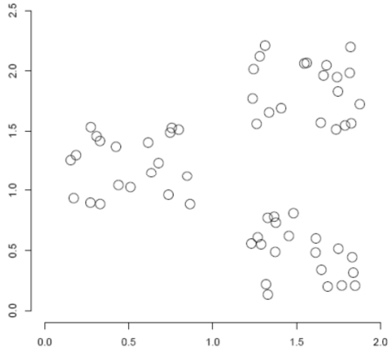
Clustering Techniques for Information Retrieval

Berlin Chen
Department of Computer Science & Information Engineering
National Taiwan Normal University

References:

1. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008. (Chapters 16 & 17)
2. *Modern Information Retrieval*, Chapters 5 & 7
3. "*A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models*," Jeff A. Bilmes, U.C. Berkeley TR-97-021

Clustering



- Place similar objects in the same group and assign dissimilar objects to different groups (typically using a distance measure, such as Euclidean distance)
  - **Word clustering**
    - Neighbor overlap: words occur with the similar left and right neighbors (such as *in* and *on*)
  - **Document clustering**
    - Documents with the similar topics or concepts are put together
- But clustering cannot give a comprehensive description of the object
  - How to label objects shown on the visual display is a difficult problem

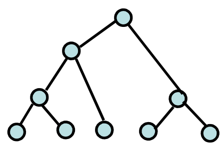
Clustering vs. Classification

- Classification is supervised and requires a set of labeled training instances for each group (class)
  - Learning with a teacher

- Clustering is unsupervised and learns without a teacher to provide the labeling information of the training data set
  - Also called automatic or unsupervised classification

Types of Clustering Algorithms

- Two types of structures produced by clustering algorithms
  - Flat or non-hierarchical clustering
  - Hierarchical clustering
- **Flat clustering**
  - Simply consisting of a certain number of clusters and the relation between clusters is often undetermined
  - Measurement: construction error minimization or probabilistic optimization
- **Hierarchical clustering**
  - A hierarchy with usual interpretation that each node stands for a subclass of its mother's node
    - The leaves of the tree are the single objects
    - Each node represents the cluster that contains all the objects of its descendants
  - Measurement: similarities of instances



Hard Assignment vs. Soft Assignment (1/2)

- Another important distinction between clustering algorithms is whether they perform soft or hard assignment

- **Hard Assignment**
  - Each object (or document in the context of IR) is assigned to one and only one cluster

- **Soft Assignment (probabilistic approach)**
  - Each object may be assigned to multiple clusters
  - An object x_i has a probability distribution $P(\cdot|x_i)$ over clusters c_j where $P(x_i|c_j)$ is the probability that x_i is a member of c_j
  - Is somewhat more appropriate in many tasks such as NLP, IR, …

Hard Assignment vs. Soft Assignment (2/2)

- Hierarchical clustering usually adopts hard assignment
- While in flat clustering, both types of assignments are common

Summarized Attributes of Clustering Algorithms (1/2)

- Hierarchical Clustering
  - Preferable for detailed data analysis
  - Provide more information than flat clustering
  - No single best algorithm (each of the algorithms only optimal for some applications)
  - Less efficient than flat clustering (minimally have to compute n x n matrix of similarity coefficients)

Summarized Attributes of Clustering Algorithms (2/2)

- Flat Clustering
  - Preferable if efficiency is a consideration or data sets are very large
  - *K*-means is the conceptually feasible method and should probably be used on a new data because its results are often sufficient
    - *K*-means assumes a simple Euclidean representation space, and so cannot be used for many data sets, e.g., nominal data like colors (or samples with features of different scales)
  - The EM algorithm is the most choice. It can accommodate definition of clusters and allocation of objects based on complex probabilistic models
    - Its extensions can be used to handle topological/hierarchical orders of samples
      - E.g., Probabilistic Latent Semantic Analysis (PLSA)

Some Applications of Clustering in IR (1/5)

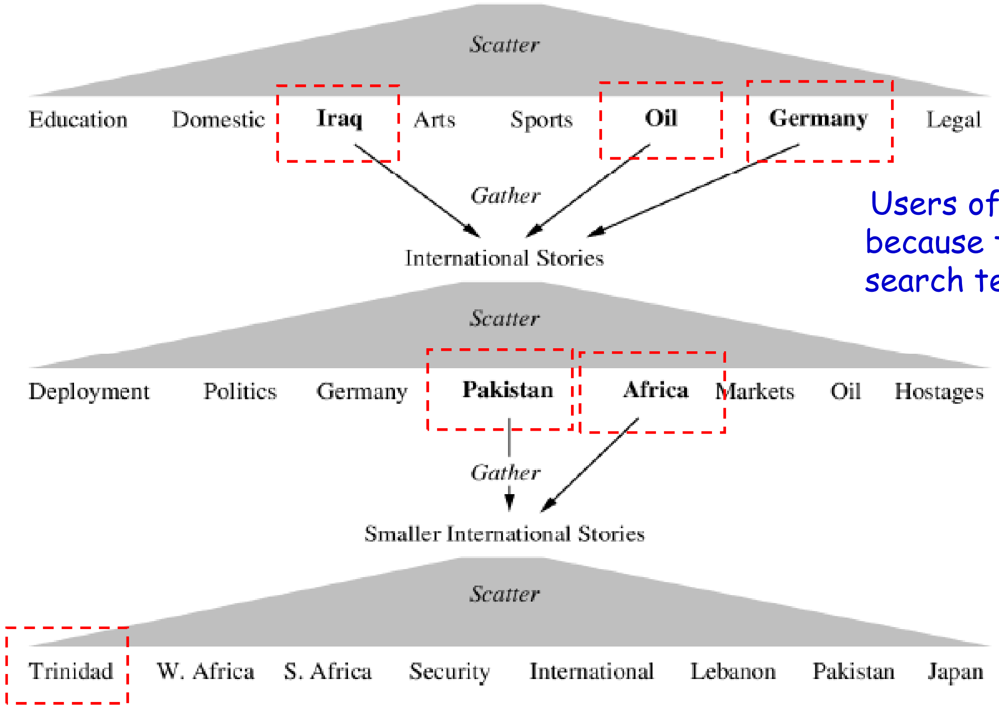
- Cluster Hypothesis (for IR): Documents in the same cluster behave similarly with respect to relevance to information needs
- Possible applications of Clustering in IR

Application	What is clustered?	Benefit	Example
Result set clustering	result set	more effective information presentation to user	Figure 16.2
Scatter-Gather	(subsets of) collection	alternative user interface: "search without typing"	Figure 16.3
Collection clustering	collection	effective information presentation for exploratory browsing	McKeown et al. (2002), http://news.google.com
Language modeling	collection	increased precision and/or recall	Liu and Croft (2004)
Cluster-based retrieval	collection	higher efficiency: faster search	Salton (1971a)

– These possible applications differ in
  - The collection of documents to be clustered
  - The aspect of the IR system to be improved

Some Applications of Clustering in IR (2/5)

1. Whole corpus analysis/navigation
   - Better user interface (users prefer browsing over searching since they are unsure about which search terms to use)
   - E.g., the *scatter-gather* approach (for a collection of New York Tin

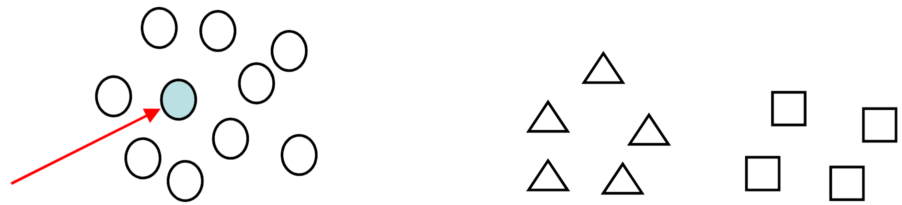


► **Figure 16.3** The Scatter-Gather user interface. A collection of New York Times news stories is clustered ("scattered") into eight clusters (top row). The user manually *gathers* three of these into a smaller collection *International Stories* and performs another scattering operation. This process repeats until a small cluster with relevant documents is found (e.g., *Trinidad*).

Users often prefer browsing over searching, because they are unsure about which search terms to use.

Some Applications of Clustering in IR (3/5)

2. Improve recall in search applications
   - Achieve better search results by
     - Alleviating the term-mismatch (synonym) problem facing the vector space model



found relevant document

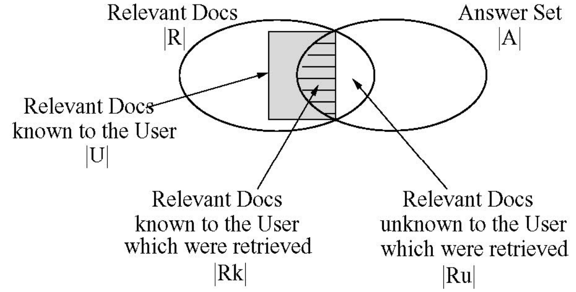
     - Estimating the collection model of the language modeling (LM) retrieval approach more accurately

$$P\left(Q\middle|\mathrm{M}_D\right)=\prod_{i=1}^{N}\left[\lambda\cdot P\left(w_i\middle|\mathrm{M}_D\right)+\left(1-\lambda\right)\cdot P\left(w_i\middle|\mathrm{M}_C\right)\right]$$

The collection model can be estimated from the cluster the document *D* belongs to, instead of the entire collection

Some Applications of Clustering in IR (4/5)

3. Better navigation of search results
   - Result set clustering
   - Effective "user recall" will be higher



http://clusty.com

Some Applications of Clustering in IR (5/5)

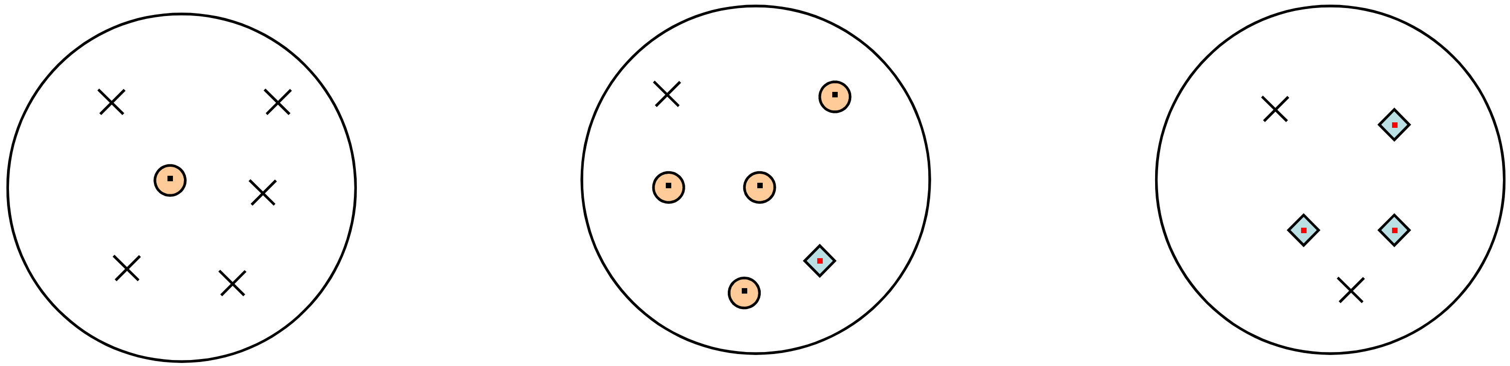
4. Speed up the search process
   - For retrieval models using exhaustive matching (computing the similarity of the query to every document) without efficient inverted index supports
     - E.g., latent semantic analysis (LSA), language modeling (LM) ?
   - Solution: cluster-based retrieval
     - First find the clusters that are closet to the query and then only consider documents from these clusters

Evaluation of Clustering (1/2)

- Internal criterion for the quality of a clustering result
  - The typical objective is to attain
    - High intra-cluster similarity (documents with a cluster are similar)
    - Low inter-cluster similarity (document from different clusters are dissimilar)
  - The measured quality depends on both the document representation and the similarity measure used
  - Good scores on an internal criterion do not necessarily translate into good effectiveness in an application

Evaluation of Clustering (2/2)

- External criterion for the quality of a clustering result
  - Evaluate how well the clustering matches the gold standard classes produced by human judges
    - That is, the quality is measured by the ability of the clustering algorithm to discover some or all of the hidden patterns or latent (true) classes



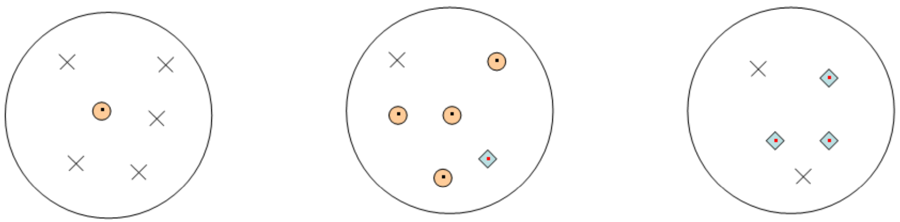
  - Two common criteria
    - Purity
    - Rand Index (RI)

Purity (1/2)

- Each cluster is first assigned to class which is most frequent in the cluster
- Then, the accuracy of the assignment is measured by counting the number of correctly assigned documents and dividing by the sample size

$$\text{Purity}\ (\Omega, \Gamma) = \frac{1}{N} \sum_{k} \max_{j} \left| \omega_j \cap c_k \right|$$

  - $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$: the set of clusters
  - $\Gamma = \{c_1, c_2, \ldots, c_J\}$: the set of classes
  - N : the sample size



$$\text{Purity}(\Omega, \Gamma) = \frac{1}{17}(5 + 4 + 3) = 0.71$$

Purity (2/2)

- High purity is easy to achieve for a large number of clusters (?)
  - Purity will be 1 if each document gets its own cluster
  - Therefore, purity cannot be used to trade off the quality of the clustering against the number of clusters

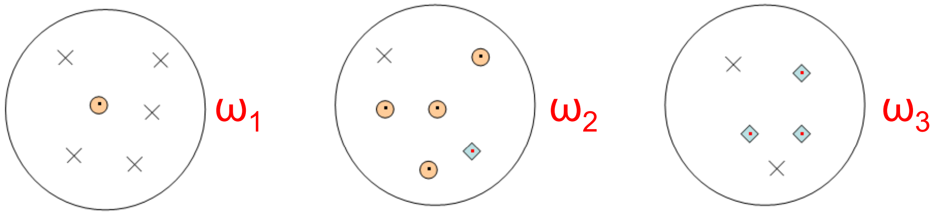
Rand Index (1/3)

- Measure the similarity between the clusters and the classes in ground truth
  – Consider the assignments of all possible $N(N-1)/2$ pairs of N distinct documents in the cluster and the true class

Number of points	Same cluster in clustering	Different clusters in clustering
Same class in ground truth	**TP (True Positive)**	**FN (False Negative)**
Different classes in ground truth	**FP (False Positive)**	**TN (True Negative)**

$$\mathrm{RI} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}}$$

Rand Index (2/3)



$$\mathrm{TP} = \underbrace{\binom{5}{2}}_{\omega_1} + \underbrace{\binom{4}{2}}_{\omega_2} + \underbrace{\left[\binom{3}{2} + \binom{2}{2}\right]}_{\omega_3} = 20$$

$$\mathrm{FP} = \underbrace{\binom{5}{1}\binom{1}{1}}_{\omega_1} + \underbrace{\left[\binom{4}{1}\binom{1}{1} + \binom{4}{1}\binom{1}{1} + \binom{1}{1}\binom{1}{1}\right]}_{\omega_2} + \underbrace{\binom{3}{1}\binom{2}{1}}_{\omega_3}$$

$$= 20$$

$$\left(= \left[\underbrace{\binom{6}{2}}_{\omega_1} + \underbrace{\binom{6}{2}}_{\omega_2} + \underbrace{\binom{5}{2}}_{\omega_3}\right] - \mathrm{TP} \right)$$

all positive pairs

$$\mathrm{TN} = \underbrace{\left[\binom{5}{1}\binom{4}{1} + \binom{5}{1}\binom{1}{1} + \binom{1}{1}\binom{1}{1} + \binom{1}{1}\binom{1}{1}\right]}_{\omega_1\omega_2} + \underbrace{\left[\binom{5}{1}\binom{3}{1} + \binom{1}{1}\binom{3}{1} + \binom{1}{1}\binom{2}{1}\right]}_{\omega_1\omega_3} + \underbrace{\left[\binom{4}{1}\binom{3}{1} + \binom{4}{1}\binom{2}{1} + \binom{1}{1}\binom{3}{1} + \binom{1}{1}\binom{2}{1}\right]}_{\omega_2\omega_3} = 72$$

$$\mathrm{FN} = \underbrace{\left[\binom{5}{1}\binom{1}{1} + \binom{1}{1}\binom{4}{1}\right]}_{\omega_1\omega_2} + \underbrace{\binom{5}{1}\binom{2}{1}}_{\omega_1\omega_3} + \underbrace{\left[\binom{1}{1}\binom{2}{1} + \binom{1}{1}\binom{3}{1}\right]}_{\omega_2\omega_3}$$

$$= 24$$

$$\left(= \left[\underbrace{\binom{6}{1}\binom{6}{1}}_{\omega_1\omega_2} + \underbrace{\binom{6}{1}\binom{5}{1}}_{\omega_1\omega_3} + \underbrace{\binom{6}{1}\binom{5}{1}}_{\omega_2\omega_3}\right] - \mathrm{TN} \right)$$

all negative pairs

	Same cluster	Different clusters
Same class	TP=20	FN=24
Different classes	FP=20	TN=72

$$\mathrm{RI} = \frac{20 + 72}{20 + 20 + 24 + 72} = 0.68$$

all pairs

$$N(N-1)/2 = 17 \times 16 / 2 = 136$$

Rand Index (3/3)

- The rand index has a value between 0 and 1
  - 0 indicates that the clusters and the classes in ground truth do not agree on any pair of points (documents)
  - 1 indicates that the clusters and the classes in ground truth are exactly the same

F-Measure Based on Rand Index

- F-Measure: harmonic mean of precision (P) and recall (R)

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \qquad R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

	Same cluster	Different clusters
Same class	TP	FN
Different classes	FP	TN

$$\mathrm{F}_b = \frac{b^2 + 1}{\frac{b^2}{R} + \frac{1}{P}} = \frac{(b^2 + 1)PR}{b^2 P + R}$$

  – If we want to penalize false negatives (FN) more strongly than false positives (FP), then we can set $b > 1$ (separating similar documents is sometimes worse than putting dissimilar documents in the same cluster)
    - That is, giving more weight to recall (R)

Normalized Mutual Information (NMI)

- NMI is an information-theoretical measure

$$\mathrm{NMI}(\Omega, \mathrm{C}) = \frac{I(\Omega; \mathrm{C})}{\left(H(\Omega) + H(C)\right)/2}$$

$$I(\Omega; \mathrm{C}) = \sum_k \sum_j p(\omega_k \cap c_j) \log \frac{p(\omega_k \cap c_j)}{p(\omega_k) p(c_j)}$$

$$= \sum_k \sum_j \frac{|\omega_k \cap c_j|}{N} \log \frac{N |\omega_k \cap c_j|}{|\omega_k| |c_j|} \quad \text{(ML estimate)}$$

$$H(\Omega) = -\sum_k p(\omega_k) \log p(\omega_k)$$

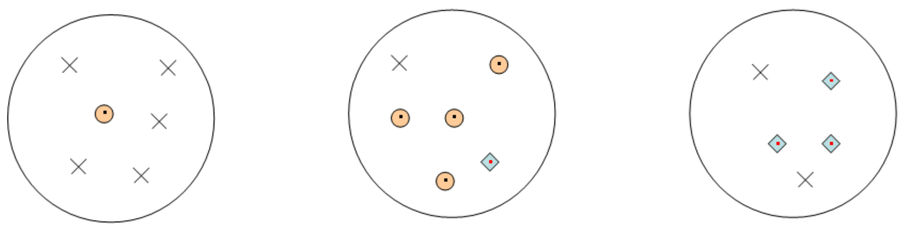
$$= -\sum_k \frac{|\omega_k|}{N} \log \frac{|\omega_k|}{N} \quad \text{(ML estimate)}$$

  - NMI will have a value between 0 and 1

Summary of External Evaluation Measures

Table 16.2 The four external evaluation measures applied to the clustering in Figure 16.4.

	purity	NMI	RI	F_5
lower bound	0.0	0.0	0.0	0.0
maximum	1.0	1.0	1.0	1.0
value for Figure 16.4	0.71	0.36	0.68	0.46



Flat Clustering

Flat Clustering

- Start out with a partition based on randomly selected seeds (one seed per cluster) and then refine the initial partition
  - In a multi-pass manner (recursion/iterations)

- **Problems** associated with non-hierarchical clustering
  - When to stop ? group average similarity, likelihood, mutual information
  - What is the right number of clusters (cluster cardinality) ?

  $k-1 \rightarrow k \rightarrow k+1$

  Hierarchical clustering also has to face this problem

- **Algorithms** introduced here
  - The *K*-means algorithm
  - The EM algorithm

The *K*-means Algorithm (1/10)

- Also called *Linde-Buzo-Gray* (LBG) in signal processing
  - A hard clustering algorithm
  - Define clusters by the **center of mass** of their members
  - Objects (e.g., documents) should be represented in vector form
- The *K*-means algorithm also can be regarded as
  - A kind of vector quantization
    - Map from a continuous space (high resolution) to a discrete space (low resolution)
  - E.g. color quantization
    - 24 bits/pixel (16 million colors) → 8 bits/pixel (256 colors)
    - A compression rate of 3

$$\boldsymbol{X} = \{\boldsymbol{x}^t\}_{t=1}^{n} \xrightarrow{\text{index } j} \boldsymbol{F} = \{\boldsymbol{m}_j\}_{j=1}^{k}$$

Dim($\boldsymbol{x}^t$)=24 → $|F|=2^8$

$\boldsymbol{m}_j$: cluster centriod or reference vector, code word, code vector

The *K*-means Algorithm (2/10)

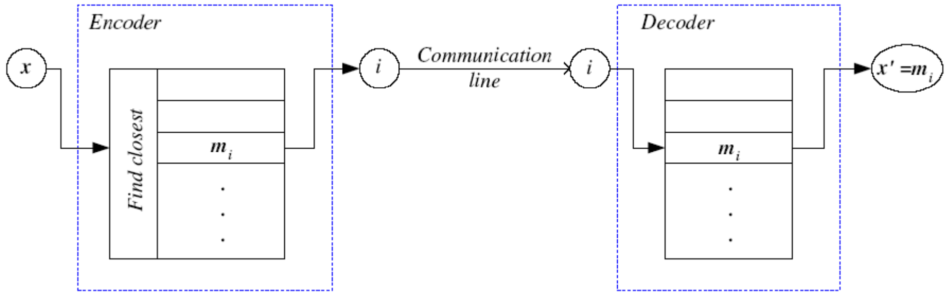


Figure 7.1: Given $\boldsymbol{x}$, the encoder sends the index of the closest code word and the decoder generates the code word with the received index as $\boldsymbol{x}'$. Error is $\|\boldsymbol{x}' - \boldsymbol{x}\|^2$.

Total reconstruction error (RSS : residual sum of squares)

automatic label

$$E\left(\{\mathbf{m}_i\}_{i=1}^{k} \middle| \mathbf{X}\right) = \sum_{t=1}^{N} \sum_{i=1}^{k} b_i^t \left\|\mathbf{x}^t - \mathbf{m}_i\right\|^2, \text{ where } b_i^t = \begin{cases} 1 & \text{if } \left\|\mathbf{x}^t - \mathbf{m}_i\right\| = \min_j \left\|\mathbf{x}^t - \mathbf{m}_j\right\| \\ 0 & \text{otherwise} \end{cases}$$

- b_i^t and $\boldsymbol{m}_i$ are unknown in advance
- b_i^t depends on $\boldsymbol{m}_i$ and this optimization problem can not be solved analytically

The *K*-means Algorithm (3/10)

- **Initialization**
  - A set of initial cluster centers is needed $\{\boldsymbol{m}_i\}_{i=1}^{k}$

- **Recursion**
  - Assign each object $\boldsymbol{x}^t$ to the cluster whose center is closest

$$b_i^t = \begin{cases} 1 & \text{if} \left\| \boldsymbol{x}^t - \boldsymbol{m}_i \right\| = \min_j \left\| \boldsymbol{x}^t - \boldsymbol{m}_j \right\| \\ 0 & \text{otherwise} \end{cases}$$

  - Then, re-compute the center of each cluster as the centroid or mean (average) of its members
    - Using the medoid as the cluster center ?

    (a medoid is one of the objects in the cluster that is closest to the centroid)

$$\boldsymbol{m}_i = \frac{\sum_{t=1}^{N} b_i^t \cdot \boldsymbol{x}^t}{\sum_{t=1}^{N} b_i^t}$$

These two steps are repeated until $\boldsymbol{m}_i$ stabilizes

The *K*-means Algorithm (4/10)

- Algorithm

Initialize $\boldsymbol{m}_i, i = 1, \ldots, k$, for example, to k random $\boldsymbol{x}^t$

Repeat

  For all $\boldsymbol{x}^t \in \mathcal{X}$

$$b_i^t \leftarrow \begin{cases} 1 & \text{if } \|\boldsymbol{x}^t - \boldsymbol{m}_i\| = \min_j \|\boldsymbol{x}^t - \boldsymbol{m}_j\| \\ 0 & \text{otherwise} \end{cases}$$

  For all $\boldsymbol{m}_i, i = 1, \ldots, k$

$$\boldsymbol{m}_i \leftarrow \sum_t b_i^t \boldsymbol{x}^t / \sum_t b_i^t$$

Until $\boldsymbol{m}_i$ converge

The *K*-means Algorithm (5/10)

- Example 1

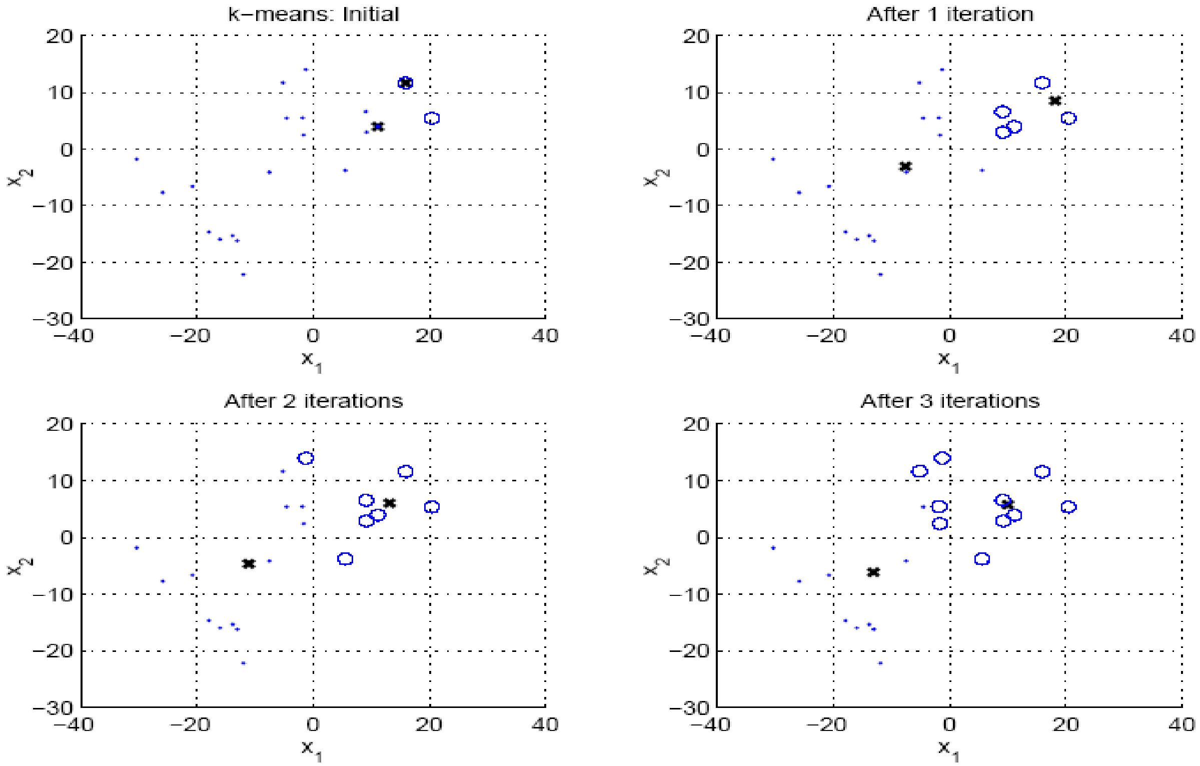


Figure 7.2: Evolution of k-means. Crosses indicate center positions. Data points are marked depending on the closest center.

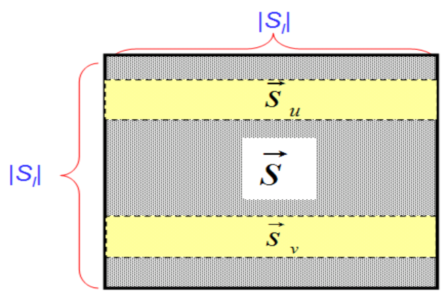
The *K*-means Algorithm (6/10)

- Example 2

Cluster	Members	
1	*ballot* (0.28), *polls* (0.28), *Gov* (0.30), *seats* (0.32)	government
2	*profit* (0.21), *finance* (0.21), *payments* (0.22)	finance
3	*NFL* (0.36), *Reds* (0.28), *Sox* (0.31), *inning* (0.33), *quarterback* (0.30), *scored* (0.30), *score* (0.33)	sports
4	*researchers* (0.23), *science* (0.23)	research
5	*Scott* (0.28), *Mary* (0.27), *Barbara* (0.27), *Edward* (0.29)	name

Table 14.4 An example of K-means clustering. Twenty words represented as vectors of co-occurrence counts were clustered into 5 clusters using K-means. The distance from the cluster centroid is given after each word.

$$s_{u,v} = \frac{\vec{s}_u \cdot \vec{s}_v}{|\vec{s}_u| \times |\vec{s}_u|}$$

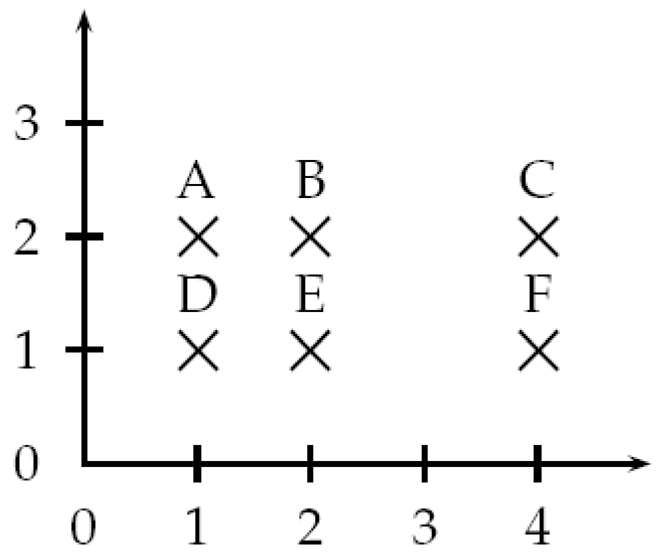


The *K*-means Algorithm (7/10)

- Complexity: O(*IKNM*)
  - I: Iterations; K: cluster number; N: object number; M: object dimensionality
- Choice of initial cluster centers (seeds) is important
  - Pick at random
  - Or, calculate the mean $\mathbf{m}$ of all data and generate *k* initial centers $\boldsymbol{m}_i$ by adding small random vector to the mean $\mathbf{m} \pm \boldsymbol{\delta}$
  - Or, project data onto the principal component (first eigenvector), divide it range into *k* equal interval, and take the mean of data in each group as the initial center $\boldsymbol{m}_i$
  - Or, use another method such as hierarchical clustering algorithm on a subset of the objects
    - E.g., buckshot algorithm uses the group-average agglomerative clustering to randomly sample of the data that has size square root of the complete set

The *K*-means Algorithm (8/10)

- Poor seeds will result in **sub-optimal** clustering

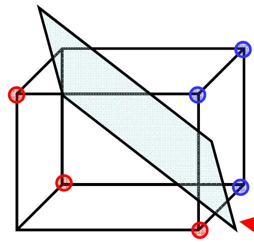


► **Figure 16.7** The outcome of clustering in k-means depends on the initial seeds. For seeds B and E, k-means converges to $\{A, B, C\}, \{D, E, F\}$, a suboptimal clustering. For seeds D and F, it converges to $\{A, B, D, E\}, \{C, F\}$, the global optimum for $K = 2$.

The *K*-means Algorithm (9/10)

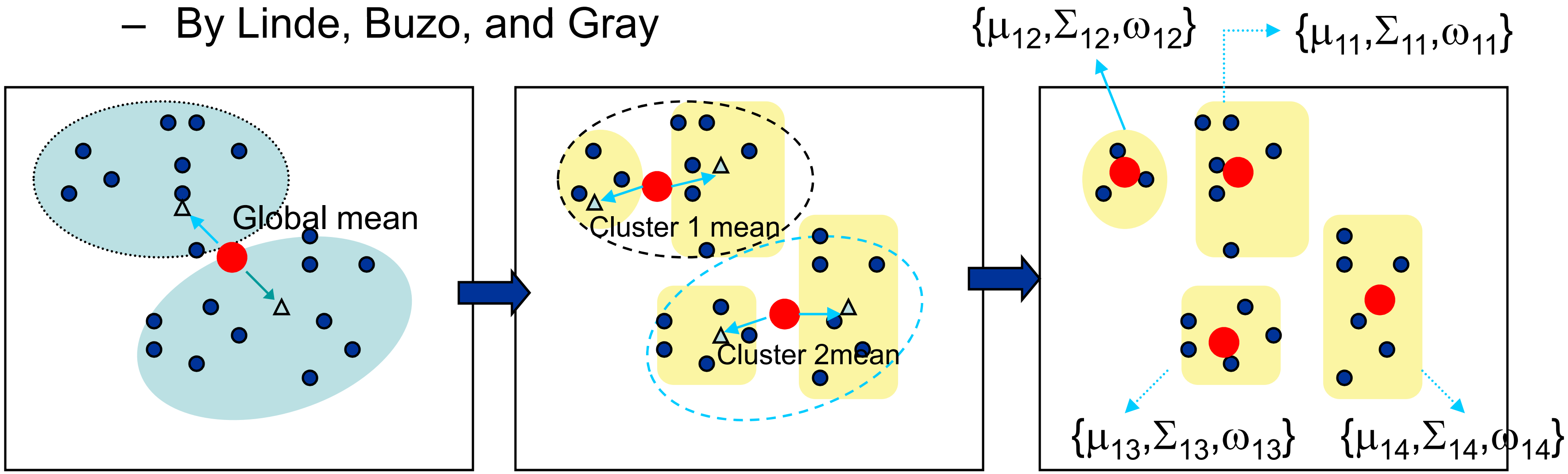
- How to break ties when in case there are several centers with the same distance from an object
  - E.g., randomly assign the object to one of the candidate clusters (or assign the object to the cluster with lowest index)
  - Or, perturb objects slightly
- Applications of the *K*-means Algorithm
  - Clustering
  - Vector quantization
  - A preprocessing stage before classification or regression
    - Map from the original space to *l*-dimensional space/hypercube

$l=\log_2 k$ (k clusters)



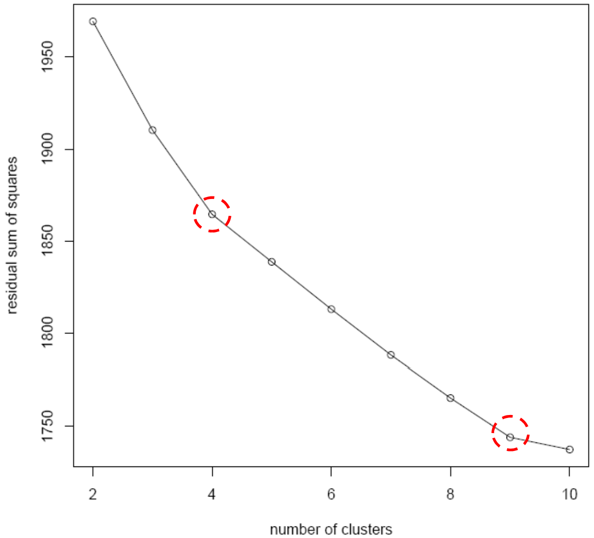
The *K*-means Algorithm (10/10)

- E.g., the LBG algorithm $M \rightarrow 2M$ at each iteration
  - By Linde, Buzo, and Gray



Total Reconstruction error
(residual sum of squares)

$$E\left(\{\mathbf{m}_i\}_{i=1}^{k} \middle| \mathbf{X}\right) = \sum_{t=1}^{N}\sum_{i=1}^{k} b_i^t \left\|\mathbf{x}^t - \mathbf{m}_i\right\|^2$$

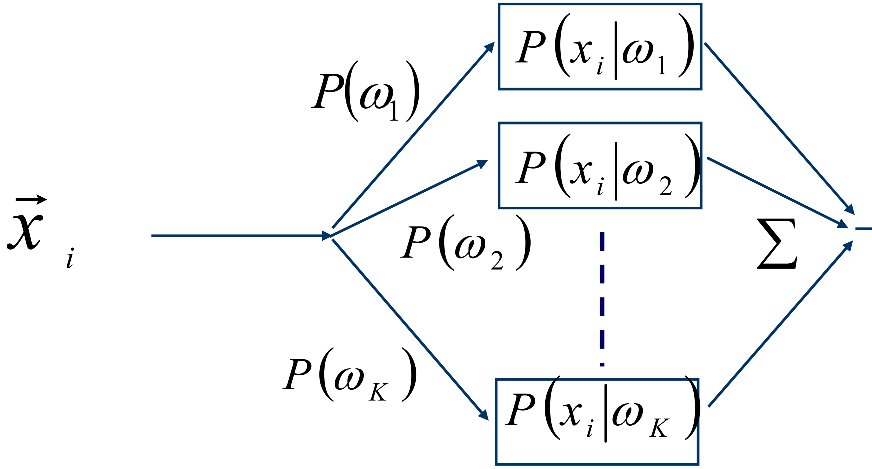


The EM Algorithm (1/3)

- EM (Expectation-Maximization) algorithm
  - A kind of model-based clustering
  - Also can be viewed as a generalization of *K*-means
  - Each cluster is a “model” for generating the data
    - The centroid is good representative for each model
    - Generate an object (e.g., document) consists of first picking a centroid at random and then adding some noise
      - If the noise is normally distributed, the procedure will result in clusters of spherical shape
- Physical Models for EM
  - Discrete: Mixture of multinomial distributions
  - Continuous: Mixture of Gaussian distributions

The EM Algorithm (2/3)

- EM is a **soft version** of *K*-mean
  - Each object could be the member of multiple clusters ω_k
  - Clustering as estimating a mixture of (continuous) probability distributions



A Mixture Gaussian HMM
(or A Mixture of Gaussians)

$$P(\vec{x}_i|\Theta)=\sum_{k=1}^{K} P(\vec{x}_i|\omega_k;\Theta)P(\omega_k|\Theta)$$

classification :

$$\max_k P(\omega_k|\vec{x}_i,\Theta)=\max_k \frac{P(\vec{x}_i|\omega_k;\Theta)P(\omega_k|\Theta)}{P(\vec{x}_i|\Theta)}$$

$$=\max_k P(\vec{x}_i|\omega_k;\Theta)P(\omega_k|\Theta)$$

Likelihood function for data samples: $\boldsymbol{X}=\vec{x}_1,\vec{x}_2,\ldots,\vec{x}_n$

Continuous case:

$$P(\vec{x}_i|\omega_k;\Theta)=\frac{1}{\sqrt{(2\pi)^m|\Sigma_k|}}\exp\left(-\frac{1}{2}(\vec{x}_i-\vec{\mu}_k)^T\Sigma_k^{-1}(\vec{x}_i-\vec{\mu}_k)\right)$$

$$P(\mathbf{X}|\Theta)=\prod_{i=1}^{n} P(\vec{x}_i|\Theta)$$

$$=\prod_{i=1}^{n}\sum_{k=1}^{K} P(\vec{x}_i|\omega_k;\Theta)P(\omega_k|\Theta)$$

$\boldsymbol{X}=\{\vec{x}_1,\vec{x}_2,\cdots,\vec{x}_n\}$

$\vec{x}_i$'s are independent identically distributed (i.i.d.)

The EM Algorithm (2/3)

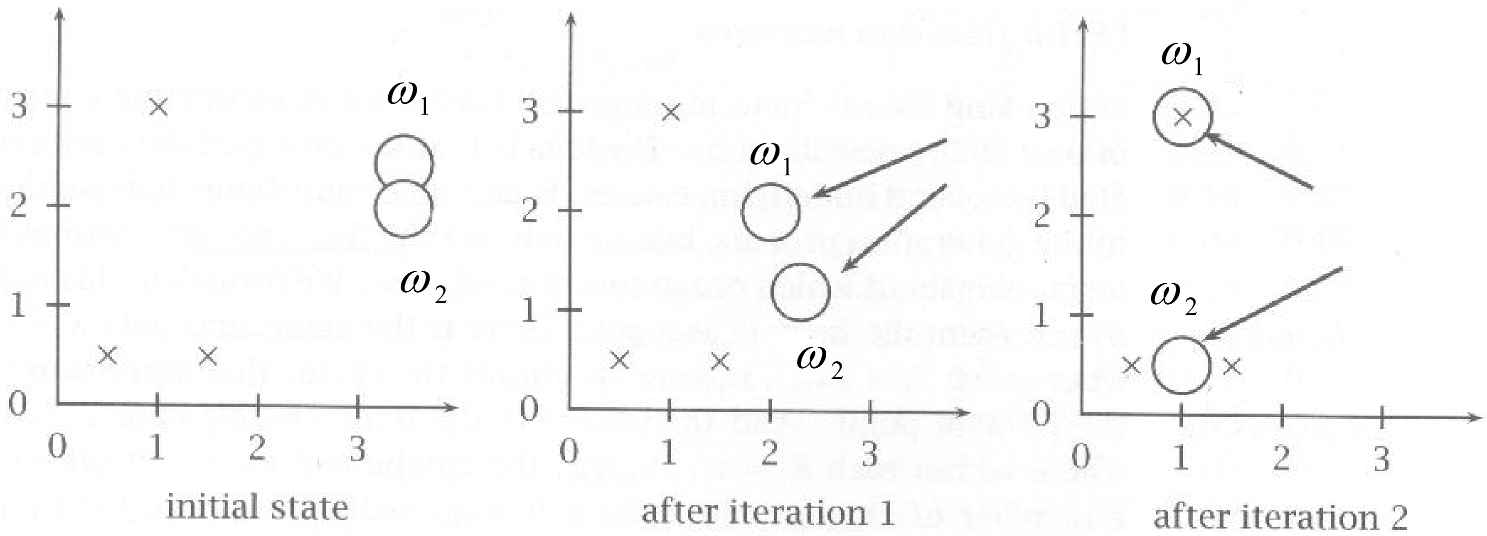
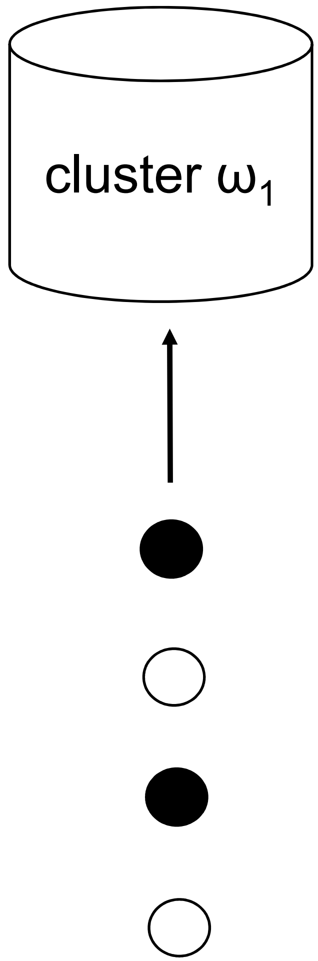


Figure 14.10 An example of using the EM algorithm for soft clustering.

Maximum Likelihood Estimation (MLE) (1/2)

- Hard Assignment



$P(B|\ \omega_1)=2/4=0.5$

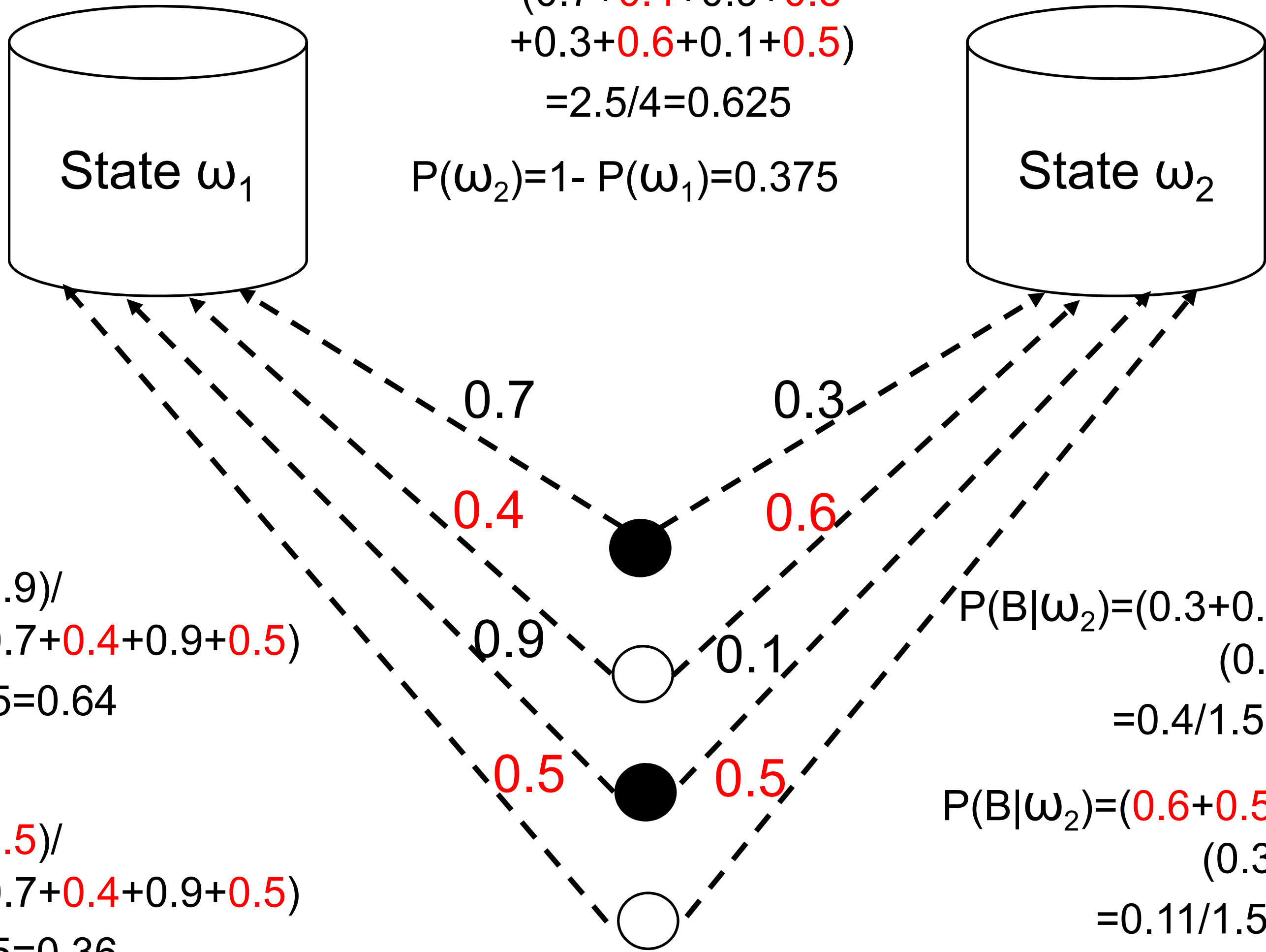
$P(W|\ \omega_1)=2/4=0.5$

Maximum Likelihood Estimation (2/2)

- Soft Assignment

$P(\omega_1)=(0.7+0.4+0.9+0.5)/$
$(0.7+0.4+0.9+0.5+0.3+0.6+0.1+0.5)$
$=2.5/4=0.625$

$P(\omega_2)=1- P(\omega_1)=0.375$



$P(B|\omega_1)=(0.7+0.9)/$
$(0.7+0.4+0.9+0.5)$
$=1.6/2.5=0.64$

$P(B|\omega_1)=(0.4+0.5)/$
$(0.7+0.4+0.9+0.5)$
$=0.9/2.5=0.36$

$P(B|\omega_2)=(0.3+0.1)/$
$(0.3+0.6+0.1+0.5)$
$=0.4/1.5=0.27$

$P(B|\omega_2)=(0.6+0.5)/$
$(0.3+0.6+0.1+0.5)$
$=0.11/1.5=0.73$

Expectation-Maximization Updating Formulas (1/3)

- Expectation

$$\gamma_{ik} = \frac{P\left(\vec{x}_i \middle| \omega_k, \Theta\right) P\left(\omega_k \middle| \Theta\right)}{\sum_{l=1}^{K} P\left(\vec{x}_i \middle| \omega_l, \Theta\right) P\left(\omega_l \middle| \Theta\right)}$$

  - Compute the likelihood that each cluster ω_k generates a document vector $\vec{x}_i$

Expectation-Maximization Updating Formulas (2/3)

- Maximization
  - Mixture Weight

$$P\left(\omega_k \middle| \hat{\Theta}\right) = \frac{\sum_{i=1}^{n} \gamma_{ik}}{\sum_{k'=1}^{K} \sum_{i=1}^{n} \gamma_{ik'}} = \frac{\sum_{i=1}^{n} \gamma_{ik}}{n}$$

  - Mean of Gaussian

$$\hat{\vec{\mu}}_k = \frac{\sum_{i=1}^{n} \gamma_{ik} \cdot \vec{x}_i}{\sum_{i'=1}^{n} \gamma_{i'k}}$$

Expectation-Maximization Updating Formulas (3/3)

- Covariance Matrix of Gaussian

$$\hat{\Sigma}_k = \frac{\sum_{i=1}^{n} \gamma_{ik} \cdot \left(\vec{x}_i - \hat{\vec{\mu}}_k\right)\left(\vec{x}_i - \hat{\vec{\mu}}_k\right)^T}{\sum_{i'=1}^{n} \gamma_{i'k}}$$

$$= \frac{\sum_{i=1}^{n} \gamma_{ik} \cdot \left(\vec{x}_i - \hat{\vec{\mu}}_k\right)\left(\vec{x}_i - \hat{\vec{\mu}}_k\right)^T}{\sum_{i'=1}^{n} \gamma_{i'k}}$$

More facts about The EM Algorithm

- The initial cluster distributions can be estimated using the *K*-means algorithm, which EM can then "soften up"
- The procedure terminates when the likelihood function $P(X|\Theta)$ is converged or maximum number of iterations is reached

Hierarchical Clustering

Hierarchical Clustering

- Can be in either bottom-up or top-down manners
  - **Bottom-up** (agglomerative) 凝集的
    - Start with individual objects and grouping the most similar ones
      - E.g., with the minimum distance apart

$$sim(x, y) = \frac{1}{1 + d(x, y)}$$

distance measures will be discussed later on

    - The procedure terminates when one cluster containing all objects has been formed
  - **Top-down** (divisive) 分裂的
    - Start with all objects in a group and divide them into groups so as to maximize within-group similarity

Hierarchical Agglomerative Clustering (HAC)

- A bottom-up approach
- Assume a similarity measure for determining the similarity of two objects
- Start with all objects in a separate cluster (a singleton) and then repeatedly joins the two clusters that have the most similarity until there is one only cluster survived
- The history of merging/clustering forms a binary tree or hierarchy

HAC: Algorithm

Given: a set $\mathcal{X} = \{x_1, \ldots x_n\}$ of objects
          a function sim: $\mathcal{P}(\mathcal{X}) \times \mathcal{P}(\mathcal{X}) \rightarrow \mathbb{R}$
for $i := 1$ **to** n **do**
    $c_i := \{x_i\}$ **end**

Initialization (for tree leaves): Each object is a cluster

$C := \{c_1, \ldots, c_n\}$
$j := n + 1$
while $|C| > 1$

cluster number

    $(c_{n_1}, c_{n_2}) := \arg\max_{(c_u, c_v) \in C \times C} \text{sim}(c_u, c_v)$
    $c_j = c_{n_1} \cup c_{n_2}$

merged as a new cluster

    $C := C \backslash \{c_{n_1}, c_{n_2}\} \cup \{c_j\}$
    $j := j + 1$

The original two clusters are removed

Figure 14.2 Bottom-up hierarchical clustering.

- c_i denotes a specific cluster here

Distance Metrics

- Euclidian Distance (L_2 norm)

$$L_2(\vec{x},\vec{y}) = \sum_{i=1}^{m} (x_i - y_i)^2$$

  - Make sure that all attributes/dimensions have the same scale (or the same variance)

- L_1 Norm (City-block distance)

$$L_1(\vec{x},\vec{y}) = \sum_{i=1}^{m} \left| x_i - y_i \right|$$

- Cosine Similarity (transform to a distance by subtracting from 1)

$$1 - \frac{\vec{x} \bullet \vec{y}}{|\vec{x}| \cdot |\vec{y}|}$$

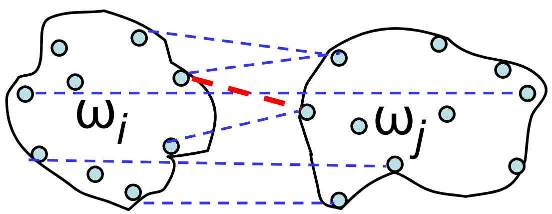
ranged between 0 and 1

Measures of Cluster Similarity (1/9)

- Especially for the bottom-up approaches

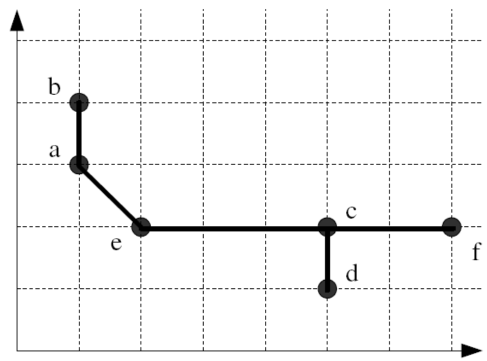
1. **Single-link** clustering
   - The similarity between two clusters is the similarity of the two closest objects in the clusters
   - Search over all pairs of objects that are from the two different clusters and select the pair with the greatest similarity
   - Elongated clusters are achieved

$$sim\left(\omega_i, \omega_j\right) = \max_{\vec{x} \in \omega_i, \vec{y} \in \omega_j} sim\left(\vec{x}, \vec{y}\right)$$



greatest similarity

cf. the minimal spanning tree



Measures of Cluster Similarity (2/9)

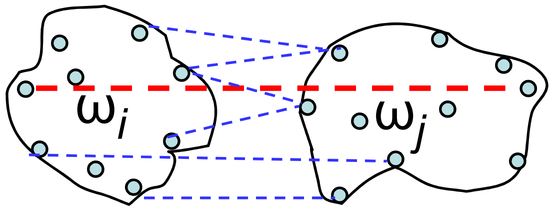
2. **Complete-link** clustering

– The similarity between two clusters is the similarity of their two most dissimilar members

– Sphere-shaped clusters are achieved

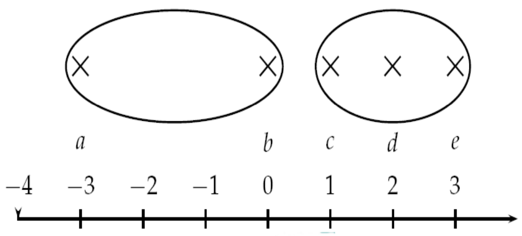
– Preferable for most IR and NLP applications

$$sim(\omega_i, \omega_j) = \min_{\vec{x} \in \omega_i, \vec{y} \in \omega_j} sim(\vec{x}, \vec{y})$$



least similarity

– More sensitive to outliers



▶ **Figure 17.6** Outliers in complete-link clustering. The four points have the coordinates $-3 + 2 \times \epsilon, 0, 1 + 2 \times \epsilon, 2$ and $3 - \epsilon$. Complete-link clustering creates the two clusters shown as ellipses. Intuitively, $\{b, c, d, e\}$ should be one cluster, but it is split by outlier a.

Measures of Cluster Similarity (3/9)

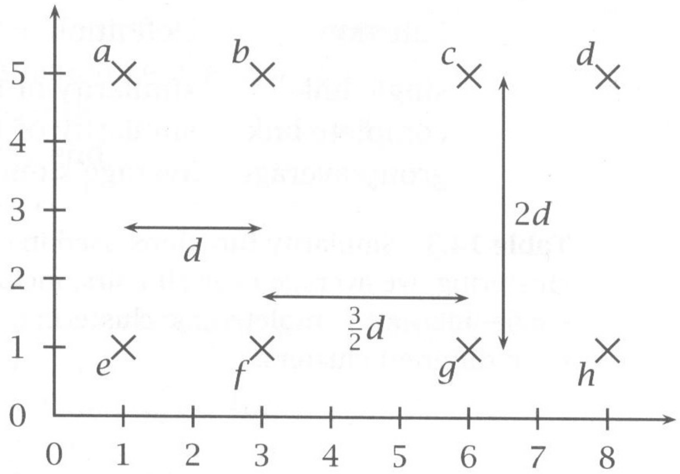


Figure 14.4 A cloud of points in a plane.

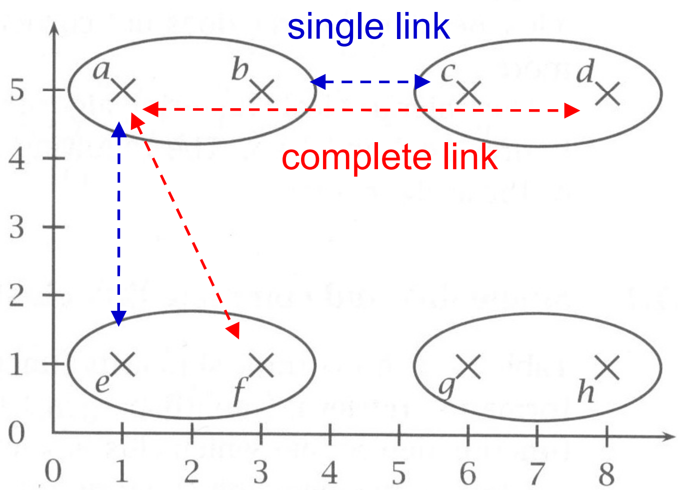


Figure 14.5 Intermediate clustering of the points in figure 14.4.

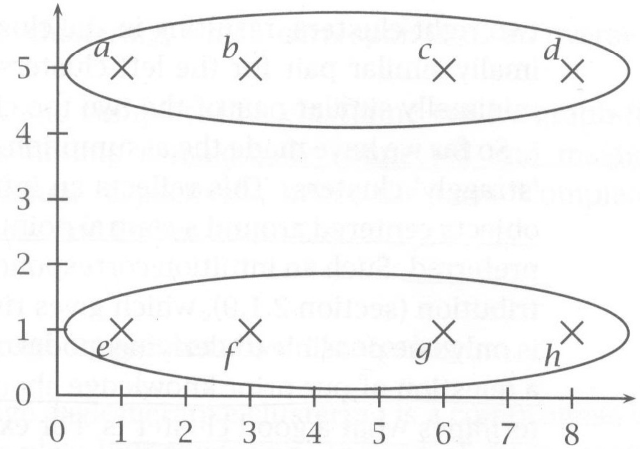


Figure 14.6 Single-link clustering of the points in figure 14.4.

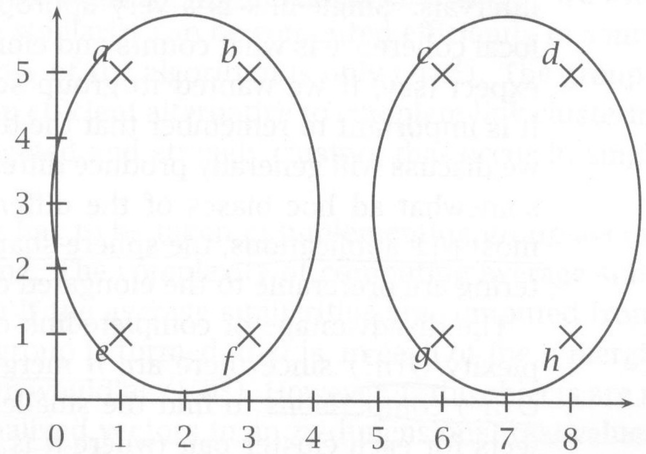
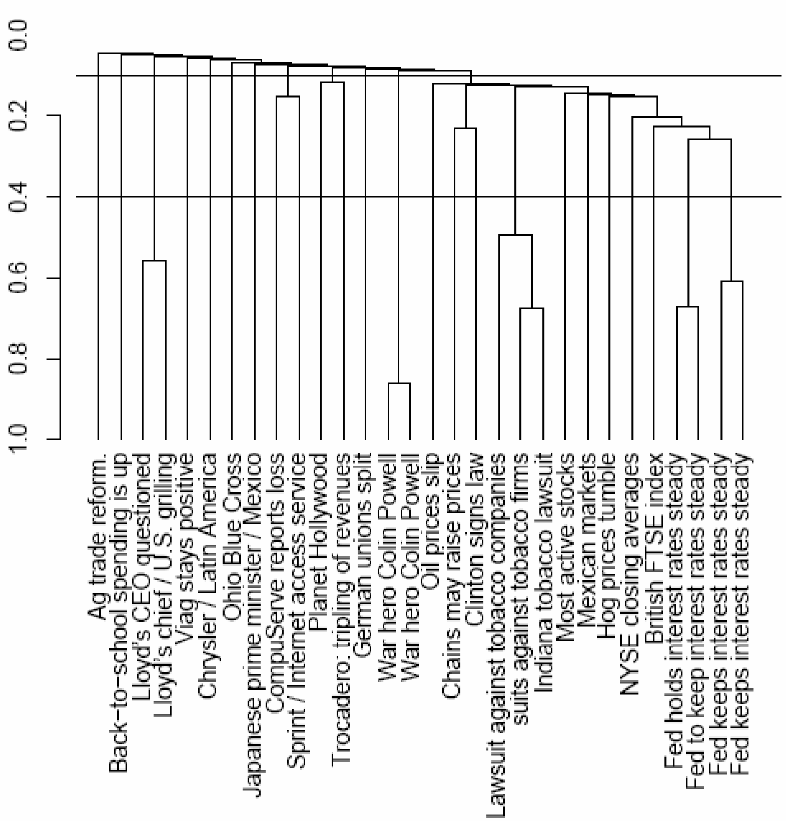
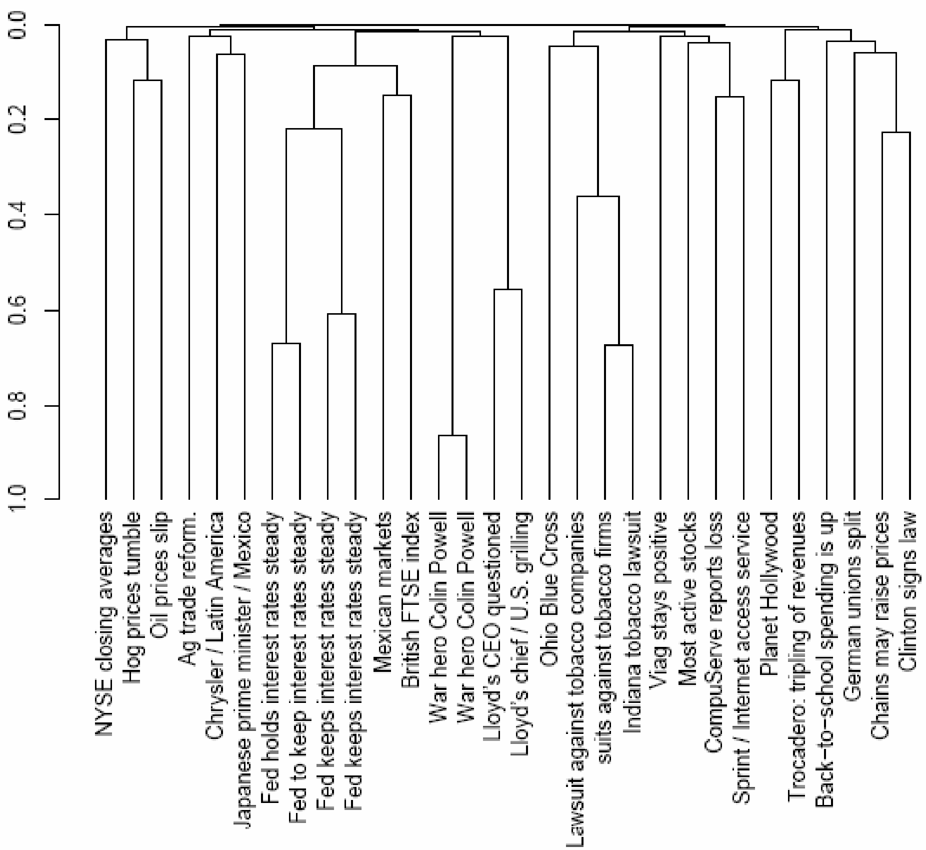


Figure 14.7 Complete-link clustering of the points in figure 14.4.

Measures of Cluster Similarity (4/9)



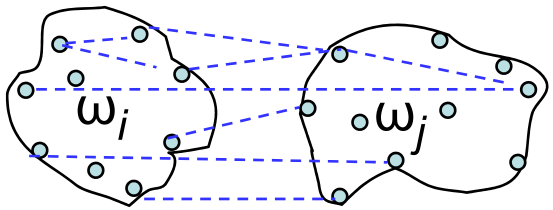
► **Figure 17.1** A dendrogram of a single-link clustering of 30 documents from Reuters-RCV1. The y-axis represents combination similarity, the similarity of the two component clusters that gave rise to the corresponding merge. For example, the combination similarity of *Lloyd's CEO questioned* and *Lloyd's chief / U.S. grilling* is ≈ 0.56. Two possible cuts of the dendrogram are shown: at 0.4 into 24 clusters and at 0.1 into 12 clusters.



► **Figure 17.4** A dendrogram of a complete-link clustering of 30 documents from Reuters-RCV1. This complete-link clustering is more balanced than the single-link clustering of the same documents in Figure 17.1. When cutting the last merger, we obtain two clusters of similar size (documents 1–16 and documents 17–30). The y-axis represents combination similarity.

Measures of Cluster Similarity (5/9)

3. **Group-average** agglomerative clustering
   - A compromise between single-link and complete-link clustering
   - The similarity between two clusters is the average similarity between members



   - If the objects are represented as length-normalized vectors and the similarity measure is the cosine
     - There exists an fast algorithm for computing the average similarity

$$sim(\vec{x}, \vec{y}) = \cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}||\vec{y}|} = \vec{x} \cdot \vec{y}$$

length-normalized vectors

Measures of Cluster Similarity (6/9)

3. **Group-average** agglomerative clustering (cont.)

- The average similarity *SIM* between vectors in a cluster ω_j is defined as

$$SIM(\omega_j)=\frac{1}{|\omega_j|(|\omega_j|-1)}\sum_{\vec{x}\in\omega_j}\sum_{\substack{\vec{y}\in\omega_j\\ \vec{y}\neq\vec{x}}} sim(\vec{x},\vec{y})=\frac{1}{|\omega_j|(|\omega_j|-1)}\sum_{\vec{x}\in\omega_j}\sum_{\substack{\vec{y}\in\omega_j\\ \vec{y}\neq\vec{x}}}\vec{x}\cdot\vec{y}$$

- The sum of members in a cluster ω_j : $\vec{s}(\omega_j)=\sum_{\vec{x}\in\omega_j}\vec{x}$
- Express $SIM(\omega_j)$ in terms of $\vec{s}(\omega_j)$

$$\vec{s}(\omega_j)\cdot\vec{s}(\omega_j)=\sum_{\vec{x}\in\omega_j}\vec{x}\cdot\vec{s}(\omega_j)=\sum_{\vec{x}\in\omega_j}\sum_{\vec{y}\in\omega_j}\vec{x}\cdot\vec{y}$$

length-normalized vector

$$=|\omega_j|(|\omega_j|-1)SIM(\omega_j)+\sum_{\vec{x}\in c_j}\boxed{\vec{x}\cdot\vec{x}}$$

=1

$$=|\omega_j|(|\omega_j|-1)SIM(\omega_j)+|\omega_j|$$

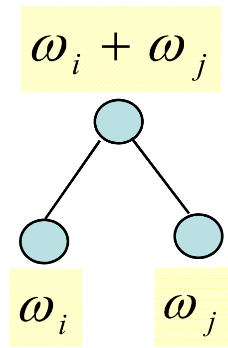
$$\therefore\ SIM(c_j)=\frac{\vec{s}(\omega_j)\cdot\vec{s}(\omega_j)-|\omega_j|}{|\omega_j|(|\omega_j|-1)}$$

Measures of Cluster Similarity (7/9)

3. **Group-average** agglomerative clustering (cont.)

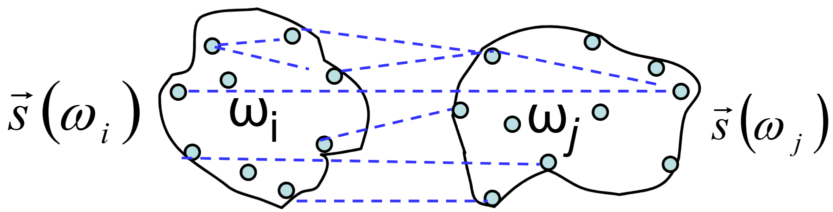
-As merging two clusters c_i and c_j, the cluster sum vectors $\vec{s}(\omega_i)$ and $\vec{s}(\omega_j)$ are known in advance

$$\Rightarrow \vec{s}(\omega_{New}) = \vec{s}(\omega_i) + \vec{s}(\omega_j), \quad |\omega_{New}| = |\omega_i| + |\omega_j|$$



– The average similarity for their union will be

$$SIM(\omega_i \cup \omega_j) = \frac{(\vec{s}(\omega_i) + \vec{s}(\omega_j)) \cdot (\vec{s}(\omega_i) + \vec{s}(\omega_j)) - (|\omega_i| + |\omega_j|)}{(|\omega_i| + |\omega_j|)(|\omega_i| + |\omega_j| - 1)}$$



Measures of Cluster Similarity (8/9)

4. **Centroid** clustering
   - The similarity of two clusters is defined as the similarity of their centroids

$$
\begin{aligned}
sim(\omega_i, \omega_j) &= \vec{\mu}(\omega_i) \cdot \vec{\mu}(\omega_j) \\
&= \left(\frac{1}{N_i} \sum_{\vec{x}_s \in \omega_i} \vec{x}_s \right) \cdot \left(\frac{1}{N_j} \sum_{\vec{x}_t \in \omega_j} \vec{x}_t \right) \\
&= \frac{1}{N_i N_j} \sum_{\vec{x}_s \in \omega_i} \sum_{\vec{x}_t \in \omega_j} \vec{x}_s \cdot \vec{x}_t
\end{aligned}
$$

Measures of Cluster Similarity (9/9)

- Graphical summary of four cluster similarity measures

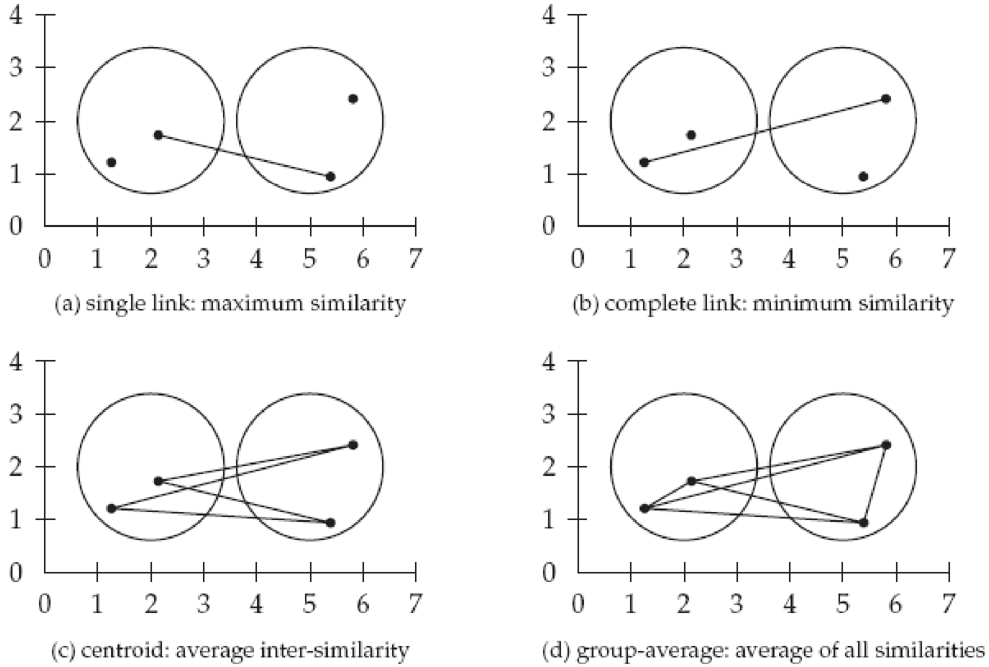


Figure 17.3 The different notions of cluster similarity used by the four HAC algorithms. An *inter-similarity* is a similarity between two documents from different clusters.

clustering algorithm	$\text{SIM}(i, k_1, k_2)$
single-link	$\max(\text{SIM}(i, k_1), \text{SIM}(i, k_2))$
complete-link	$\min(\text{SIM}(i, k_1), \text{SIM}(i, k_2))$
centroid	$(\frac{1}{N_m}\vec{v}_m) \cdot (\frac{1}{N_i}\vec{v}_i)$
group-average	$\frac{1}{(N_m+N_i)(N_m+N_i-1)}[(\vec{v}_m + \vec{v}_i)^2 - (N_m + N_i)]$

Example: Word Clustering

- Words (objects) are described and clustered using a set of features and values
  - E.g., the left and right neighbors of tokens of words

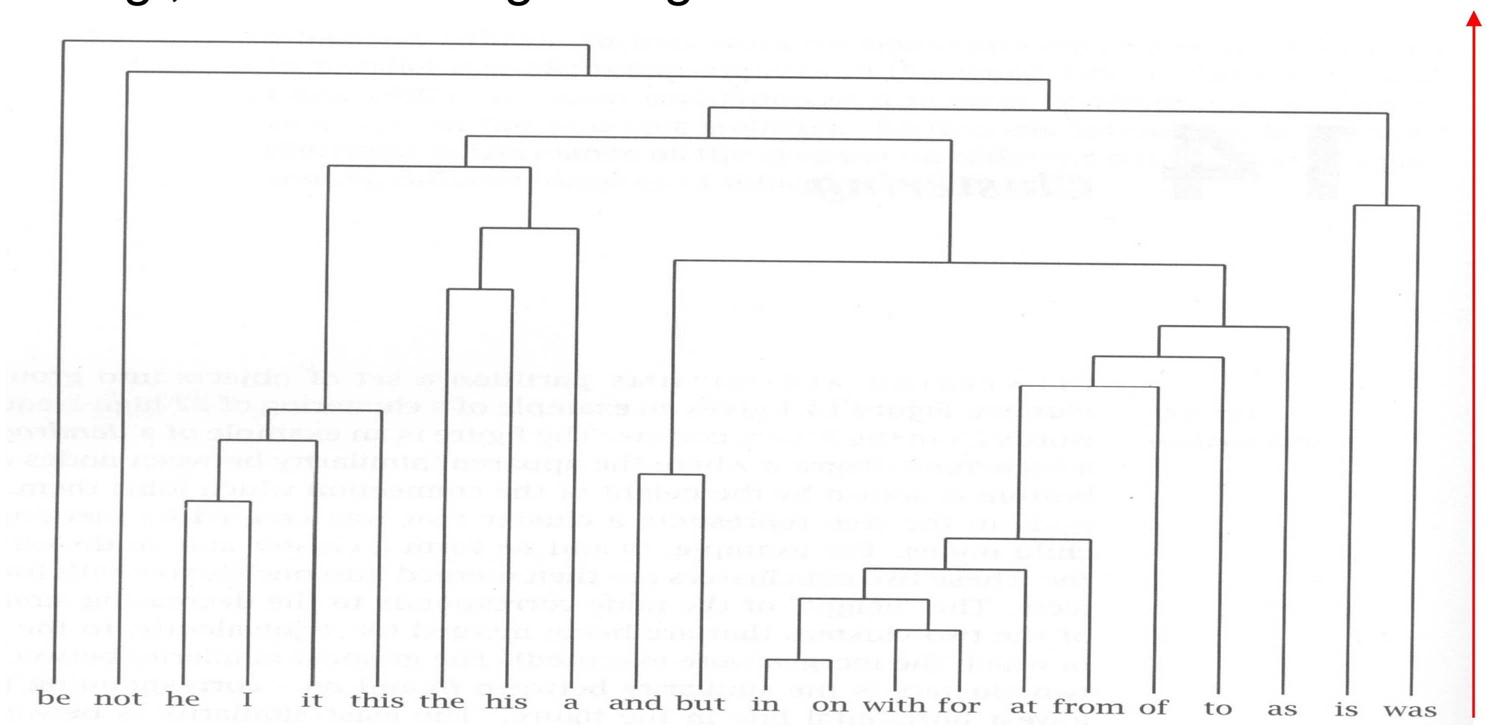


Figure 14.1 A single-link clustering of 22 frequent English words represented as a dendrogram.

higher nodes: decreasing of similarity

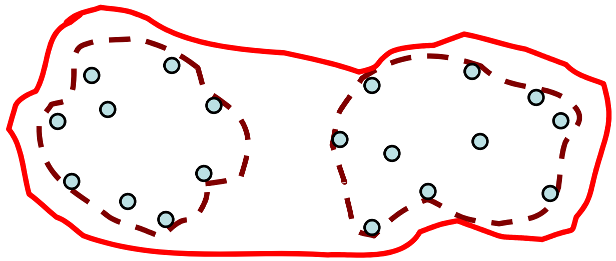
"be" has least similarity with the other 21 words !

Divisive Clustering (1/2)

- A top-down approach
- Start with all objects in a single cluster
- At each iteration, select the least **coherent** cluster and **split** it
- Continue the iterations until a predefined criterion (e.g., the cluster number) is achieved
- The history of clustering forms a binary tree or hierarchy

Divisive Clustering (2/2)

- To select the least coherent cluster, the measures used in bottom-up clustering (e.g. HAC) can be used again here
  - Single link measure
  - Complete-link measure
  - Group-average measure



- How to split a cluster
  - Also is a clustering task (finding two sub-clusters)
  - Any clustering algorithm can be used for the splitting operation, e.g.,
    - Bottom-up (agglomerative) algorithms
    - Non-hierarchical clustering algorithms (e.g., *K*-means)

Divisive Clustering: Algorithm

Given: a set $\mathcal{X} = \{x_1, \ldots x_n\}$ of objects
a function coh: $\mathcal{P}(\mathcal{X}) \rightarrow \mathbb{R}$
a function split: $\mathcal{P}(\mathcal{X}) \rightarrow \mathcal{P}(\mathcal{X}) \times \mathcal{P}(\mathcal{X})$
$C := \{\mathcal{X}\} \quad (= \{c_1\})$
$j := 1$
while $\exists c_i \in C$ s.t. $|c_i| > 1$
$\quad c_u := \arg\min_{c_v \in C} \mathrm{coh}(c_v)$
$\quad (c_{j+1}, c_{j+2}) := \mathrm{split}(c_u)$
$\quad C := C \backslash \{c_u\} \cup \{c_{j+1}, c_{j+2}\}$
$\quad j := j + 2$

split the least coherent cluster

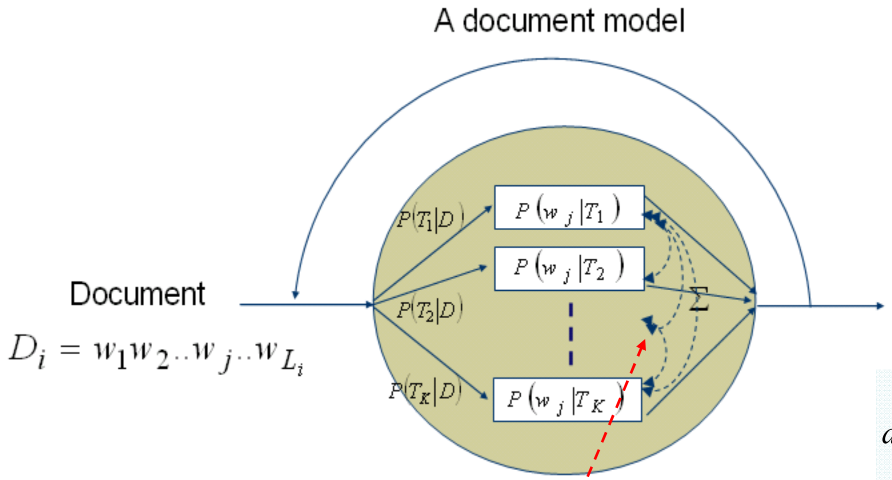
Generate two new clusters and remove the original one

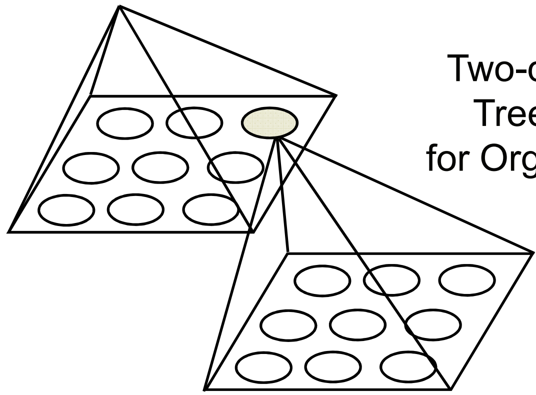
Figure 14.3 Top-down hierarchical clustering.

- c_u denotes a specific cluster here

Hierarchical Document Organization (1/7)

- Explore the Probabilistic Latent Topical Information
  - TMM/PLSA approach





$$P(w_j|D_i) = \sum_{k=1}^{K} P(T_k|D_i)\left[\sum_{l=1}^{K} \boxed{P(T_l|Y_k)}\, P(w_j|T_l)\right]$$

$$dist(T_i,T_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \qquad E(T_l,T_k) = \frac{1}{\sqrt{2\pi}\,\sigma}\exp\left[-\frac{dist(T_k,T_l)^2}{2\sigma^2}\right]$$

$$P(T_l|Y_k) = \frac{E(T_l,T_k)}{\sum_{s=1}^{K} E(T_s,T_k)}$$

- Documents are clustered by the latent topics and organized in a two-dimensional tree structure, or a two-layer map
- Those related documents are in the same cluster and the relationships among the clusters have to do with the distance on the map
- When a cluster has many documents, we can further analyze it into an other map on the next layer

Hierarchical Document Organization (2/7)

- The model can be trained by maximizing the total log-likelihood of all terms observed in the document collection

$$L_T = \sum_{i=1}^{N} \sum_{n=1}^{J} c\left(w_j, D_i\right) \log P\left(w_j \middle| D_i\right)$$
$$= \sum_{i=1}^{N} \sum_{n=1}^{J} c\left(w_j, D_i\right) \log \left\{ \sum_{k=1}^{K} P\left(T_k \middle| D_i\right) \left[\sum_{l=1}^{K} P\left(T_l \middle| Y_k\right) P\left(w_j \middle| T_l\right) \right] \right\}$$

  – EM training can be performed

$$\hat{P}\left(w_j \mid T_k\right) = \frac{\sum_{i=1}^{N} c\left(w_j, D_i\right) P\left(T_k \mid w_j, D_i\right)}{\sum_{j'=1}^{J} \sum_{i'=1}^{N} c\left(w_{j'}, D_{i'}\right) P\left(T_k \mid w_{j'}, D_{i'}\right)}$$

$$\hat{P}\left(T_k \mid D_i\right) = \frac{\sum_{j=1}^{J} c\left(w_j, D_i\right) P\left(T_k \mid w_j, D_i\right)}{c\left(D_i\right)}$$

where

$$P'\left(T_k \mid w_j, D_i\right) = \frac{\left[\sum_{l=1}^{K} P\left(w_j \mid T_l\right) P\left(T_l \mid T_k\right) \right] \cdot P\left(T_k \mid D_i\right)}{\sum_{k'=1}^{K} \left\{ \left[\sum_{l'=1}^{K} P\left(w_j \mid T_{l'}\right) P\left(T_{l'} \mid T_{k'}\right) \right] \cdot P\left(T_{k'} \mid D_i\right) \right\}}$$

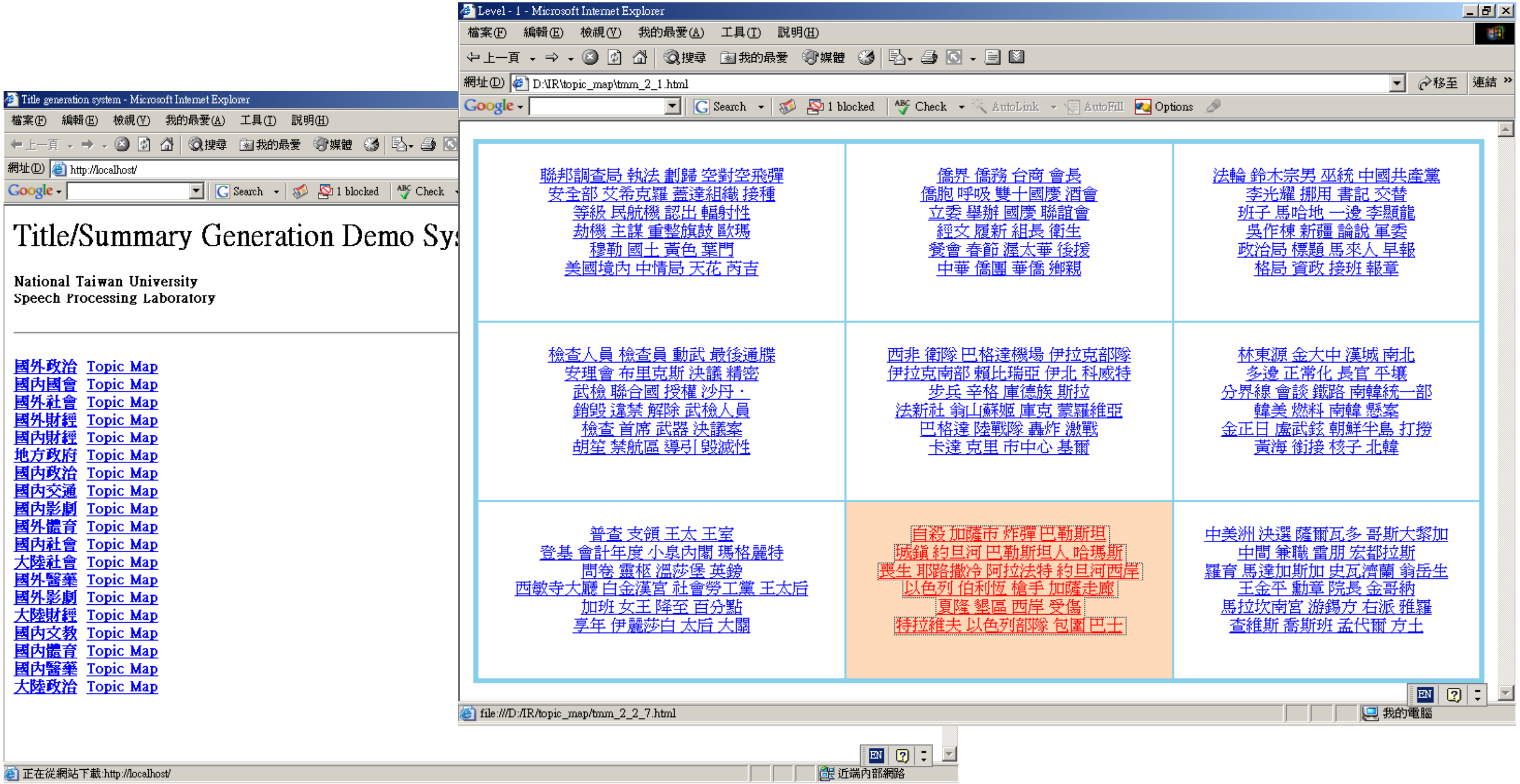
Hierarchical Document Organization (3/7)

- Criterion for Topic Word Selecting

$$S\left(w_j, T_k\right) = \frac{\sum_{i=1}^{N} c\left(w_j, D_i\right) P\left(T_k \mid D_i\right)}{\sum_{i'=1}^{N} c\left(w_j, D_{i'}\right)\left[1 - P\left(T_k \mid D_{i'}\right)\right]}$$

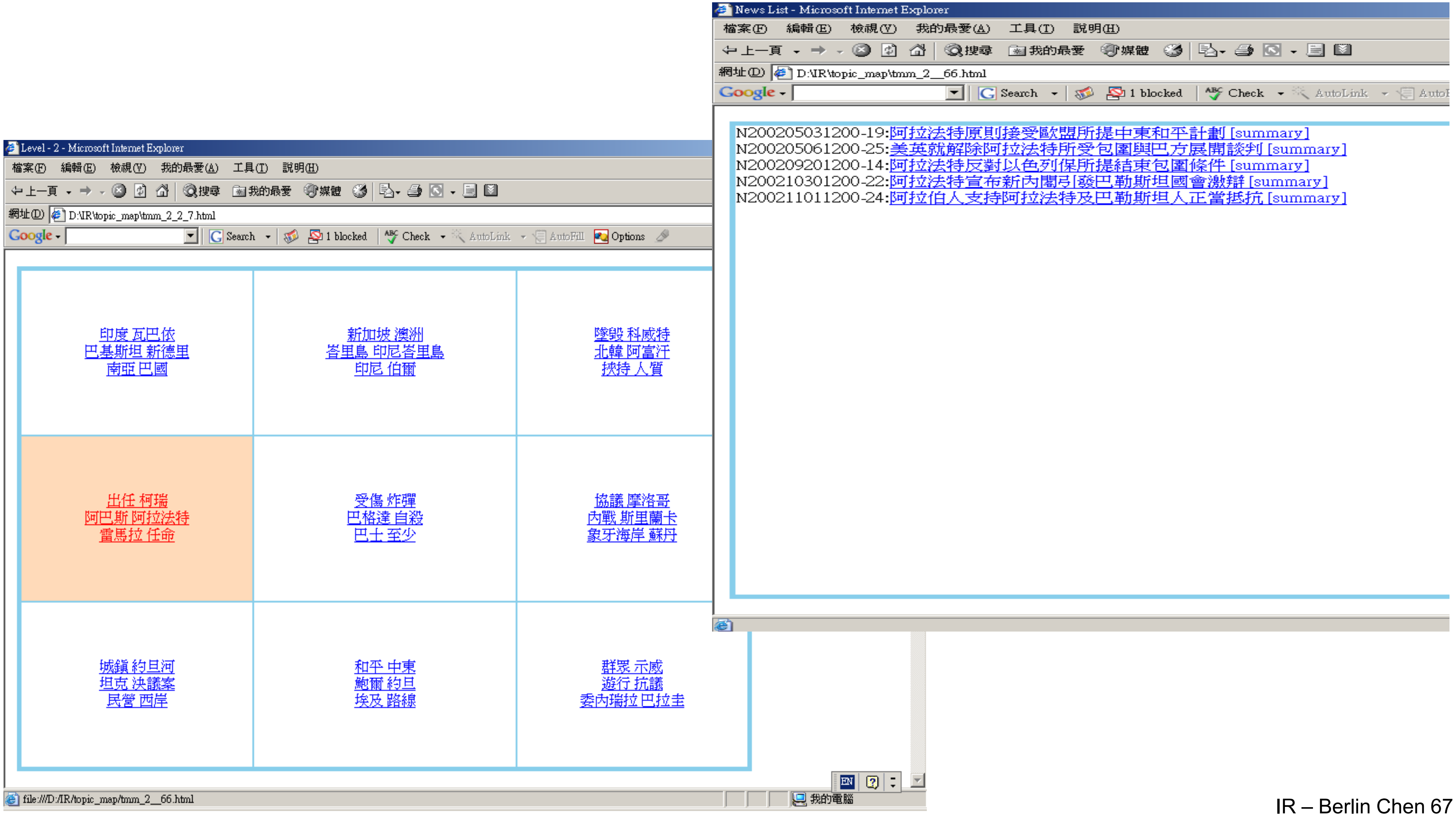
Hierarchical Document Organization (4/7)

- Example



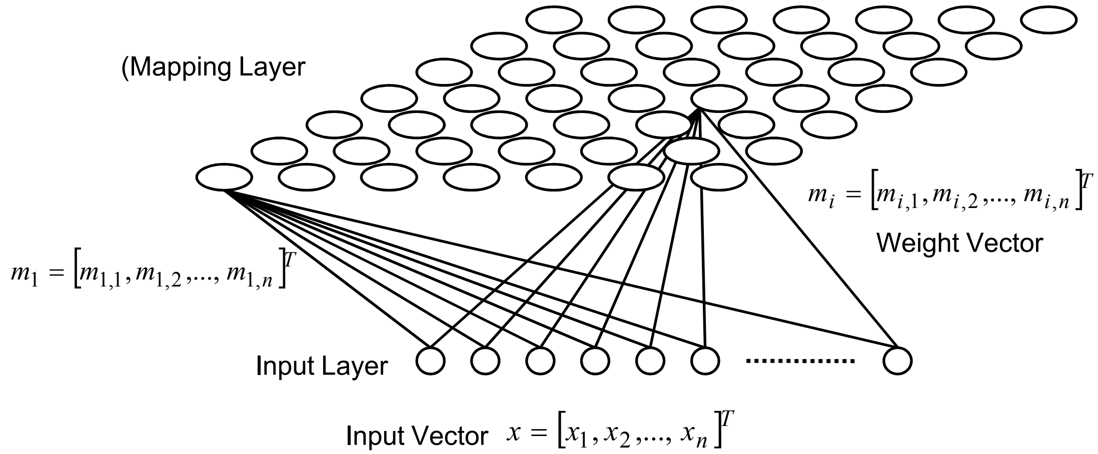
Hierarchical Document Organization (5/7)

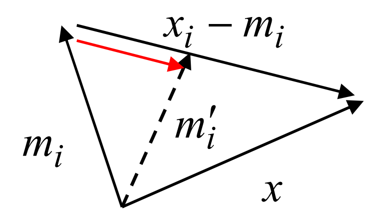
- Example (cont.)



Hierarchical Document Organization (6/7)

- Self-Organization Map (SOM)
  - A recursive regression process





$$m_i(t+1) = m_i(t) + h_{c(x),i}(t)[x(t) - m_i(t)]$$

$$c(x) = \arg \min_{i'} \|x - m_{i'}\|$$

where

$$\|x - m_{i'}\| = \sqrt{\sum_n (x_n - m_{i',n})^2}$$

$$h_{c(x),i}(t) = \alpha(t) \exp\left(- \frac{\|r_i - r_{c(x)}\|^2}{2\sigma^2(t)} \right)$$

Hierarchical Document Organization (7/7)

- Results

Model	Iterations	$dist_{Between}/dist_{Within}$
TMM	10	1.9165
	20	2.0650
	30	1.9477
	40	1.9175
SOM	100	2.0604

$$R_{Dist} = \frac{dist_{Between}}{dist_{Within}}$$

where

$$dist_{Between} = \frac{\sum_{i=1}^{|D|}\sum_{j=i+1}^{|D|} f_{Between}(i,j)}{\sum_{i=1}^{|D|}\sum_{j=i+1}^{|D|} C_{Between}(i,j)}$$

$$f_{Between}(i,j) = \begin{cases} dist_{Map}(i,j) & T_{r,i} \neq T_{r,j} \\ 0 & otherwise \end{cases}$$

$$dist_{Map}(i,j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

$$C_{Between}(i,j) = \begin{cases} 1 & T_{r,i} \neq T_{r,j} \\ 0 & otherwise \end{cases}$$

$$dist_{Within} = \frac{\sum_{i=1}^{|D|}\sum_{j=i+1}^{|D|} f_{Within}(i,j)}{\sum_{i=1}^{|D|}\sum_{j=i+1}^{|D|} C_{Within}(i,j)}$$

$$f_{Within}(i,j) = \begin{cases} dist_{Map}(i,j) & T_{r,i} = T_{r,j} \\ 0 & otherwise \end{cases}$$

$$C_{Within}(i,j) = \begin{cases} 1 & T_{r,i} = T_{r,j} \\ 0 & otherwise \end{cases}$$