Machine Learning

Clustering

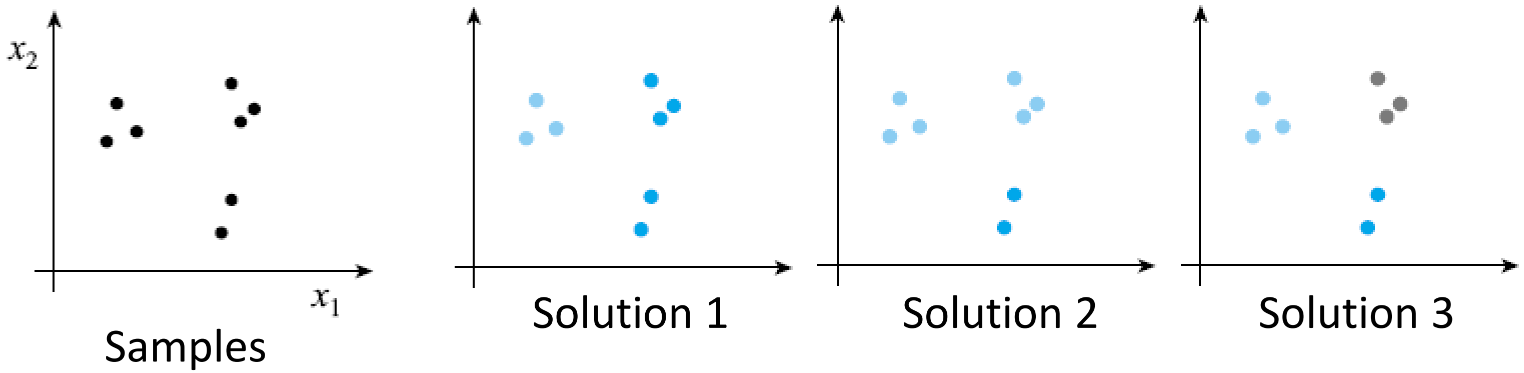
Professor: Cheolsoo Park

Supervised vs Unsupervised Learning

- **Supervised Learning**
  - With label
  - Neural Network
  - Support Vector Machine
- **Unsupervised Learning**
  - Without label
  - Clustering

Clustering

- Cluster depending on situations



Samples

Solution 1

Solution 2

Solution 3

Clustering

- Samples $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_\mathrm{N}\}$, where $\mathbf{x}_{\boldsymbol{i}} = (x_{\boldsymbol{i1}}, x_{\boldsymbol{i2}}, \dots, x_{\boldsymbol{id}})^{\boldsymbol{T}}$
- Find a clustering solution $C = \{c_1, c_2, \dots, c_k\}, k < N$
- Depending on a situation, k is known or unknown

$$\left.\begin{array}{l} c_i \neq \varnothing, i = 1, \cdots, k \\ \cup_{i=1,k}\, c_i = X \\ c_i \cap c_j = \varnothing, i \neq j \end{array}\right\}$$

- Make sample differences in the same group closer, and those in the other groups further

➔ sample difference is important for the clustering

Distance

- Minkowski distance
  - Between two samples, $\mathbf{x}_i=(x_{i1},\ldots,x_{id})^T$ and $\mathbf{x}_j=(x_{j1},\ldots,x_{jd})^T$
  - Euclidean distance (p=2), Manhattan distance (p=1)

$$d_{ij} = \left(\sum_{k=1}^{d} \left| x_{ik} - x_{jk} \right|^{p} \right)^{1/p}$$ Minkowski distance

Euclidean distance $(p=2)$ $d_{ij} = \sqrt{\sum_{k=1}^{d} (x_{ik} - x_{jk})^2}$

Manhattan distance $(p=1)$ $d_{ij} = \sum_{k=1}^{d} \left| x_{ik} - x_{jk} \right|$

- Hamming distance
  - for the binary vector (count the number of different bits)
  - Hamming distance between $(1,0,1,0,0,0,1,1)^T$ and $(1,0,0,1,0,0,1,0)^T$ is 3

Distance and similarity

- Cosine similarity
  - Used for text mining

$$s_{ij} = \cos\theta = \frac{\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j}{\|\mathbf{x}_i\|\|\mathbf{x}_j\|}$$

- Similarity between two binary feature vector

$$s_{ij} = \frac{n_{00} + n_{11}}{n_{00} + n_{11} + n_{01} + n_{10}}$$

$$s_{ij} = \frac{n_{11}}{n_{11} + n_{01} + n_{10}}$$

, where

n_{00} *and* n_{11} : both bits from two vectors are the same as 0 or 1

n_{01} *and* n_{10} : both bits from two vectors are different each other

Group distance

- Distance between a sample $\mathbf{x}_{\boldsymbol{i}}$ and a group c_j, $D(\mathbf{x}_i, c_j)$

- Distance between two groups $c_i\ and\ c_j, D(c_i, c_j)$

- Distance between a sample and a group

$$D_{\max}(\mathbf{x}_i, c_j) = \max_{\mathbf{y}_k \in c_j} d_{ik}$$

$$D_{\min}(\mathbf{x}_i, c_j) = \min_{\mathbf{y}_k \in c_j} d_{ik}$$

$$D_{\mathrm{ave}}(\mathbf{x}_i, c_j) = \frac{1}{\left|c_j\right|} \sum_{\mathbf{y}_k \in c_j} d_{ik}$$, where $\left|C_j\right|$ is # of samples

- $D_{max}\ and\ D_{min}$ are good for outlier detection

Group distance

- Distance between a sample and a group (use one sample in the group as a representative of the group)
  - Use a mean value as a representative of the group

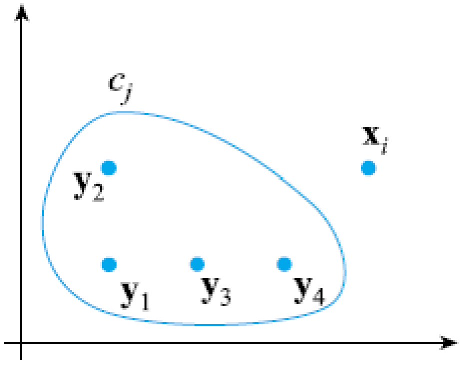
$$\left.\begin{aligned} & D_{\mathrm{mean}}(\mathbf{x}_i, c_j) = d_{i,mean} \\ & \text{, where } \quad \mathbf{y}_{mean} = \frac{1}{|c_j|} \sum_{\mathbf{y}_k \in c_j} \mathbf{y}_k \end{aligned}\right\}$$

  - Use a closest sample in a group among all

$$\left.\begin{aligned} & D_{\mathrm{rep}}(\mathbf{x}_i, c_j) = d_{i,rep} \\ & \text{, where } \quad \sum_{\mathbf{y}_k \in c_j} d_{rep,k} \le \sum_{\mathbf{y}_k \in c_j} d_{lk}, \forall \mathbf{y}_l \in c_j \end{aligned}\right\}$$

Group distance

- Example



$$c_j = \{\mathbf{y}_1 = (1,1)^T,\ \mathbf{y}_2 = (1,2)^T,\ \mathbf{y}_3 = (2,1)^T,\ \mathbf{y}_4 = (3,1)^T\},\ \mathbf{x}_i = (4,2)^T$$

* Use Euclidean distance

$$D_{\max} = \max(3.162,\ 3.0,\ 2.236,\ 1.414) = 3.162$$

$$D_{\min} = \min(3.162,\ 3.0,\ 2.236,\ 1.414) = 1.414$$

$$D_{\text{ave}} = (3.162+3.0+2.236+1.414)/4 = 2.453$$

$$\mathbf{y}_{\text{mean}} = ((1,1)^T + (1,2)^T + (2,1)^T + (3,1)^T)/4 = (1.75,1.25)^T$$

$$D_{\text{mean}} = d_{i,mean} = 2.372$$

$$\sum_{\mathbf{y}_k \in c_j} d_{1k} = 1.0 + 1.0 + 2.0 = 4.0$$

$$\sum_{\mathbf{y}_k \in c_j} d_{2k} = 1.0 + 1.414 + 2.236 = 4.65$$

$$\sum_{\mathbf{y}_k \in c_j} d_{3k} = 1.0 + 1.414 + 1.0 = 3.414$$

$$\sum_{\mathbf{y}_k \in c_j} d_{4k} = 2.0 + 2.236 + 2.0 = 6.236$$

Choose y_3
as a representative

$$D_{\text{rep}}(\mathbf{x}_i, c_j) = d_{i,rep} = 2.236$$

Group distance

- Group distance

$$D_{\max}(c_i, c_j) = \max_{\mathbf{x}_k \in c_i, \mathbf{y}_l \in c_j} d_{kl}$$

$$D_{\min}(c_i, c_j) = \min_{\mathbf{x}_k \in c_i, \mathbf{y}_l \in c_j} d_{kl}$$

$$D_{\text{ave}}(c_i, c_j) = \frac{1}{|c_i||c_j|} \sum_{\mathbf{x}_k \in c_i} \sum_{\mathbf{y}_l \in c_j} d_{kl}$$

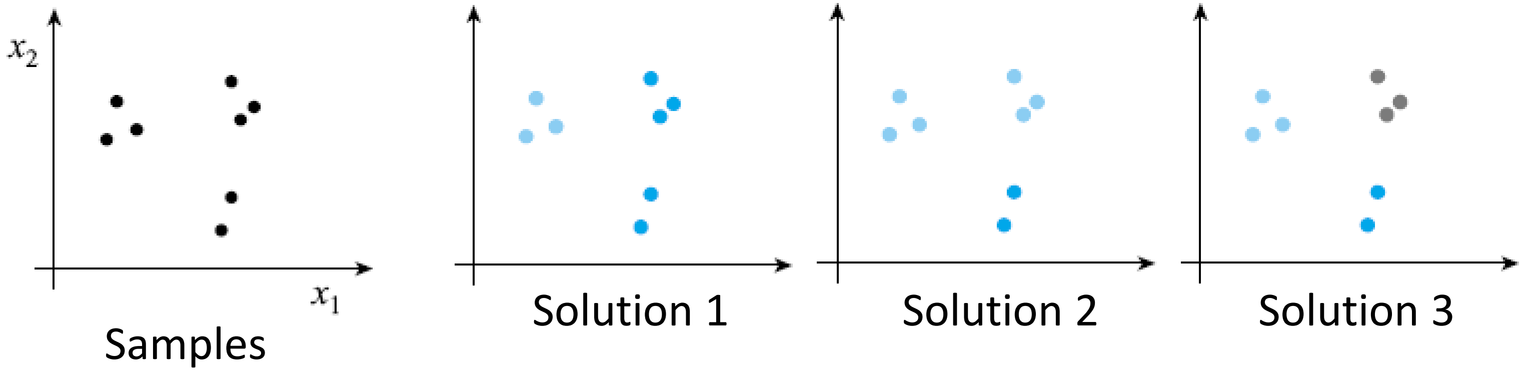
(c_i and c_j are the number of samples in the groups)

$$\left.\begin{array}{l} D_{\text{mean}}(c_i, c_j) = d_{mean1, mean2} \\ \text{, where } \quad \mathbf{x}_{mean1} = \frac{1}{|c_i|} \sum_{\mathbf{x}_k \in c_i} \mathbf{x}_k, \ \mathbf{y}_{mean2} = \frac{1}{|c_j|} \sum_{\mathbf{y}_l \in c_j} \mathbf{y}_l \end{array}\right\}$$

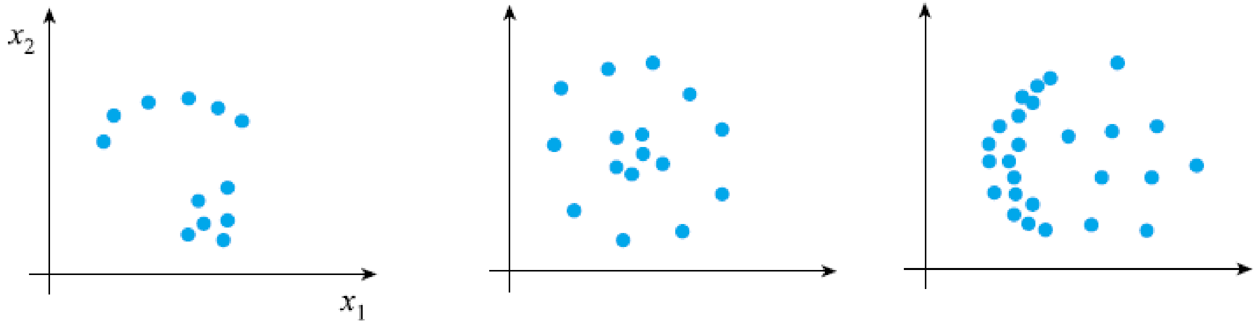
$$\left.\begin{array}{l} D_{\text{rep}}(c_i, c_j) = d_{rep1, rep2} \\ \text{, where } \quad \sum_{\mathbf{x}_k \in c_i} d_{rep1,k} \le \sum_{\mathbf{x}_k \in c_i} d_{pk}, \forall \mathbf{x}_p \in c_i, \sum_{\mathbf{y}_l \in c_j} d_{rep2,l} \le \sum_{\mathbf{y}_l \in c_j} d_{pl}, \forall \mathbf{y}_p \in c_j \end{array}\right\}$$

Clustering Algorithms

- Various type of clustering algorithms



Subjective results

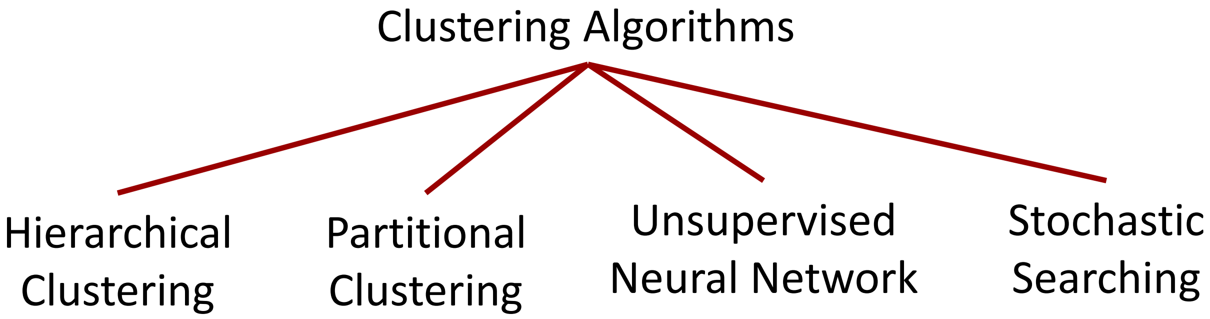


Various situations

- One algorithm works well for a dataset, but won't for the other

→ Very important to fully understand the algorithm to optimize it to data

Clustering Algorithms



Clustering Algorithms

Hierarchical Clustering

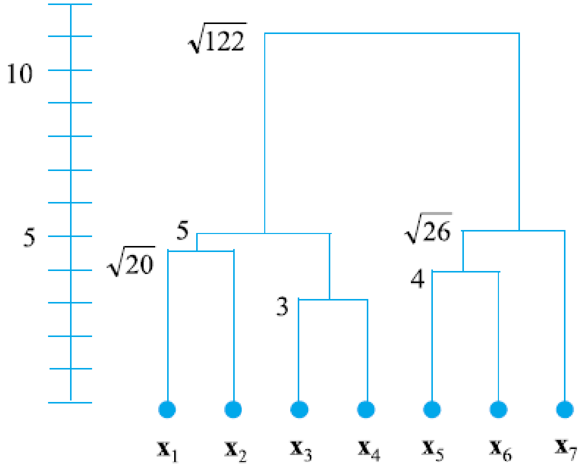
Partitional Clustering

Unsupervised Neural Network

Stochastic Searching

Hierarchical Clustering

- Hierarchy of the cluster
  - When $C_1 = \{\{\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_6\}, \{\mathbf{x}_2, \mathbf{x}_4, \mathbf{x}_5\}\}\ and\ C_2 = \{\{\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_6\}, \{\mathbf{x}_2\}, \{\mathbf{x}_4, \mathbf{x}_5\}\},\ then\ C_1 \ni C_2$
- Agglomerative : small groups → large groups
- Divisive : large groups → small groups



dendrogram

Agglomerative Hierarchical Clustering

- Each sample is a group → agglomerate small groups into large groups

- Algorithm 1
  - Input : samples X
  - Output : Dendrogram displaying the group hierarchy
  - Algorithm

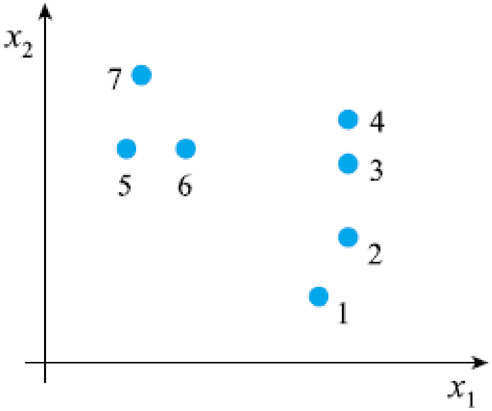
```
$C_0 = \{c_1 = \{x_1\}, c_2 = \{x_2\}, \dots, c_N = \{x_N\}$;  // each sample is considered as a group
for (t=1 to N-1) {
    $D(c_p, c_q) = \min_{c_i, c_j \in C_{t-1}} D(c_i, c_j)$        // find the most closest pair
    $c_r = c_p \cup c_q$                                  // combine $c_p$ $and$ $c_q$ into a group, $c_r$
    $C_t = (C_{t-1} - c_p - c_q) \cup c_r$             // remove $c_p$ $and$ $c_q$ and add the new group, $c_r$
}
```

Agglomerative Hierarchical Clustering

- Example

$\mathbf{x}_1 = (18,5)^T$, $\mathbf{x}_2 = (20,9)^T$, $\mathbf{x}_3 = (20,14)^T$, $\mathbf{x}_4 = (20,17)^T$, $\mathbf{x}_5 = (5,15)^T$, $\mathbf{x}_6 = (9,15)^T$,

$\mathbf{x}_7 = (6,20)^T$

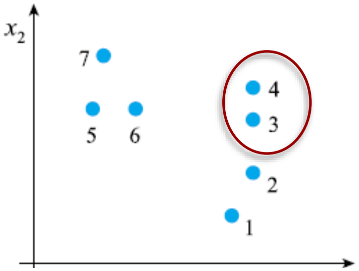


Step 1: $C_0 = \{c_1 = \{\mathbf{x}_1\},\ c_2 = \{\mathbf{x}_2\},\ c_3 = \{\mathbf{x}_3\},\ c_4 = \{\mathbf{x}_4\},\ c_5 = \{\mathbf{x}_5\},\ c_6 = \{\mathbf{x}_6\},\ c_7 = \{\mathbf{x}_7\}\}$

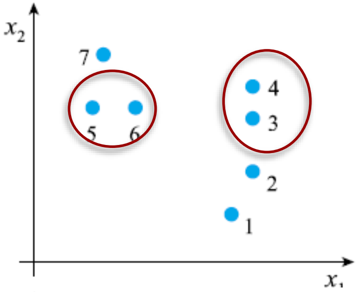
Agglomerative Hierarchical Clustering

(repeat the loop in Algorithm 1 with using Euclidean Dist. and D_{min})

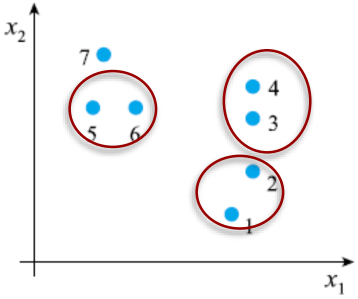
$C_1 = \{c_1 = \{\mathbf{x}_1\},\ c_2 = \{\mathbf{x}_2\},\ c_3 = \{\mathbf{x}_3, \mathbf{x}_4\},\ c_4 = \{\mathbf{x}_5\},\ c_5 = \{\mathbf{x}_6\},\ c_6 = \{\mathbf{x}_7\}\}$

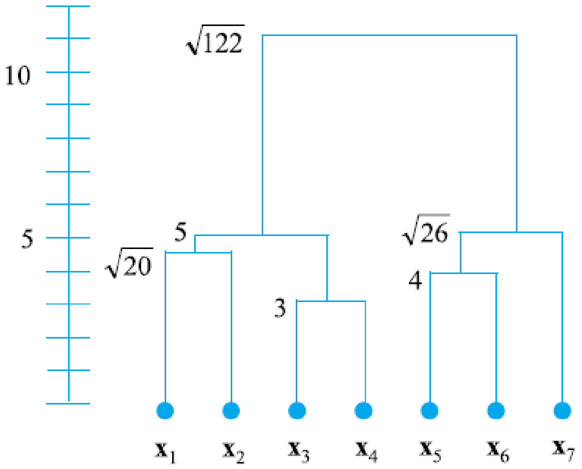


$C_2 = \{c_1 = \{\mathbf{x}_1\},\ c_2 = \{\mathbf{x}_2\},\ c_3 = \{\mathbf{x}_3, \mathbf{x}_4\},\ c_4 = \{\mathbf{x}_5, \mathbf{x}_6\},\ c_5 = \{\mathbf{x}_7\}\}$



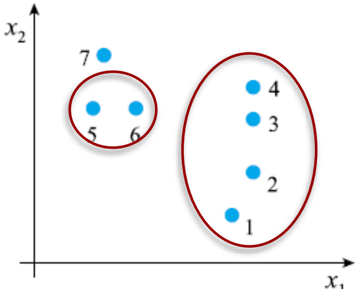
$C_3 = \{c_1 = \{\mathbf{x}_1, \mathbf{x}_2\},\ c_2 = \{\mathbf{x}_3, \mathbf{x}_4\},\ c_3 = \{\mathbf{x}_5, \mathbf{x}_6\},\ c_4 = \{\mathbf{x}_7\}\}$



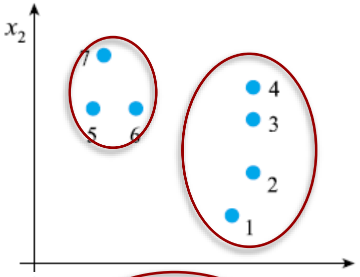


Agglomerative Hierarchical Clustering

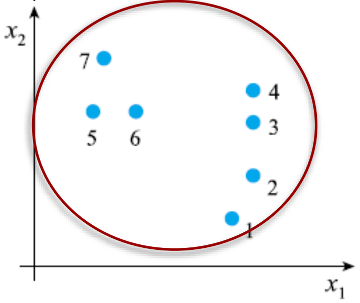
$$C_4 = \{c_1 = \{\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\mathbf{x}_4\},\ c_2 = \{\mathbf{x}_5,\mathbf{x}_6\},\ c_3 = \{\mathbf{x}_7\}\}$$

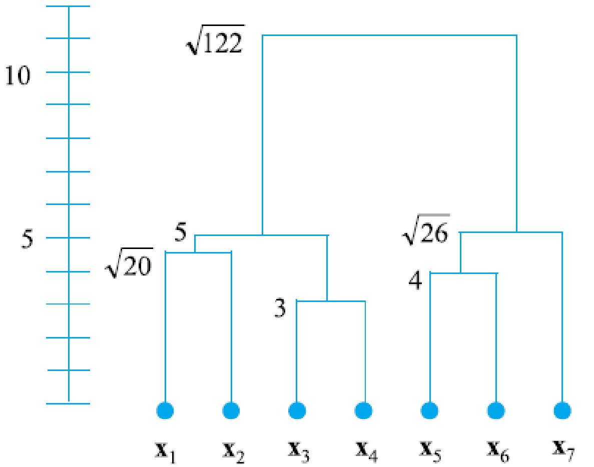


$$C_5 = \{c_1 = \{\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\mathbf{x}_4\},\ c_2 = \{\mathbf{x}_5,\mathbf{x}_6,\mathbf{x}_7\}\}$$



$$C_6 = \{c_1 = \{\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\mathbf{x}_4,\mathbf{x}_5,\mathbf{x}_6,\mathbf{x}_7\}\}$$



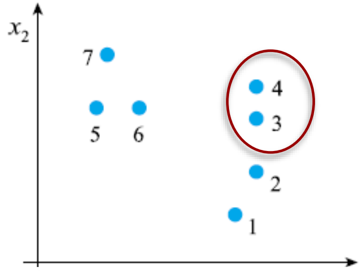


Agglomerative Hierarchical Clustering

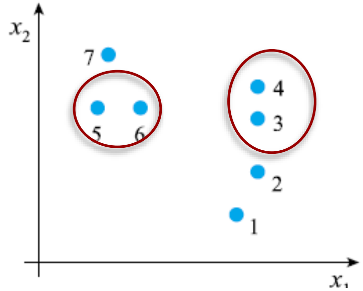
- When we use D_{max} for the distance measurement

$C_0 = \{c_1 = \{x_1\}, c_2 = \{x_2\}, c_3 = \{x_3\}, c_4 = \{x_4\}, c_5 = \{x_5\}, c_6 = \{x_6\}, c_7 = \{x_7\}\}$

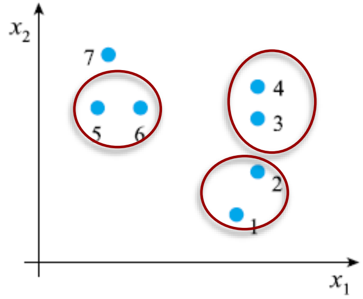
$C_1 = \{c_1 = \{x_1\}, c_2 = \{x_2\}, c_3 = \{x_3, x_4\}, c_4 = \{x_5\}, c_5 = \{x_6\}, c_6 = \{x_7\}\}$



$C_2 = \{c_1 = \{x_1\}, c_2 = \{x_2\}, c_3 = \{x_3, x_4\}, c_4 = \{x_5, x_6\}, c_5 = \{x_7\}\}$

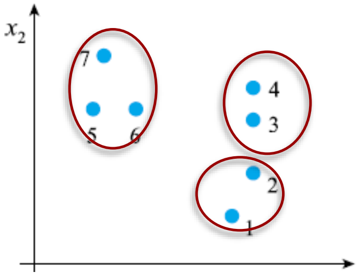


$C_3 = \{c_1 = \{x_1, x_2\}, c_2 = \{x_3, x_4\}, c_3 = \{x_5, x_6\}, c_4 = \{x_7\}\}$

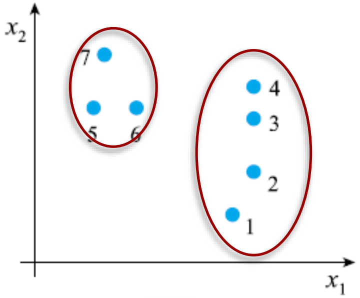


Agglomerative Hierarchical Clustering

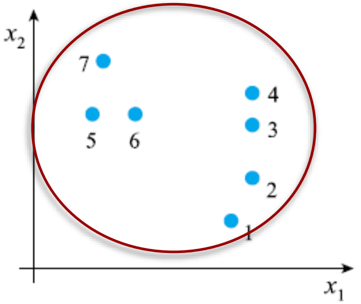
$C_4 = \{c_1 = \{x_1, x_2\}, c_2 = \{x_3, x_4\}, c_3 = \{x_5, x_6, x_7\}\}$

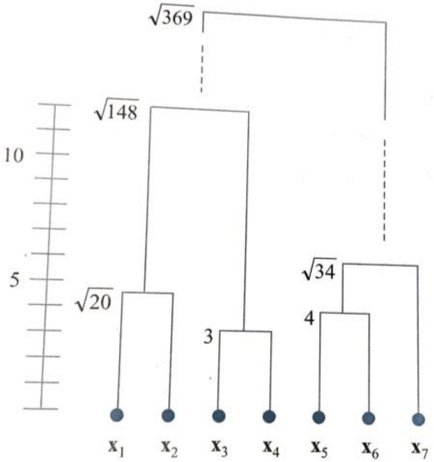


$C_5 = \{c_1 = \{x_1, x_2, x_3, x_4\}, c_2 = \{x_5, x_6, x_7\}\}$



$C_6 = \{c_1 = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}\}$



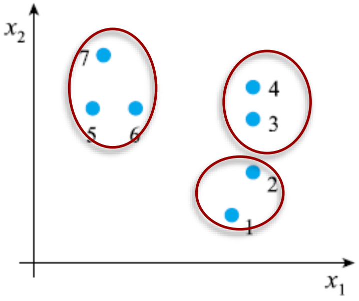


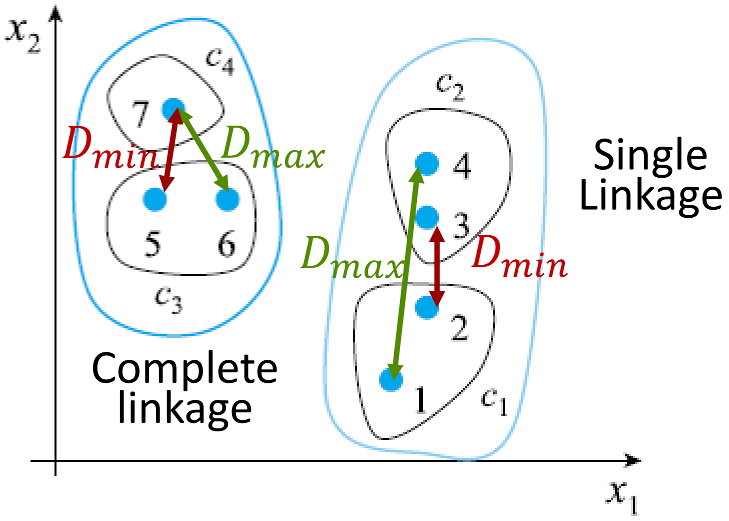
Agglomerative Hierarchical Clustering

- Single-linkage using D_{min}
- Complete-linkage using D_{max}
- Average-linkage using D_{ave}
- Single-linkage for long-shaped cluster, complete-linkage for round-shaped cluster, and average-linkage for in-between

t=4

Complete linkage



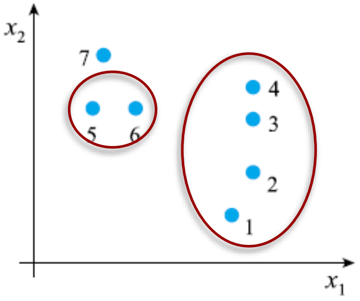


Single Linkage

Complete linkage

t=4

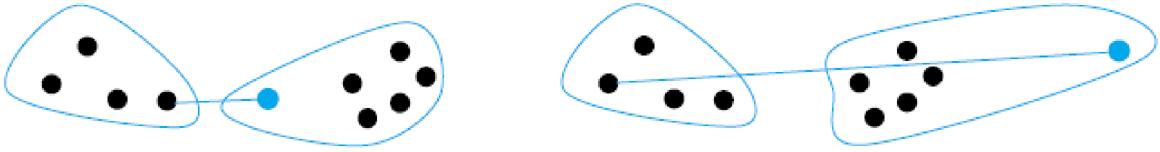
Single Linkage



Agglomerative Hierarchical Clustering

- How to get the number of clusters?
  - General issue for most clustering algorithms
  - User would define the number or be defined automatically, but automatic way is hard

- Single/complete-linkages are sensitive to outliers, while average-linkage is less



Single-linkage Complete-linkage

- Computational complexity
  - O(N^3), where N denotes # of samples
  - High complexity

Partitional Clustering

- Sequential algorithm
- K-means algorithm
- MST algorithm
- GMM algorithm
- …

k-means algorithm

- Most popular clustering algorithm
- Intuitive and easy implementation
- ***We need to provide # of clusters***
- Algorithm 2
  - Input : sample $X = \{x_1, x_2, \dots, x_N\}$ and # of clusters, k
  - Output : clusters C
  - Algorithm

```
Initialize the center of clusters $Z = \{z_1, z_2, \dots, z_k\}$
while (TRUE) {
    for (i=1 to N) assign $x_i$ to the nearest cluster
    for (j=1 to k) calculate the cluster center with considering newly assigned samples
    if (the centers are the same as the previous one) break;
}
```

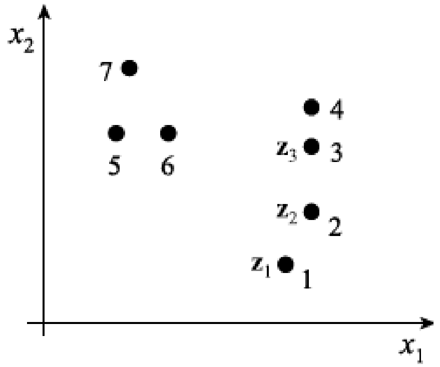
k-means algorithm

- Example

cluster 7 samples into 3 clusters (k=3)

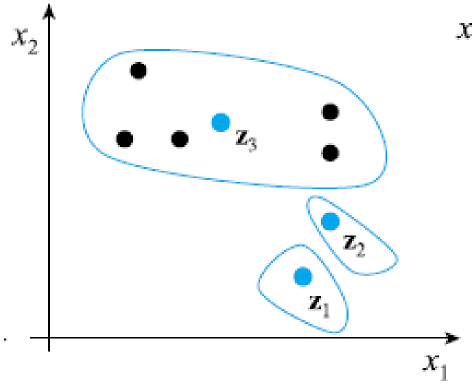
$\mathbf{x}_1 = (18,5)^T,\ \mathbf{x}_2 = (20,9)^T,\ \mathbf{x}_3 = (20,14)^T,\ \mathbf{x}_4 = (20,17)^T,\ \mathbf{x}_5 = (5,15)^T,\ \mathbf{x}_6 = (9,15)^T,$
$\mathbf{x}_7 = (6,20)^T$

arbitrary initialize the center of clusters as $\{x_1\}, \{x_2\}, \{x_3\}$



1st loop

$$z_1 = (18,5)^T$$
$$z_2 = (20,9)^T$$
$$z_3 = \frac{x_3 + x_4 + x_5 + x_6 + x_7}{5} = (12,16.2)^T$$



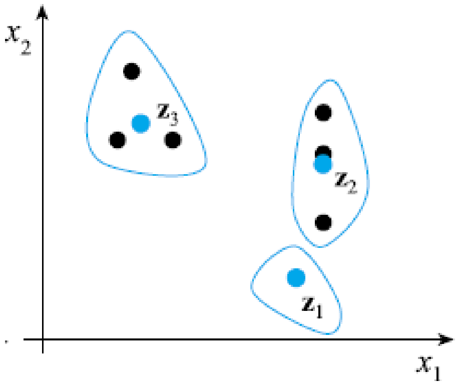
k-means algorithm

2nd loop

$$\mathbf{z}_1 = \mathbf{x}_1 = (18,5)^T$$

$$\mathbf{z}_2 = (\mathbf{x}_2+\mathbf{x}_3+\mathbf{x}_4)/3 = (20,13.333)^T$$

$$\mathbf{z}_3 = (\mathbf{x}_5+\mathbf{x}_6+\mathbf{x}_7)/3 = (6.667,16.667)^T$$



3rd loop

same as the previous one, then break

Final Output

$$C = \{\{\mathbf{x}_1\},\ \{\mathbf{x}_2,\mathbf{x}_3,\mathbf{x}_4\},\ \{\mathbf{x}_5,\mathbf{x}_6,\mathbf{x}_7\}\}$$

k-means algorithm

- Always converged to local minimum
- Sensitive to the initial centers
- Fast
- Sensitive to the outliers

Difference between Hierarchical and Partitional Clustering

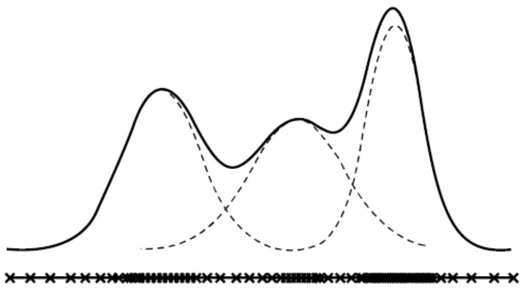
- In conclusion, the main differences between Hierarchical and Partitional Clustering are that each cluster starts as singletons or individual clusters . With every iteration, the closest clusters get merged. This process repeats until one single cluster remains for Hierarchical clustering.
  - *An example of Hierarchical clustering is the Two-Step clustering method.*

- Whereas, Partitional clustering requires the analyst to define K number of clusters before running the algorithm and objects closest to the clusters are grouped. With every iteration, the distance of the clusters shifts. This process continues until there is no more movement in the centroid of each cluster or until the stopping criterion is met.
  - *An example of Partitional clustering is the K-Means clustering method.*

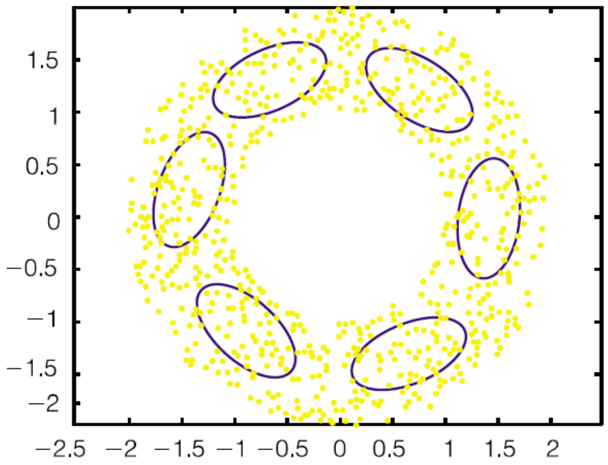
https://medium.com/@lzpdatascience/what-is-the-difference-between-hierarchical-and-partitional-clustering-edc0d488c7c4

Gaussian Mixture Model

- Gaussian Mixture Model (GMM)
  - Estimate PDF of data using multiple Gaussian functions
  - Consider each PDF as a kernel of the mixture model



1 dimension



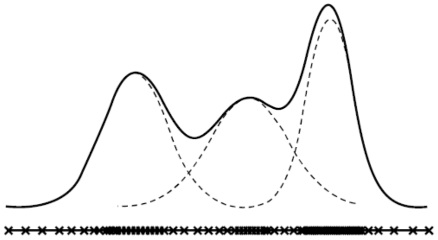
2 dimension

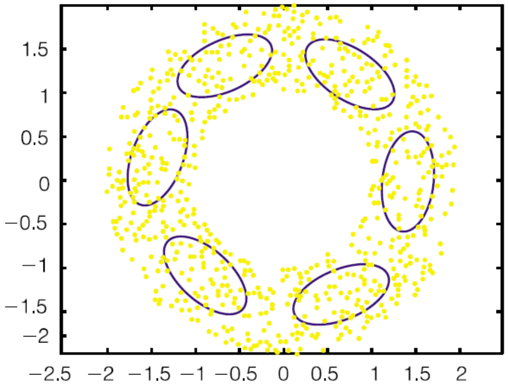
Gaussian Mixture Model

$$\theta_{\mathrm{i}} = (\mu_1, \mu_2, \cdots, \mu_M, \sigma_1^2, \sigma_2^2, \cdots, \sigma_M^2, \alpha_1, \alpha_2, \cdots, \alpha_M)$$

$$p(X \mid \theta) = \sum_{i=1}^{M} p(X \mid \omega_i, \theta_i) P(\omega_i) \qquad (1)$$

$$0 \le \alpha_i \le 1 \quad \text{and} \quad \sum_{i=1}^{M} \alpha_i = 1 \qquad (2)$$





EM Algorithm

[Example]

- PDF of students' grade in Machine Learning class was determined with a parameter μ like below
  - ω_1 : grade A, P(A) = ½
  - ω_2 : grade B, P(B) = μ
  - ω_3 : grade C, P(C) = 2 μ
  - ω_4 : grade D, P(D) = ½-3μ
  - Subject to $0 \leq \mu \leq 1/6$
- Let's estimate the parameter μ, when the student numbers of the grades are 'a' for A grade, 'b' for B grade, 'c' for C grade and 'd' for D grade

EM Algorithm

Likelihood function → $P(a,b,c,d \mid \mu) = K(\tfrac{1}{2})^{a}(\mu)^{b}(2\mu)^{c}(\tfrac{1}{2} - 3\mu)^{d}$

Log- Likelihood function → $\log P(a,b,c,d \mid \mu) = \log K + a\log \tfrac{1}{2} + b\log \mu + c\log 2\mu + d\log(\tfrac{1}{2} - 3\mu)$

Estimate the parameter μ using maximum likelihood method

$$\frac{\partial \log P(a,b,c,d \mid \mu)}{\partial \mu} = \frac{b}{\mu} + \frac{2c}{2\mu} - \frac{3d}{1/2 - 3\mu} = 0 \longrightarrow \mu = \frac{b+c}{6(b+c+d)}$$

if A:14, B:6, C:9, D:10 , $\mu = \frac{1}{10}$

However, if we know ‘c’ and ‘d’, and only ‘*h=20*’ which is the summation of ‘a’ and ‘b’ (‘a’ and ‘b’ are hidden), how can we estimate all numbers of the students for the grades?

$$a = \frac{1/2}{1/2 + \mu}h, \quad b = \frac{\mu}{1/2 + \mu}h \quad \rightleftarrows \quad \mu = \frac{b+c}{6(b+c+d)}$$

EM

(Expectation-Maximisation)

EM Algorithm

- Why do we need EM?
  - EM algorithm repeats E step and M step in order to estimate the unknown parameter μ and unknown variables a and b

Initial seed : μ(0)

E-Step :

$$b(t) = \frac{\mu(t)h}{1/2 + \mu(t)} = E[b \mid \mu(t)]$$

M-Step :

$$\mu(t+1) = \frac{b(t)+c}{6(b(t)+c+d)} =$$

Choose the maximum parameter μ, given *b(t)*

EM Algorithm

- ❖ Results of EM algorithm

M

$$\mu(t+1)=\frac{b(t)+c}{6(b(t)+c+d)}$$

E

$$b(t)=\frac{\mu(t)h}{1/2+\mu(t)}=E[b\mid\mu(t)]$$

c=9, d=10, h=20

t	μ(t)	b(t)
0	0	0
1	0.0833	2.857
2	0.0937	3.158
3	0.0947	3.185
4	0.0948	3.187
5	0.0948	3.187
6	0.0948	3.187

EM Algorithm

- Objective : Find the value $\widehat{\theta}$ that maximizes f(y|θ)

$$\widehat{\theta} = \overset{argmax}{\theta} \log(f(y;\theta))$$

- EM is an iterative method that attempts to find the maximum likelihood estimator of a parameter θ

- Two main applications of the EM
  - When the data has missing values due to limitations of the observation
  - When optimizing the likelihood function is analytically intractable

EM Algorithm

- Why the maximum likelihood (ML) problem is not trivial?
  - When the likelihood function has multiple local maxima, the ML does not in general have a closed form

- Nice properties of EM algorithm
  - Always converges to a local maximum of the likelihood function
  - If it has only one local maximum, it will always converge to it

EM Algorithm

- Observation y, and its distribution $f(y;\Theta) = \mathrm{P}(\mathrm{Y}|\Theta)$, which is parameterized by a nonrandom and unknown quantity Θ

$$\widehat{\Theta} = \underset{\Theta}{argmax} \log(f(y;\Theta))$$

It's **HARD** to evaluate without a latent variable x during training

This latent variable becomes a missing variable in EM algorithm

ex) how can we define the probability distribution of EEG without knowing sleep stages?

- A joint distribution $q(x,y;\Theta) = P(X,Y|\Theta)$, where we consider x as a missing variable

$$f(y;\Theta) = \int_x q(x,y;\Theta)\, dx$$

EM Algorithm

- KEY EQUATION OF EM algorithm

$$\begin{aligned} L(\theta) &\triangleq \log f(y;\theta) \\ &= \log \int_x q(\boxed{x}, y;\theta)\,dx \\ &= \log\left[\int_x \boxed{z(x)}\frac{q(x,y;\theta)}{z(x)}\right]dx \\ &\geq \boxed{\int_x z(x)\log\left[\frac{q(x,y;\theta)}{z(x)}\right]dx} \\ &\triangleq F(z,\theta) \end{aligned}$$

Latent variable `x' is introduced.

(missing variable!!!)

Arbitary pdf of `x',

Jensen's inequality

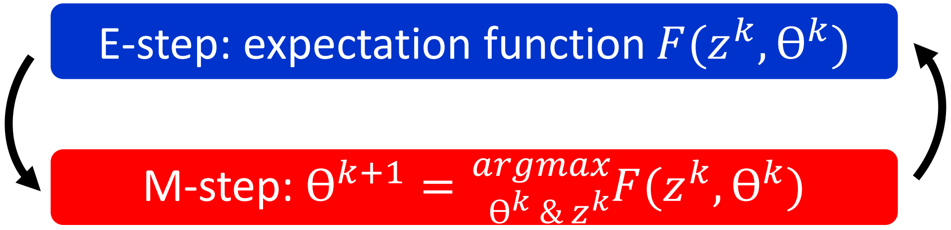
Equation of **EXPECTATION**

cf. $E[W(x)]=\int_x W(x)z(x)dx$,

($W(x) = \log\left[\frac{q(x,y;\theta)}{z(x)}\right]$,

y *and* ϴ are constant)

- Look for ϴ, which maximizes $F(z,\Theta)$



E-step: expectation function $F(z^k,\Theta^k)$

M-step: $\Theta^{k+1} = \underset{\Theta^k\ \&\ z^k}{argmax} F(z^k,\Theta^k)$

EM Algorithm

- **E-step: Implement EXPECTATION equation**

$$\int_x z(x) \log\left[\frac{q(x,y;\Theta)}{z(x)}\right] dx$$

Ex) students' grade example

$$a = \frac{1/2}{1/2+\mu}h \,,\quad b = \frac{\mu}{1/2+\mu}h$$

EM Algorithm

- M-step: find $z^k(x)$ and θ^{k+1} with maximizing the EXPECTATION from E-step

1) Calculate $z^k(x)$

Let us now define

$$w\left(x|y;\theta^k\right) \triangleq \frac{q\left(x,y;\theta^k\right)}{f\left(y;\theta^k\right)} = \frac{q\left(x,y;\theta^k\right)}{\int_x q\left(x,y;\theta^k\right)dx}$$

then note that

$$\begin{aligned} F(z,\theta^k) &\triangleq \int_x z(x)\log\left[\frac{w\left(x|y;\theta^k\right) f\left(y;\theta^k\right)}{z(x)}\right]dx \\ &= \log f\left(y;\theta^k\right) - D\left(z(x)\|w\left(x|y;\theta^k\right)\right) \end{aligned}$$

where

$$D\left(z_1(x)\|z_2(x)\right) \triangleq \int_x z_1(x)\log\left[\frac{z_1(x)}{z_2(x)}\right]dx$$

(D is the Kullback-Leibler distance)

In order to maximize $F(z,\Theta^k)$,

$$z^k(x) = w\left(x|y;\Theta^k\right) = \boldsymbol{P(X|Y,\Theta^k)},$$

a priori **=** **a posteriori**

EM Algorithm

2) find the optimal θ^{k+1}

It was a posteriori from the previous slide

$$\begin{aligned}
\theta^{k+1} &= \arg\max_{\theta^k} F(\boxed{z^k}, \theta) \\
&= \arg\max_{\theta^k} \int_x z^k(x) \log\left[\frac{q(x,y;\theta^k)}{z^k(x)}\right] dx \\
&= \arg\max_{\theta^k} \int_x z^k(x) \log\left[q(x,y;\theta^k)\right] - \underline{\int_x z^k(x) \log\left[z^k(x)\right]} \\
&= \arg\max_{\theta^k} \boxed{\int_x z^k(x) \log q(x,y;\theta^k)\, dx}
\end{aligned}$$

no Θ, then ignore

Equation of **EXPECTATION**

cf. $E[W(x)]=\int_x W(x)z(x)dx$,
($W(x) = \mathbf{log}\,[\boldsymbol{q}(\boldsymbol{x},\boldsymbol{y};\Theta)]$,
y *and* Θ are constant)

COULD BE DONE IN CLOSED FORM

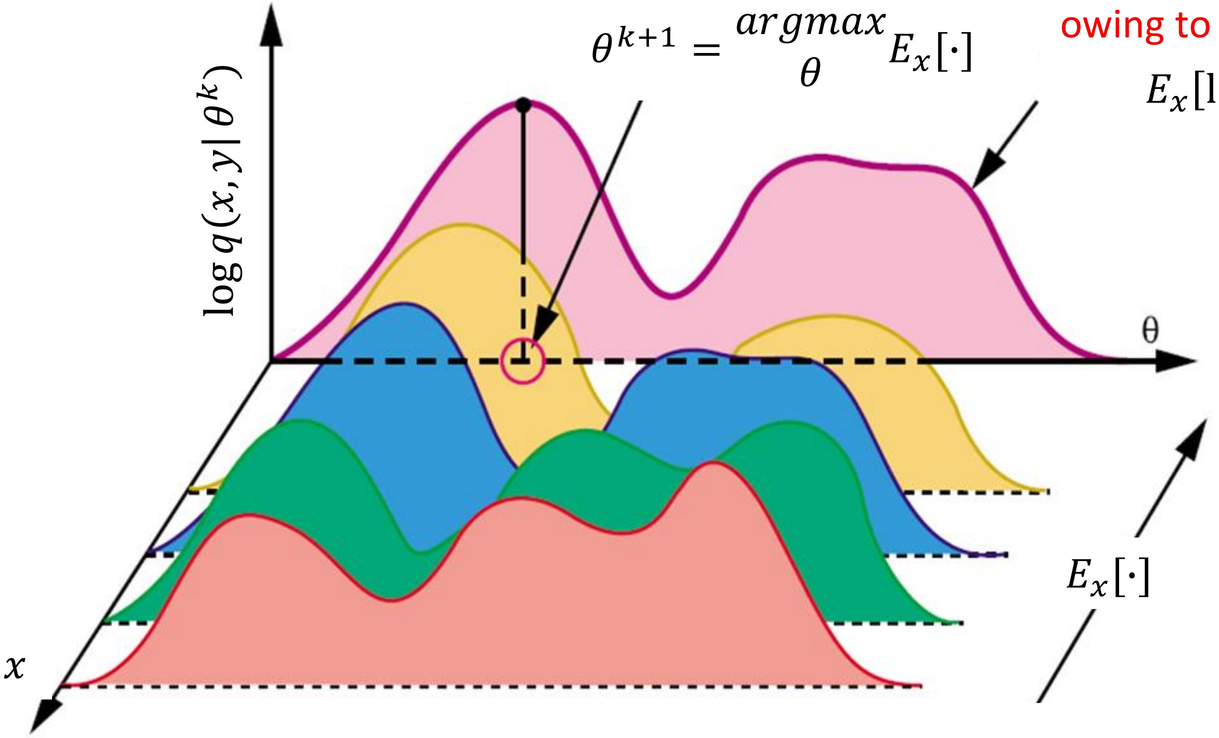
EM Algorithm

Maximum log-likelihood

$$\theta^{k+1} = \underset{\theta}{argmax} E_x[\cdot]$$

It becomes log-likelihood with respect to y owing to the marginalization across x.

$$E_x[\log q(x, y|\, \theta^k)] \approx \log(y|\theta^k)$$



- Since x is unknown, we could just maximize the marginalized log-likelihood function across all possible x.

Gaussian Mixture Model

- GMM training
  - Parameter $(\mu_j, \sigma_j^2, \alpha_j = P(\omega_j))$ estimation with maximizing the likelihood function of sample data $x = \{x_1, x_2, \ldots, x_N\}$
  - Log-likelihood function

$$E = -\log L(\boldsymbol{\theta}) = -\sum_{n=1}^{N} \log p(\mathbf{x}_n \mid \boldsymbol{\theta})$$

  - Maximum Likelihood Estimation (MLE)

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \arg\max[p(\mathbf{x} \mid \boldsymbol{\theta})] \\ &= \arg\max[\sum_{n=1}^{\mathrm{N}} \log p(\mathbf{x}_n \mid \boldsymbol{\theta})] \\ &= \arg\max[\sum_{n=1}^{\mathrm{N}} \log \sum_{j=1}^{M} p(\mathbf{x}_n \mid \boldsymbol{\theta}_{\mathbf{j}}) \mathrm{P}(\boldsymbol{\omega}_{\mathbf{j}})]\end{aligned}$$

Expectation form
with respect to ω

given ω_i

Gaussian Mixture Model

Now, its maximization

$$\frac{\partial}{\partial \boldsymbol{\mu}_{\mathrm{j}}} E = -\frac{\partial}{\partial \boldsymbol{\mu}_{\mathrm{j}}} \sum_{n=1}^{N} \log p(\mathbf{x}_n \mid \boldsymbol{\theta})$$

For *j* case

$$= -\sum_{n=1}^{N} \frac{1}{p(\mathbf{x}_n \mid \boldsymbol{\theta})} \frac{\partial}{\partial \boldsymbol{\mu}_{\mathrm{j}}} p(\mathbf{x}_n \mid \boldsymbol{\theta})$$

$$p(\mathbf{x}_n \mid \boldsymbol{\theta}) = \sum_{j=1}^{M} p(\mathbf{x}_n \mid \boldsymbol{\omega}_{\mathrm{j}}, \boldsymbol{\theta}) \mathrm{P}(\boldsymbol{\omega}_{\mathrm{j}} \mid \boldsymbol{\theta})$$

$$= \sum_{\mathrm{j}=1}^{M} p(\mathbf{x}_n \mid \boldsymbol{\mu}_{\mathrm{j}}, \boldsymbol{\sigma}_{\mathrm{j}}^2) \alpha_j$$

$$= -\sum_{n=1}^{N} \frac{1}{p(\mathbf{x}_n \mid \boldsymbol{\theta})} \frac{\partial}{\partial \boldsymbol{\mu}_{\mathrm{j}}} \sum_{j=1}^{M} p(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j^2) \alpha_j$$

$$p(\mathbf{x}_{\mathrm{n}} \mid \boldsymbol{\mu}_{\mathrm{j}}, \boldsymbol{\sigma}_{\mathrm{j}}^2) = \frac{1}{\sqrt{2\pi\boldsymbol{\sigma}_{\mathrm{j}}}} \exp\left(-\frac{(\mathbf{x}_{\mathrm{n}} - \boldsymbol{\mu}_{\mathrm{j}})^2}{2\boldsymbol{\sigma}_{\mathrm{j}}^2} \right)$$

$$= -\sum_{n=1}^{N} \frac{1}{p(\mathbf{x}_n \mid \boldsymbol{\theta})} \frac{1}{\sqrt{2\pi\boldsymbol{\sigma}_{\mathrm{j}}}} \exp\left(-\frac{(\mathbf{x}_{\mathrm{n}} - \boldsymbol{\mu}_{\mathrm{j}})^2}{2\boldsymbol{\sigma}_{\mathrm{j}}^2} \right) \frac{-2(\mathbf{x}_{\mathrm{n}} - \boldsymbol{\mu}_{\mathrm{j}})}{2\boldsymbol{\sigma}_{\mathrm{j}}^2} (-1)\, \alpha_j$$

$$= -\sum_{n=1}^{N} \frac{1}{p(\mathbf{x}_n \mid \boldsymbol{\theta})} p(\mathbf{x}_n \mid \boldsymbol{\mu}_{\mathrm{j}}, \boldsymbol{\sigma}_{\mathrm{j}}^2) \frac{(\mathbf{x}_{\mathrm{n}} - \boldsymbol{\mu}_{\mathrm{j}})}{\boldsymbol{\sigma}_{\mathrm{j}}^2} \alpha_j$$

$$\mathrm{P}(\boldsymbol{\omega}_{\mathrm{j}} \mid \mathbf{x}_{\mathrm{n}}, \boldsymbol{\theta}) = \frac{p(\mathbf{x}_{\mathrm{n}} \mid \boldsymbol{\omega}_{\mathrm{j}}, \boldsymbol{\theta}) \mathrm{P}(\boldsymbol{\omega}_{\mathrm{j}} \mid \boldsymbol{\theta})}{p(\mathbf{x}_n \mid \boldsymbol{\theta})}$$

$$= \frac{p(\mathbf{x}_n \mid \boldsymbol{\mu}_{\mathrm{j}}, \boldsymbol{\sigma}_{\mathrm{j}}^2) \alpha_j}{p(\mathbf{x}_n \mid \boldsymbol{\theta})}$$

$$= -\sum_{\mathbf{n}=1}^{\mathbf{N}} \mathrm{P}(\boldsymbol{\omega}_{\mathrm{j}} \mid \mathbf{x}_{\mathrm{n}}, \boldsymbol{\theta}) \frac{(\mathbf{x}_{\mathrm{n}} - \boldsymbol{\mu}_{\mathrm{j}})}{\boldsymbol{\sigma}_{\mathrm{j}}^2}$$

Gaussian Mixture Model

$$0 \overset{!}{=} \frac{\partial}{\partial \mu_j} E$$

$$0 = -\sum_{n=1}^{N} P(\omega_j \mid x_n, \theta) \frac{(x_n - \mu_j)}{\sigma_j^2}$$

$$0 = -\sum_{n=1}^{N} P(\omega_j \mid x_n, \theta) x_n + \mu_j \sum_{n=1}^{N} P(\omega_j \mid x_n, \theta)$$

Posterior

$$\hat{\mu}_j = \frac{\sum_{n=1}^{N} P(\omega_j \mid x_n, \theta) x_n}{\sum_{n=1}^{N} P(\omega_j \mid x_n, \theta)}$$

Gaussian Mixture Model

- Parameter Estimation using EM algorithm

$$\frac{\partial}{\partial \boldsymbol{\mu}_j}[\cdot] = 0 \quad \Rightarrow \quad \hat{\boldsymbol{\mu}}_j = \frac{\sum_{n=1}^{N} P(\omega_j \mid \mathbf{x}_n, \theta)\mathbf{x}_n}{\sum_{n=1}^{N} P(\omega_j \mid \mathbf{x}_n, \theta)}$$

$$\frac{\partial}{\partial \boldsymbol{\sigma}_j}[\cdot] = 0 \quad \Rightarrow \quad \hat{\boldsymbol{\sigma}}_j^2 = \frac{1}{\mathbf{d}} \frac{\sum_{n=1}^{N} P(\omega_j \mid \mathbf{x}_n, \theta)\left\|\mathbf{x}_n - \hat{\boldsymbol{\mu}}_j\right\|^2}{\sum_{n=1}^{N} P(\omega_j \mid \mathbf{x}_n, \theta)}$$

$$\frac{\partial}{\partial \alpha_j}[\cdot] = 0 \quad \Rightarrow \quad \hat{\alpha}_j = \hat{\mathbf{P}}(\omega_j) = \frac{1}{\mathbf{N}} \sum_{n=1}^{N} P(\omega_j \mid \mathbf{x}_n, \theta)$$

k-mean algorithm using EM

Input : training set X, number of Gaussian K

Output : (μ_j, Σ_j), 1≤j ≤K, and $\pi (==\alpha)$

Algorithm

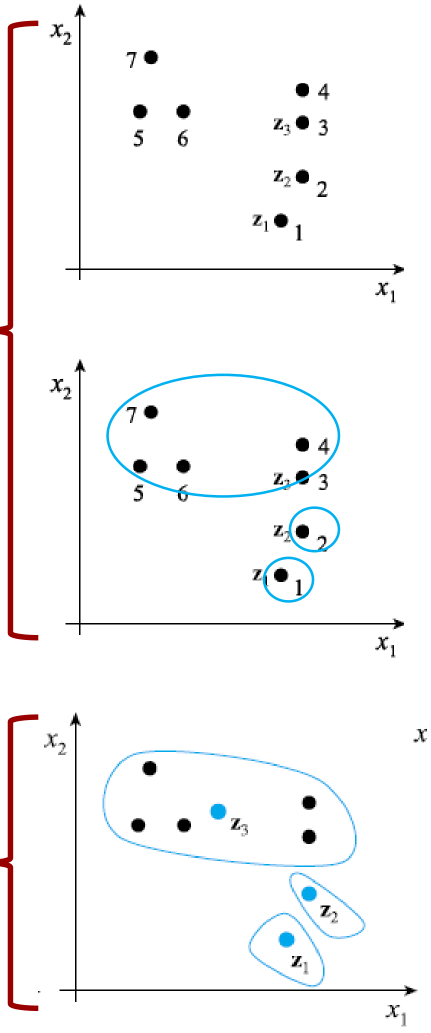
ω_j

```
μ  Initialize $\mu_j, \Sigma_j$ , 1≤j ≤K, and $\pi$
repeat {
    // E-step (estimate the cluster for each sample)
    for ($i = 1$ to $N$)
        for ($j = 1$ to $K$)
            $P(z_j | \mathbf{x}_i) = \frac{\pi_j N(\mathbf{x}_i | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_{k=1}^{K} \pi_k N(\mathbf{x}_i | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$    // (3.25)
    // M-step (parameter estimation)
    for ($j = 1$ to $K$) {
        $N_j = \sum_{i=1}^{N} P(z_j | \mathbf{x}_i)$;    // (3.27)
        $\boldsymbol{\mu}_j = \frac{1}{N_j} \sum_{i=1}^{N} P(z_j | \mathbf{x}_i) \mathbf{x}_i$ ;    // (3.26)
        $\boldsymbol{\Sigma}_j = \frac{1}{N_j} \sum_{i=1}^{N} P(z_j | \mathbf{x}_i)(\mathbf{x}_i - \boldsymbol{\mu}_j)(\mathbf{x}_i - \boldsymbol{\mu}_j)^{\mathrm{T}}$ ;    // (3.28)
        $\pi_j = \frac{N_j}{N}$;    // (3.29)
    }
} until  (meet the stopping criteria);
```

Posterior!!!

$P(z_j)$ Statistical distance between x and $z \Rightarrow P(x_i | z \rightarrow \theta_j = \mu_j, \Sigma_j)$

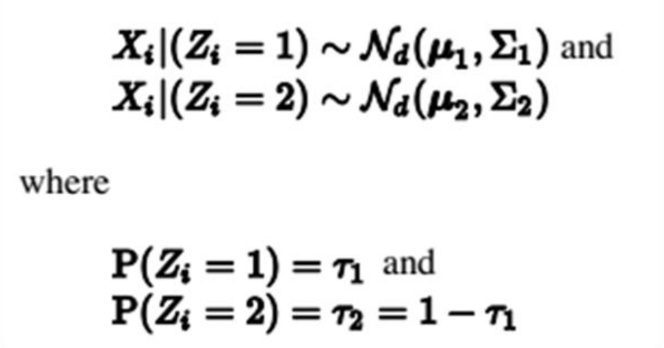
$P(z_k)$ $P(x_i | z_k, \theta_k)$



Supplementary material

EM Algorithm

- Let $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ be a sample of n independent observations from a mixture of two multivariate normal distributions of dimension d, and let $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n)$ be the latent variables that determine the component from which the observation originates



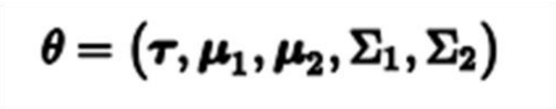
$$X_i|(Z_i = 1) \sim \mathcal{N}_d(\boldsymbol{\mu}_1, \Sigma_1) \text{ and}$$
$$X_i|(Z_i = 2) \sim \mathcal{N}_d(\boldsymbol{\mu}_2, \Sigma_2)$$

where

$$\mathbf{P}(Z_i = 1) = \tau_1 \text{ and}$$
$$\mathbf{P}(Z_i = 2) = \tau_2 = 1 - \tau_1$$

* z_i has the information of the class

- The aim is to estimate the unknown parameters representing the "mixing" value between the Gaussians and the means and covariances of each:



$$\theta = (\boldsymbol{\tau}, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \Sigma_1, \Sigma_2)$$

https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm

EM Algorithm

$$\boldsymbol{\theta} = (\boldsymbol{\tau}, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \Sigma_1, \Sigma_2)$$

where the incomplete-data likelihood function is

$$L(\boldsymbol{\theta}; \mathbf{x}) = \prod_{i=1}^{n} \sum_{j=1}^{2} \tau_j \, f(\mathbf{x}_i; \boldsymbol{\mu}_j, \Sigma_j),$$

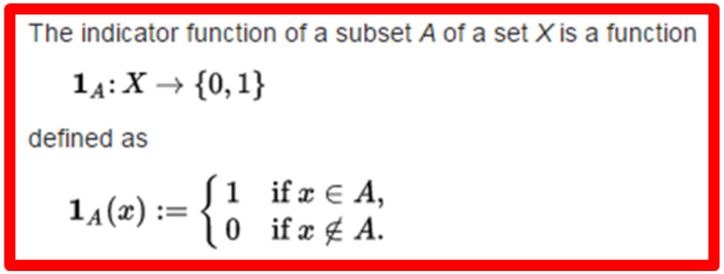
And the complete-data likelihood function is

$$L(\boldsymbol{\theta}; \mathbf{x}, \mathbf{z}) = p(\mathbf{x}, \mathbf{z} | \boldsymbol{\theta}) = \prod_{i=1}^{n} \sum_{j=1}^{2} \mathbb{I}(z_i = j) \, f(\mathbf{x}_i; \boldsymbol{\mu}_j, \Sigma_j) \tau_j$$

or

$$L(\boldsymbol{\theta}; \mathbf{x}, \mathbf{z}) = \exp\left\{ \sum_{i=1}^{n} \sum_{j=1}^{2} \mathbb{I}(z_i = j) \left[\log \tau_j - \tfrac{1}{2} \log |\Sigma_j| - \tfrac{1}{2} (\mathbf{x}_i - \boldsymbol{\mu}_j)^\top \Sigma_j^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_j) - \tfrac{d}{2} \log(2\pi) \right] \right\}.$$

where **I** is an indicator function and ***f*** is the probability density function of a multivariate normal



The indicator function of a subset *A* of a set *X* is a function

$$\mathbf{1}_A \colon X \to \{0, 1\}$$

defined as

$$\mathbf{1}_A(x) := \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{if } x \notin A. \end{cases}$$

EM Algorithm

❖ E step

- Given our current estimate of the parameters $\theta^{(t)}$, the conditional distribution of the Z_i is determined by Byes theorem to be the proportional height of the normal density weighted by τ:

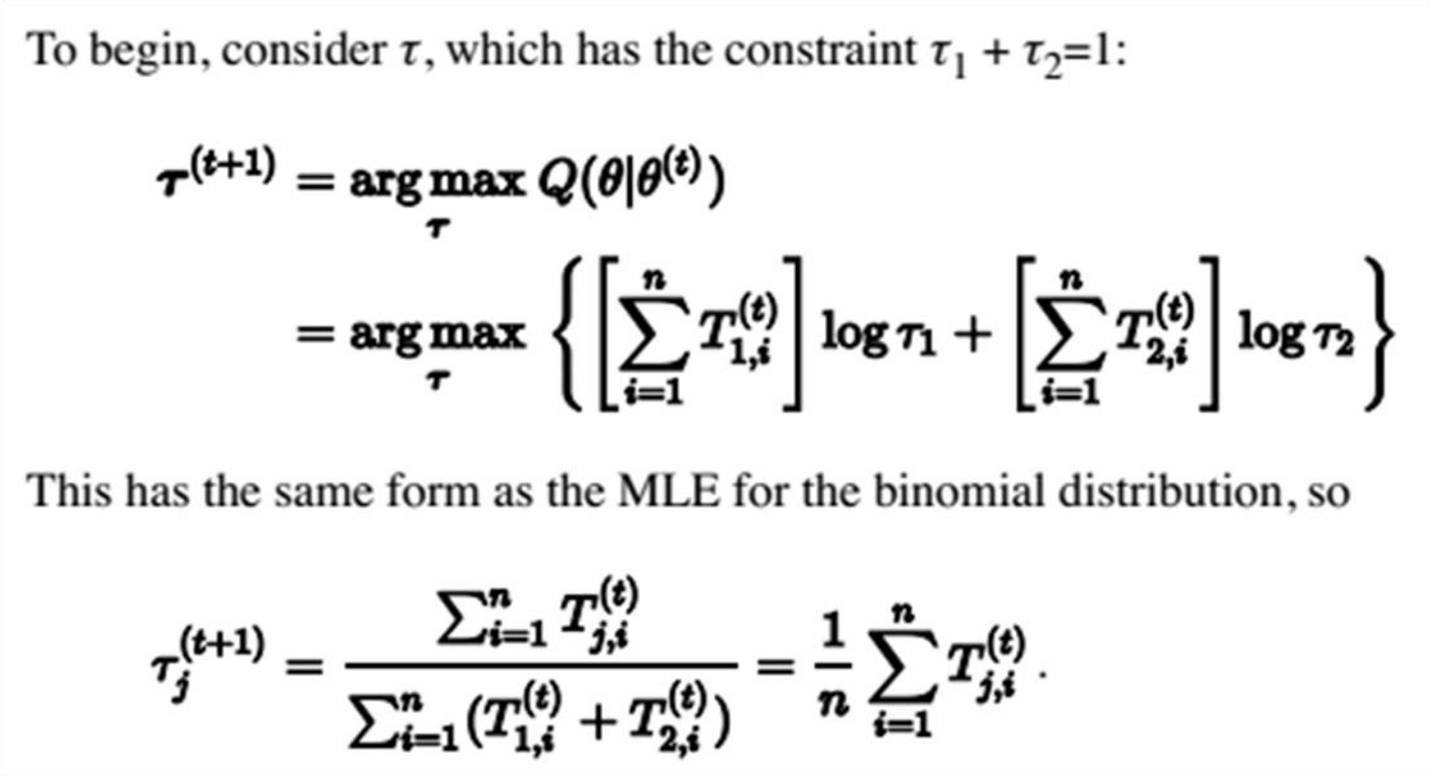
$$T_{j,i}^{(t)} := \mathrm{P}(Z_i = j | X_i = \mathbf{x}_i; \theta^{(t)}) = \frac{\tau_j^{(t)}\ f(\mathbf{x}_i; \boldsymbol{\mu}_j^{(t)}, \Sigma_j^{(t)}) \rightarrow P(X_i, Z_i|\theta)}{\tau_1^{(t)}\ f(\mathbf{x}_i; \boldsymbol{\mu}_1^{(t)}, \Sigma_1^{(t)}) + \tau_2^{(t)}\ f(\mathbf{x}_i; \boldsymbol{\mu}_2^{(t)}, \Sigma_2^{(t)}) \rightarrow P(X_i|\ \theta)}.$$

- These are called the "membership probabilities", the output of the E step (although this is not the Q function of below)
- E step

$$\begin{aligned}
Q(\theta|\theta^{(t)}) &= \mathrm{E}_{\mathbf{Z}|\mathbf{X},\theta^{(t)}}[\log L(\theta; \mathbf{x}, \mathbf{Z})] \\
&= \mathrm{E}_{\mathbf{Z}|\mathbf{X},\theta^{(t)}}[\log \prod_{i=1}^{n} L(\theta; \mathbf{x}_i, \mathbf{z}_i)] \\
&= \mathrm{E}_{\mathbf{Z}|\mathbf{X},\theta^{(t)}}[\sum_{i=1}^{n} \log L(\theta; \mathbf{x}_i, \mathbf{z}_i)] \\
&= \sum_{i=1}^{n} \mathrm{E}_{\mathbf{Z}|\mathbf{X};\theta^{(t)}}[\log L(\theta; \mathbf{x}_i, \mathbf{z}_i)] \\
&= \sum_{i=1}^{n} \sum_{j=1}^{2} P(Z_i = j | X_i = \mathbf{x}_i; \theta^{(t)}) \log L(\theta_j; \mathbf{x}_i, \mathbf{z}_i) \\
&= \sum_{i=1}^{n} \sum_{j=1}^{2} T_{j,i}^{(t)} [\log \tau_j - \tfrac{1}{2} \log |\Sigma_j| - \tfrac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_j)^\top \Sigma_j^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_j) - \tfrac{d}{2} \log(2\pi)]
\end{aligned}$$

EM Algorithm

- M step



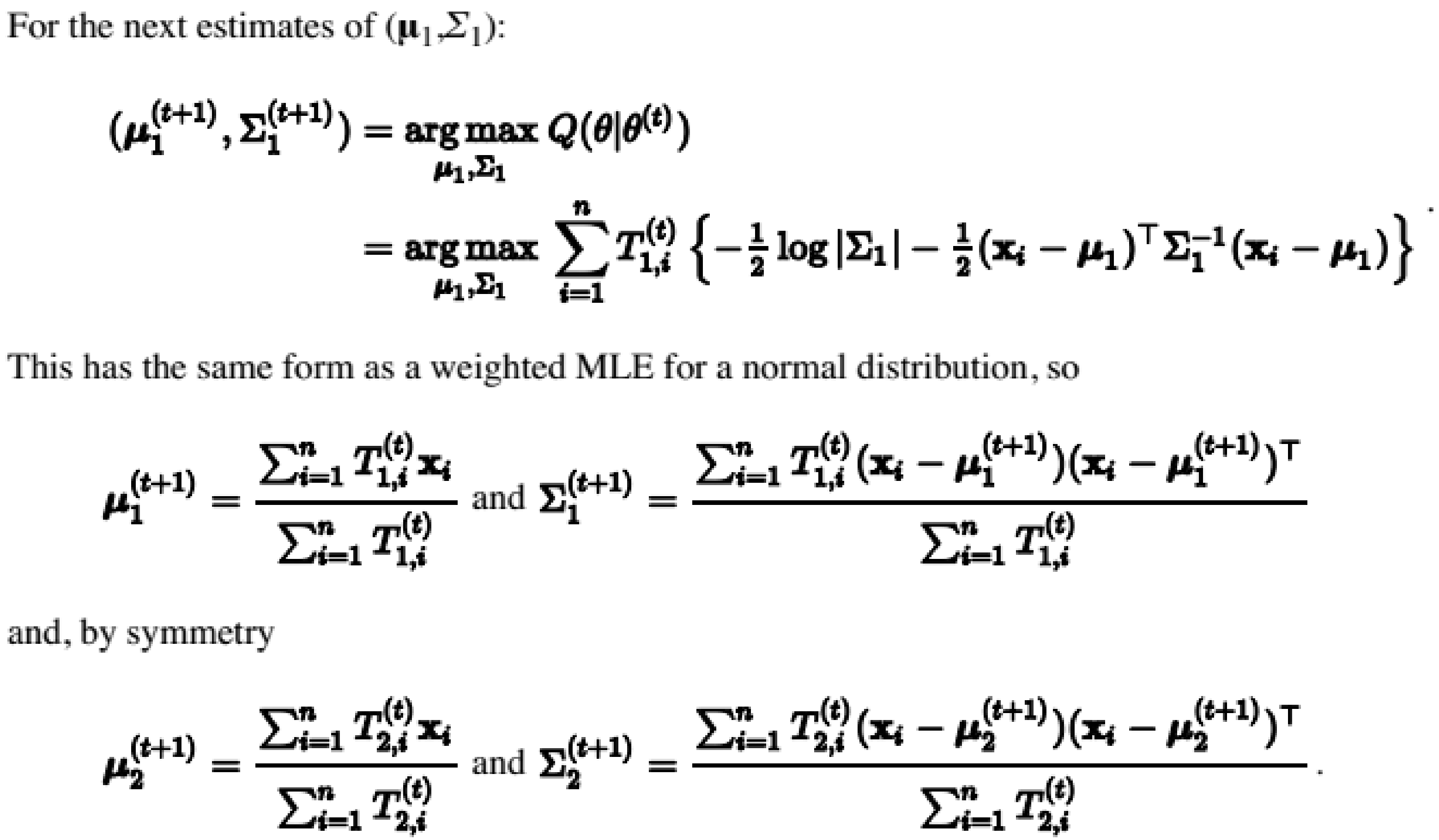
To begin, consider τ, which has the constraint $\tau_1 + \tau_2$=1:

$$\begin{aligned}\boldsymbol{\tau}^{(t+1)} &= \arg\max_{\boldsymbol{\tau}} Q(\theta|\theta^{(t)}) \\ &= \arg\max_{\boldsymbol{\tau}} \left\{ \left[\sum_{i=1}^{n} T_{1,i}^{(t)} \right] \log \tau_1 + \left[\sum_{i=1}^{n} T_{2,i}^{(t)} \right] \log \tau_2 \right\}\end{aligned}$$

This has the same form as the MLE for the binomial distribution, so

$$\tau_j^{(t+1)} = \frac{\sum_{i=1}^{n} T_{j,i}^{(t)}}{\sum_{i=1}^{n} (T_{1,i}^{(t)} + T_{2,i}^{(t)})} = \frac{1}{n} \sum_{i=1}^{n} T_{j,i}^{(t)}.$$

EM Algorithm



For the next estimates of $(\boldsymbol{\mu}_1, \Sigma_1)$:

$$(\boldsymbol{\mu}_1^{(t+1)}, \Sigma_1^{(t+1)}) = \underset{\boldsymbol{\mu}_1, \Sigma_1}{\arg\max}\, Q(\theta|\theta^{(t)})$$

$$= \underset{\boldsymbol{\mu}_1, \Sigma_1}{\arg\max} \sum_{i=1}^{n} T_{1,i}^{(t)} \left\{ -\tfrac{1}{2}\log|\Sigma_1| - \tfrac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_1)^\top \Sigma_1^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_1) \right\}.$$

This has the same form as a weighted MLE for a normal distribution, so

$$\boldsymbol{\mu}_1^{(t+1)} = \frac{\sum_{i=1}^{n} T_{1,i}^{(t)} \mathbf{x}_i}{\sum_{i=1}^{n} T_{1,i}^{(t)}} \text{ and } \Sigma_1^{(t+1)} = \frac{\sum_{i=1}^{n} T_{1,i}^{(t)} (\mathbf{x}_i - \boldsymbol{\mu}_1^{(t+1)})(\mathbf{x}_i - \boldsymbol{\mu}_1^{(t+1)})^\top}{\sum_{i=1}^{n} T_{1,i}^{(t)}}$$

and, by symmetry

$$\boldsymbol{\mu}_2^{(t+1)} = \frac{\sum_{i=1}^{n} T_{2,i}^{(t)} \mathbf{x}_i}{\sum_{i=1}^{n} T_{2,i}^{(t)}} \text{ and } \Sigma_2^{(t+1)} = \frac{\sum_{i=1}^{n} T_{2,i}^{(t)} (\mathbf{x}_i - \boldsymbol{\mu}_2^{(t+1)})(\mathbf{x}_i - \boldsymbol{\mu}_2^{(t+1)})^\top}{\sum_{i=1}^{n} T_{2,i}^{(t)}}.$$