Clustering Techniques

Berlin Chen 2003

References:

1. Modern Information Retrieval, chapters 5, 7
2. Foundations of Statistical Natural Language Processing, Chapter 14

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
- Clustering is a way of learning

Clustering vs. Classification

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

- 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
- **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

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**
  - 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

- Hierarchical clustering usually adopts hard assignment
- While in flat clustering both types of clustering 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 x n matrix of similarity coefficients)

Summarized Attributes of Clustering Algorithms

- 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
  - The EM algorithm is the most choice. It can accommodate definition of clusters and allocation of objects based on complex probabilistic models

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 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

Hierarchical Agglomerative Clustering (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

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

$c_j = c_{n_1} \cup c_{n_2}$

merged as a new cluster

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

The original two clusters are removed

$j := j + 1$

Figure 14.2 Bottom-up hierarchical clustering.

Distance Metrics

- Euclidian Distance (*L_2* norm)

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

- *L_1* Norm

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

- 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

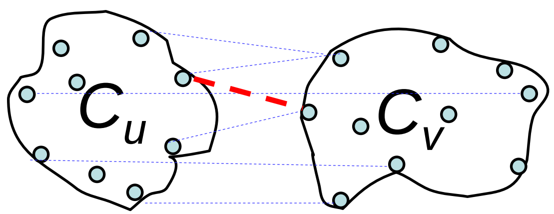
Measures of Cluster Similarity

- 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

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

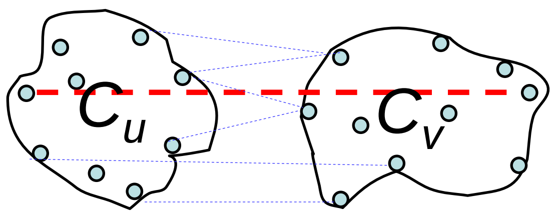


greatest similarity

Measures of Cluster Similarity

- **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

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



least similarity

Measures of Cluster Similarity

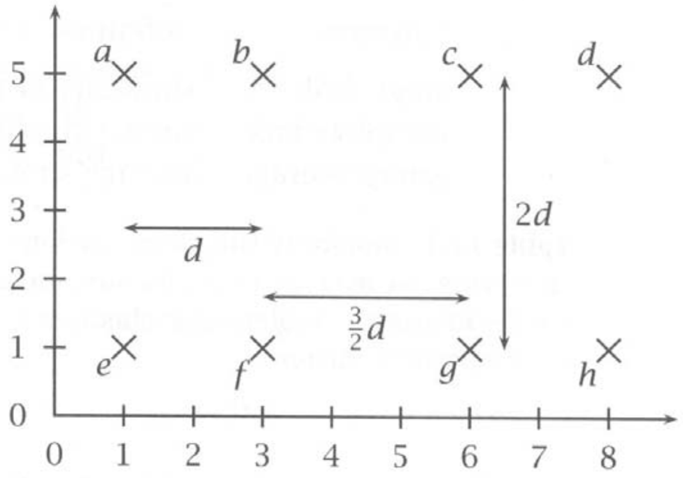


Figure 14.4 A cloud of points in a plane.

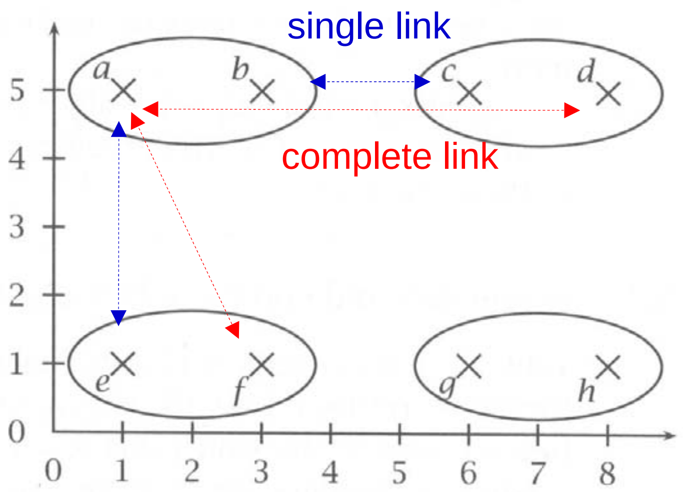


Figure 14.5 Intermediate clustering of the points in figure 14.4.

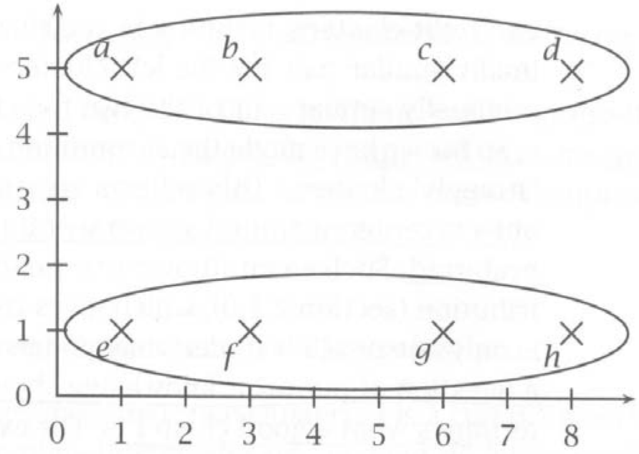


Figure 14.6 Single-link clustering of the points in figure 14.4.

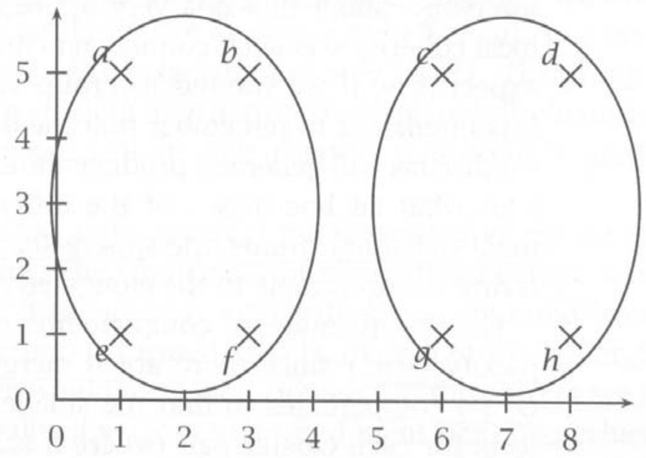


Figure 14.7 Complete-link clustering of the points in figure 14.4.

Measures of Cluster Similarity

- **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

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

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

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

  – 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 $\vec{s}(c_j)$

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

[length-normalized vector]

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

[=1]

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

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

Measures of Cluster Similarity

- **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

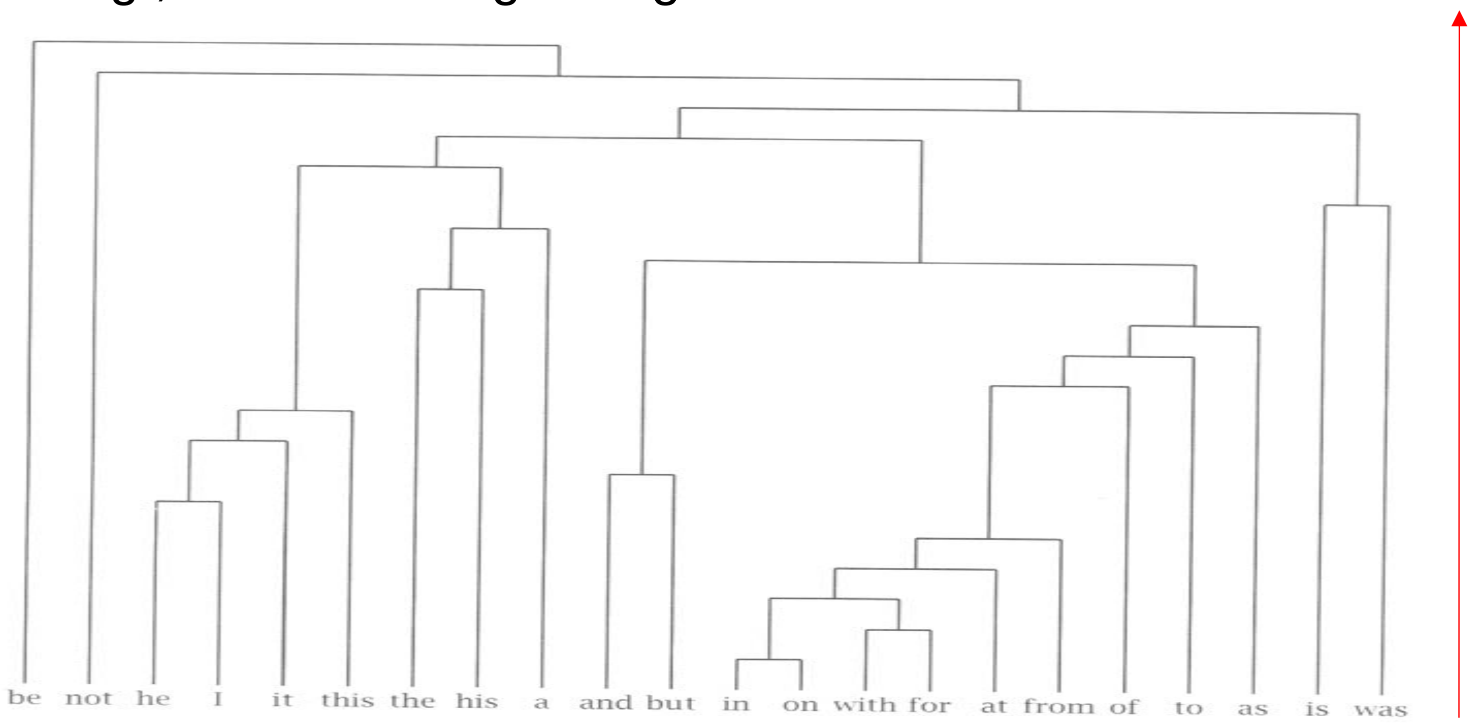
$$\Rightarrow \quad \vec{s}(c_{New}) = \vec{s}(c_i) + \vec{s}(c_j), \quad |c_{New}| = |c_i| + |c_j|$$

– 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



higher nodes: decreasing of similarity

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

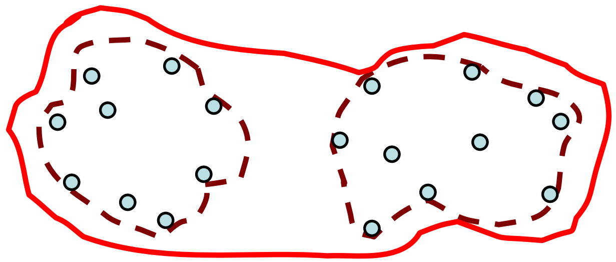
"be" has least similarity with the other 21 words !

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

- To select the least coherent cluster, the measures used in bottom-up clustering 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. $|c_i| > 1$
$c_u := \arg\min_{c_v \in C} \mathrm{coh}(c_v)$
$(c_{j+1}, c_{j+2}) := \mathrm{split}(c_u)$
$C := C \backslash \{c_u\} \cup \{c_{j+1}, c_{j+2}\}$
$j := j + 2$

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

- **Problems** associated non-hierarchical clustering
  - When to stop   MI, group average similarity, likelihood
  - What is the right number of clusters

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

Hierarchical clustering also has to face this problem

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

The *K*-means Algorithm

- A hard clustering algorithm
- Define clusters by the **center of mass** of their members
- **Initialization**
  - A set of initial cluster centers is needed
- **Recursion**
  - Assign each object to the cluster whose center is closet
  - 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)

The *K*-means Algorithm

- Algorithm

Given: a set $\mathcal{X} = \{\vec{x}_1, \ldots, \vec{x}_n\} \subseteq \mathbb{R}^m$

a distance measure $d : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}$

a function for computing the mean $\mu : \mathcal{P}(\mathbb{R}) \rightarrow \mathbb{R}^m$

Select k initial centers $\vec{f}_1, \ldots, \vec{f}_k$

while stopping criterion is not true **do**

  for all clusters c_j **do**

    $c_j = \{\vec{x}_i \mid \forall \vec{f}_l \; d(\vec{x}_i, \vec{f}_j) \leq d(\vec{x}_i, \vec{f}_l)\}$

  end

  for all means $\vec{f}_j$ **do**

    $\vec{f}_j = \mu(c_j)$

  end

end

cluster

centroid

cluster assignment

calculation of new centroids

Figure 14.8 The K-means clustering algorithm.

The *K*-means Algorithm

- Example 1

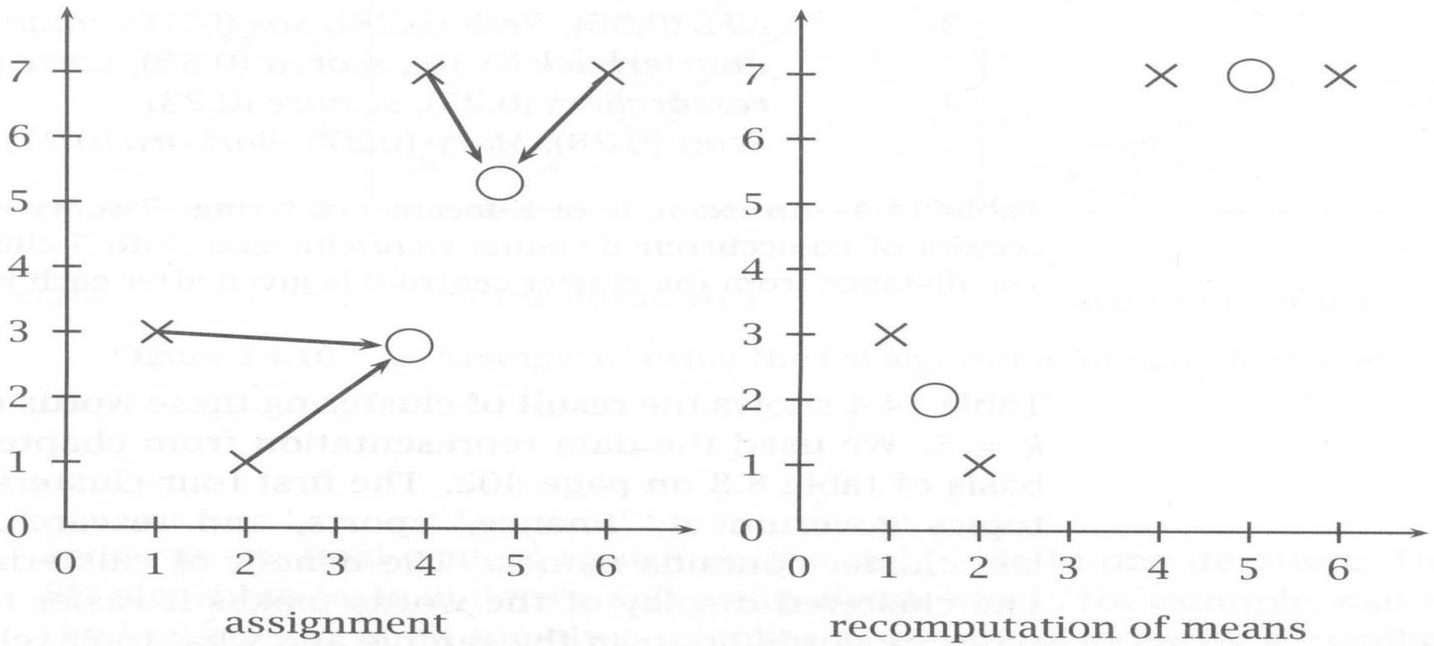


Figure 14.9 One iteration of the K-means algorithm. The first step assigns objects to the closest cluster mean. Cluster means are shown as circles. The second step recomputes cluster means as the center of mass of the set of objects that are members of the cluster.

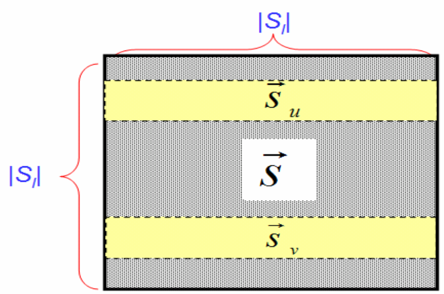
The *K*-means Algorithm

- 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.

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

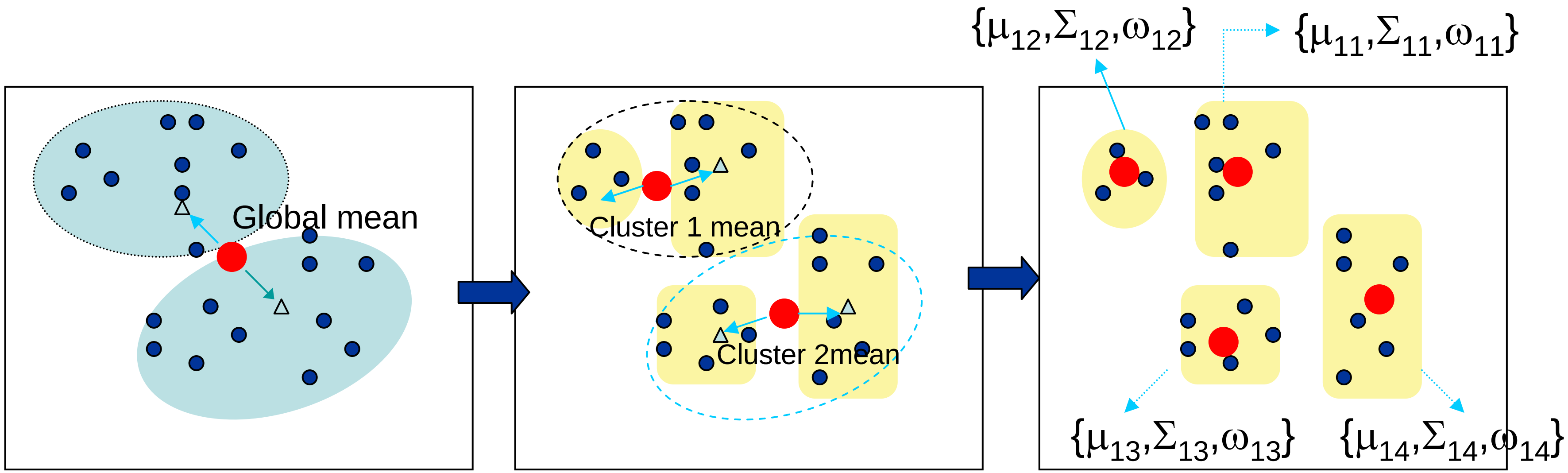


The *K*-means Algorithm

- Choice of initial cluster centers (seeds) is important
  - Pick at random
  - 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
- 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

The *K*-means Algorithm

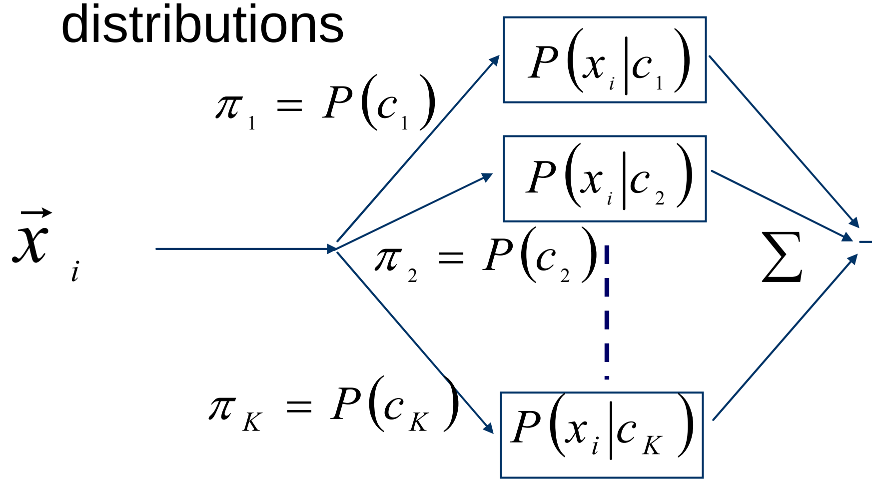
- E.g., the LBG algorithm
  - By Linde, Buzo, and Gray



M→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



A Mixture Gaussian HMM
(or A Mixture of Gaussians)

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

classification:

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

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

Continuous case:

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

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

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

The EM Algorithm

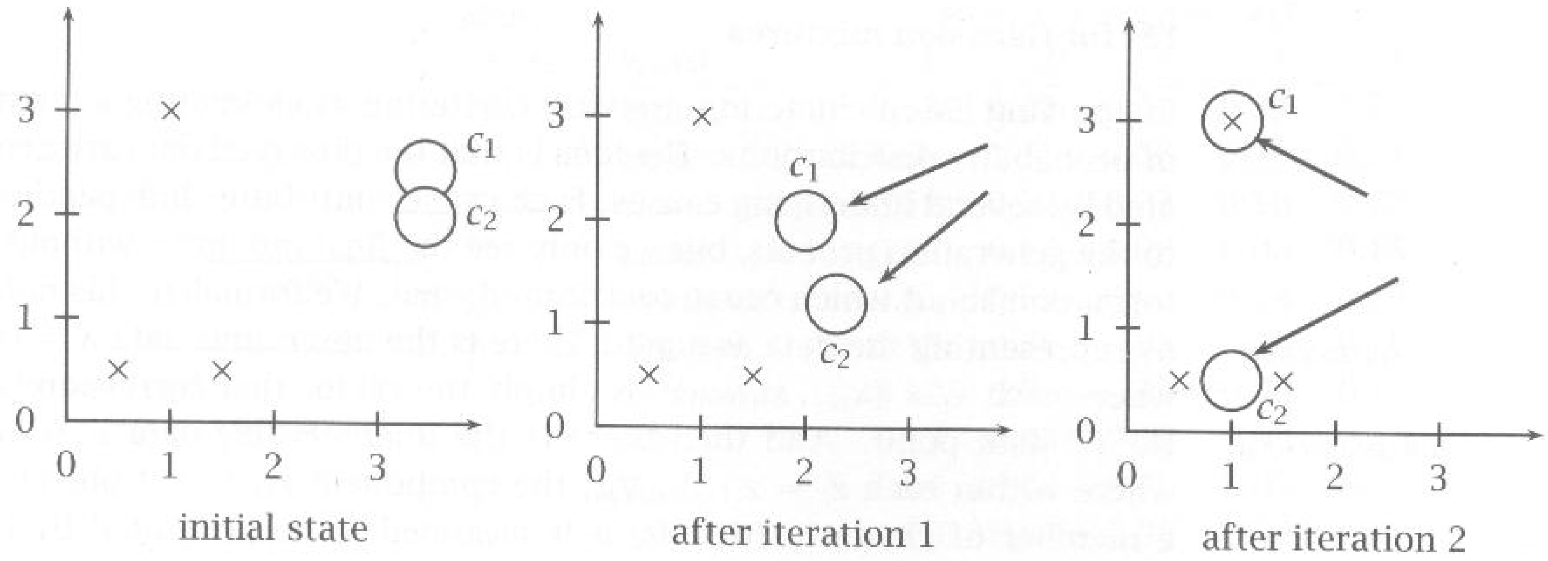


Figure 14.10 An example of using the EM algorithm for soft clustering.

The EM Algorithm

Note :

$$\prod_{t=1}^{T}\left(\sum_{k_t=1}^{M} a_{tk_t}\right)$$
$$=(a_{11}+a_{12}+...+a_{1M})(a_{21}+a_{22}+...+a_{2M})...(a_{T1}+a_{T2}+...+a_{TM})$$
$$=\sum_{k_1=1}^{M}\sum_{k_2=1}^{M}...\sum_{k_T=1}^{M}\prod_{t=1}^{T} a_{tk_t}$$

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

likelihood function

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

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

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

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

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

$$\boldsymbol{X}=\vec{x}_1\vec{x}_2\cdots\vec{x}_{n-1}\vec{x}_n$$
$$\boldsymbol{C}=c_{k_1}c_{k_2}\cdots c_{k_{n-1}}c_{k_n}$$

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

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

$$=\sum_{\boldsymbol{C}}\left[P\left(\boldsymbol{X},\boldsymbol{C}\middle|\hat{\Theta}\right)\right]$$

the complete data likelihood function

The EM Algorithm

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

$$\Phi\left(\Theta, \hat{\Theta}\right) = E\left[\log L^{CM}\right]_{\boldsymbol{C}|\boldsymbol{X},\Theta} = E\left[\log P\left(\boldsymbol{X}, \boldsymbol{C} \middle| \hat{\Theta}\right)\right]_{\boldsymbol{C}|\boldsymbol{X},\Theta}$$

$$= \sum_{\boldsymbol{C}} P\left(\boldsymbol{C} \middle| \boldsymbol{X}, \Theta\right) \log P\left(\boldsymbol{X}, \boldsymbol{C} \middle| \hat{\Theta}\right)$$

$$= \sum_{\boldsymbol{C}} \frac{P\left(\boldsymbol{X}, \boldsymbol{C} \middle| \Theta\right)}{P\left(\boldsymbol{X} \middle| \Theta\right)} \log P\left(\boldsymbol{X}, \boldsymbol{C} \middle| \hat{\Theta}\right)$$

  - Maximize the log likelihood function $\log P\left(\boldsymbol{X} \middle| \hat{\Theta}\right)$ by maximizing the expectation of the log complete likelihood function $\Phi\left(\Theta, \hat{\Theta}\right)$
    - We have shown this when deriving the HMM-based retrieval model

The EM Algorithm

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

$$\Phi\left(\Theta,\hat{\Theta}\right)=\sum_{\boldsymbol{C}}\frac{P\left(\boldsymbol{X},\boldsymbol{C}\middle|\Theta\right)}{P\left(\boldsymbol{X}\middle|\Theta\right)}\log P\left(\boldsymbol{X},\boldsymbol{C}\middle|\hat{\Theta}\right)$$

$$=\sum_{\boldsymbol{C}=c_{k_1}c_{k_2}\cdots c_{k_n}}\left[\prod_{j=1}^{n}\frac{P\left(\vec{x}_j,c_{k_j}\middle|\Theta\right)}{P\left(\vec{x}_j\middle|\Theta\right)}\right]\left[\log\prod_{i=1}^{n}P\left(\vec{x}_i,c_{k_i}\middle|\hat{\Theta}\right)\right]$$

$$=\sum_{\boldsymbol{C}=c_{k_1}c_{k_2}\cdots c_{k_n}}\left[\prod_{j=1}^{n}P\left(c_{k_j}\middle|\vec{x}_j,\Theta\right)\right]\left[\sum_{i=1}^{n}\log P\left(\vec{x}_i,c_{k_i}\middle|\hat{\Theta}\right)\right]$$

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

$$=\sum_{\boldsymbol{C}=c_{k_1}c_{k_2}\cdots c_{k_n}}\sum_{k=1}^{m}\left\{\delta_{k,k_i}\left[\sum_{i=1}^{n}\log P\left(\vec{x}_i,c_k\middle|\hat{\Theta}\right)\right]\left[\prod_{j=1}^{n}P\left(c_{k_j}\middle|\vec{x}_j,\Theta\right)\right]\right\}$$

$$=\sum_{k=1}^{m}\sum_{i=1}^{n}\left\{\left[\log P\left(\vec{x}_i,c_k\middle|\hat{\Theta}\right)\right]\sum_{\boldsymbol{C}=c_{k_1}c_{k_2}\cdots c_{k_n}}\delta_{k,k_i}\left[\prod_{j=1}^{n}P\left(c_{k_j}\middle|\vec{x}_j,\Theta\right)\right]\right\}$$

?

$$=\sum_{k=1}^{m}\sum_{i=1}^{n}\left\{\left[\log P\left(\vec{x}_i,c_k\middle|\hat{\Theta}\right)\right]P\left(c_k\middle|\vec{x}_i,\Theta\right)\right\}$$

$$=\sum_{k=1}^{m}\sum_{i=1}^{n}\left\{P\left(c_{k_j}\middle|\vec{x}_j,\Theta\right)\log P\left(\vec{x}_i,c_k\middle|\hat{\Theta}\right)\right\}$$

$$=\sum_{k=1}^{m}\sum_{i=1}^{n}\left\{P\left(c_{k_j}\middle|\vec{x}_j,\Theta\right)\log\left[P\left(\vec{x}_i\middle|c_k,\hat{\Theta}\right)P\left(c_k\middle|\hat{\Theta}\right)\right]\right\}$$

$$=\sum_{k=1}^{m}\sum_{i=1}^{n}\left\{P\left(c_{k_j}\middle|\vec{x}_j,\Theta\right)\log P\left(c_k\middle|\hat{\Theta}\right)\right\}+\sum_{k=1}^{m}\sum_{i=1}^{n}\left\{P\left(c_{k_j}\middle|\vec{x}_j,\Theta\right)\log P\left(\vec{x}_i\middle|c_k,\hat{\Theta}\right)\right\}$$

The EM Algorithm

Note :

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

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

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

– Note that

$$\sum_{C=c_{k_1}c_{k_2}\ldots c_{k_n}} \delta_{k,k_i}\left[\prod_{j=1}^{n} P\left(c_{k_j}\middle|\vec{x}_j,\Theta\right)\right]$$

$$=\sum_{c_{k_1}=1}^{m}\sum_{c_{k_2}=1}^{m}\cdots\sum_{c_{k_n}=1}^{m}\prod_{j=1}^{n}\left[\delta_{k,k_i} P\left(c_{k_j}\middle|\vec{x}_j,\Theta\right)\right]$$

$$=\sum_{c_{k_1}=1}^{m}\sum_{c_{k_2}=1}^{m}\cdots\sum_{c_{k_n}=1}^{m}\prod_{j=1}^{n}\left[\delta_{k,k_i} P\left(c_{k_j}\middle|\vec{x}_j,\Theta\right)\right]$$

$$=\left[\prod_{j=1,\,j\neq i}^{n}\left[\sum_{k_j=1}^{m} P\left(c_{k_j}\middle|\vec{x}_j,\Theta\right)\right]\right]\left[\sum_{c_{k_i}=1}^{m}\delta_{k,k_i} P\left(c_{k_i}\middle|\vec{x}_i,\Theta\right)\right]$$

$$=\left[\prod_{j=1,\,j\neq i}^{n} 1\right] P\left(c_k\middle|\vec{x}_i,\Theta\right)$$

$\vec{x}_i$ can only be aligned to c_k

$$=P\left(c_k\middle|\vec{x}_i,\Theta\right)$$

The EM Algorithm

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

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

where

$$\Phi_a\left(\Theta, \hat{\Theta}\right) = \sum_{i=1}^{n}\sum_{k=1}^{K} P\left(c_k \middle| \vec{x}_i, \Theta\right) \log P\left(c_k \middle| \hat{\Theta}\right)$$

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

auxiliary function for mixture weights

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

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

auxiliary function for cluster distributions

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

The EM Algorithm

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

By applying Lagrange Multiplier ℓ

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

Constraint

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

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

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

Note :

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

The EM Algorithm

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

auxiliary function for
mixture weights (or priors for Gaussians)

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

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

w_k y_k

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

The EM Algorithm

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

auxiliary function for
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 $\log P(\vec{x}_i | c_k; \Theta) =$

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

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

constant (D)

The EM Algorithm

- M-step (Maximization)
  - Maximize $\Phi_b\left(\Theta, \hat{\Theta}\right)$ with respect to $\vec{\mu}_k$

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

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

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

and Σ_k^{-1} is symmetric here

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

The EM Algorithm

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

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

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

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

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

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

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

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

and Σ_k is symmetric here

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

The EM Algorithm

- The initial cluster distributions can be estimated using the *K*-means algorithm

- The procedure terminates when the likelihood function $P\left(X \mid \Theta\right)$ is converged or maximum number of iterations is reached