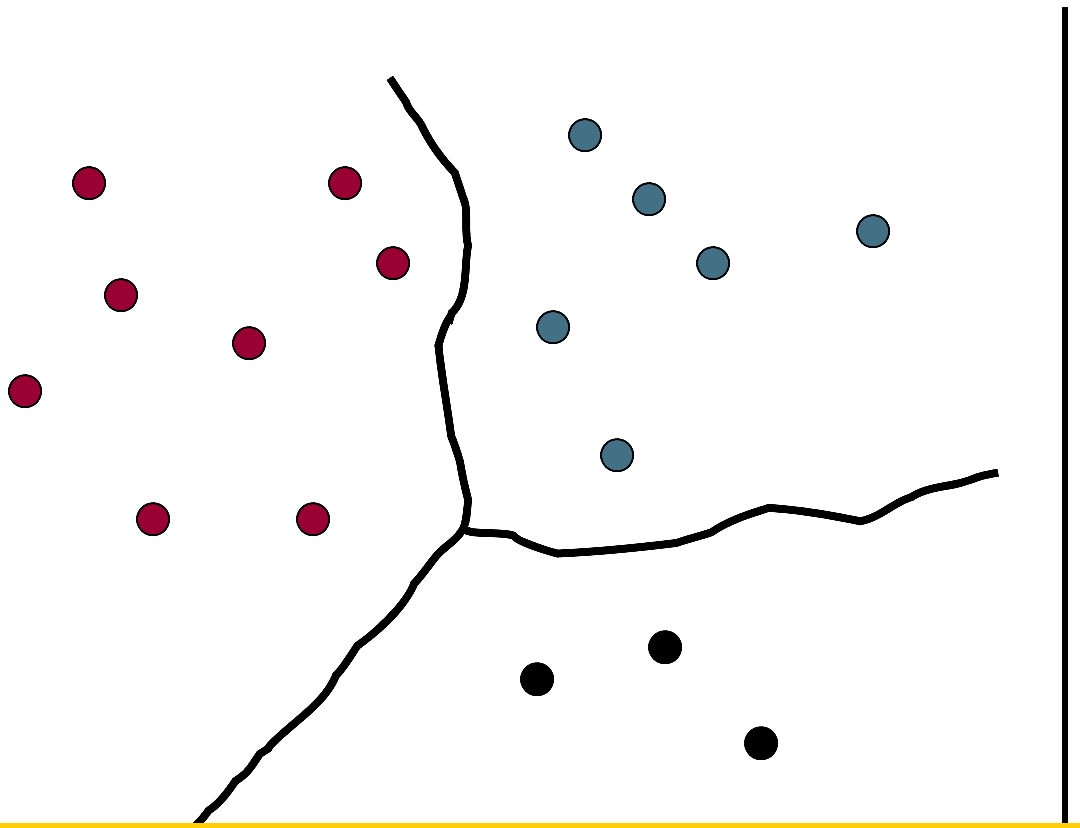
Nearest-Neighbor Learning Algorithm

- Learning is just storing the representations of the training examples in *D*.
- Testing instance *x (under 1NN)*:
  - Compute similarity between *x* and all examples in *D*.
  - Assign *x* the category of the most similar example in *D*.
- Does not explicitly compute a generalization or category prototypes.
- Also called:
  - Case-based learning
  - Memory-based learning
  - Lazy learning
- Rationale of kNN: contiguity hypothesis

k Nearest Neighbor

- Using only the closest example (1NN) to determine the class is subject to errors due to:
  - A single atypical example.
  - Noise (i.e., an error) in the category label of a single training example.
- More robust alternative is to find the *k* most-similar examples and return the majority category of these *k* examples.
- Value of *k* is typically odd to avoid ties; 3 and 5 are most common.

kNN decision boundaries



Boundaries are in principle arbitrary surfaces – but usually polyhedra

● Government

● Science

● Arts

kNN gives locally defined decision boundaries between classes – far away points do not influence each classification decision (unlike in Naïve Bayes, Rocchio, etc.)

Similarity Metrics

- Nearest neighbor method depends on a similarity (or distance) metric.
- Simplest for continuous *m*-dimensional instance space is *Euclidean distance*.
- Simplest for *m*-dimensional binary instance space is *Hamming distance* (number of feature values that differ).
- For text, cosine similarity of tf.idf weighted vectors is typically most effective.

Nearest Neighbor with Inverted Index

- Naively finding nearest neighbors requires a linear search through $|D|$ documents in collection
- But 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.
- Use standard vector space inverted index methods to find the *k* nearest neighbors.
- Testing Time: $O(B|V_t|)$ where *B* is the average number of training documents in which a test-document word appears.
  - Typically $B << |D|$

kNN: Discussion

- No feature selection necessary
- Scales well with large number of classes
  - Don’t need to train *n* classifiers for *n* classes
- Classes can influence each other
  - Small changes to one class can have ripple effect
- Scores can be hard to convert to probabilities
- No training necessary
  - Actually: perhaps not true. (Data editing, etc.)
- May be expensive at test time
- In most cases it’s more accurate than NB or Rocchio

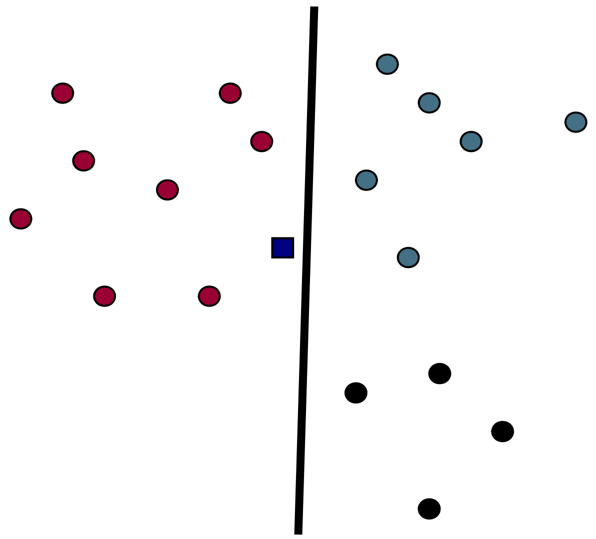
Linear classifiers and binary and multiclass classification

- Consider 2 class problems
    - Deciding between two classes, perhaps, government and non-government
        - One-versus-rest classification
- How do we define (and find) the separating surface?
- How do we decide which region a test doc is in?

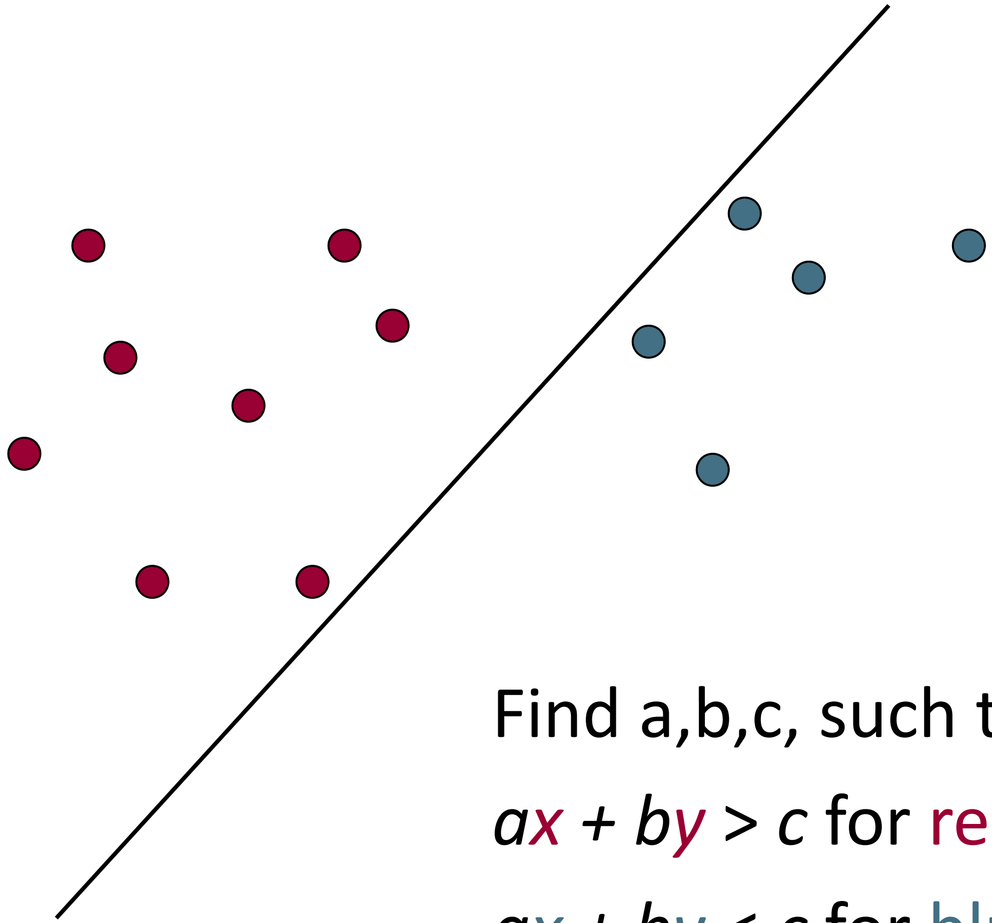
Separation by Hyperplanes

- A strong high-bias assumption is *linear separability*:
  - in 2 dimensions, can separate classes by a line
  - in higher dimensions, need hyperplanes
- Can find separating hyperplane by *linear programming*

  (or can iteratively fit solution via perceptron):
  - separator can be expressed as *ax + by = c*



Linear programming / Perceptron

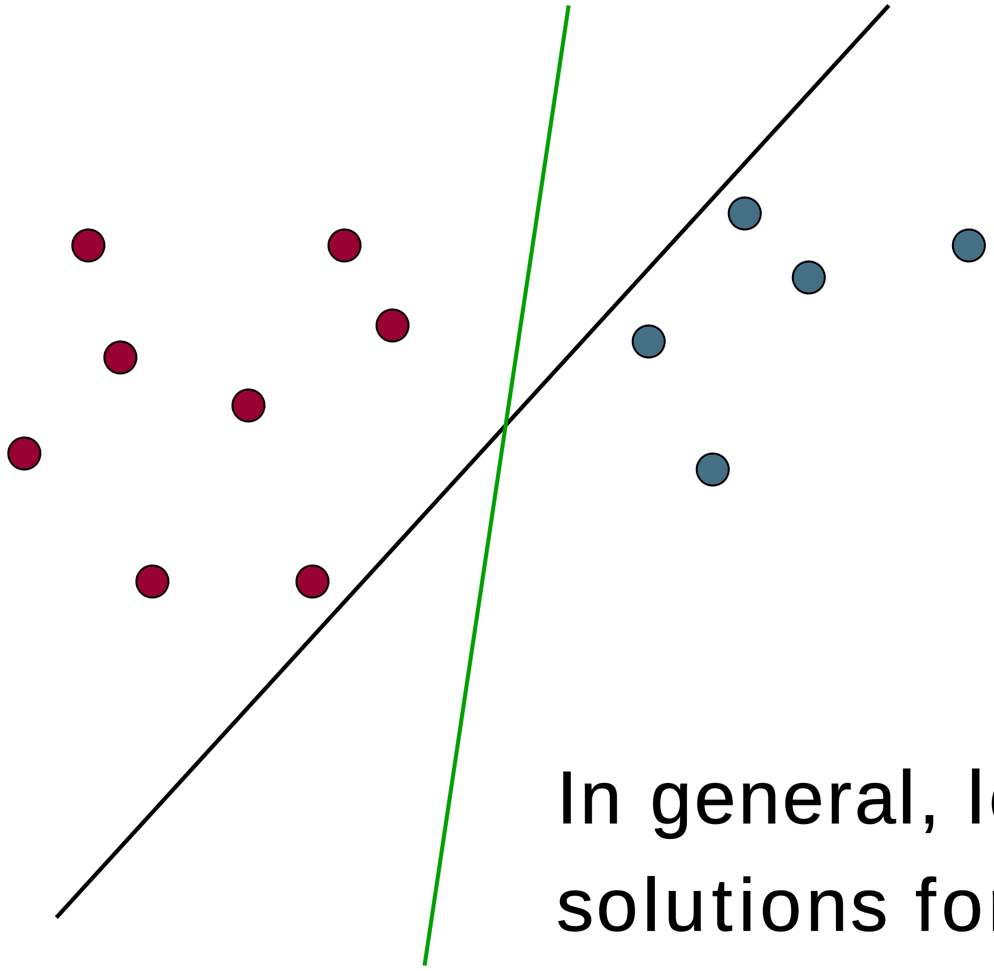


Find a,b,c, such that

$ax + by > c$ for red points

$ax + by < c$ for blue points.

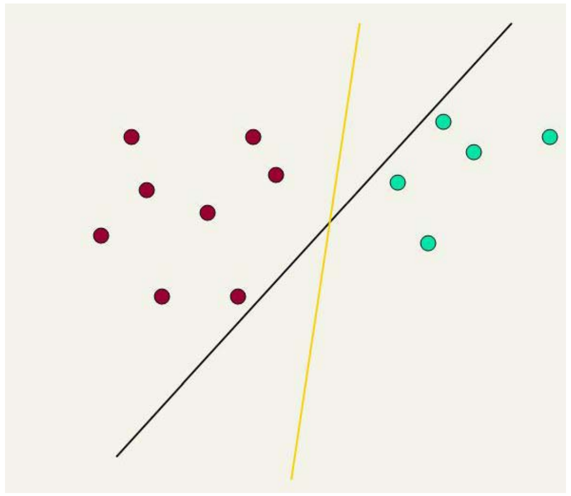
Which Hyperplane?



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 [according to some criterion of expected goodness]
- Most methods find an optimal separating hyperplane
- Which points should influence optimality?
  - All points
    - Linear/logistic regression
    - Naïve Bayes
  - Only "difficult points" close to decision boundary
    - Support vector machines



Linear Classifiers

- Many common text classifiers are linear classifiers
  - Naïve Bayes
  - Perceptron
  - Rocchio
  - Logistic regression
  - Support vector machines (with linear kernel)
  - Linear regression with threshold
- Despite this similarity, noticeable performance differences
  - For separable problems, there is an infinite number of separating hyperplanes. Which one do you choose?
  - What to do for non-separable problems?
  - Different training methods pick different hyperplanes
- Classifiers more powerful than linear often don't perform better on text problems. Why?

Two-class Rocchio as a linear classifier

- Line or hyperplane defined by:

$$\sum_{i=1}^{M} w_i d_i = b$$

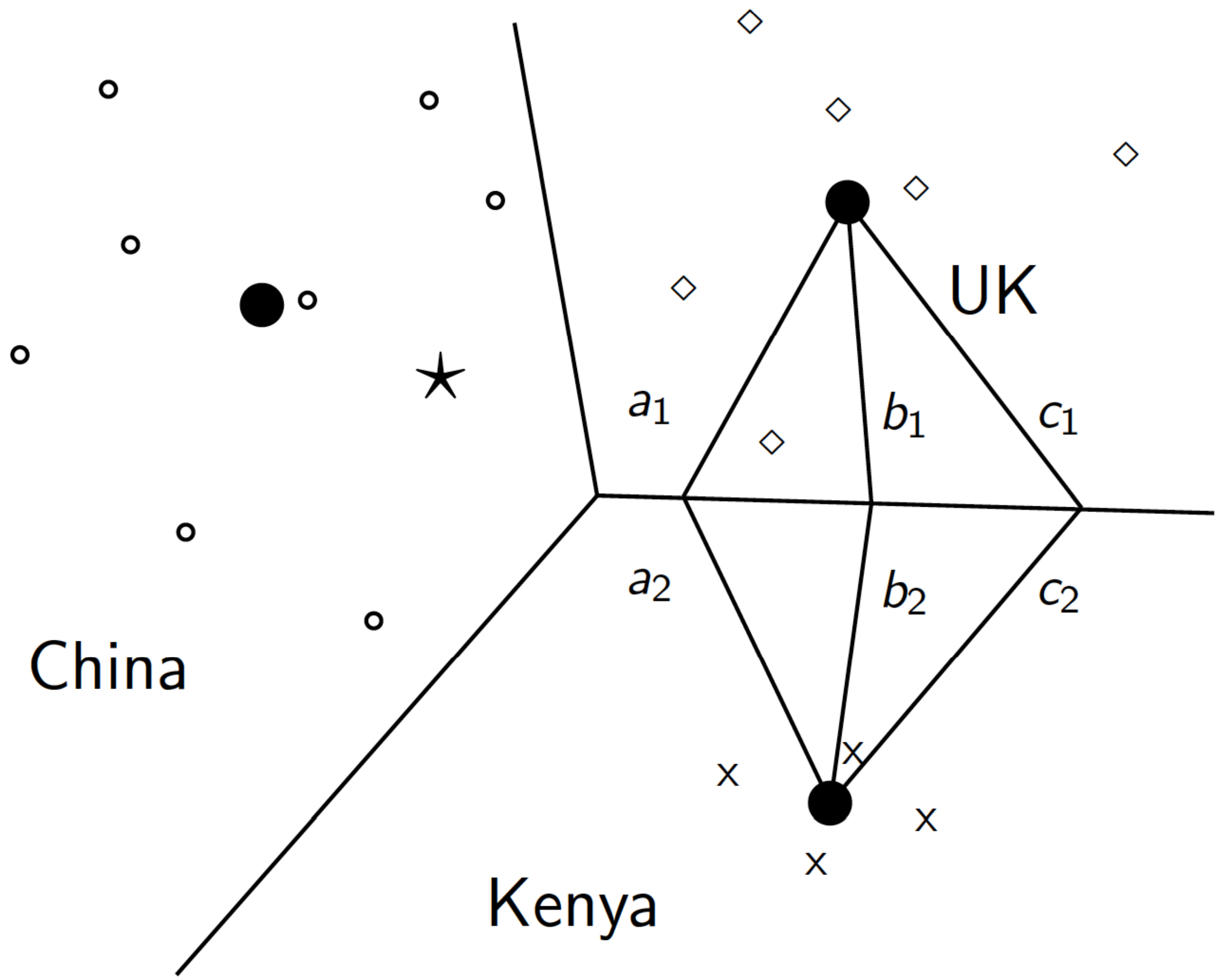
- For Rocchio, set:

$$\vec{w} = \vec{\mu}(c_1) - \vec{\mu}(c_2)$$

$$b = 0.5 \times (|\vec{\mu}(c_1)|^2 - |\vec{\mu}(c_2)|^2)$$

[Aside for ML/stats people: Rocchio classification is a simplification of the classic Fisher Linear Discriminant where you don’t model the variance (or assume it is spherical).]

Rocchio is a linear classifier



Naive Bayes is a linear classifier

- Two-class Naive Bayes. We compute:

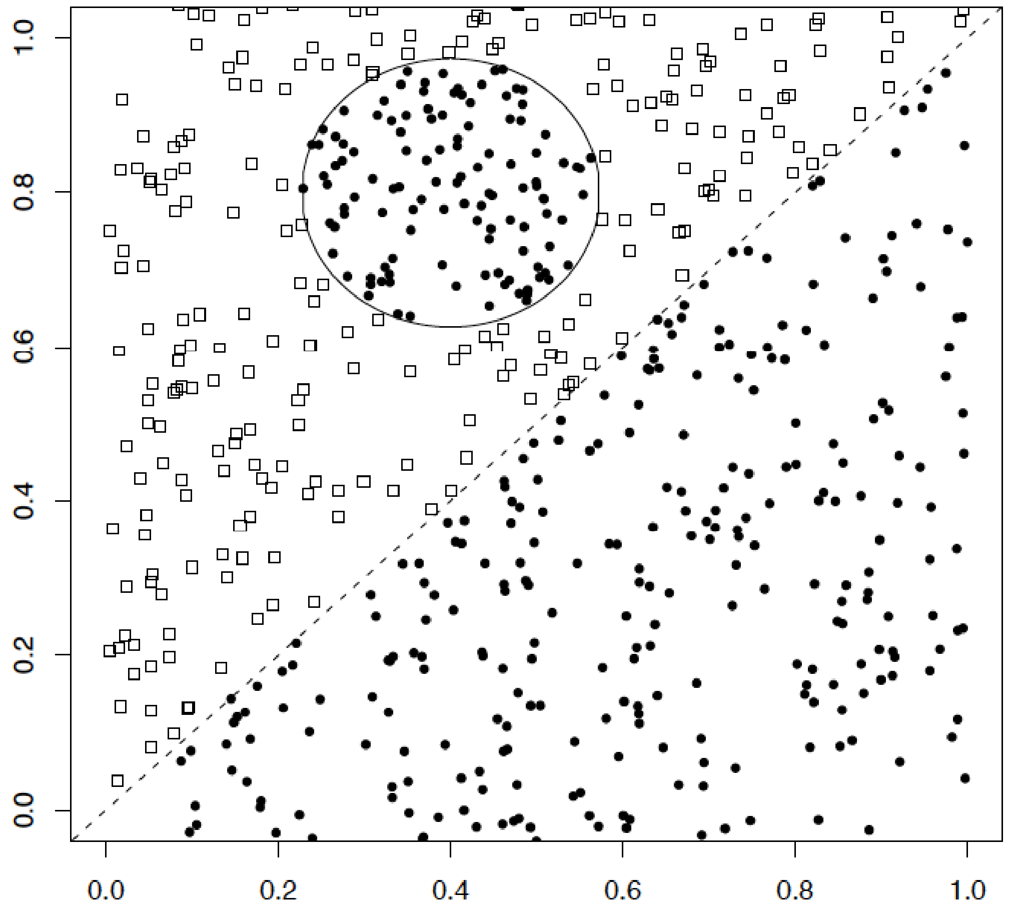
$$\log \frac{P(C|d)}{P(\overline{C}|d)} = \log \frac{P(C)}{P(\overline{C})} + \sum_{w \in d} \log \frac{P(w|C)}{P(w|\overline{C})}$$

- Decide class *C* if the odds is greater than 1, i.e., if the log odds is greater than 0.
- So decision boundary is hyperplane:

$$\alpha + \sum\nolimits_{w \in V} \beta_w \times n_w = 0 \quad \text{where } \alpha = \log \frac{P(C)}{P(\overline{C})};$$

$$\beta_w = \log \frac{P(w|C)}{P(w|\overline{C})}; \quad n_w = \#\text{ of occurrences of } w \text{ in } d$$

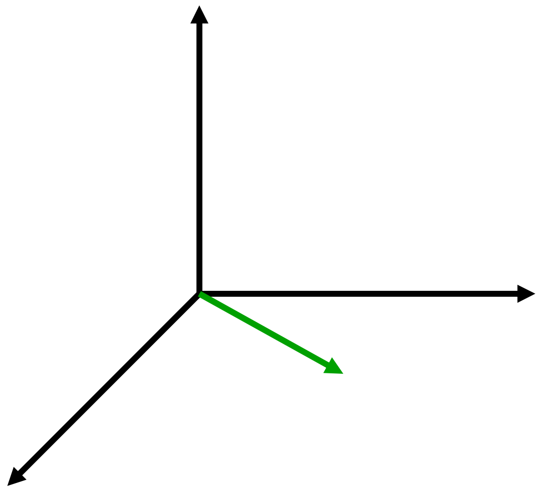
A nonlinear problem



- A linear classifier like Naïve Bayes does badly on this task
- kNN will do very well (assuming enough training data)

High Dimensional Data

- Pictures like the one at right are absolutely misleading!
- 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)
- In classification terms: often document sets are separable, for most any classification
- This is part of why linear classifiers are quite successful in this domain



More Than Two Classes

- Any-of or multivalue classification
  - Classes are independent of each other.
  - A document can belong to 0, 1, or >1 classes.
  - Decompose into *n* binary problems
  - Quite common for documents
- One-of or multinomial or polytomous classification
  - Classes are mutually exclusive.
  - Each document belongs to exactly one class
  - E.g., digit recognition is polytomous classification
    - Digits are mutually exclusive

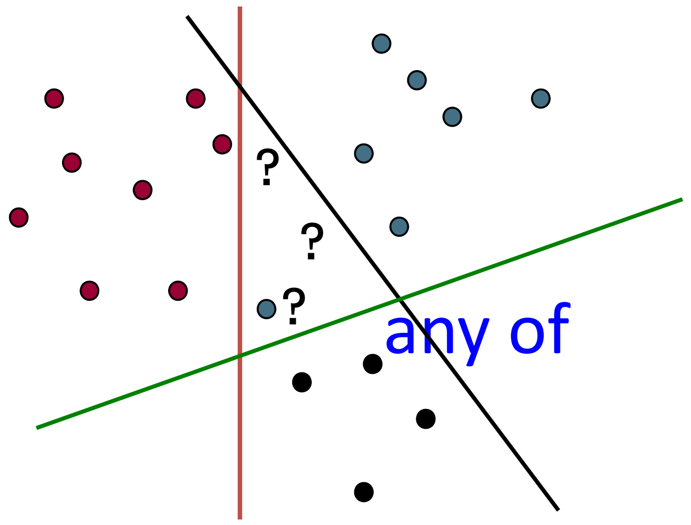
Set of Binary Classifiers: Any of

- Build a classifier between each class and its complementary set (docs from all other classes).
- Given test doc, evaluate it for membership in each class.
- Apply decision criterion of classifiers independently
- Done

    - Though maybe you could do better by considering dependencies between categories

Set of Binary Classifiers: One of

- Build a classifier between each class and its complementary set (docs from all other classes).
- Given test doc, evaluate it for membership in each class.
- Assign document to class with:
  - maximum score
  - maximum confidence
  - maximum probability
- Why different from multiclass/ any of classification?



Summary: Representation of Text Categorization Attributes

- Representations of text are usually very high dimensional (one feature for each word)
- High-bias algorithms that prevent overfitting in high-dimensional space should generally work best*
- For most text categorization tasks, there are many relevant features and many irrelevant ones
- Methods that combine evidence from many or all features (e.g. naive Bayes, kNN) often tend to work better than ones that try to isolate just a few relevant features*

*Although the results are a bit more mixed than often thought

Which classifier do I use for a given text classification problem?

- Is there a learning method that is optimal for all text classification problems?
- No, because there is a tradeoff between bias and variance.
- Factors to take into account:
  - How much training data is available?
  - How simple/complex is the problem? (linear vs. nonlinear decision boundary)
  - How noisy is the data?
  - How stable is the problem over time?
    - For an unstable problem, it's better to use a simple and robust classifier.

Evaluating Categorization

- Evaluation must be done on test data that are independent of the training data (usually a disjoint set of instances).
  - Sometimes use cross-validation (averaging results over multiple training and test splits of the overall data)
- 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, F1, classification accuracy
- *Classification accuracy*: c/n where n is the total number of test instances and c is the number of test instances correctly classified by the system.
  - Adequate if one class per document
  - Otherwise F measure for each class

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

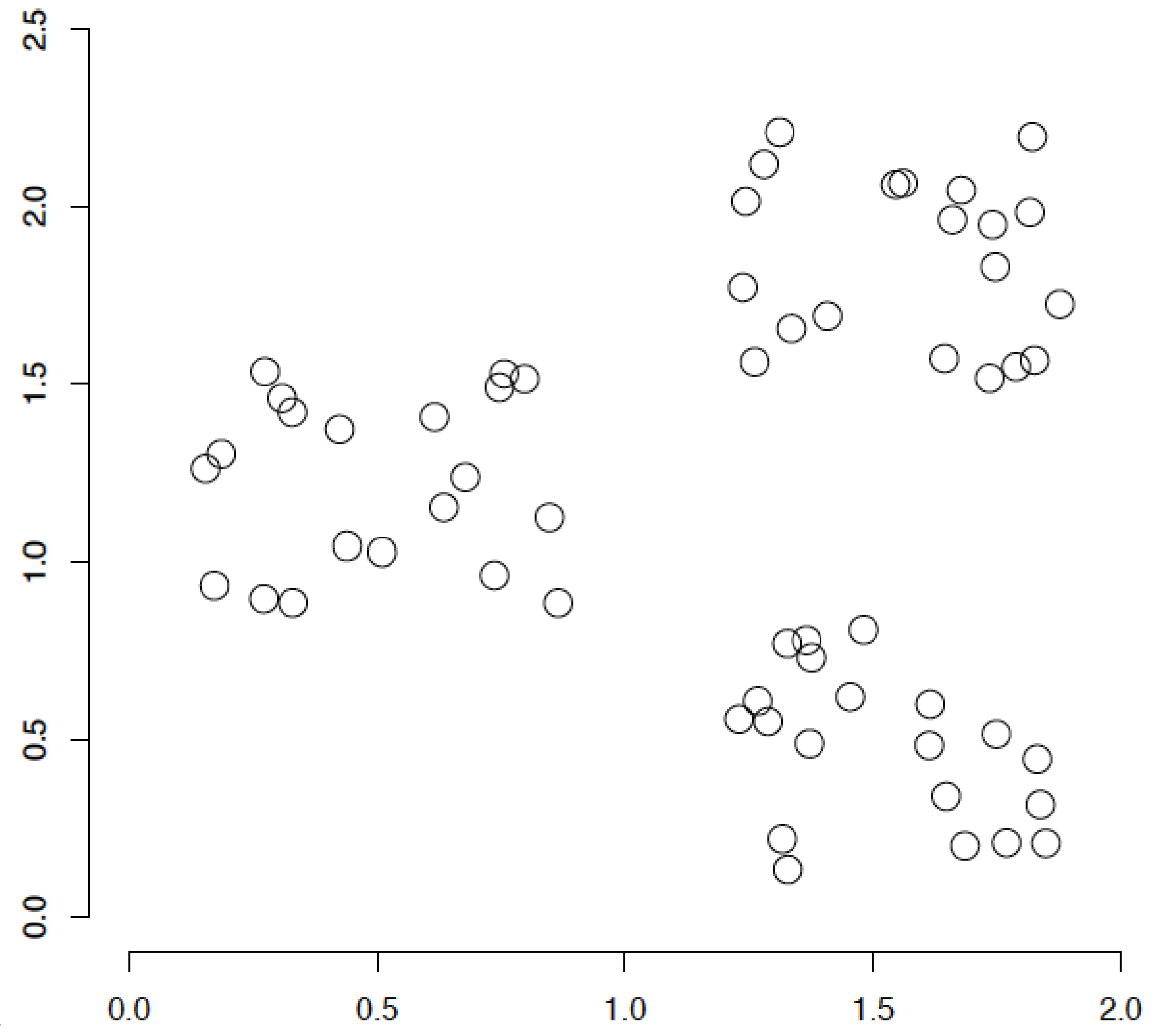
Evaluation measure: F_1

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

What is clustering?

- Clustering: the process of grouping a set of objects into classes of similar objects
  - Documents within a cluster should be similar.
  - Documents from different clusters should be dissimilar.
- The commonest 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

A data set with clear cluster structure

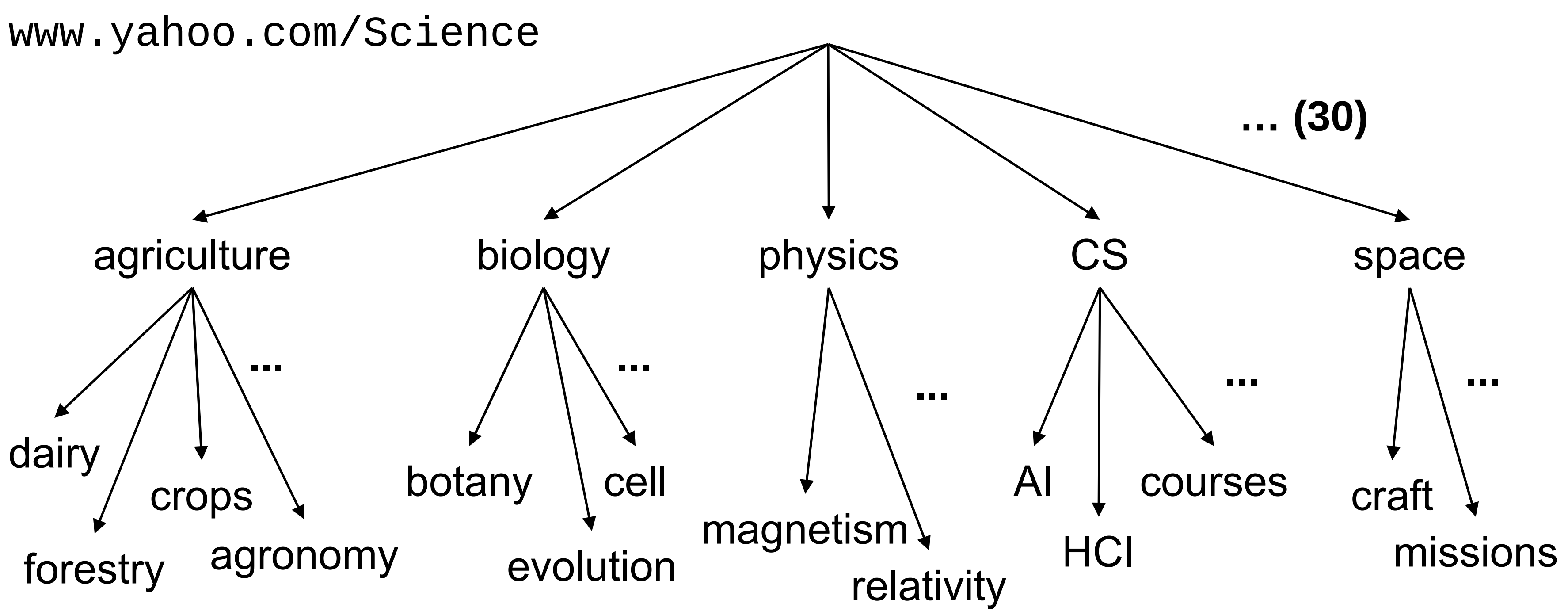


- How would you design an algorithm for finding the three clusters in this case?

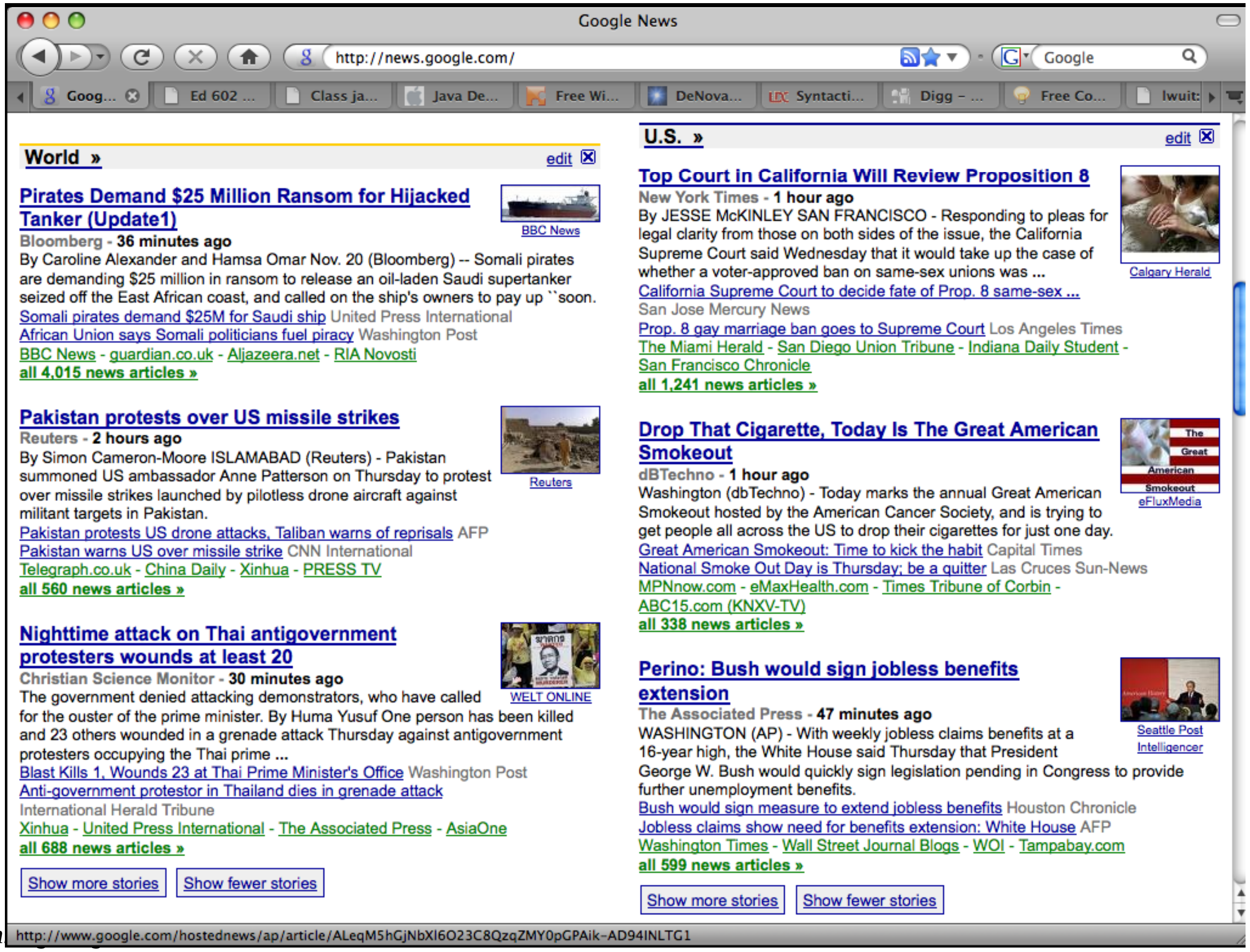
Applications of clustering in IR

- Whole corpus analysis/navigation
  - Better user interface: search without typing
- For improving recall in search applications
  - Better search results (like pseudo RF)
- For better navigation of search results
  - Effective “user recall” will be higher
- For speeding up vector space retrieval
  - Cluster-based retrieval gives faster search

Yahoo! Hierarchy *isn't* clustering but *is* the kind of output you want from clustering



Google News: automatic clustering gives an effective news presentation metaphor


Google News
http://news.google.com/
Google
Goog... Ed 602 ... Class ja... Java De... Free Wi... DeNova... Syntacti... Digg – ... Free Co... lwuit:

World » edit

Pirates Demand $25 Million Ransom for Hijacked Tanker (Update1)
BBC News
Bloomberg - **36 minutes ago**
By Caroline Alexander and Hamsa Omar Nov. 20 (Bloomberg) -- Somali pirates are demanding $25 million in ransom to release an oil-laden Saudi supertanker seized off the East African coast, and called on the ship's owners to pay up ``soon.
Somali pirates demand $25M for Saudi ship United Press International
African Union says Somali politicians fuel piracy Washington Post
BBC News - guardian.co.uk - Aljazeera.net - RIA Novosti
all 4,015 news articles »

Pakistan protests over US missile strikes
Reuters
Reuters - **2 hours ago**
By Simon Cameron-Moore ISLAMABAD (Reuters) - Pakistan summoned US ambassador Anne Patterson on Thursday to protest over missile strikes launched by pilotless drone aircraft against militant targets in Pakistan.
Pakistan protests US drone attacks, Taliban warns of reprisals AFP
Pakistan warns US over missile strike CNN International
Telegraph.co.uk - China Daily - Xinhua - PRESS TV
all 560 news articles »

Nighttime attack on Thai antigovernment protesters wounds at least 20
WELT ONLINE
Christian Science Monitor - **30 minutes ago**
The government denied attacking demonstrators, who have called for the ouster of the prime minister. By Huma Yusuf One person has been killed and 23 others wounded in a grenade attack Thursday against antigovernment protesters occupying the Thai prime ...
Blast Kills 1, Wounds 23 at Thai Prime Minister's Office Washington Post
Anti-government protestor in Thailand dies in grenade attack International Herald Tribune
Xinhua - United Press International - The Associated Press - AsiaOne
all 688 news articles »

Show more stories | Show fewer stories

U.S. » edit

Top Court in California Will Review Proposition 8
Calgary Herald
New York Times - **1 hour ago**
By JESSE McKINLEY SAN FRANCISCO - Responding to pleas for legal clarity from those on both sides of the issue, the California Supreme Court said Wednesday that it would take up the case of whether a voter-approved ban on same-sex unions was ...
California Supreme Court to decide fate of Prop. 8 same-sex ... San Jose Mercury News
Prop. 8 gay marriage ban goes to Supreme Court Los Angeles Times
The Miami Herald - San Diego Union Tribune - Indiana Daily Student - San Francisco Chronicle
all 1,241 news articles »

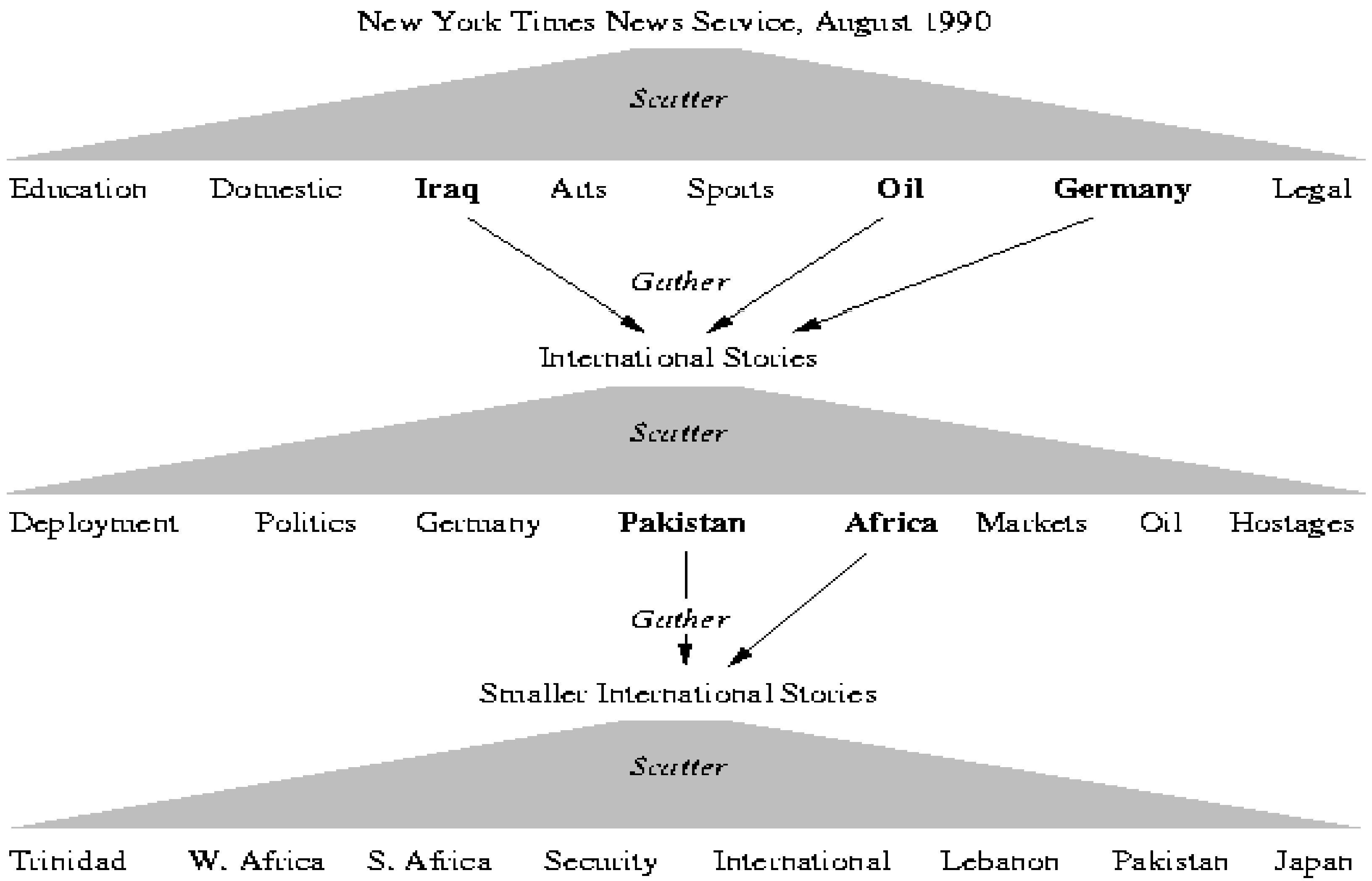
Drop That Cigarette, Today Is The Great American Smokeout
eFluxMedia
dBTechno - **1 hour ago**
Washington (dbTechno) - Today marks the annual Great American Smokeout hosted by the American Cancer Society, and is trying to get people all across the US to drop their cigarettes for just one day.
Great American Smokeout: Time to kick the habit Capital Times
National Smoke Out Day is Thursday; be a quitter Las Cruces Sun-News
MPNnow.com - eMaxHealth.com - Times Tribune of Corbin - ABC15.com (KNXV-TV)
all 338 news articles »

Perino: Bush would sign jobless benefits extension
Seattle Post Intelligencer
The Associated Press - **47 minutes ago**
WASHINGTON (AP) - With weekly jobless claims benefits at a 16-year high, the White House said Thursday that President George W. Bush would quickly sign legislation pending in Congress to provide further unemployment benefits.
Bush would sign measure to extend jobless benefits Houston Chronicle
Jobless claims show need for benefits extension: White House AFP
Washington Times - Wall Street Journal Blogs - WOI - Tampabay.com
all 599 news articles »

Show more stories | Show fewer stories

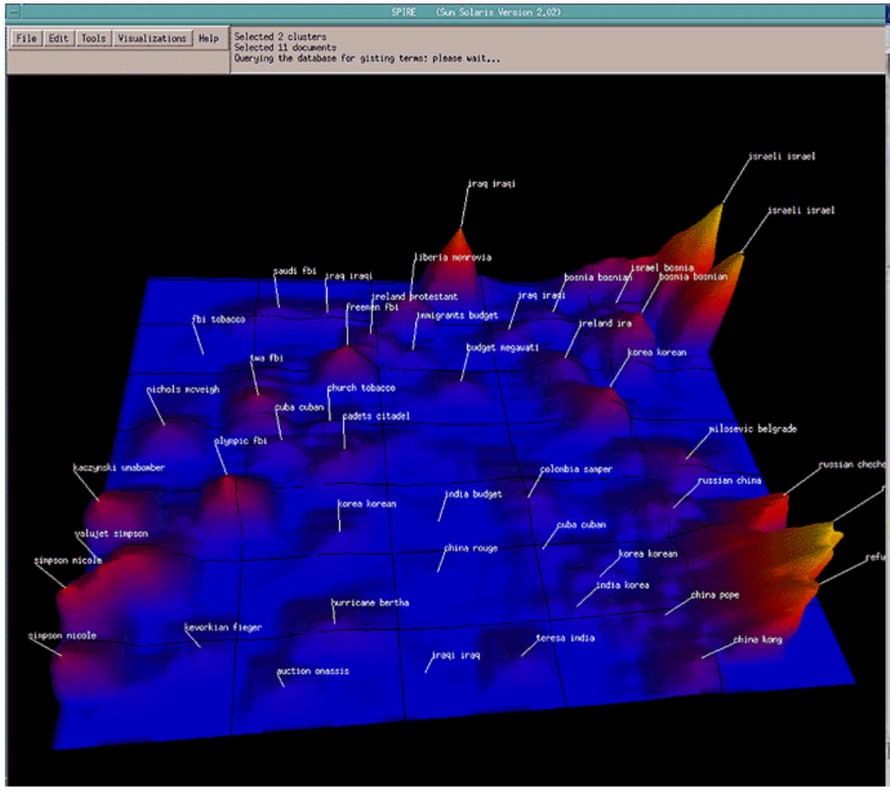
http://www.google.com/hostednews/ap/article/ALeqM5hGjNbXI6O23C8QzqZMY0pGPAik-AD94INLTG1

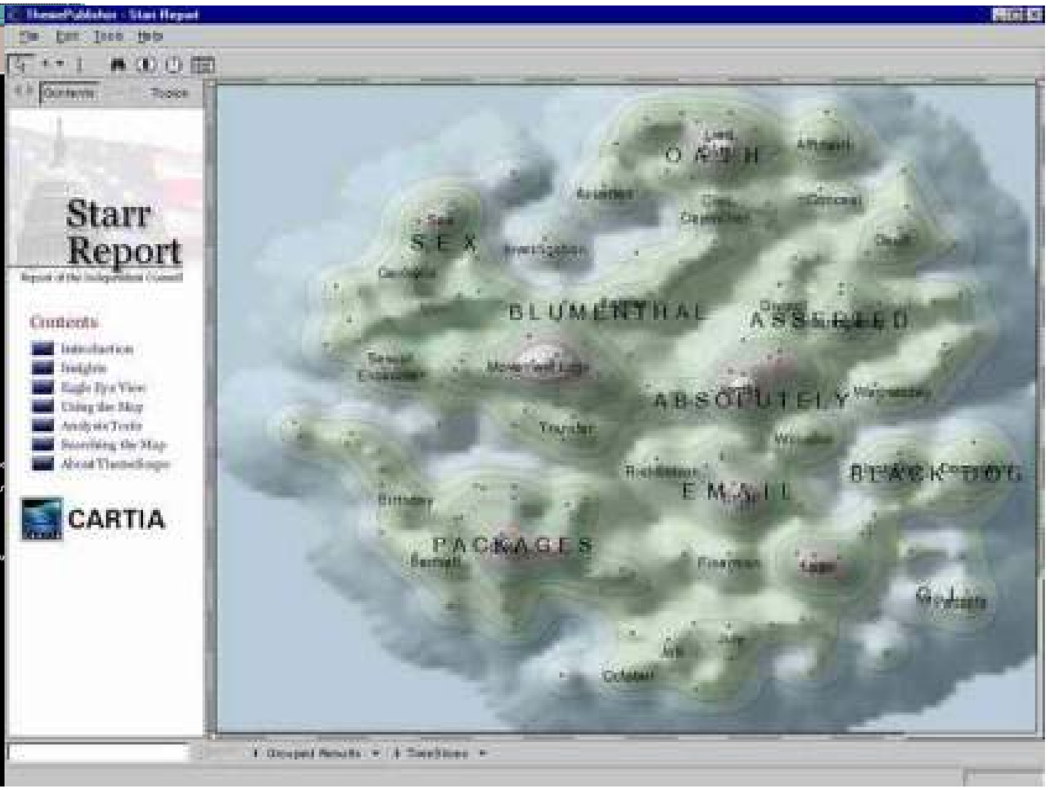
Scatter/Gather: Cutting, Karger, and Pedersen



For visualizing a document collection and its themes

- Wise et al, “Visualizing the non-visual” PNNL
- ThemeScapes, Cartia
  - [Mountain height = cluster size]





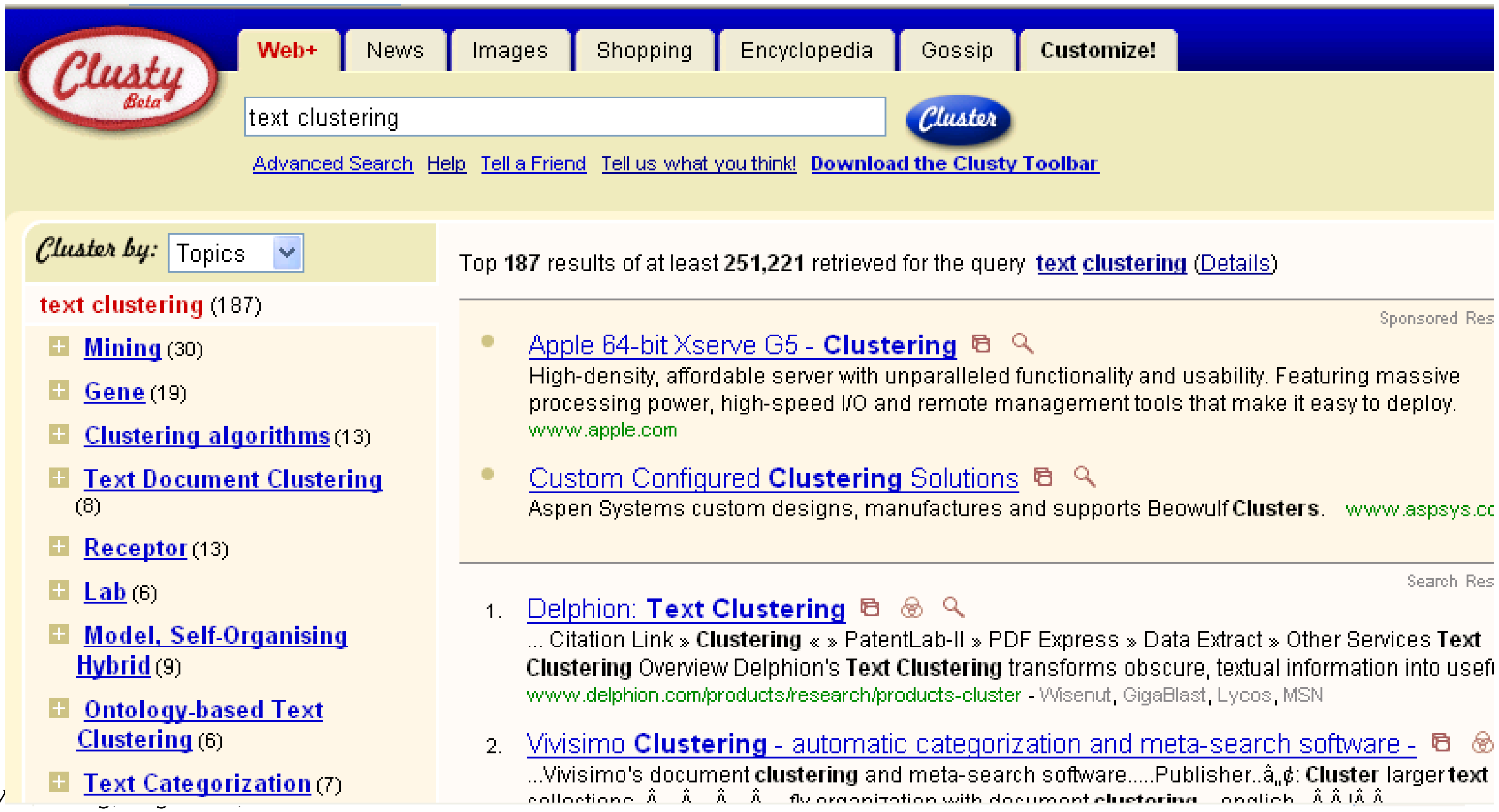
For improving search recall

- *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.*

Why might this happen?

For better navigation of search results

- For grouping search results thematically
  - clusty.com / Vivisimo



text clustering (187)
- Mining (30)
- Gene (19)
- Clustering algorithms (13)
- Text Document Clustering (8)
- Receptor (13)
- Lab (6)
- Model, Self-Organising Hybrid (9)
- Ontology-based Text Clustering (6)
- Text Categorization (7)

Top **187** results of at least **251,221** retrieved for the query **text clustering** (Details)

Sponsored Res

- Apple 64-bit Xserve G5 - **Clustering**
  High-density, affordable server with unparalleled functionality and usability. Featuring massive processing power, high-speed I/O and remote management tools that make it easy to deploy.
  www.apple.com
- Custom Configured **Clustering** Solutions
  Aspen Systems custom designs, manufactures and supports Beowulf **Clusters**. www.aspsys.co

Search Res

1. Delphion: **Text Clustering**
   ... Citation Link » **Clustering** « » PatentLab-II » PDF Express » Data Extract » Other Services **Text Clustering** Overview Delphion's **Text Clustering** transforms obscure, textual information into usef
   www.delphion.com/products/research/products-cluster - Wisenut, GigaBlast, Lycos, MSN
2. Vivisimo **Clustering** - automatic categorization and meta-search software -
   ...Vivisimo's document **clustering** and meta-search software.....Publisher..â„¢: **Cluster** larger **text**

Slides by

Issues for clustering

- Representation for clustering
  - Document representation
    - Vector space? Normalization?
      - Centroids aren't length normalized
  - 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.

Notion of similarity/distance

- Ideal: semantic similarity.
- Practical: term-statistical similarity
  - We will use cosine similarity.
  - Docs as vectors.
  - For many algorithms, easier to think in terms of a *distance* (rather than similarity) between docs.
  - We will mostly speak of Euclidean distance
    - But real implementations use cosine similarity

Clustering Algorithms

- Flat algorithms
  - Usually start with a random (partial) partitioning
  - Refine it iteratively
    - *K* means clustering
    - (Model based clustering)
- Hierarchical algorithms
  - Bottom-up, agglomerative
  - (Top-down, divisive)

Hard vs. soft clustering

- Hard clustering: Each document belongs to exactly one cluster
  - More common and easier to do
- Soft clustering: A document can belong to more than one cluster.
  - Makes more sense for applications like creating browsable hierarchies
  - You may want to put a pair of sneakers in two clusters: (i) sports apparel and (ii) shoes
  - You can only do that with a soft clustering approach.
- We won’t do soft clustering today

Partitioning Algorithms

- Partitioning method: Construct a partition of *n* documents into a set of *K* clusters
- Given: a set of documents and the number *K*
- Find: a partition of *K* clusters that optimizes the chosen partitioning criterion
  - Globally optimal
    - Intractable for many objective functions
    - Ergo, exhaustively enumerate all partitions
  - Effective heuristic methods: *K*-means and *K*-medoids algorithms

K-Means

- Assumes documents are real-valued vectors.
- Clusters based on *centroids* (aka the *center of gravity* or mean) of points in a cluster, *c*:

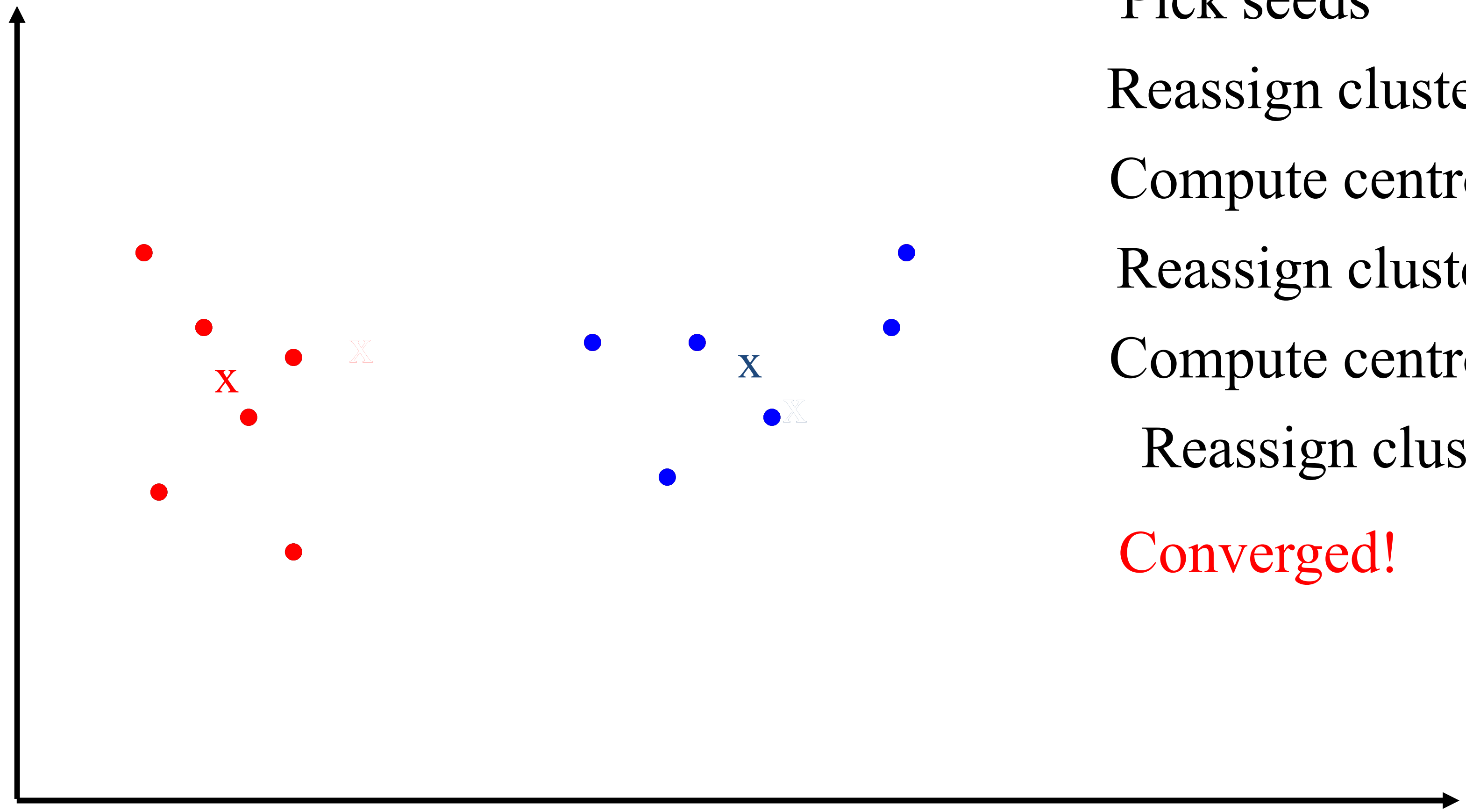
$$\vec{\mu}(\mathrm{c}) = \frac{1}{|c|} \sum_{\vec{x} \in c} \vec{x}$$

- Reassignment of instances to clusters is based on distance to the current cluster centroids.
  - (Or one can equivalently phrase it in terms of similarities)

K-Means Algorithm

```
Select $K$ random docs $\{s_1, s_2,\ldots s_K\}$ as seeds.
Until clustering *converges* (or other stopping criterion):
    For each doc $d_i$:
        Assign $d_i$ to the cluster $c_j$ such that $dist(x_i, s_j)$ is minimal.
    (*Next, update the seeds to the centroid of each cluster*)
    For each cluster $c_j$
        $s_j = \mu(c_j)$
```

K Means Example (*K*=2)



Pick seeds

Reassign clusters

Compute centroids

Reassign clusters

Compute centroids

Reassign clusters

Converged!

Termination conditions

- Several possibilities, e.g.,
  - A fixed number of iterations.
  - Doc partition unchanged.
  - Centroid positions don’t change.

Does this mean that the docs in a cluster are unchanged?

Convergence

- Why should the *K*-means algorithm ever reach a *fixed point*?
  - A state in which clusters don’t change.
- *K*-means is a special case of a general procedure known as the *Expectation Maximization (EM) algorithm*.
  - EM is known to converge.
  - Number of iterations could be large.
    - But in practice usually isn’t

Lower case!

Convergence of *K*-Means

- Define goodness measure of cluster *k* as sum of squared distances from cluster centroid:
  - $G_k = \Sigma_i (d_i - c_k)^2$ (sum over all d_i in cluster k)
- $G = \Sigma_k G_k$
- Reassignment monotonically decreases G since each vector is assigned to the closest centroid.

Convergence of *K*-Means

- Recomputation monotonically decreases each G_k since (m_k is number of members in cluster k):
  - $\Sigma (d_i - a)^2$ reaches minimum for:
  - $\Sigma\, 2(d_i - a) = 0$
  - $\Sigma\, d_i = \Sigma\, a$
  - $m_K\, a = \Sigma\, d_i$
  - $a = (1/ m_k)\, \Sigma\, d_i = c_k$
- *K*-means typically converges quickly

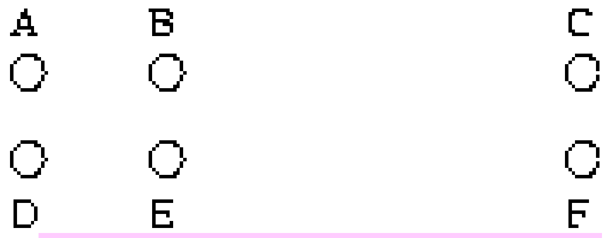
Time Complexity

- Computing distance between two docs is O(*M)* where *M* is the dimensionality of the vectors.
- Reassigning clusters: O(*KN)* distance computations, or O(*KNM).*
- Computing centroids: Each doc gets added once to some centroid: O(*NM).*
- Assume these two steps are each done once for *I* iterations: O(*IKNM).*

Seed Choice

- Results can vary based on random seed selection.
- Some seeds can result in poor convergence rate, or convergence to sub-optimal clusterings.
  - Select good seeds using a heuristic (e.g., doc least similar to any existing mean)
  - Try out multiple starting points
  - Initialize with the results of another method.

Example showing sensitivity to seeds



In the above, if you start with B and E as centroids you converge to {A,B,C} and {D,E,F}
If you start with D and F you converge to {A,B,D,E} {C,F}

K-means issues, variations, etc.

- Recomputing the centroid after every assignment (rather than after all points are re-assigned) can improve speed of convergence of *K*-means
- Assumes clusters are spherical in vector space
    - Sensitive to coordinate changes, weighting etc.
- Disjoint and exhaustive
    - Doesn’t have a notion of “outliers” by default
    - But can add outlier filtering

How Many Clusters?

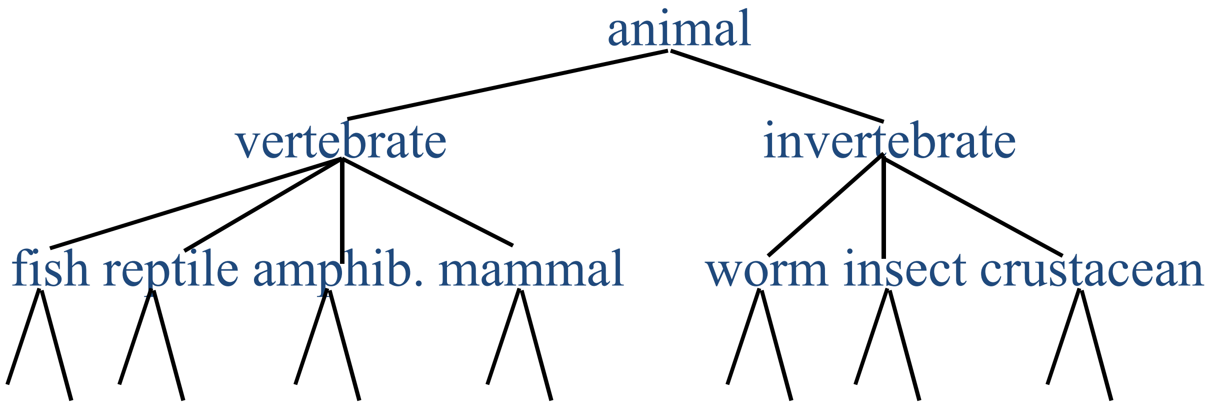
- Number of clusters *K* is given
    - Partition *n* docs into predetermined number of clusters
- Finding the “right” number of clusters is part of the problem
    - Given docs, partition into an “appropriate” number of subsets.
    - E.g., for query results - ideal value of *K* not known up front - though UI may impose limits.
- Can usually take an algorithm for one flavor and convert to the other.

K not specified in advance

- Say, the results of a query.
- Solve an optimization problem: penalize having lots of clusters
  - application dependent, e.g., compressed summary of search results list.
- Tradeoff between having more clusters (better focus within each cluster) and having too many clusters

Hierarchical Clustering

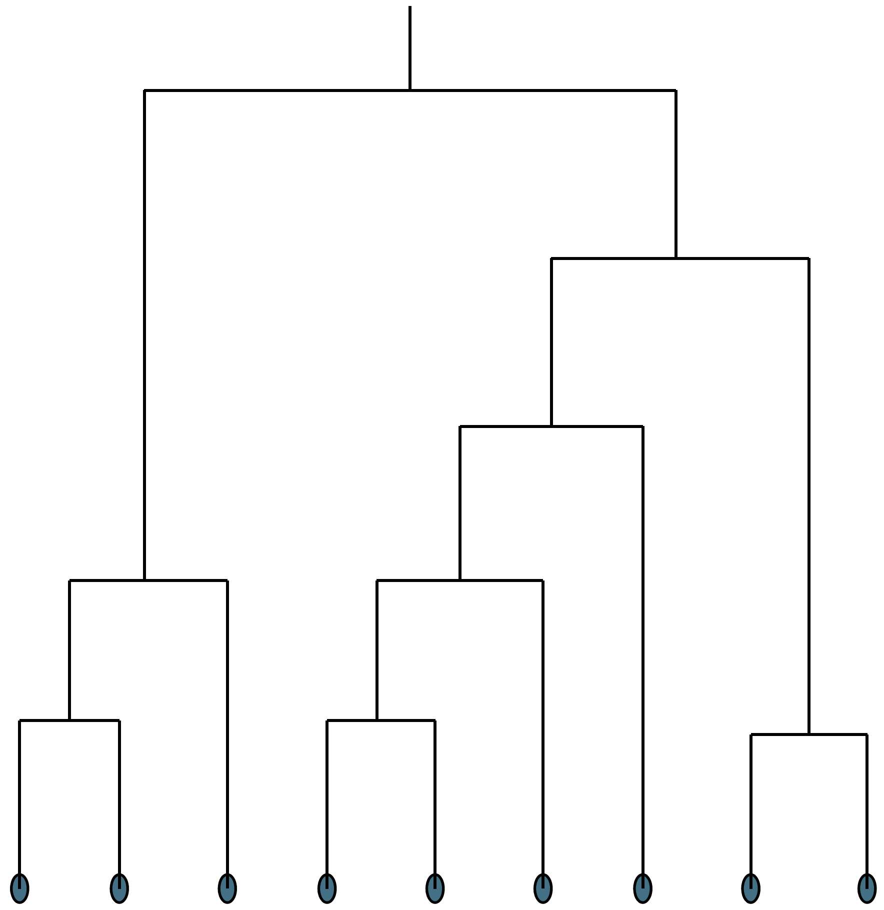
- Build a tree-based hierarchical taxonomy (*dendrogram*) from a set of documents.



- One approach: recursive application of a partitional clustering algorithm.

Dendrogram: Hierarchical Clustering

- Clustering obtained by cutting the dendrogram at a desired level: each connected component forms a cluster.



Hierarchical Agglomerative Clustering (HAC)

- Starts with each doc in a separate cluster
  - then repeatedly joins the *closest pair* of clusters, until there is only one cluster.
- The history of merging forms a binary tree or hierarchy.

Closest pair of clusters

- Many variants to defining closest pair of clusters
- **Single-link**
  - Similarity of the *most* cosine-similar (single-link)
- **Complete-link**
  - Similarity of the "furthest" points, the *least* cosine-similar
- **Centroid**
  - Clusters whose centroids (centers of gravity) are the most cosine-similar
- **Average-link**
  - Average cosine between pairs of elements

Single Link Agglomerative Clustering

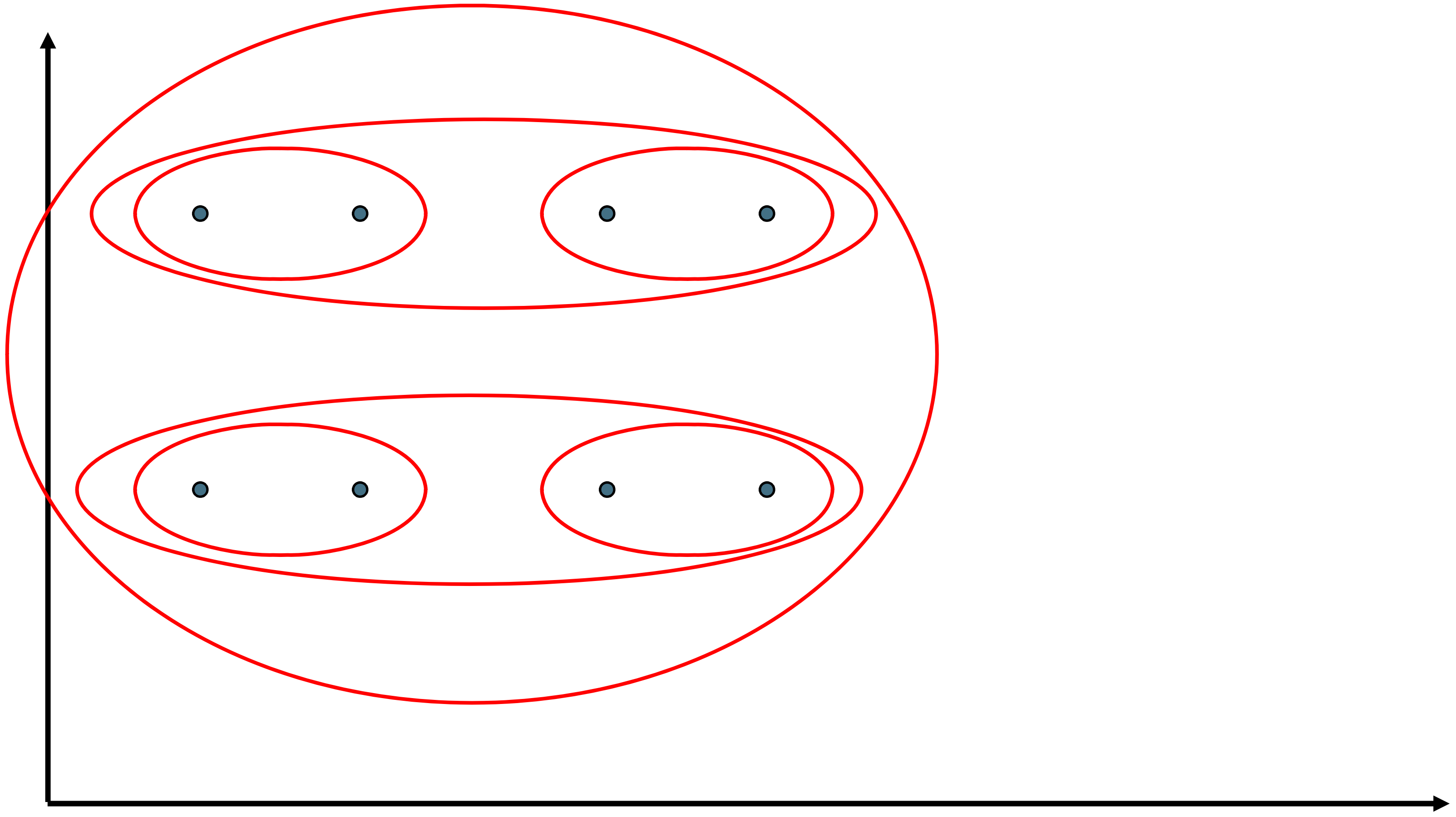
- Use maximum similarity of pairs:

$$sim(c_i, c_j) = \max_{x \in c_i, y \in c_j} sim(x, y)$$

- Can result in "straggly" (long and thin) clusters due to chaining effect.
- After merging c_i and c_j, the similarity of the resulting cluster to another cluster, c_k, is:

$$sim((c_i \cup c_j), c_k) = \max(sim(c_i, c_k), sim(c_j, c_k))$$

Single Link Example



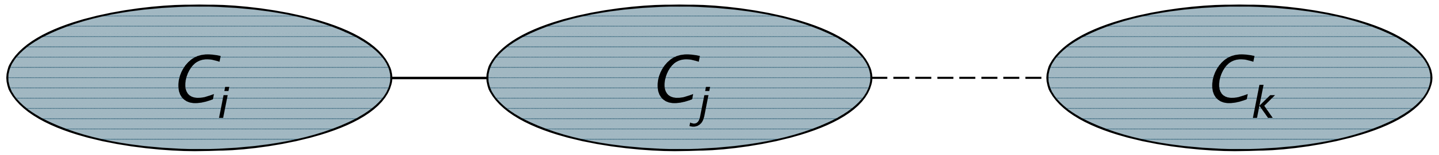
Complete Link

- Use minimum similarity of pairs:

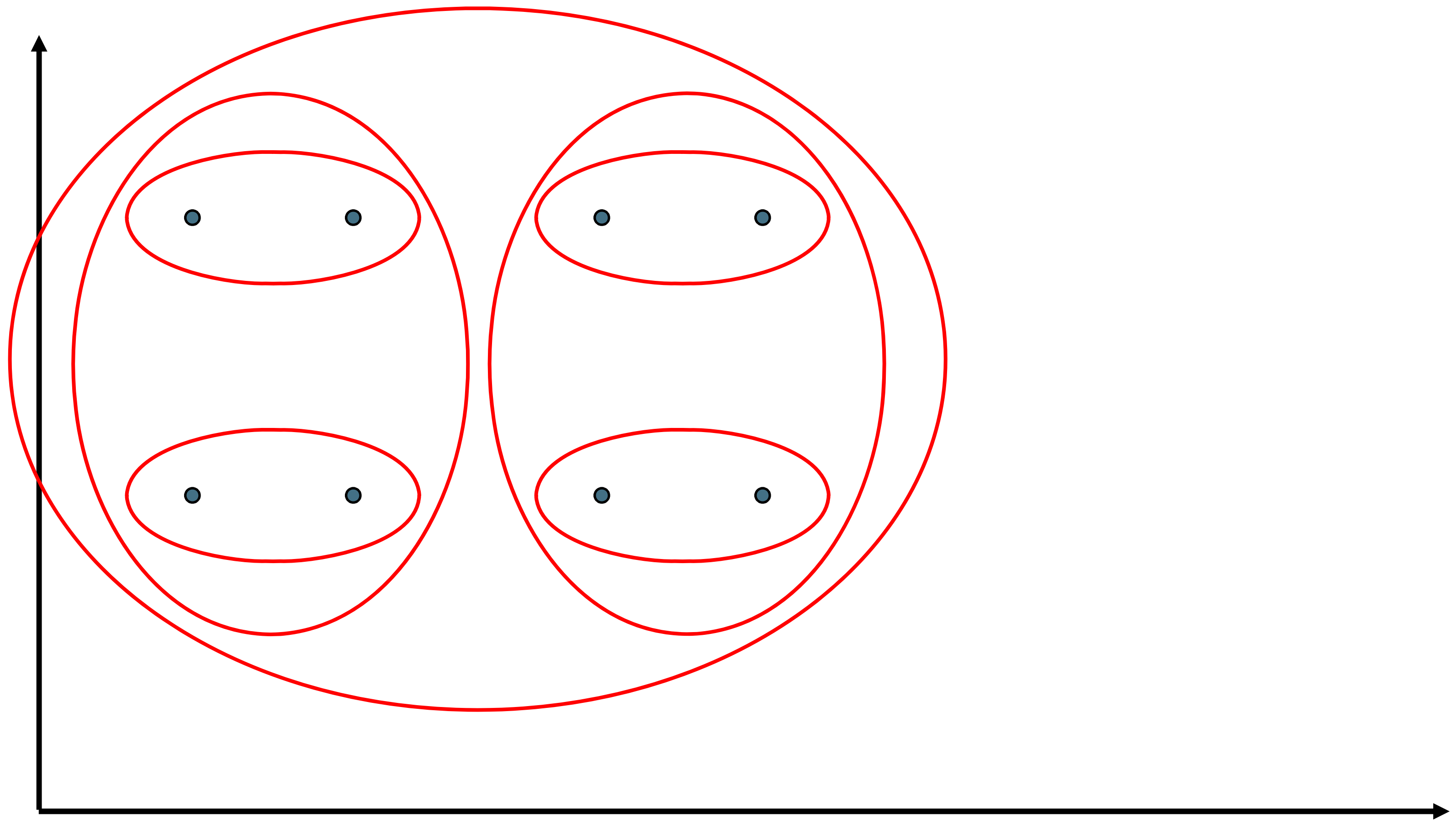
$$sim(c_i, c_j) = \min_{x \in c_i, y \in c_j} sim(x, y)$$

- Makes "tighter," spherical clusters that are typically preferable.
- After merging c_i and c_j, the similarity of the resulting cluster to another cluster, c_k, is:

$$sim((c_i \cup c_j), c_k) = \min(sim(c_i, c_k), sim(c_j, c_k))$$



Complete Link Example



Computational Complexity

- In the first iteration, all HAC methods need to compute similarity of all pairs of N initial instances, which is O(N^2).
- In each of the subsequent N–2 merging iterations, compute the distance between the most recently created cluster and all other existing clusters.
- In order to maintain an overall O(N^2) performance, computing similarity to each other cluster must be done in constant time.
  - Often O(N^3) if done naively or O(N^2 log N) if done more cleverly

Group Average

- Similarity of two clusters = average similarity of all pairs within merged cluster.

$$sim(c_i, c_j) = \frac{1}{\left|c_i \cup c_j\right|\left(\left|c_i \cup c_j\right| - 1\right)} \sum_{\vec{x} \in (c_i \cup c_j)} \sum_{\vec{y} \in (c_i \cup c_j) : \vec{y} \neq \vec{x}} sim(\vec{x}, \vec{y})$$

- Compromise between single and complete link.
- Two options:
  - Averaged across all ordered pairs in the merged cluster
  - Averaged over all pairs *between* the two original clusters
- No clear difference in efficacy

Computing Group Average Similarity

- Always maintain sum of vectors in each cluster.

$$\vec{s}(c_j) = \sum_{\vec{x} \in c_j} \vec{x}$$

- Compute similarity of clusters in constant time:

$$sim(c_i, c_j) = \frac{(\vec{s}(c_i) + \vec{s}(c_j)) \bullet (\vec{s}(c_i) + \vec{s}(c_j)) - (|c_i| + |c_j|)}{(|c_i| + |c_j|)(|c_i| + |c_j| - 1)}$$

What Is A Good Clustering?

- Internal criterion: A good clustering will produce high quality clusters in which:
  - the intra-class (that is, intra-cluster) similarity is high
  - the inter-class similarity is low
  - The measured quality of a clustering depends on both the document representation and the similarity measure used

External criteria for clustering quality

- Quality measured by its ability to discover some or all of the hidden patterns or latent classes in gold standard data
- Assesses a clustering with respect to ground truth … requires *labeled data*
- Assume documents with *C* gold standard classes, while our clustering algorithms produce *K* clusters, $\omega_1, \omega_2, \ldots, \omega_K$ with n_i members.

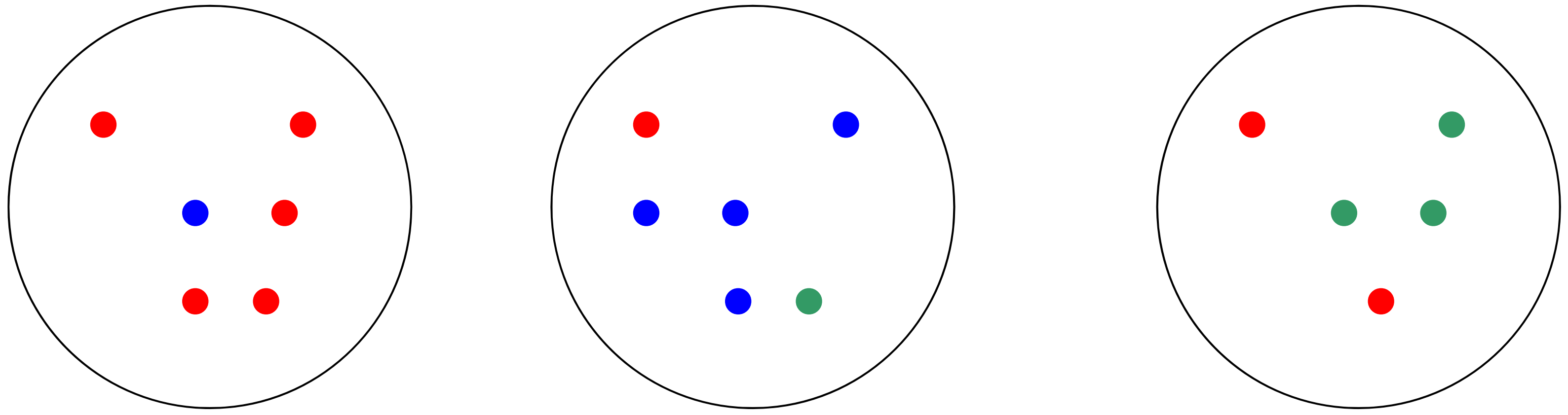
External Evaluation of Cluster Quality

- Simple measure: purity, the ratio between the dominant class in the cluster π_i and the size of cluster ω_i

$$Purity(\omega_i) = \frac{1}{n_i} \max_j (n_{ij}) \quad j \in C$$

- Biased because having *n* clusters maximizes purity
- Others are entropy of classes in clusters (or mutual information between classes and clusters)

Purity example



Cluster I   Cluster II   Cluster III

Cluster I: Purity = 1/6 (max(5, 1, 0)) = 5/6

Cluster II: Purity = 1/6 (max(1, 4, 1)) = 4/6

Cluster III: Purity = 1/5 (max(2, 0, 3)) = 3/5

Rand Index measures between pair decisions. Here RI = 0.68

Number of points	Same Cluster in clustering	Different Clusters in clustering
Same class in ground truth	20	24
Different classes in ground truth	20	72

Rand index and Cluster F-measure

$$RI = \frac{A+D}{A+B+C+D}$$

Compare with standard Precision and Recall:

$$P = \frac{A}{A+B} \qquad R = \frac{A}{A+C}$$

People also define and use a cluster F-measure, which is probably a better measure.

Final word and resources

- In clustering, clusters are inferred from the data without human input (unsupervised learning)
- However, in practice, it's a bit less clear: there are many ways of influencing the outcome of clustering: number of clusters, similarity measure, representation of documents, . . .
- Resources

Resources for today's lecture

- IIR 13 except 13.4
- IIR 14
- IIR 16 except 16.5
- IIR 17.1–17.3
  - Fabrizio Sebastiani. Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 34(1):1-47, 2002.
  - Yiming Yang & Xin Liu, A re-examination of text categorization methods. *Proceedings of SIGIR*, 1999.
  - Trevor Hastie, Robert Tibshirani and Jerome Friedman, *Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer-Verlag, New York.
  - Open Calais: Automatic Semantic Tagging
    - Free provided by Thompson/Reuters
  - Weka: A data mining software package that includes an implementation of many ML algorithms