Lecture 21 - Mar 20

Gaussian Mixture Clustering (by EM algorithm)
Hierarchical Clustering

References

Data Mining and Machine Learning
Ch 13 - Representation-based Clustering
Ch 14 - Hierarchical Clustering

Elements of Statistical Learning
14.3.7 Gaussian Mixtures as Soft K-Means Clustering
14.3.12 Hierarchical Clustering

Generalize k-means to soft assignment where each point has a probability of being in each cluster.

Assume each cluster C_i is characterized by a normal distribution

Assume pdf of $x \in X$ is a Gaussian mixture over the cluster normal densities

$$f(x) = \sum_{i=1}^{k} f_i(x) P(C_i) = \sum_{i=1}^{k} f_i(x \mid \mu_i, \Sigma_i) P(C_i)$$

priors = "mixture parameters"

Gaussian mixture model

Assume each cluster C_i is characterized by a multivariate normal distribution

multivariate normal distribution

$$f_i(x) = f(x \mid \mu_i, \Sigma_i) = \frac{1}{(2\pi)^{d/2}|\Sigma_i|^{1/2}} \exp\left\{ -\frac{(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)}{2} \right\}$$

unknown parameters

Assume pdf of $x \in X$ is a Gaussian mixture over the cluster normal densities

$$f(x) = \sum_{i=1}^{k} f_i(x) P(C_i) = \sum_{i=1}^{k} f_i(x \mid \mu_i, \Sigma_i) P(C_i)$$

priors = "mixture parameters"

Model parameters $\Theta = \{ \mu_i, \Sigma_i, P(C_i) \mid i = 1, 2, \ldots, k \}$

Gaussian Mixture Clustering in 1D

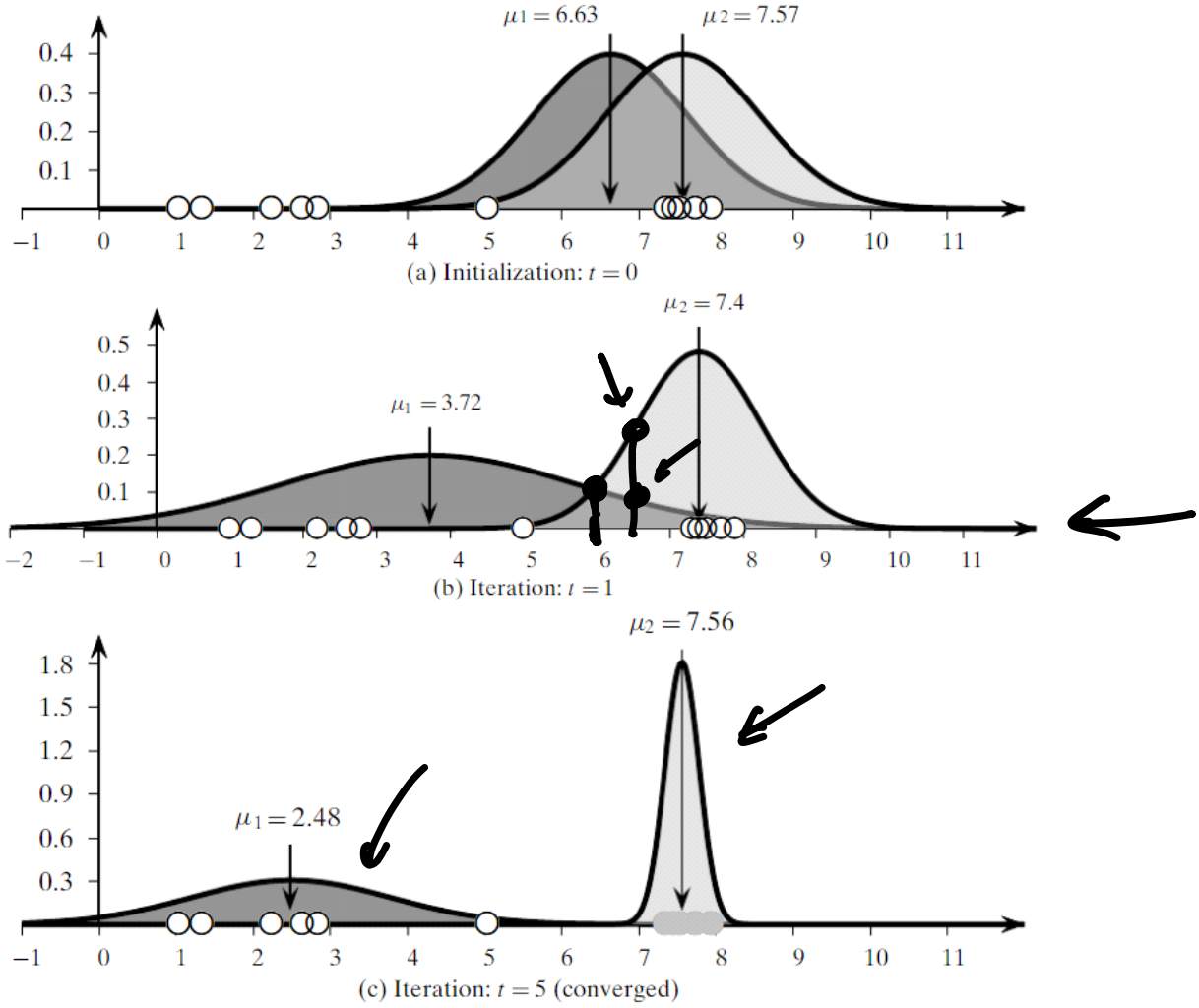


Figure 13.4. EM in one dimension.

Gaussian Mixture Clustering in 2D

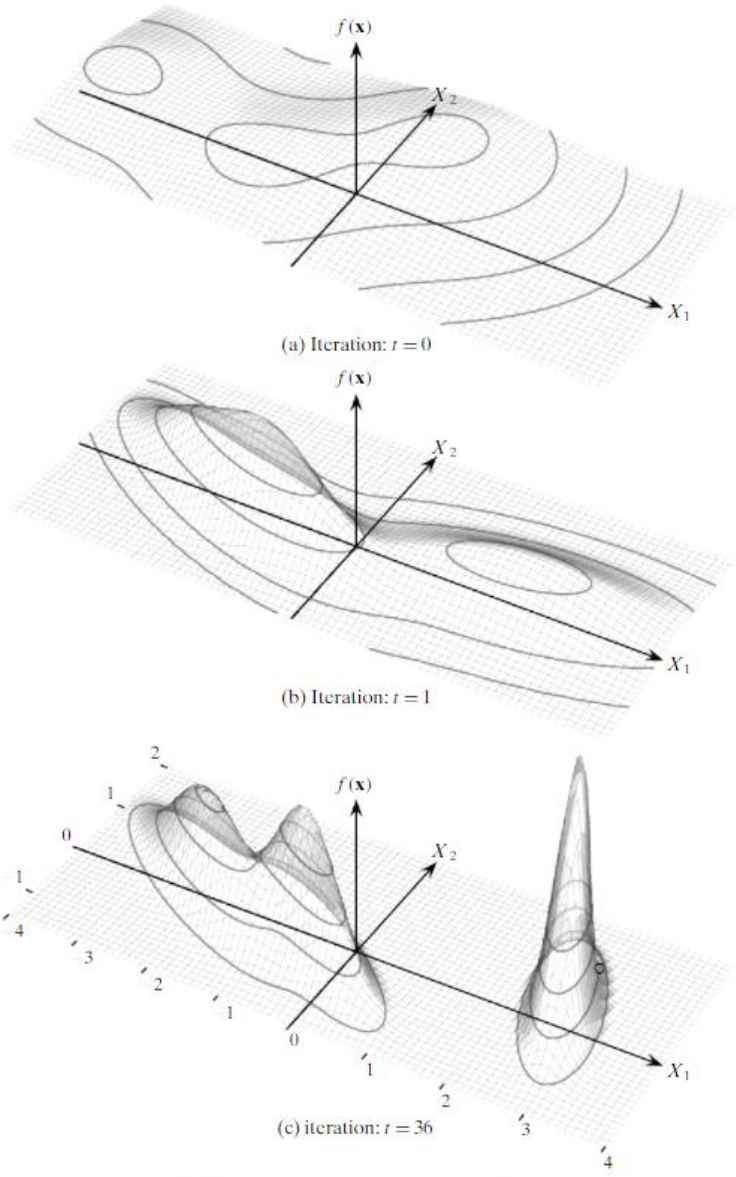


Figure 13.5. EM algorithm in two dimensions: mixture of $k = 3$ Gaussians.

Ch 12 lab

Maximum Likelihood Estimation of Θ

Assuming $x_1, \ldots, x_n$ from a sample, likelihood of Θ is

$$\mathbb{P}(X|\Theta) = \prod_{j=1}^{n} f(x_j)$$

Goal: $\Theta^* = \arg\max_{\Theta} \{\mathbb{P}(X|\Theta)\} = \arg\max_{\Theta} \{\ln(\mathbb{P}(X|\Theta))\}$

$$\ln(\mathbb{P}(X|\Theta)) = \sum_{j=1}^{n} \ln(f(x_j))$$

$$= \sum_{j=1}^{n} \ln\left(\sum_{i=1}^{k} f(x_j|\mu_i, \Sigma_i)\mathbb{P}(C_i)\right)$$

Maximizing by hand is typically infeasible, so EM algorithm can iteratively seek Θ with two steps:

0. Make an initial estimate for Θ

1. Given estimated Θ, compute posterior probabilities with

1. Given estimated θ, compute posterior P

Bayes' Theorem

"expectation step"

$$P(C_i|x_j) = \frac{P(x_j|C_i)P(C_i)}{\sum_{a=1}^{k} P(x_j|C_a)P(C_a)}$$

using $P(x_j|C_i) \approx 2\Sigma f_i(x)$

$$\Rightarrow P(C_i|x_j) \approx \frac{f_i(x_j)P(C_i)}{\sum_{a=1}^{k} f_a(x_j)P(C_a)}$$

2. Given weights $P(C_i|x_j)$, estimate $\mu_i, \Sigma_i, P(C_i)$ for each C_i.

"maximization step"

Initialization

initialize μ_i uniformly randomly in feature space

$$\Sigma_i = I_d$$

$$P(C_i) = \frac{1}{k}$$

Expectation Step

Expectation Step

Compute posterior probability $\underbrace{P(C_i|x_j)}_{w_{ij}}$ for all i, j.

as $$w_{ij} = \frac{f_i(x_j)P(C_i)}{\sum_{a=1}^{k} f_a(x_j)P(C_a)}$$

Maximization Step

Use posterior probabilities w_{ij} to re-compute $\mu_i, \Sigma_i, P(C_i)$

$$\mu_i = \frac{\sum_{j=1}^{n} w_{ij} x_j}{\sum_{j=1}^{n} w_{ij}}$$

$$\Sigma_i = \frac{\sum_{j=1}^{n} w_{ij} (x_j - \mu_i)(x_j - \mu_i)^T}{\sum_{j=1}^{n} w_{ij}}$$

Considering the pairwise attribute view, the covariance between dimensions X_a and X_b is estimated as

$$\sigma_{ab}^{i} = \frac{\sum_{j=1}^{n} w_{ij}(x_{ja} - \mu_{ia})(x_{jb} - \mu_{ib})}{\sum_{j=1}^{n} w_{ij}}$$

$$P(C_i) = \frac{\sum_{j=1}^{n} w_{ij}}{n}$$

K-means as a Special Case of EM

Instead of multivariate normal cluster distribution, let

$$P(x_j|C_i) = \begin{cases} 1, & \text{if } \underset{C_a}{\operatorname{argmax}}\{\|x_j - m_a\|^2\} \\ 0, & \text{otherwise} \end{cases}$$

posterior probabilities are

$$P(C_i|x_j) = \frac{P(x_j|C_i)P(C_i)}{\sum_{a=1}^{k} P(x_j|C_a)P(C_a)}$$

(binary; 0 if 0; $\frac{1 \cdot P(C_i)}{1 \cdot P(C_i)} = 1$ if 1)

$$= \begin{cases} 0, & \text{if } i \neq \underset{i}{\operatorname{argmax}}\{\|x_j - m_i\|^2\} \\ 1, & \text{if } i = \underset{i}{\operatorname{argmax}}\{\|x_j - m_i\|^2\} \end{cases}$$

$\Rightarrow$ only m_i + $P(C_i)$ are parameters

$\Rightarrow$ simplifies to K-mean

$X \in \mathbb{R}^{n \times d}$, $x_i \in \mathbb{R}^d$, $\mathcal{E} = \{E_1, \dots, E_k\}$ set of clusters (a "clustering")

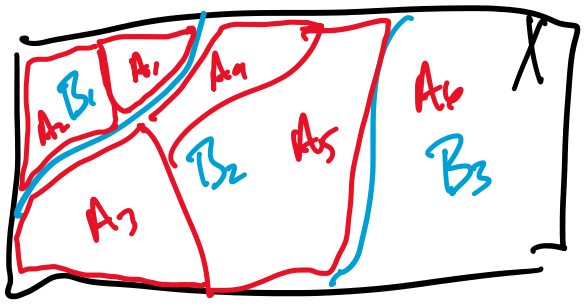
($E_1, \dots, E_k$ partition X)

$E_i \cap E_j = \emptyset$ for $i \neq j$

$E_1 \cup \dots \cup E_k = X$

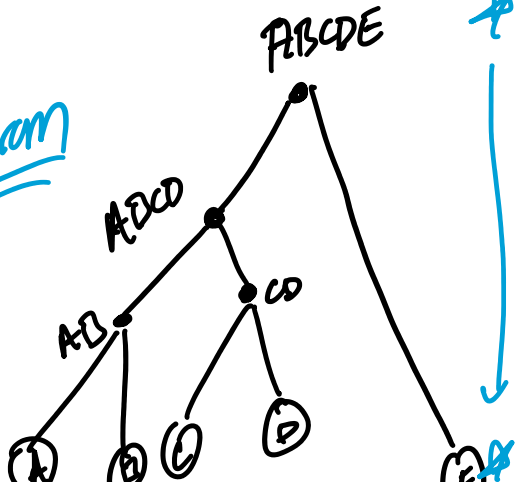
Consider clusterings $\mathcal{A} = \{A_1, \dots, A_r\}$ and $\mathcal{B} = \{B_1, \dots, B_s\}$

$\mathcal{A}$ is <u>nested</u> in $\mathcal{B}$ if for $r > s$, each $A_i \subseteq B_j$ for some i, j



Hierarchical clustering produces a hierarchy of clusterings from $\mathcal{E}_1 = \{\{x_1\}, \{x_2\}, \dots, \{x_n\}\}$ up to $\mathcal{E}_n = \{\{x_1, \dots, x_n\}\}$ where each $\mathcal{E}_{t-1}$ is nested in $\mathcal{E}_t$

dendogram



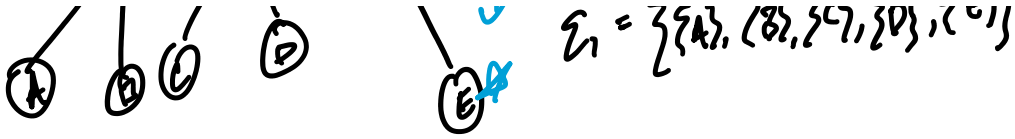
$\mathcal{E}_5 = \{\{ABCDE\}\}$

$\mathcal{E}_4 = \{\{ABCD\}, \{E\}\}$

$\mathcal{E}_3 = \{\{AB\}, \{CD\}, \{E\}\}$

$\mathcal{E}_2 = \{\{AB\}, \{C\}, \{D\}, \{E\}\}$ arbitrary choice $\mathcal{E}_2 = \{\{A\}, \{B\}, \{CD\}, \{E\}\}$

$\mathcal{E}_1 = \{\{A\}, \{B\}, \{C\}, \{D\}, \{E\}\}$



The cluster dendogram captures the nesting structure.
If $E_i \subset E_j$, we draw an edge between them.

Number of Hierarchical Clusterings : $(2n-3)!!$ → enumerative techniques fail

$(2n-3)(2n-5)(2n-7)\cdots 1$

Agglomerative techniques - "bottom-up" methods start with each pt. in its own cluster and merges them based on some criteria

Repeatedly merge the two closest clusters. If $\Sigma = \{E_1, \ldots, E_m\}$, then if E_i, E_j are the closest, $\Sigma \to \Sigma \setminus \{E_i, E_j\} \cup \{E_i \cup E_j\}$

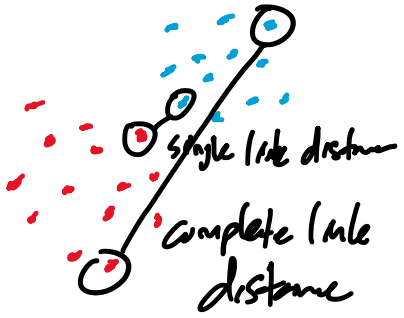
E_{ij}

Repeat until Σ is one cluster of all points or until $|\Sigma| = k$ (prespecified)

Algorithm 14.1: Agglomerative Hierarchical Clustering Algorithm

```
AGGLOMERATIVECLUSTERING(D, k):
1 C ← {C_i = {x_i} | x_i ∈ D} // Each point in separate cluster
2 Δ ← {||x_i − x_j||: x_i, x_j ∈ D} // Compute distance matrix
3 repeat
4     Find the closest pair of clusters C_i, C_j ∈ C
5     C_ij ← C_i ∪ C_j // Merge the clusters
6     C ← (C \ {C_i, C_j}) ∪ {C_ij} // Update the clustering
7     Update distance matrix Δ to reflect new clustering
8 until |C| = k
```

Several distance measures exist. - based on distance between points, e.g. l^2 distance.



Single Link $\delta(E_i, E_j) = \min\{\|x-y\| : x \in E_i, y \in E_j\}$

min dist. between a point in E_i + a point in E_j

Complete Link $\delta(E_i, E_j) = \max\{\|x-y\| : x \in E_i, y \in E_j\}$

max dist. between a pt in E_i + a point in E_j. If we connected points from clusters with distance at most $\delta(E_i, E_j)$, we would connect <u>all</u> pairs.

Group Average $\delta(E_i, E_j) = \dfrac{\sum_{x \in E_i} \sum_{y \in E_j} \|x-y\|}{n_i \cdot n_j}$

Average pairwise distance $(|C_i| = n_i)$

Mean Distance $\delta(E_i, E_j) = \|\mu_i - \mu_j\|$

Mean dist between centroids

Min Variance: Ward's Method

$$SSE_i = \sum_{x \in E_i} \|x-\mu_i\|^2 = \sum_{x \in E_i} \|x\|^2 - n_i \|\mu_i\|^2$$

$\frac{m}{\sum} SSE_i$

SSE of Σ: $\sum_{i=1}^{m} SSE_i$

$\Rightarrow \delta(E_i, E_j) = \Delta SSE_{ij} = SSE_{ij} - SSE_i - SSE_j$

Net change in SSE if E_i, E_j are merged

$$\delta(E_i, E_j) = n_i \|M_i\|^2 + n_j \|M_j\|^2 - (n_i + n_j)\|M_{ij}\|^2$$

where $M_{ij} = \frac{n_i M_i + n_j M_j}{n_i + n_j}$ (weighted average)

$$\delta(E_i, E_j) = \frac{n_i n_j}{n_i + n_j} \|M_i - M_j\|^2$$

When E_i, E_j merge to form E_{ij}, we need to update the distance from E_{ij} to each E_r, $r \neq i, j$

Lance-Williams formula allows recomputing distances for all the cluster proximity measures

$$\delta(E_{ij}, E_r) = \alpha_i \delta(E_i, E_r) + \alpha_j \delta(E_j, E_r) + \beta \delta(E_i, E_j) + \gamma |\delta(E_i, E_r) - \delta(E_j, E_r)|$$

Measure	α_i	α_j	β	γ
Single Link	$\frac{1}{2}$	$\frac{1}{2}$	0	$-\frac{1}{2}$
Complete Link	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$
Group Average	$\frac{n_i}{n_i + n_j}$	$\frac{n_j}{n_i + n_j}$	0	0
Mean Distance	$\frac{n_i}{n_i + n_j}$	$\frac{n_j}{n_i + n_j}$	$\frac{-n_i n_j}{(n_i + n_j)^2}$	0

Mean Distance	$\frac{n_i}{n_i+n_j}$	$\frac{n_j}{n_i+n_j}$	$\frac{-n_i n_j}{(n_i+n_j)^2}$	0
Ward's Measure	$\frac{n_i+n_r}{n_i+n_j+n_r}$	$\frac{n_j+n_r}{n_i+n_j+n_r}$	$\frac{-n_r}{n_i+n_j+n_r}$	0

Computational Complexity: $O(n^2 \log n)$