

Clustering Techniques

Berlin Chen 2005

References:

1. *Introduction to Machine Learning* , Chapter 7
2. *Modern Information Retrieval*, Chapters 5, 7
3. *Foundations of Statistical Natural Language Processing*, Chapter 14
4. "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

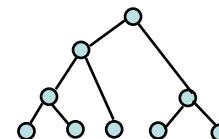
- Place similar objects in the same group and assign dissimilar objects to different groups
 - **Word clustering**
 - Neighbor overlap: words occur with the similar left and right neighbors (such as *in* and *on*)
 - **Document clustering**
 - Documents with the similar topics or concepts are put together
- But clustering cannot give a comprehensive description of the object
 - How to label objects shown on the visual display
- Regarded as a kind of semiparametric learning approach
 - Allow a mixture of distributions to be used for estimating the input samples (a parametric model for each group of samples)

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

- Another important distinction between clustering algorithms is whether they perform soft or hard assignment
- **Hard Assignment**
 - Each object 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 (cont.)

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

Summarized Attributes of Clustering Algorithms

- 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 $n \times n$ matrix of similarity coefficients)

Summarized Attributes of Clustering Algorithms (cont.)

- Flat Clustering
 - Preferable if efficiency is a consideration or data sets are very large
 - *K*-means is the conceptually 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
 - Probabilistic Latent Semantic Analysis (PLSA), Topic Mixture Model (TMM), etc.

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

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

← 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 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 (cont.)

- Algorithm

```
1 Given: a set  $\mathcal{X} = \{x_1, \dots, x_n\}$  of objects
2       a function  $\text{sim}: \mathcal{P}(\mathcal{X}) \times \mathcal{P}(\mathcal{X}) \rightarrow \mathbb{R}$ 
3 for  $i := 1$  to  $n$  do   Initialization (for tree leaves):
4    $c_i := \{x_i\}$  end   Each object is a cluster
5  $C := \{c_1, \dots, c_n\}$ 
6  $j := n + 1$ 
7 while  $|C| > 1$    cluster number
8    $(c_{n_1}, c_{n_2}) := \arg \max_{(c_u, c_v) \in C \times C} \text{sim}(c_u, c_v)$ 
9    $c_j = c_{n_1} \cup c_{n_2}$    merged as a new cluster
10   $C := C \setminus \{c_{n_1}, c_{n_2}\} \cup \{c_j\}$    The original two clusters
11   $j := j + 1$    are removed
```

Figure 14.2 Bottom-up hierarchical clustering.

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)

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

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

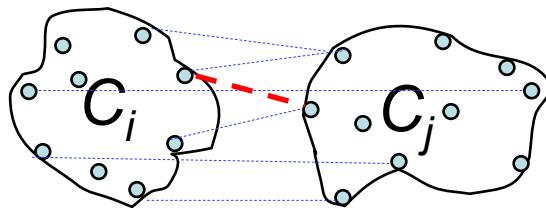
$$1 - \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| \cdot |\vec{y}|}$$

ranged between 0 and 1

Measures of Cluster Similarity

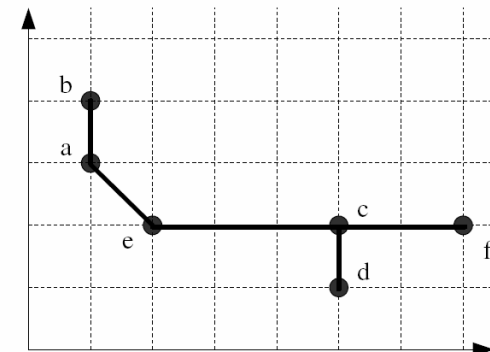
- Especially for the bottom-up approaches
- **Single-link** clustering
 - The similarity between two clusters is the similarity of the two closest objects in the clusters
 - Search over all pairs of objects that are from the two different clusters and select the pair with the greatest similarity
 - Elongated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

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



greatest similarity

cf. the minimal
spanning tree

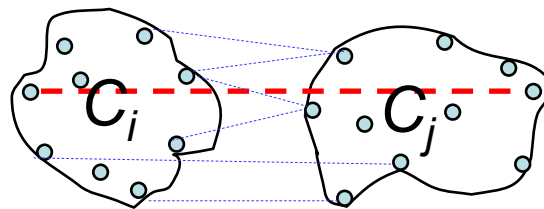


Measures of Cluster Similarity (cont.)

- **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

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



least similarity

Measures of Cluster Similarity (cont.)

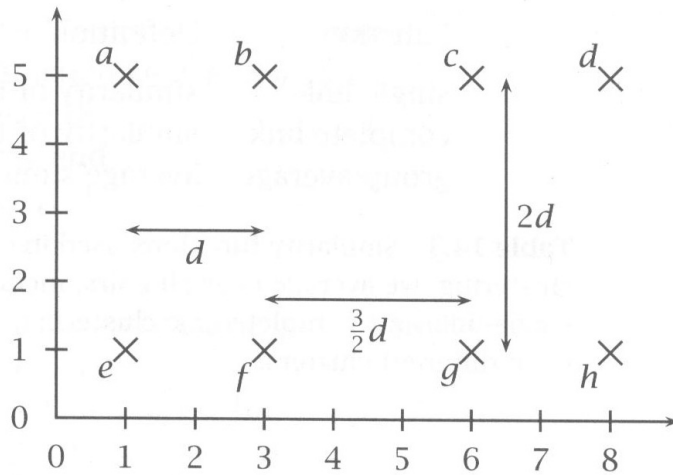


Figure 14.4 A cloud of points in a plane.

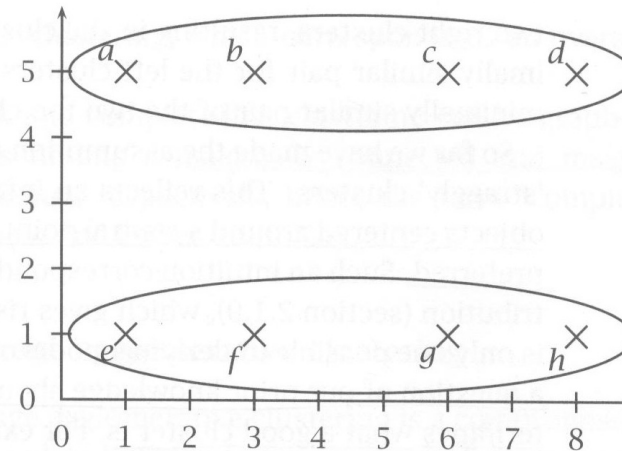


Figure 14.6 Single-link clustering of the points in figure 14.4.

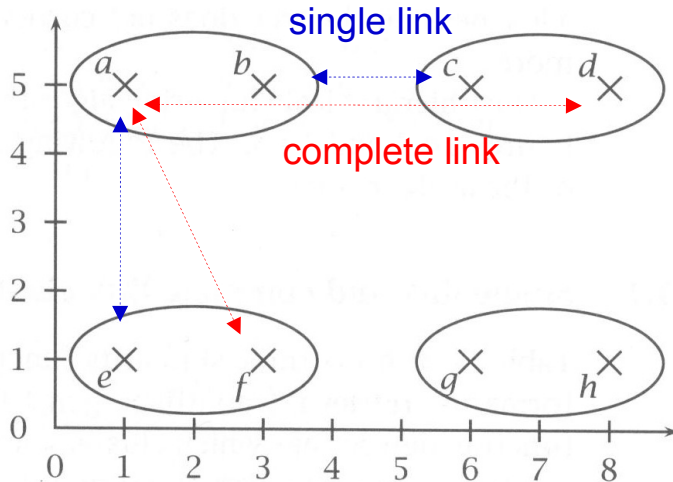


Figure 14.5 Intermediate clustering of the points in figure 14.4.

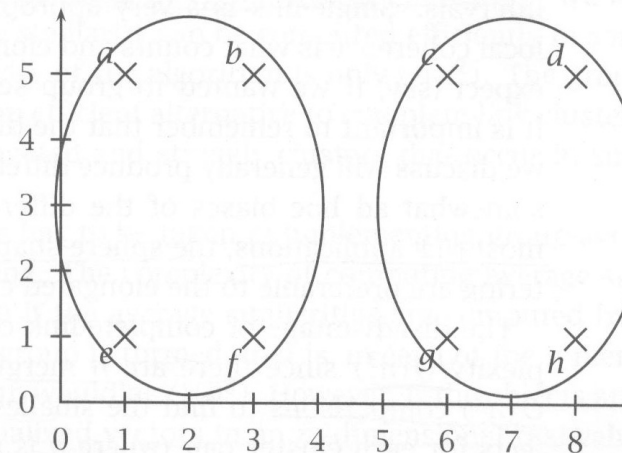
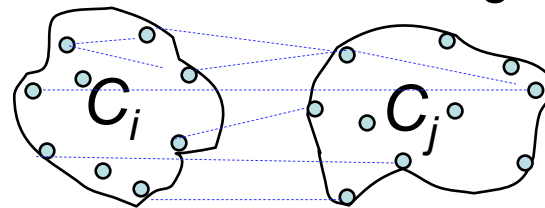


Figure 14.7 Complete-link clustering of the points in figure 14.4.

Measures of Cluster Similarity (cont.)

- **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

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

length-normalized vectors

Measures of Cluster Similarity (cont.)

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

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

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

- The sum of members in a cluster c_j : $\vec{s}(c_j) = \sum_{\vec{x} \in c_j} \vec{x}$

- Express $SIM(c_j)$ in terms of $\vec{s}(c_j)$

$$\begin{aligned} \vec{s}(c_j) \cdot \vec{s}(c_j) &= \sum_{\vec{x} \in c_j} \vec{x} \cdot \vec{s}(c_j) = \sum_{\vec{x} \in c_j} \sum_{\vec{y} \in c_j} \vec{x} \cdot \vec{y} \quad \text{length-normalized vector} \\ &= |c_j|(|c_j| - 1)SIM(c_j) + \sum_{\vec{x} \in c_j} \vec{x} \cdot \vec{x} = 1 \end{aligned}$$

$$= |c_j|(|c_j| - 1)SIM(c_j) + |c_j|$$

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

Measures of Cluster Similarity (cont.)

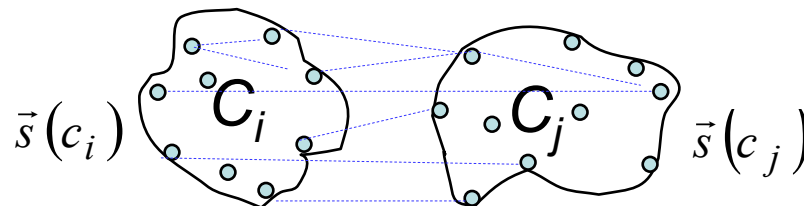
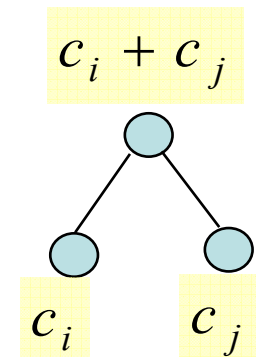
- **Group-average** agglomerative clustering (cont.)

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

$$\Rightarrow \vec{s}(c_{New}) = \vec{s}(c_i) + \vec{s}(c_j), \quad |c_{New}| = |c_i| + |c_j|$$

- The average similarity for their union will be

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



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

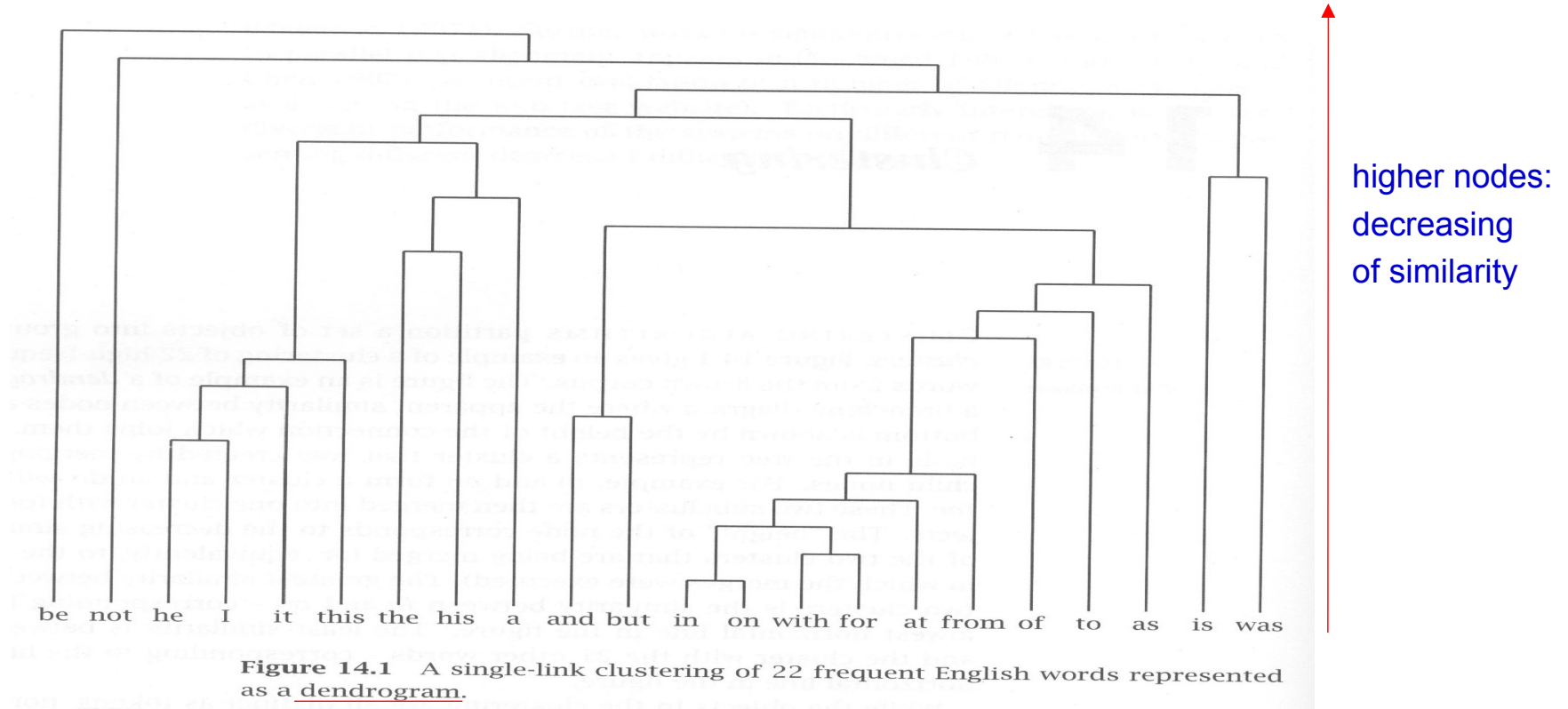


Figure 14.1 A single-link clustering of 22 frequent English words represented as a dendrogram.

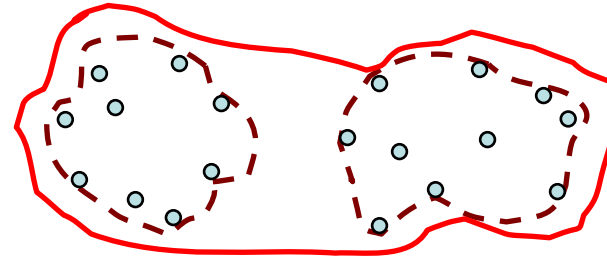
"be" has least similarity with the other 21 words !

Divisive Clustering

- 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 (cont.)

- 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 (cont.)

- Algorithm

```
1 Given: a set  $\mathcal{X} = \{x_1, \dots, x_n\}$  of objects
2       a function  $\text{coh}: \mathcal{P}(\mathcal{X}) \rightarrow \mathbb{R}$ 
3       a function  $\text{split}: \mathcal{P}(\mathcal{X}) \rightarrow \mathcal{P}(\mathcal{X}) \times \mathcal{P}(\mathcal{X})$ 
4  $C := \{\mathcal{X}\}$  ( $= \{c_1\}$ )
5  $j := 1$ 
6 while  $\exists c_i \in C$  s.t.  $|c_i| > 1$ 
7      $c_u := \arg \min_{c_v \in C} \text{coh}(c_v)$ 
8      $(c_{j+1}, c_{j+2}) := \text{split}(c_u)$ 
9      $C := C \setminus \{c_u\} \cup \{c_{j+1}, c_{j+2}\}$ 
10     $j := j + 2$ 
```

split the least coherent cluster

Generate two new clusters and remove the original one

Figure 14.3 Top-down hierarchical clustering.

Non-Hierarchical Clustering

Non-hierarchical 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
 - **Algorithms** introduced here
 - The *K*-means algorithm
 - The EM algorithm
- k-1 → k → k+1*
- Hierarchical clustering also has to face this problem*

The *K*-means Algorithm

- Also called *Linde-Buzo-Gray* (LBG) in signal processing
 - A **hard clustering** algorithm
 - Define clusters by the **center of mass** of their members
- 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) \rightarrow 8 bits/pixel (256 colors)
 - A compression rate of 3

$$\mathbf{X} = \left\{ \mathbf{x}^t \right\}_{t=1}^n \xrightarrow{\text{index } j} \mathbf{F} = \left\{ \mathbf{m}_j \right\}_{j=1}^k \quad \text{Dim}(\mathbf{x}^t)=24 \rightarrow k=2^8$$

\mathbf{m}_j : cluster centroid or reference vector, code word, code vector

The K -means Algorithm (cont.)

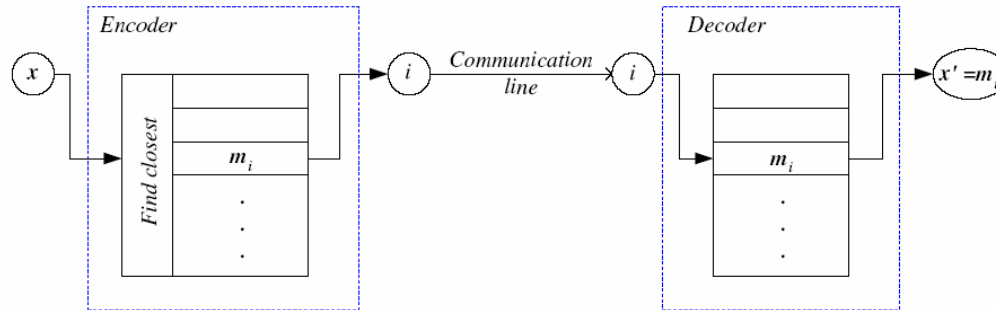


Figure 7.1: Given \mathbf{x} , the encoder sends the index of the closest code word and the decoder generates the code word with the received index as \mathbf{x}' . Error is $\|\mathbf{x}' - \mathbf{x}\|^2$.

Total Reconstruction error

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

↖ label

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

The K -means Algorithm (cont.)

- **Initialization**

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

- **Recursion**

- Assign each object \mathbf{x}^t to the cluster whose center is closest

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

- 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)

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

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

The K -means Algorithm (cont.)

- Algorithm

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

Repeat

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

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

For all $\mathbf{m}_i, i = 1, \dots, k$

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

Until \mathbf{m}_i converge

The K -means Algorithm (cont.)

- Example 1

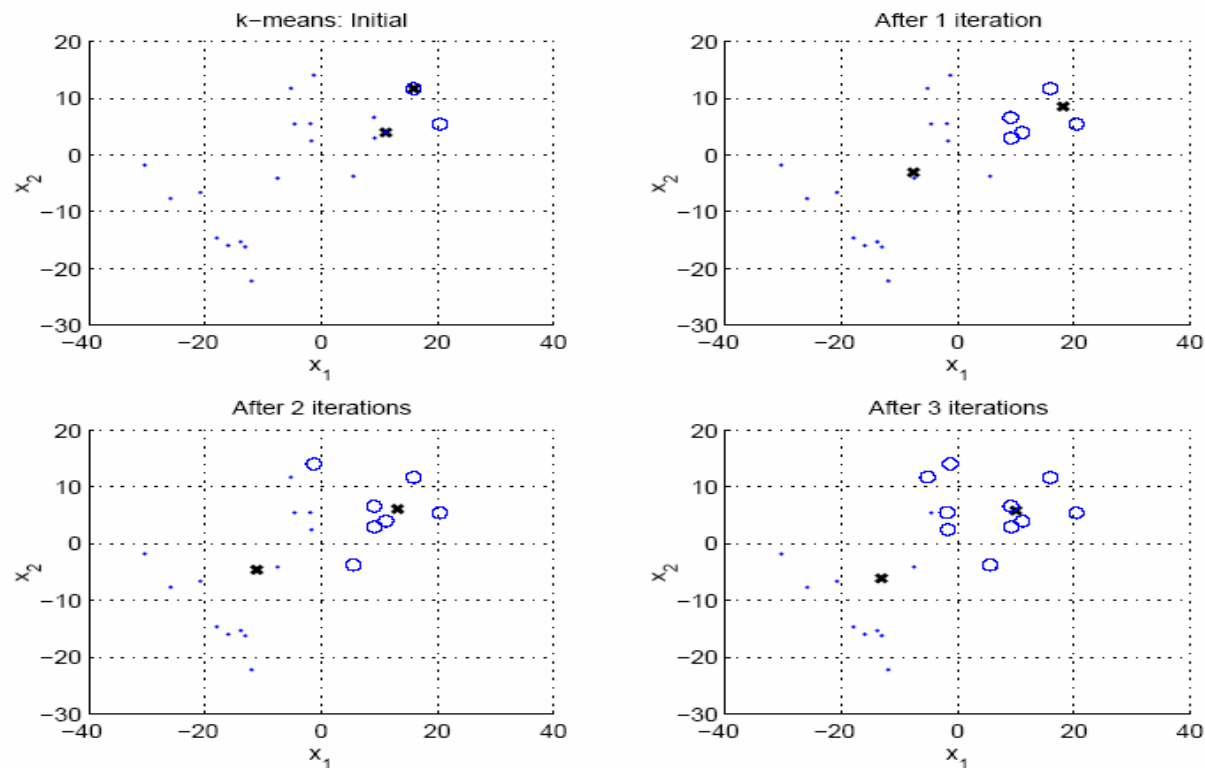


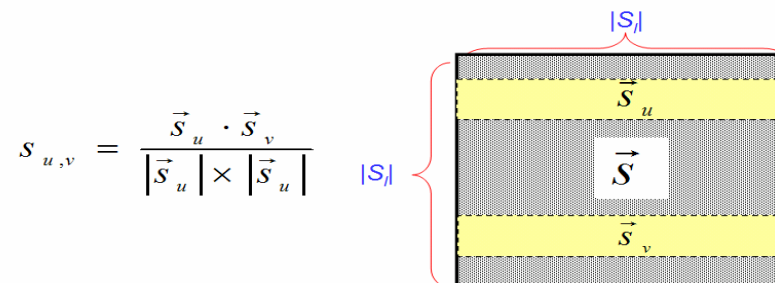
Figure 7.2: Evolution of k -means. Crosses indicate center positions. Data points are marked depending on the closest center.

The *K*-means Algorithm (cont.)

- Example 2

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



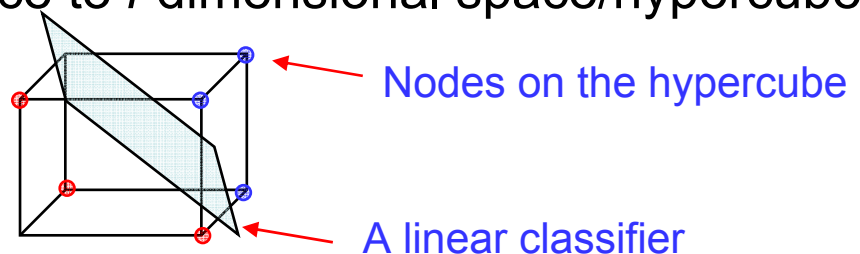
The K -means Algorithm (cont.)

- Choice of initial cluster centers (seeds) is important
 - Pick at random
 - Calculate the mean of all data and generate k initial centers m_i by adding small random vector to the mean $m_i \pm \delta$
 - Project data onto the principal component (first eigenvector), divide its range into k equal intervals, and take the mean of data in each group as the initial center 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
- Poor seeds will result in **sub-optimal clustering**

The K -means Algorithm (cont.)

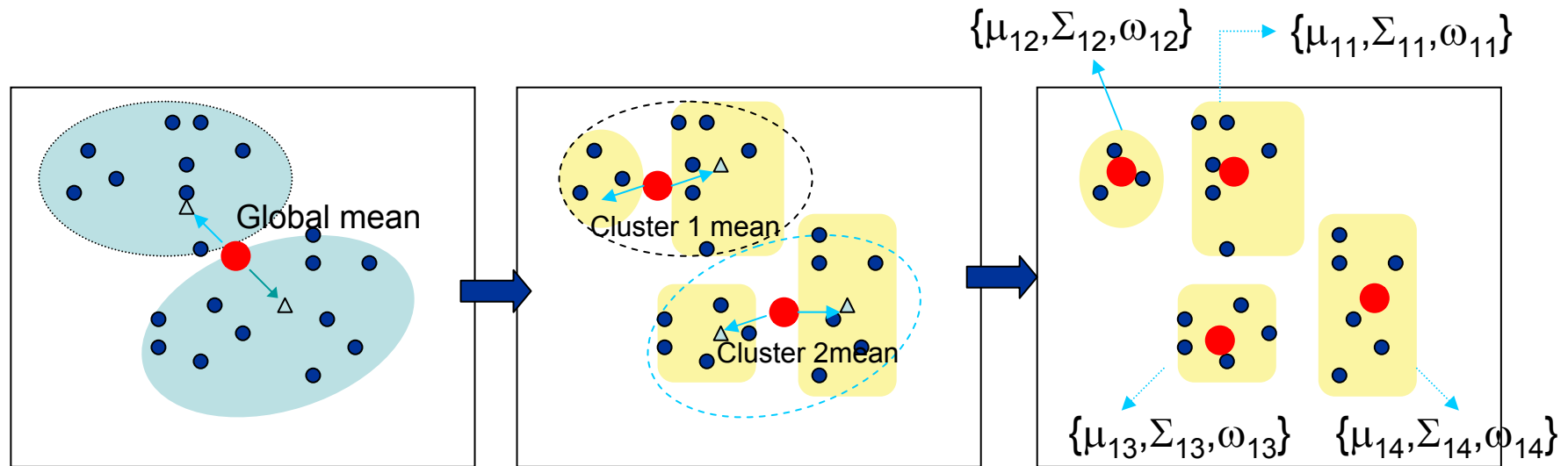
- How to break ties when in case there are several centers with the same distance from an object
 - Randomly assign the object to one of the candidate clusters
 - Or, perturb objects slightly
- Applications of the K -means Algorithm
 - Clustering
 - Vector quantization
 - A preprocessing stage before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

$$l = \log_2 k \quad (k \text{ clusters})$$



The K -means Algorithm (cont.)

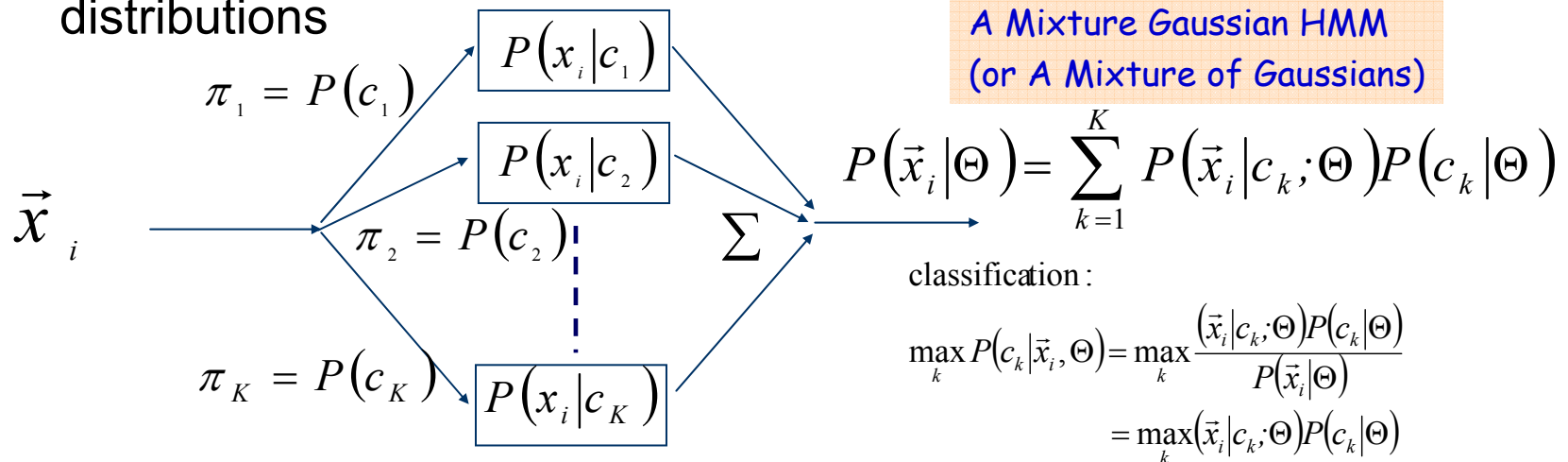
- E.g., the LBG algorithm
 - By Linde, Buzo, and Gray



$M \rightarrow 2M$ at each iteration

The EM Algorithm

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



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

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

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

Continuous case:

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

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

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

The EM Algorithm (cont.)

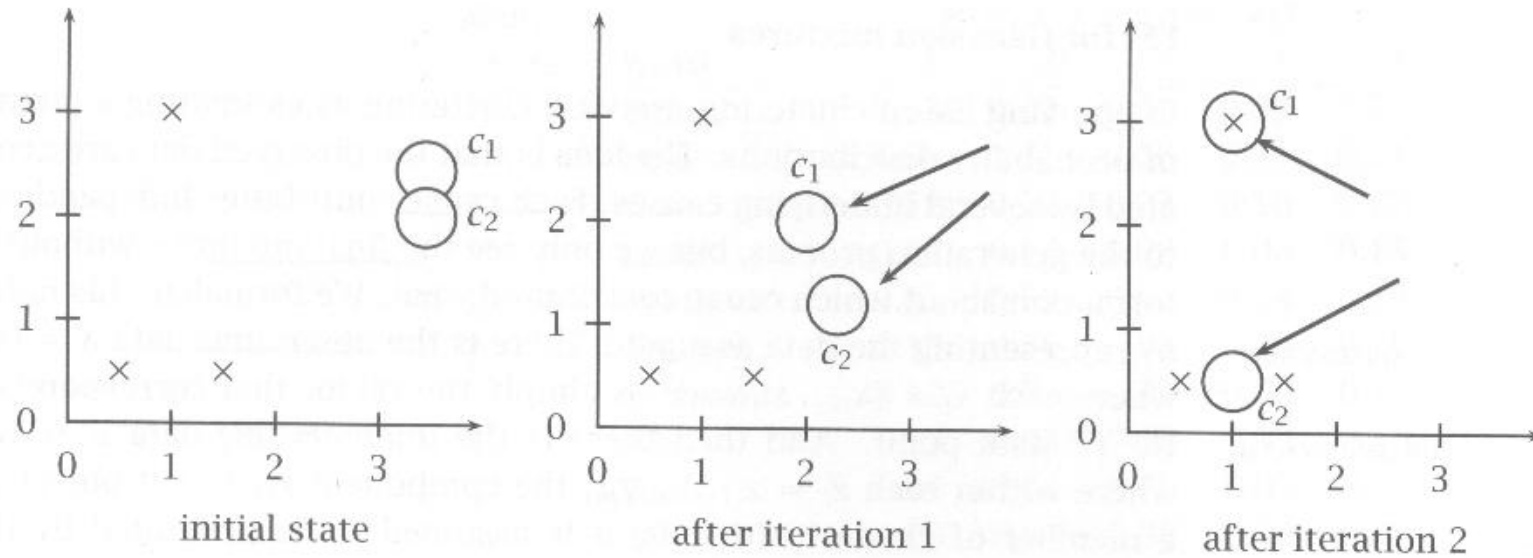
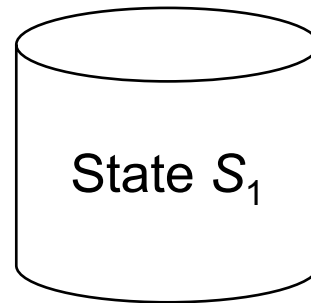


Figure 14.10 An example of using the EM algorithm for soft clustering.

Maximum Likelihood Estimation

- Hard Assignment



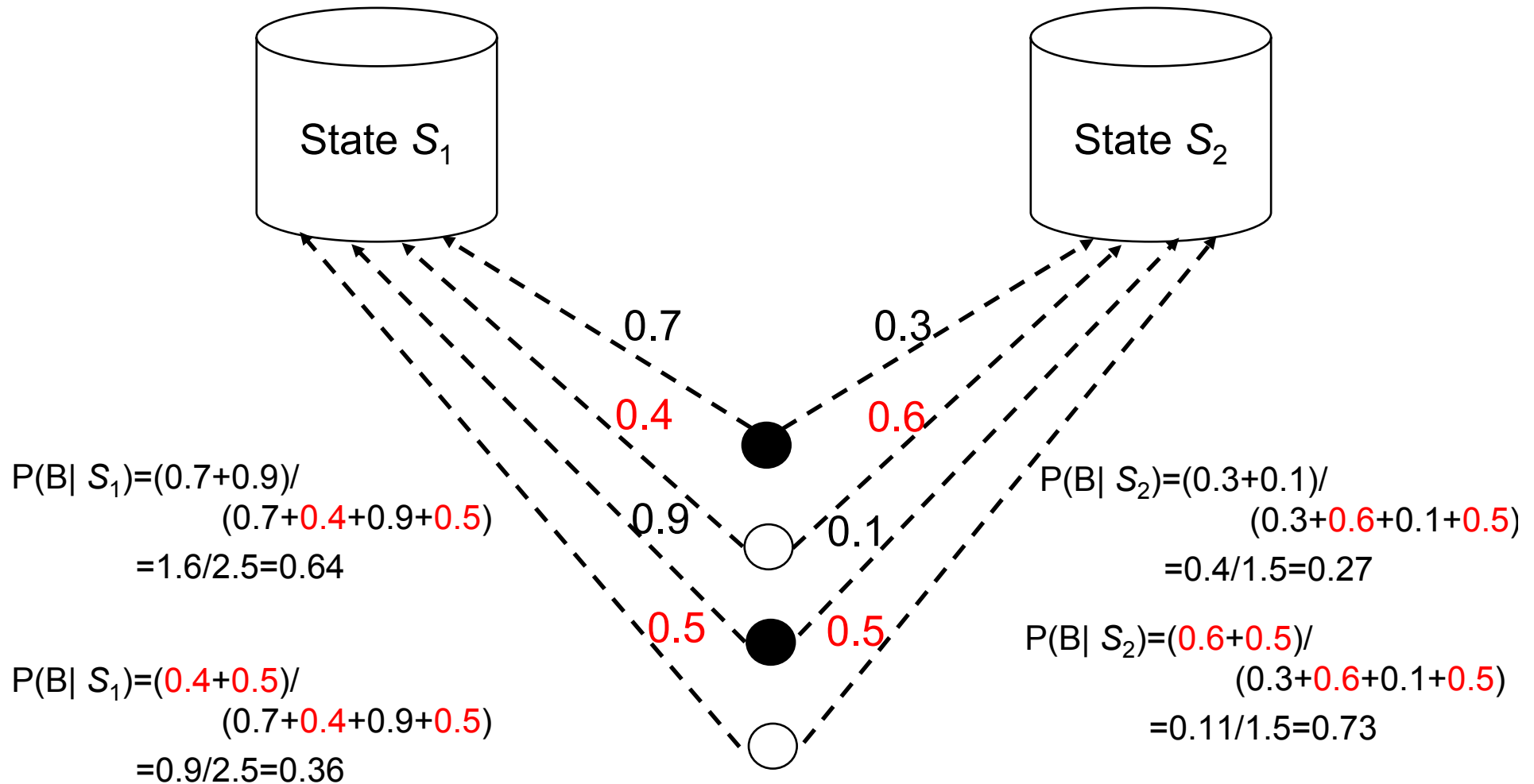
$$P(B | S_1) = 2/4 = 0.5$$

$$P(W | S_1) = 2/4 = 0.5$$



Maximum Likelihood Estimation

- Soft Assignment



The EM Algorithm (cont.)

Note :

$$\prod_{t=1}^T \left(\sum_{k_i=1}^M a_{tk_i} \right)$$

$$= (a_{11} + a_{12} + \dots + a_{1M})(a_{21} + a_{22} + \dots + a_{2M}) \dots (a_{T1} + a_{T2} + \dots + a_{TM})$$

$$= \sum_{k_1=1}^M \sum_{k_2=1}^M \dots \sum_{k_T=1}^M \prod_{t=1}^T a_{tk_t}$$

- E-step (Expectation)
 - Derive the complete data likelihood function

likelihood function

$$P(\mathbf{X} | \hat{\Theta}) = \prod_{i=1}^n P(\vec{x}_i | \hat{\Theta}) = \prod_{i=1}^n \sum_{k_i=1}^K P(\vec{x}_i | c_{k_i}; \hat{\Theta}) P(c_{k_i} | \hat{\Theta})$$

$$= \left(P(\vec{x}_1 | c_1; \hat{\Theta}) P(c_1 | \hat{\Theta}) + \dots + P(\vec{x}_1 | c_K; \hat{\Theta}) P(c_K | \hat{\Theta}) \right) \times \dots$$

$$\times \left(P(\vec{x}_n | c_1; \hat{\Theta}) P(c_1 | \hat{\Theta}) + \dots + P(\vec{x}_n | c_K; \hat{\Theta}) P(c_K | \hat{\Theta}) \right)$$

$$= \sum_{k_1=1}^K \sum_{k_2=1}^K \dots \sum_{k_n=1}^K \prod_{i=1}^n \left[P(\vec{x}_i | c_{k_i}; \hat{\Theta}) P(c_{k_i} | \hat{\Theta}) \right]$$

$$= \sum_{k_1=1}^K \sum_{k_2=1}^K \dots \sum_{k_n=1}^K \prod_{i=1}^n \left[P(\vec{x}_i, c_{k_i} | \hat{\Theta}) \right]$$

$$= \sum_{k_1=1}^K \sum_{k_2=1}^K \dots \sum_{k_n=1}^K \left[P(\vec{x}_1, c_{k_1}, \vec{x}_2, c_{k_2}, \dots, \vec{x}_n, c_{k_n} | \hat{\Theta}) \right]$$

$$= \sum_{k_1=1}^K \sum_{k_2=1}^K \dots \sum_{k_n=1}^K \left[P(\vec{x}_1, c_{k_1}, \vec{x}_2, c_{k_2}, \dots, \vec{x}_n, c_{k_n} | \hat{\Theta}) \right]$$

$$= \sum_{\mathbf{C}} \left[P(\mathbf{X}, \mathbf{C} | \hat{\Theta}) \right] \quad \text{the complete data likelihood function}$$

$$\mathbf{X} = \vec{x}_1 \vec{x}_2 \dots \vec{x}_{n-1} \vec{x}_n$$

$$\mathbf{C} = c_{k_1} c_{k_2} \dots c_{k_{n-1}} c_{k_n}$$

How many kinds of \mathbf{C} ? (K^n kinds)

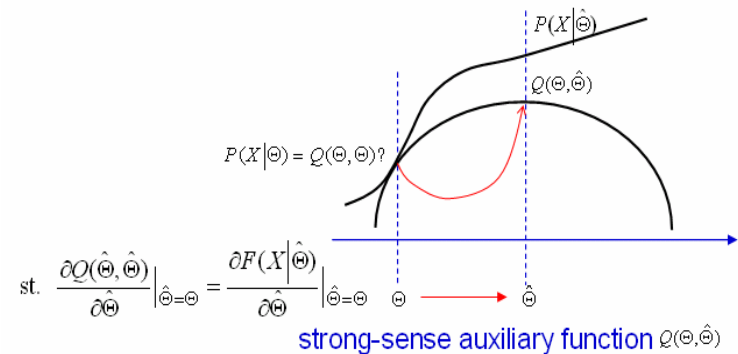
The EM Algorithm (cont.)

- E-step (Expectation)
 - Define the **auxiliary function** $\Phi(\Theta, \hat{\Theta})$ as the expectation of the **log complete likelihood function** L^{CM} with respect to the hidden/latent variable \mathbf{C} conditioned on known data (\mathbf{X}, Θ)

known unknown

$$\begin{aligned} \Phi(\Theta, \hat{\Theta}) &= E[\log L^{CM}]_{\mathbf{C}|\mathbf{X}, \Theta} = E[\log P(\mathbf{X}, \mathbf{C} | \hat{\Theta})]_{\mathbf{C}|\mathbf{X}, \Theta} \\ &= \sum_{\mathbf{C}} P(\mathbf{C} | \mathbf{X}, \Theta) \log P(\mathbf{X}, \mathbf{C} | \hat{\Theta}) \\ &= \sum_{\mathbf{C}} \frac{P(\mathbf{X}, \mathbf{C} | \Theta)}{P(\mathbf{X} | \Theta)} \log P(\mathbf{X}, \mathbf{C} | \hat{\Theta}) \end{aligned}$$

$$Q(\Theta, \hat{\Theta}) - Q(\hat{\Theta}, \hat{\Theta}) \leq P(\mathbf{X} | \Theta) - P(\mathbf{X} | \hat{\Theta})$$



- Maximize the log likelihood function $\log P(\mathbf{X} | \hat{\Theta})$ by maximizing the expectation of the log complete likelihood function $\Phi(\Theta, \hat{\Theta})$

- We have shown this property when deriving the HMM-based retrieval model

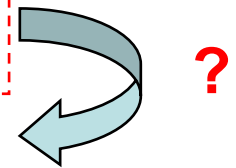
$$\Phi(\Theta, \hat{\Theta}) \nearrow \implies \log P(\mathbf{X} | \hat{\Theta}) \nearrow$$

The EM Algorithm (cont.)

- E-step (Expectation)
 - The auxiliary function $\Phi(\Theta, \hat{\Theta})$

$$\begin{aligned}
 \Phi(\Theta, \hat{\Theta}) &= \sum_{\mathbf{C}} \frac{P(\mathbf{X}, \mathbf{C} | \Theta)}{P(\mathbf{X} | \Theta)} \log P(\mathbf{X}, \mathbf{C} | \hat{\Theta}) \\
 &= \sum_{\mathbf{C} = c_{k_1} c_{k_2} \dots c_{k_n}} \left[\prod_{j=1}^n \frac{P(\bar{x}_j, c_{k_j} | \Theta)}{P(\bar{x}_j | \Theta)} \right] \left[\log \prod_{i=1}^n P(\bar{x}_i, c_{k_i} | \hat{\Theta}) \right] \\
 &= \sum_{\mathbf{C} = c_{k_1} c_{k_2} \dots c_{k_n}} \left[\prod_{j=1}^n P(c_{k_j} | \bar{x}_j, \Theta) \right] \left[\sum_{i=1}^n \log P(\bar{x}_i, c_{k_i} | \hat{\Theta}) \right] \\
 &= \sum_{\mathbf{C} = c_{k_1} c_{k_2} \dots c_{k_n}} \sum_{k=1}^m \left\{ \delta_{k, k_i} \left[\sum_{i=1}^n \log P(\bar{x}_i, c_k | \hat{\Theta}) \right] \left[\prod_{j=1}^n P(c_{k_j} | \bar{x}_j, \Theta) \right] \right\} \\
 &= \sum_{k=1}^m \sum_{i=1}^n \left\{ \left[\log P(\bar{x}_i, c_k | \hat{\Theta}) \right] \sum_{\mathbf{C} = c_{k_1} c_{k_2} \dots c_{k_n}} \delta_{k, k_i} \left[\prod_{j=1}^n P(c_{k_j} | \bar{x}_j, \Theta) \right] \right\} \\
 &= \sum_{k=1}^m \sum_{i=1}^n \left\{ \left[\log P(\bar{x}_i, c_k | \hat{\Theta}) \right] P(c_k | \bar{x}_i, \Theta) \right\} \\
 &= \sum_{k=1}^m \sum_{i=1}^n \left\{ P(c_{k_j} | \bar{x}_j, \Theta) \log P(\bar{x}_i, c_k | \hat{\Theta}) \right\} \\
 &= \sum_{k=1}^m \sum_{i=1}^n \left\{ P(c_{k_j} | \bar{x}_j, \Theta) \log \left[P(\bar{x}_i | c_k, \hat{\Theta}) P(c_k | \hat{\Theta}) \right] \right\} \\
 &= \sum_{k=1}^m \sum_{i=1}^n \left\{ P(c_{k_j} | \bar{x}_j, \Theta) \log P(c_k | \hat{\Theta}) \right\} + \sum_{k=1}^m \sum_{i=1}^n \left\{ P(c_{k_j} | \bar{x}_j, \Theta) \log P(\bar{x}_i | c_k, \hat{\Theta}) \right\}
 \end{aligned}$$

$$\delta_{k, k_i} = \begin{cases} 1 & \text{if } k_i = k \\ 0 & \text{otherwise} \end{cases}$$



See Next Slide

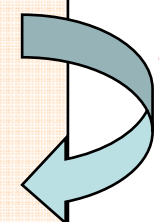
The EM Algorithm (cont.)

– Note that

Note :

$$\begin{aligned} & \prod_{t=1}^T \left(\sum_{k_i=1}^M a_{tk_i} \right) \\ &= (a_{11} + a_{12} + \dots + a_{1M})(a_{21} + a_{22} + \dots + a_{2M}) \dots (a_{T1} + a_{T2} + \dots + a_{TM}) \\ &= \sum_{k_1=1}^M \sum_{k_2=1}^M \dots \sum_{k_T=1}^M \prod_{t=1}^T a_{tk_t} \end{aligned}$$

$$\begin{aligned} & \sum_{c = c_{k_1} c_{k_2} \dots c_{k_n}} \delta_{k, k_i} \left[\prod_{j=1}^n P(c_{k_j} | \vec{x}_j, \Theta) \right] \\ &= \sum_{c_{k_1}=1}^m \sum_{c_{k_2}=1}^m \dots \sum_{c_{k_n}=1}^m \prod_{j=1}^n \left[\delta_{k, k_i} P(c_{k_j} | \vec{x}_j, \Theta) \right] \\ &= \sum_{c_{k_1}=1}^m \sum_{c_{k_2}=1}^m \dots \sum_{c_{k_n}=1}^m \prod_{j=1}^n \left[\delta_{k, k_i} P(c_{k_j} | \vec{x}_j, \Theta) \right] \\ &= \left[\prod_{j=1, j \neq i}^n \left[\sum_{k_j=1}^m P(c_{k_j} | \vec{x}_j, \Theta) \right] \right] \left[\sum_{c_{k_i}=1}^m \delta_{k, k_i} P(c_{k_i} | \vec{x}_i, \Theta) \right] \\ &= \left[\prod_{j=1, j \neq i}^n 1 \right] P(c_k | \vec{x}_i, \Theta) \quad \leftarrow \vec{x}_i \text{ can only be aligned to } c_k \\ &= P(c_k | \vec{x}_i, \Theta) \end{aligned}$$



The EM Algorithm (cont.)

- E-step (Expectation)
 - The auxiliary function can also be divided into two:

$$\Phi(\Theta, \hat{\Theta}) = \Phi_a(\Theta, \hat{\Theta}) + \Phi_b(\Theta, \hat{\Theta})$$

where

$$\begin{aligned} \Phi_a(\Theta, \hat{\Theta}) &= \sum_{i=1}^n \sum_{k=1}^K P(c_k | \vec{x}_i, \Theta) \log P(c_k | \hat{\Theta}) \\ &= \sum_{i=1}^n \sum_{k=1}^K \frac{P(\vec{x}_i | c_k, \Theta) P(c_k | \Theta)}{P(\vec{x}_i | \Theta)} \log P(c_k | \hat{\Theta}) \\ &= \sum_{i=1}^n \sum_{k=1}^K \frac{P(\vec{x}_i | c_k, \Theta) P(c_k | \Theta)}{\sum_{l=1}^K P(\vec{x}_i | c_l, \Theta) P(c_l | \Theta)} \log P(c_k | \hat{\Theta}) \end{aligned}$$

auxiliary function for mixture weights

$$\begin{aligned} \Phi_b(\Theta, \hat{\Theta}) &= \sum_{i=1}^n \sum_{k=1}^K P(c_k | \vec{x}_i, \Theta) \log P(\vec{x}_i | c_k, \hat{\Theta}) \\ &= \sum_{i=1}^n \sum_{k=1}^K \frac{P(\vec{x}_i | c_k, \Theta) P(c_k | \Theta)}{\sum_{l=1}^K P(\vec{x}_i | c_l, \Theta) P(c_l | \Theta)} \log P(\vec{x}_i | c_k, \hat{\Theta}) \end{aligned}$$

auxiliary function for cluster distributions

The EM Algorithm (cont.)

- M-step (Maximization)
 - Remember that
 - Maximize a function F with a constraint by applying Lagrange multiplier

By applying Lagrange Multiplier ℓ

Suppose that $F = \sum_{j=1}^N w_j \log y_j \Rightarrow \hat{F} = \sum_{j=1}^N w_j \log y_j + \ell \left(\sum_{j=1}^N y_j - 1 \right)$

$$\frac{\partial \hat{F}}{\partial y_j} = \frac{w_j}{y_j} + \ell = 0 \Rightarrow \ell = -\frac{w_j}{y_j} \quad \forall j$$

$$\ell \sum_{j=1}^N y_j = -\sum_{j=1}^N w_j \Rightarrow \ell = -\sum_{j=1}^N w_j$$

$$\therefore y_j = \frac{w_j}{\sum_{j=1}^N w_j}$$

Constraint

Note :

$$\frac{\partial \log y_j}{\partial y_j} = \frac{1}{y_j}$$

The EM Algorithm (cont.)

- M-step (Maximization)

auxiliary function for mixture weights (or priors for Gaussians)

- Maximize $\Phi_a(\Theta, \hat{\Theta})$

$$\bar{\Phi}_a(\Theta, \hat{\Theta}) = \Phi_a(\Theta, \hat{\Theta}) + l \left(\sum_{k=1}^K P(c_k | \hat{\Theta}) - 1 \right)$$

$$= \sum_{k=1}^K \sum_{i=1}^n \frac{P(\vec{x}_i | c_k, \Theta) P(c_k | \Theta)}{\sum_{l=1}^K P(\vec{x}_i | c_l, \Theta) P(c_l | \Theta)} \log P(c_k | \hat{\Theta}) + l \left(\sum_{k=1}^K P(c_k | \hat{\Theta}) - 1 \right)$$

w_k y_k

r_k^i : the expected number of times that \vec{x}_i falls in class c_k

$$\Rightarrow \hat{\pi}_k = P(c_k | \hat{\Theta}) = \frac{w_k}{\sum_{k=1}^K w_k} = \frac{\sum_{i=1}^n \frac{P(\vec{x}_i | c_k, \Theta) P(c_k | \Theta)}{\sum_{l=1}^K P(\vec{x}_i | c_l, \Theta) P(c_l | \Theta)}}{\sum_{k=1}^K \sum_{i=1}^n \frac{P(\vec{x}_i | c_k, \Theta) P(c_k | \Theta)}{\sum_{l=1}^K P(\vec{x}_i | c_l, \Theta) P(c_l | \Theta)}} = \frac{\sum_{i=1}^n \frac{P(\vec{x}_i | c_k, \Theta) P(c_k | \Theta)}{\sum_{l=1}^K P(\vec{x}_i | c_l, \Theta) P(c_l | \Theta)}}{n}$$

The EM Algorithm (cont.)

- M-step (Maximization)

- Maximize $\Phi_b(\Theta, \hat{\Theta})$

auxiliary function for
(multivariate) Gaussian Means and Variances

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

$$\Phi_b(\Theta, \hat{\Theta}) = \sum_{i=1}^n \sum_{k=1}^K \frac{P(\vec{x}_i | c_k, \Theta) P(c_k | \Theta)}{\sum_{l=1}^K P(\vec{x}_i | c_l, \Theta) P(c_l | \Theta)} \log P(\vec{x}_i | c_k, \hat{\Theta})$$

Let $w_{k,i} = \frac{P(\vec{x}_i | c_k, \Theta) P(c_k | \Theta)}{\sum_{l=1}^K P(\vec{x}_i | c_l, \Theta) P(c_l | \Theta)}$ and $\log P(\vec{x}_i | c_k; \Theta) =$

$$-m/2 \cdot \log(2\pi) - 1/2 \log |\Sigma_k| - \frac{1}{2} (\vec{x}_i - \vec{\mu}_k)^T \Sigma_k^{-1} (\vec{x}_i - \vec{\mu}_k)$$

$$\Rightarrow \Phi_b(\Theta, \hat{\Theta}) = - \sum_{i=1}^n \sum_{k=1}^K w_{k,i} \left[\frac{1}{2} \log |\Sigma_k| + \frac{1}{2} (\vec{x}_i - \hat{\vec{\mu}}_k)^T \hat{\Sigma}_k^{-1} (\vec{x}_i - \hat{\vec{\mu}}_k) \right] + D$$

constant

The EM Algorithm (cont.)

- M-step (Maximization)

- Maximize $\Phi_b(\Theta, \hat{\Theta})$ with respect to $\vec{\mu}_k$

$$\frac{d(\mathbf{x}^T \mathbf{C} \mathbf{x})}{d\mathbf{x}} = (\mathbf{C} + \mathbf{C}^T) \mathbf{x}$$

and Σ_k^{-1} is symmetric here

$$\Phi_b(\Theta, \hat{\Theta}) = - \sum_{i=1}^n \sum_{k=1}^K w_{k,i} \left[\frac{1}{2} \log |\hat{\Sigma}_k| + \frac{1}{2} (\vec{x}_i - \hat{\mu}_k)^T \hat{\Sigma}_k^{-1} (\vec{x}_i - \hat{\mu}_k) \right] + D$$

$$\frac{\partial \Phi_b(\Theta, \hat{\Theta})}{\partial \hat{\mu}_k} = - \sum_{i=1}^n w_{k,i} \cdot \frac{1}{2} \cdot (2) \cdot \hat{\Sigma}_k^{-1} (\vec{x}_i - \hat{\mu}_k) (-1) = 0$$

$$\Rightarrow \hat{\mu}_k = \frac{\sum_{i=1}^n w_{k,i} \cdot \vec{x}_i}{\sum_{i=1}^n w_{k,i}} = \frac{\sum_{i=1}^n \frac{P(\vec{x}_i | c_k, \Theta) P(c_k | \Theta)}{\sum_{l=1}^K P(\vec{x}_i | c_l, \Theta) P(c_l | \Theta)} \cdot \vec{x}_i}{\sum_{i=1}^n \frac{P(\vec{x}_i | c_k, \Theta) P(c_k | \Theta)}{\sum_{l=1}^K P(\vec{x}_i | c_l, \Theta) P(c_l | \Theta)}}$$

r_k^i : the expected number of times that \vec{x}_i falls in class c_k

The EM Algorithm (cont.)

- M-step (Maximization)

- Maximize $\Phi_b(\Theta, \hat{\Theta})$ with respect to Σ_k

$$\Phi_b(\Theta, \hat{\Theta}) = - \sum_{i=1}^n \sum_{k=1}^K w_{k,i} \left[\frac{1}{2} \log |\hat{\Sigma}_k| + \frac{1}{2} (\bar{x}_i - \hat{\mu}_k)^T \hat{\Sigma}_k^{-1} (\bar{x}_i - \hat{\mu}_k) \right] + D$$

$$\frac{\partial \Phi_b(\Theta, \hat{\Theta})}{\partial \hat{\Sigma}_k} = - \sum_{i=1}^n \frac{1}{2} \cdot w_{k,i} \left[\frac{1}{|\hat{\Sigma}_k|} \cdot |\hat{\Sigma}_k| \cdot \hat{\Sigma}_k^{-1} - \Sigma_k^{-1} (\bar{x}_i - \hat{\mu}_k) (\bar{x}_i - \hat{\mu}_k)^T \Sigma_k^{-1} \right] = 0$$

$$\Rightarrow \sum_{i=1}^n w_{k,i} \cdot \hat{\Sigma}_k^{-1} = \sum_{i=1}^n w_{k,i} \cdot \hat{\Sigma}_k^{-1} (\bar{x}_i - \hat{\mu}_k) (\bar{x}_i - \hat{\mu}_k)^T \hat{\Sigma}_k^{-1}$$

$$\Rightarrow \sum_{i=1}^n w_{k,i} \cdot \hat{\Sigma}_k \hat{\Sigma}_k^{-1} \hat{\Sigma}_k = \sum_{i=1}^n w_{k,i} \cdot \hat{\Sigma}_k \hat{\Sigma}_k^{-1} (\bar{x}_i - \hat{\mu}_k) (\bar{x}_i - \hat{\mu}_k)^T \hat{\Sigma}_k^{-1} \hat{\Sigma}_k$$

$$\Rightarrow \sum_{i=1}^n w_{k,i} \cdot \hat{\Sigma}_k = \sum_{i=1}^n w_{k,i} \cdot (\bar{x}_i - \hat{\mu}_k) (\bar{x}_i - \hat{\mu}_k)^T$$

$$\Rightarrow \hat{\Sigma}_k = \frac{\sum_{i=1}^n w_{k,i} \cdot (\bar{x}_i - \hat{\mu}_k) (\bar{x}_i - \hat{\mu}_k)^T}{\sum_{i=1}^n w_{k,i}} = \frac{\sum_{i=1}^n \frac{P(\bar{x}_i | c_k, \Theta) P(c_k | \Theta)}{\sum_{l=1}^K P(\bar{x}_i | c_l, \Theta) P(c_l | \Theta)} \cdot (\bar{x}_i - \hat{\mu}_k) (\bar{x}_i - \hat{\mu}_k)^T}{\sum_{i=1}^n \frac{P(\bar{x}_i | c_k, \Theta) P(c_k | \Theta)}{\sum_{l=1}^K P(\bar{x}_i | c_l, \Theta) P(c_l | \Theta)}}$$

$$\frac{d[\det(\mathbf{X})]}{d\mathbf{X}} = \det(\mathbf{X}) \cdot \mathbf{X}^{-T}$$

and Σ_k is symmetric here

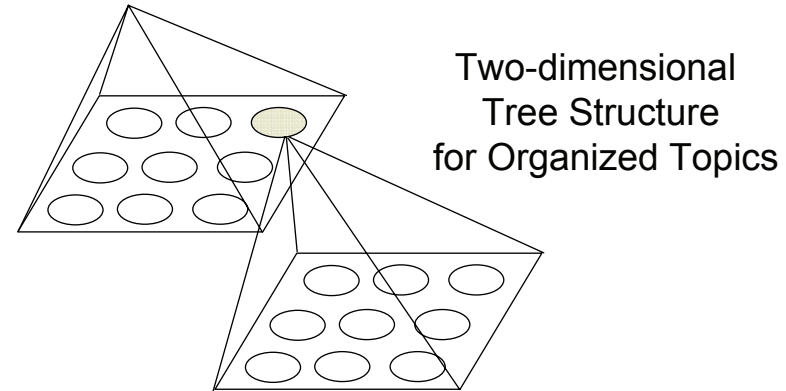
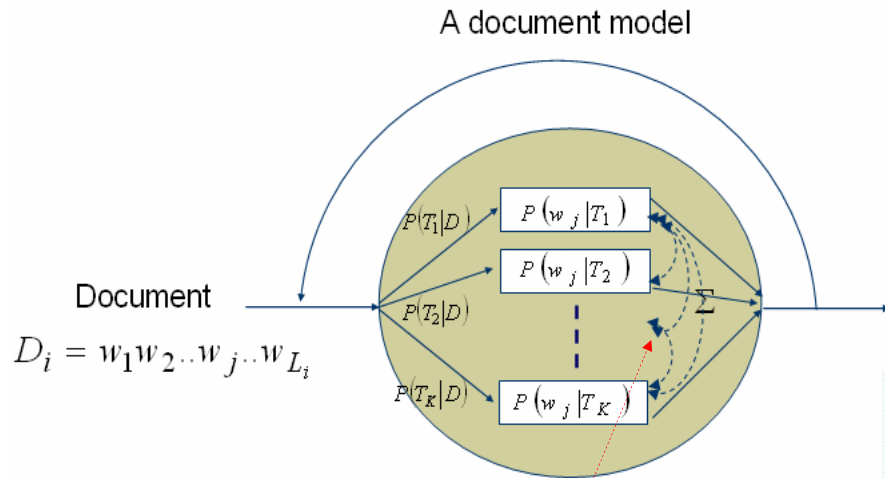
$$\frac{d(\mathbf{a}^T \mathbf{X}^{-1} \mathbf{b})}{d\mathbf{X}} = -\mathbf{X}^{-1} \mathbf{a} \mathbf{b}^T \mathbf{X}^{-1}$$

The EM Algorithm (cont.)

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

Hierarchical Document Organization

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



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

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

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

- 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 (cont.)

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

$$L_T = \sum_{i=1}^N \sum_{n=1}^J c(w_j, D_i) \log P(w_j | D_i)$$

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

- EM training can be performed

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

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

where

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

Hierarchical Document Organization (cont.)

- Criterion for Topic Word Selecting

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

Hierarchical Document Organization (cont.)

- Example

The screenshot shows a web browser window with the following content:

Left Sidebar: Title/Summary Generation Demo System
 National Taiwan University
 Speech Processing Laboratory

Navigation Menu:
 國外政治 Topic Map
 國內國會 Topic Map
 國外社會 Topic Map
 國外財經 Topic Map
 國內財經 Topic Map
 地方政府 Topic Map
 國內政治 Topic Map
 國內交通 Topic Map
 國內影劇 Topic Map
 國外體育 Topic Map
 國內社會 Topic Map
 大陸社會 Topic Map
 國外醫藥 Topic Map
 國外影劇 Topic Map
 大陸財經 Topic Map
 國內文教 Topic Map
 國內體育 Topic Map
 國內醫藥 Topic Map
 大陸政治 Topic Map

Main Content Area: 3x3 Grid of Topic Maps

<p>聯邦調查局 執法 劃歸 空對空飛彈 安全部 艾希克羅 蓋達組織 接種 等級 民航機 認出 輻射性 劫機 主謀 重整旗鼓 歐瑪 穆勒 國土 黃色 葉門 美國境內 申請局 天花 芮吉</p>	<p>僑界 僑務 台商 會長 僑胞 呼吸 雙十國慶 酒會 立委 舉辦 國慶 聯誼會 經文 履新 組長 衛生 餐會 春節 滬太華 後援 中華 僑團 華僑 鄉親</p>	<p>法輪 鈴木宗男 巫統 中國共產黨 李光耀 挪用 書記 交替 班子 馬哈地 一邊 李顯龍 吳作棟 新疆 論說 軍委 政治局 標題 馬來人 早報 格局 資政 接班 報章</p>
<p>檢查人員 檢查員 動武 最後通牒 安理會 布里斯克斯 決議 精密 武檢 聯合國 授權 沙丹· 銷毀 違禁 解除 武檢人員 檢查 首席 武器 決議案 胡笨 禁航區 導引 毀滅性</p>	<p>西非 衛隊 巴格達機場 伊拉克部隊 伊拉克南部 賴比瑞亞 伊北 科威特 步兵 辛格 庫德族 斯拉 法新社 翁山蘇姬 庫克 蒙羅維亞 巴格達 陸戰隊 轟炸 激戰 卡達 克里 市中心 暴亂</p>	<p>林東源 金大中 漢城 南北 多邊 正常化 長官 平壤 分界線 會談 鐵路 南韓 一部 韓美 燃料 南韓 懸案 金正日 盧武鉉 朝鮮 半島 打撈 黃海 銜接 核子 北韓</p>
<p>普查 支領 王太 王室 登基 會計年度 小泉內閣 瑪格麗特 問卷 靈樞 溫莎堡 英鎊 西敏寺 大廳 白金漢宮 社會勞工黨 王太后 加班 女王 降至 百分點 享年 伊麗莎白 太后 大關</p>	<p>自殺 加薩市 炸彈 巴勒斯坦 賊鎮 約旦河 巴勒斯坦人 哈瑪斯 喪生 耶路撒冷 阿拉法特 約旦河西岸 以色列 伯利恆 槍手 加薩走廊 夏隆 颶風 西岸 受傷 特拉維夫 以色列部隊 包圍 巴士</p>	<p>中美洲 決選 薩爾瓦多 哥斯大黎加 中間 兼職 雷朋 宏都拉斯 羅賓 馬達加斯加 史瓦濟蘭 翁岳生 王金平 勳章 院長 金哥納 馬拉坎南宮 游錫方 右派 雅羅 查維斯 哥斯班 孟代爾 方士</p>

Hierarchical Document Organization (cont.)

- Example (cont.)

Level - 2 - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
← 上一頁 → 搜尋 我的最愛 媒體
網址(AD) D:\R\topic_map\trmm_2_7.html
Google Search 1 blocked Check AutoLink Options

印度 瓦巴依 巴基斯坦 新德里 南亞 巴國	新加坡 澳洲 峇里島 印尼 峇里島 印尼 伯爾	墜毀 科威特 北韓 阿富汗 挾持人質
出任 柯瑞 阿巴斯 阿拉法特 雷馬拉 任命	受傷 炸彈 巴格達 自殺 巴士 至少	協議 摩洛哥 內戰 斯里蘭卡 象牙海岸 蘇丹
城鎮 約旦河 坦克 決議案 民營 西岸	和平 中東 鮑爾 約旦 埃及 路線	群眾 示威 遊行 抗議 委內瑞拉 巴拉圭

file://D:/R/topic_map/trmm_2_66.html

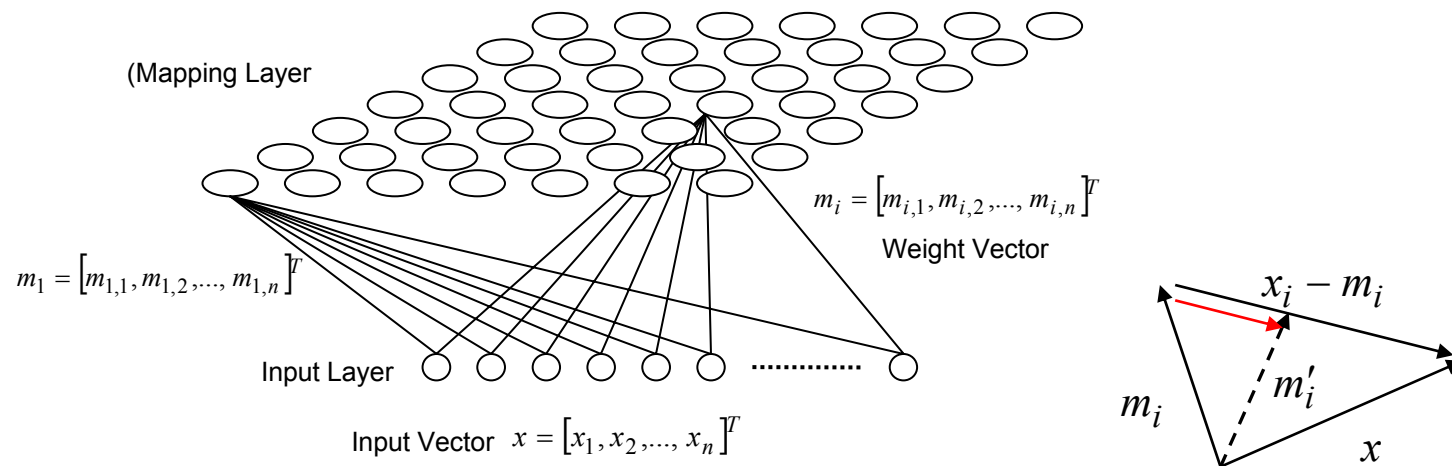
News List - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
← 上一頁 → 搜尋 我的最愛 媒體
網址(AD) D:\R\topic_map\trmm_2_66.html
Google Search 1 blocked Check AutoLink Auto

N200205031200-19: [阿拉法特原則接受歐盟所提中東和平計劃 \[summary\]](#)
N200205061200-25: [美英就解除阿拉法特所受包圍與巴方展開談判 \[summary\]](#)
N200209201200-14: [阿拉法特反對以色列保所提結束包圍條件 \[summary\]](#)
N200210301200-22: [阿拉法特宣布新內閣弓\[發巴勒斯坦國會激辯 \[summary\]](#)
N200211011200-24: [阿拉伯人支持阿拉法特及巴勒斯坦人正當抵抗 \[summary\]](#)

我的電腦

Hierarchical Document Organization (cont.)

- 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 (cont.)

- Results

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

$$R_{\text{Dist}} = \frac{\text{dist}_{\text{Between}}}{\text{dist}_{\text{Within}}}$$

where

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

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

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

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

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

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

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