Expectation-Maximization

Course policies

출석: 10주 이상 참석 (2/3) (학칙)

모든 수업은 zoom사용 예정

수업자료: 포털/과목 메인 페이지

모든 공지는 포털/공지시항에 포스팅 함

수업: video on, audio off

중간, 기말: 반드시 참석 (하나라도 치르지 않으면 F)

학칙이 허용하는 예외 사유만 허용

기말 시험은 offline (14-16주 사이)

중간 시험은 offline이 가능하면 실시 (7-9주 사이).

불가능하면 기말에 같이 실시

질문: 수업 중 마이크 사용하여 직접, 채팅창에서

수업 후 메일로 (타이틀에 [지능형생물정보학] 포함)

Review: Expectation-Maximization (EM) vs. MLE

? H T T T H H T H T H

? H H H H T H H H H H

? H T H H H H H T H H

? H T H T T T H H T T

? T H H H T H H H T H

$\hat{\theta}_A = \ ?$

$\hat{\theta}_B = \ ?$

? H H H H T T H T H T

Review: Expectation-Maximization (EM)

EM is a procedure for learning hidden variables from partially observed data

X: observed variable

Z: hidden variable

θ : parameters for model

assign arbitrary values for parameters θ

iterate until convergence

E step: estimate the values of hidden variable Z by using θ and X

$$Z = \text{argmax } P(Z \mid X, \theta)$$

M step: obtain more accurate parameters θ using observed variable X and estimated Z

(use MLE for parameters)

$$\theta = \text{argmax } P(D \mid \theta_k)$$

Review: EM: coin example for hard assignment

$\theta_A^{(0)} = 0.6, \quad \theta_B^{(0)} = 0.5$

$\theta_A^{(1)} = 0.8, \quad \theta_B^{(1)} = 0.45$

$Z = \text{argmax } P(Z \mid X, \theta)$

	X	A	B	Z
1	5	0.1	0.9	B
2	9	0.98	0.02	A
3	8			A
4	4			A
5	7			A

→

	A	B
1		5H5T
2	9H1T	
3	8H2T	
4	4H6T	
5	7H3T	

$P(d_1 \mid \theta_A^{(1)}) = {}_{10}C_5 \ 0.8^5 \ 0.2^5 = 0.026$

$P(d_1 \mid \theta_B^{(1)}) = {}_{10}C_5 \ 0.45^5 \ 0.55^5 = 0.234$

$$P(z^1 = A \mid d_1) = \frac{P(d_1 \mid \theta_A^{(1)})}{P(d_1 \mid \theta_A^{(1)}) + P(d_1 \mid \theta_B^{(1)})} = 0.1$$

$\theta_A^{(2)} = 28 / (28+12) = 0.7$

$\theta_B^{(2)} = 5 / (5+5) = 0.5$

E-step: assign the expected values to the hidden variable

M-step: update the parameters that maximize the probability

Review: EM: coin example for soft assignment

Z
B
A
A
B
A

	A	B
1		5H5T
2	9H1T	
3	8H2T	
4		4H6T
5	7H3T	

randomly assigned for the first iteration

$\theta_A^{(0)} = 0.6, \quad \theta_B^{(0)} = 0.5$

$Z = P(Z \mid X, \theta)$

	X	P_A	P_B	Z
1	5	0.45	0.55	
2	9	0.80	0.20	
3	8	0.73	0.27	
4	4	0.35	0.65	
5	7	0.65	0.35	

	A	B	
1	2.2H 2.2T	2.8H 2.8H	5H5T
2	7.2H 0.8T	1.8H 0.2T	9H1T
3	5.9H 1.5T	2.1H 0.5T	8H2T
4	1.4H 2.1H	2.6H 3.9T	4H6T
5	4.5H 1.9T	2.5H 1.1T	7H3T

x is the number of heads

z is the type of coin

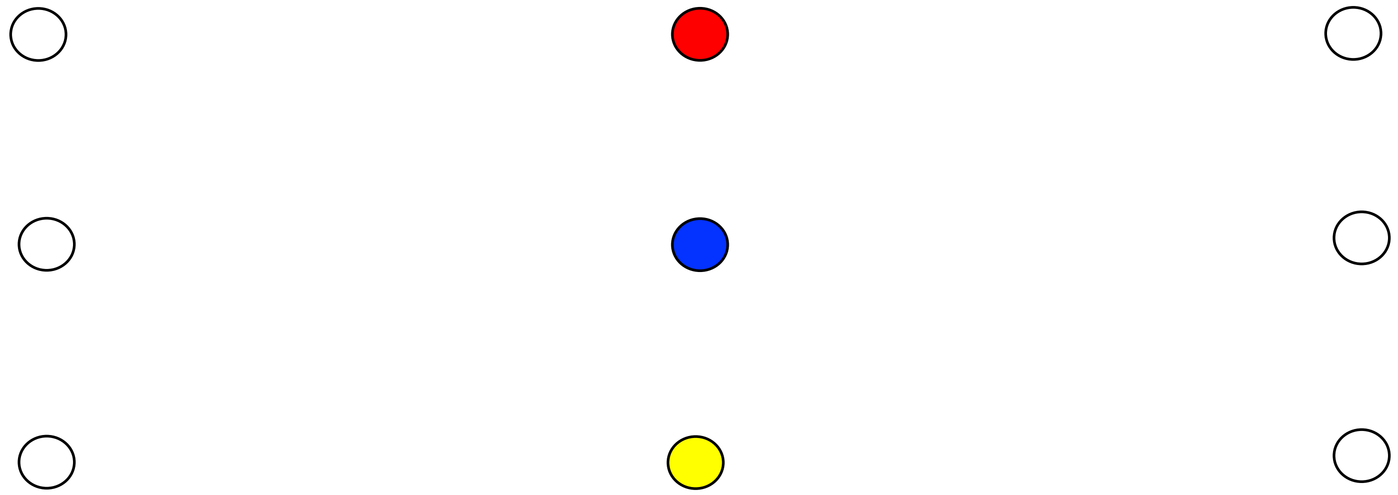
$\theta_A^{(1)} = 21.3 / (21.3+8.6) = 0.71$

$\theta_B^{(1)} = 11.7 / (11.7+8.4) = 0.58$

E-step: assign the expected values to the hidden variable based on the given model

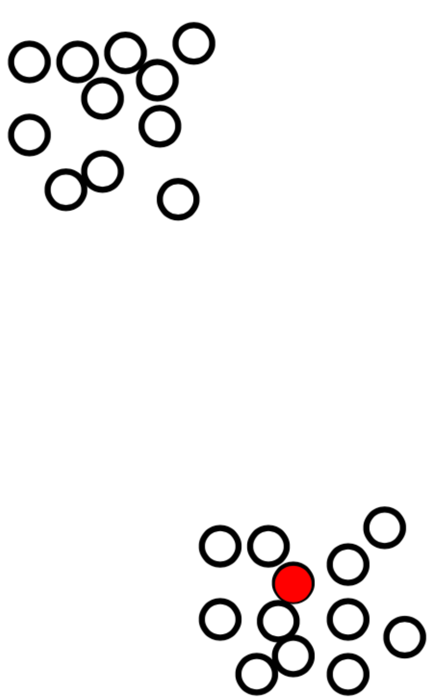
M-step: update the parameters that maximize the probability

K-means clustering



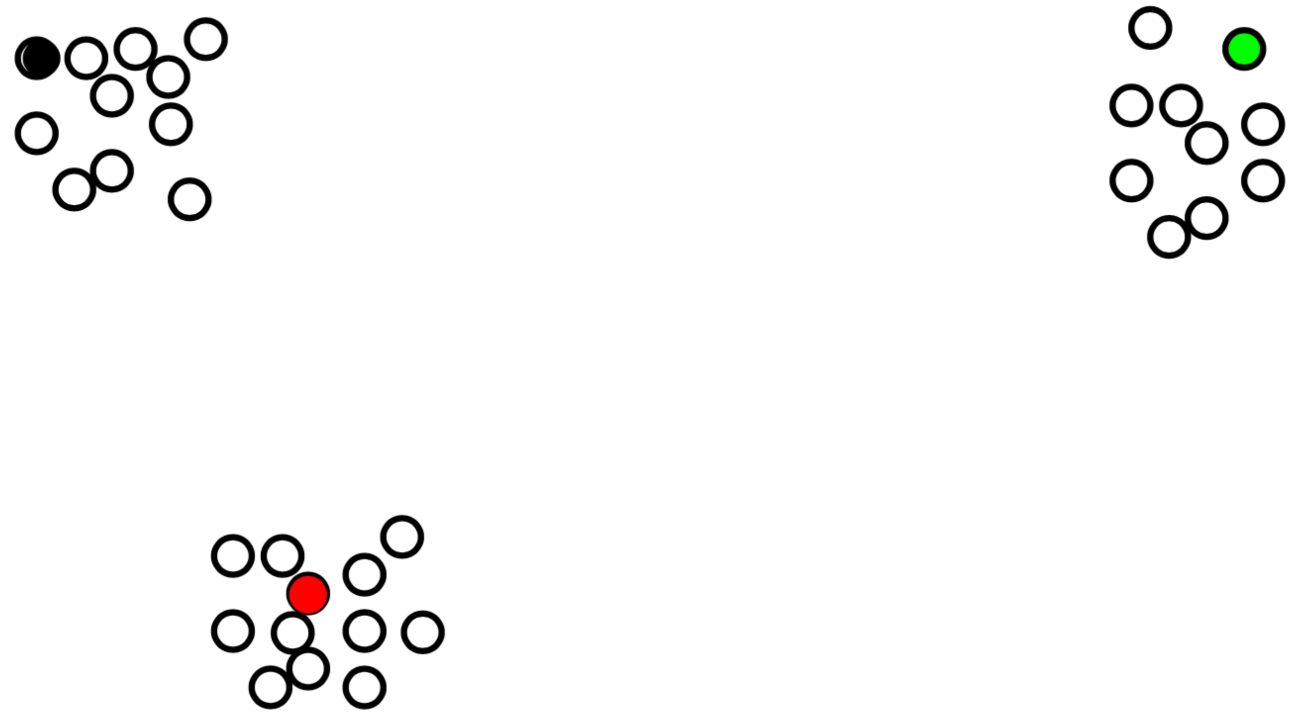
Local optimum: every point is assigned to its nearest center and every center is the mean value of its points

K-means clustering



Local optimum: every point is assigned to its nearest center and every center is the mean value of its points

K-means clustering



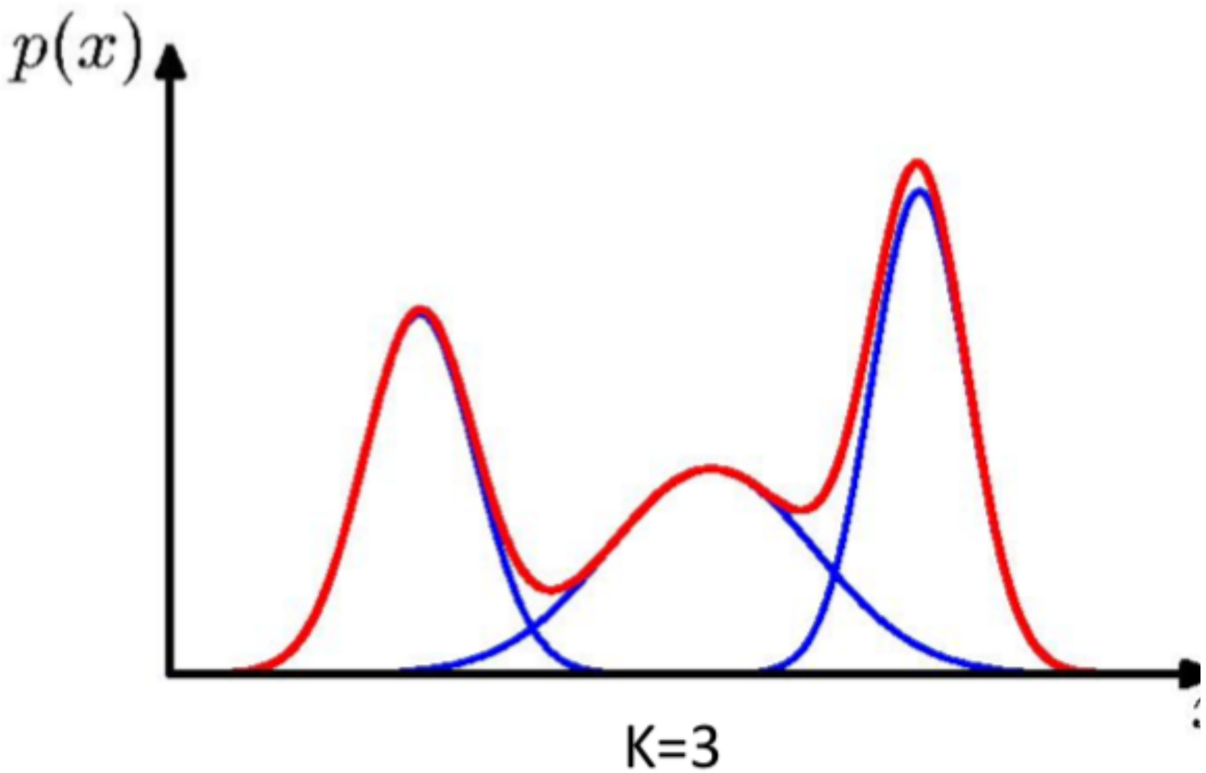
One approach is to pick furthest points (farthest point cluster, k-means ++)

→ Pick the initial point at random

→ Each subsequent point is picked from the remaining points with probability proportional to its squared distance to the points's closest cluster center

→ might be sensitive to outliers

Mixture models



$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \underbrace{\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}_{\text{Component}}$$

π_k: Mixing coefficient (weight)

$$\forall k : \pi_k \geqslant 0 \qquad \sum_{k=1}^{K} \pi_k = 1$$

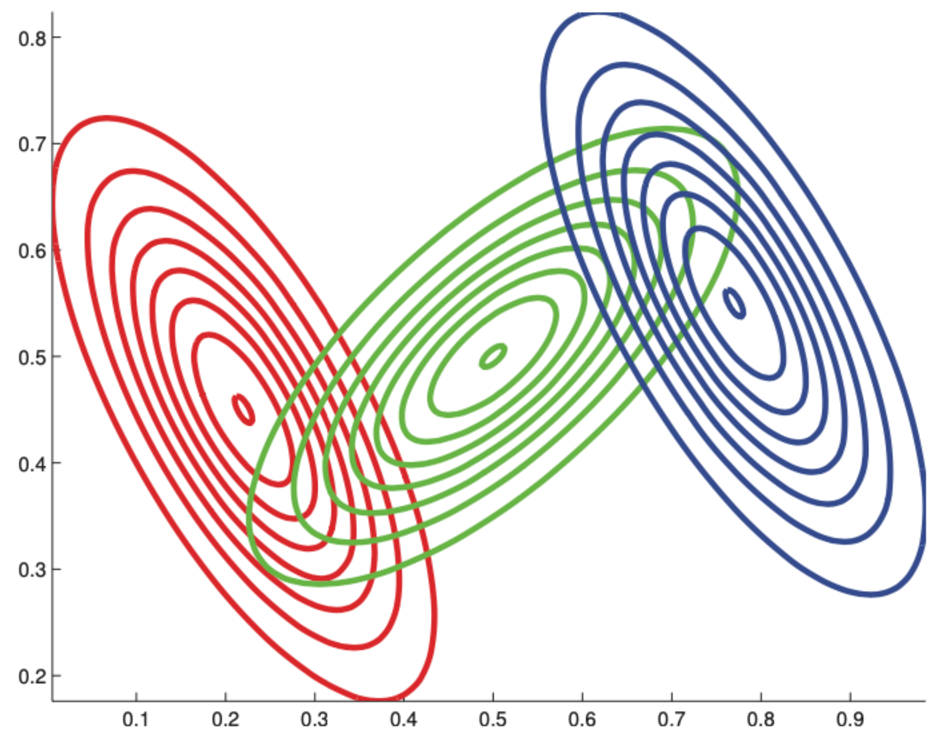
- A probabilistic model for representing the presence of subpopulations within an overall population

- way of doing soft clustering

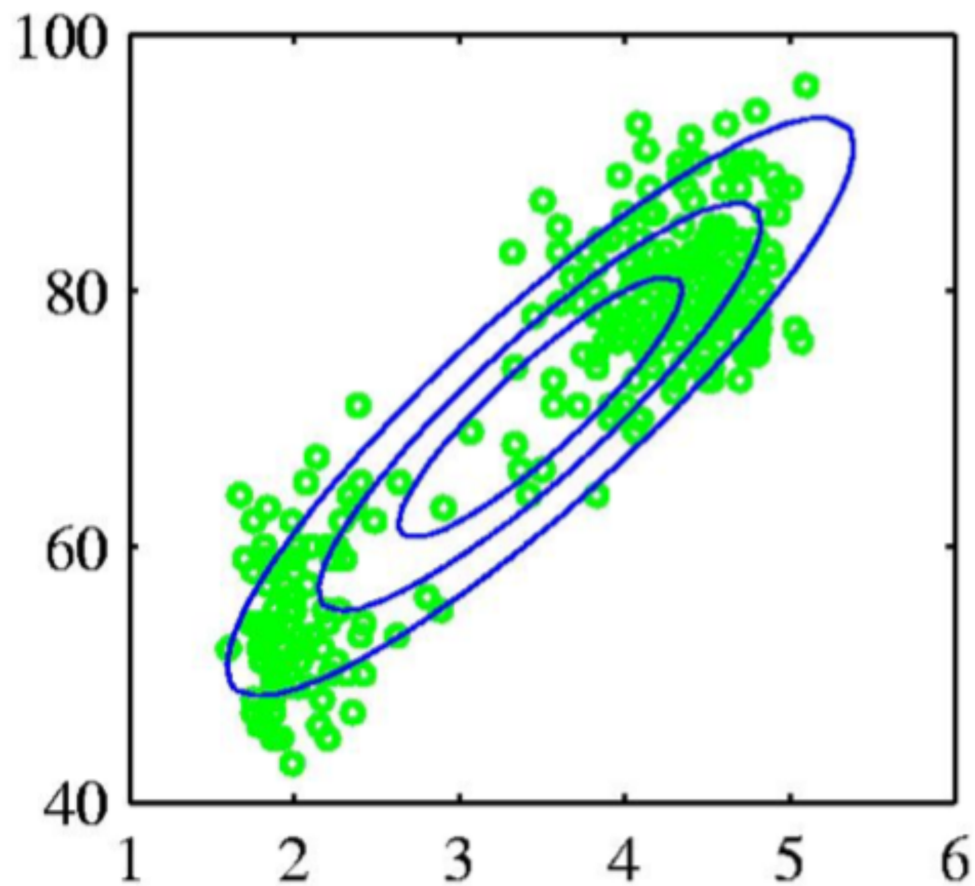
- each cluster has a generative model such as Gaussian

  (Gaussian mixture model; GMM)

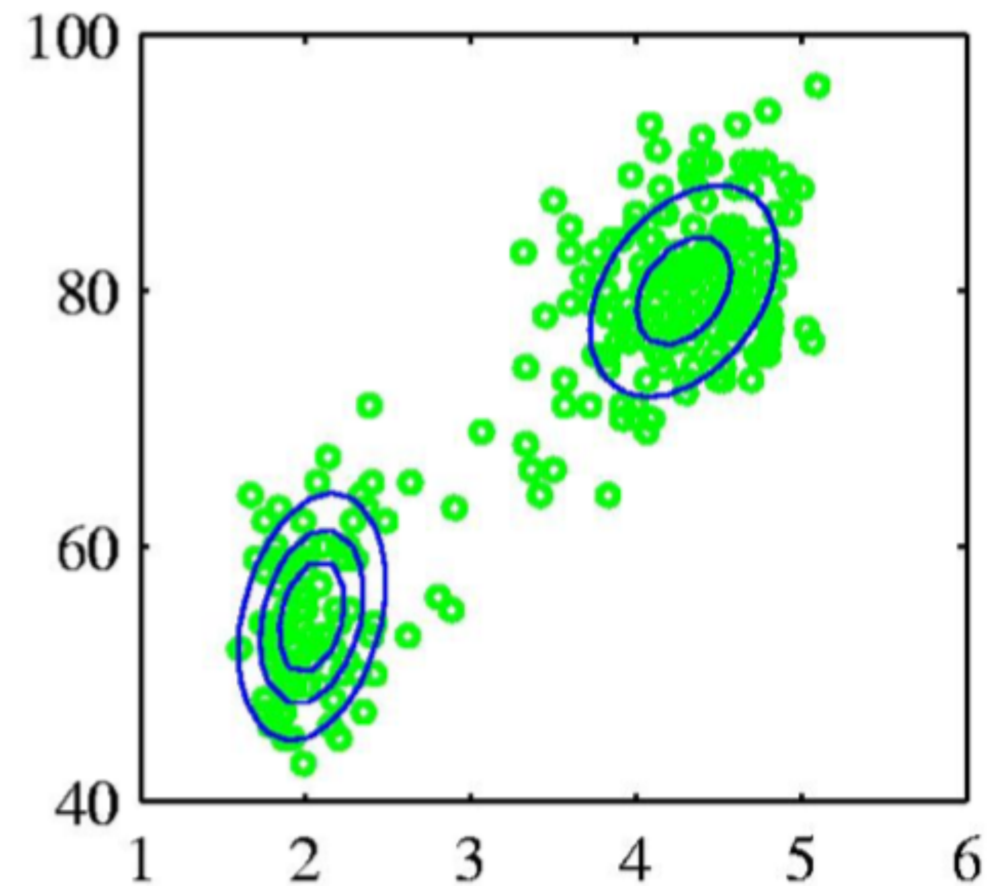
Mixture models with Multivariate Gaussian



Mixture models with Multivariate Gaussian

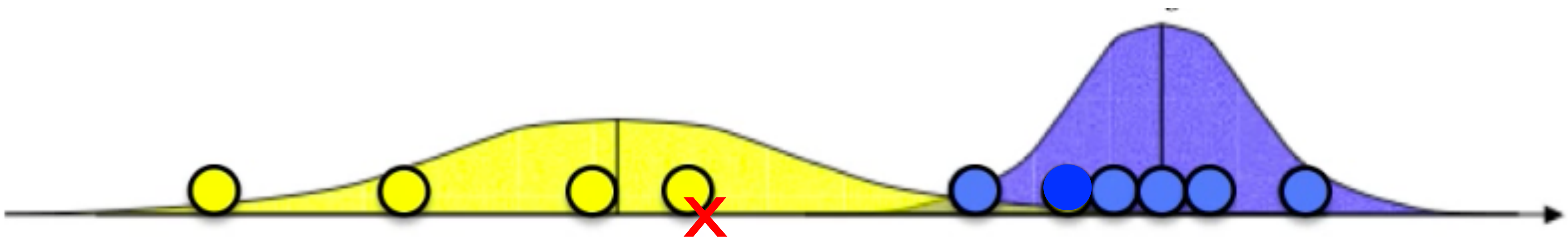


Single Gaussian



Mixture of two Gaussians

Supervised learning



Univariate Gaussian

$$\mu_{MLE} = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad \sigma^2_{MLE} = \frac{1}{N}\sum_{i=1}^{N} (x_i - \widehat{\mu})^2$$

Review: K-means clustering

- K-means clustering uses EM approach
  - choose an initial values for μ_k
  - repeat two steps
    - E-step: assign each example to the nearest prototype by minimizing J;

      → determine r_{nk}

$$\boxed{z_i^* = \arg\min_k ||\mathbf{x}_i - \boldsymbol{\mu}_k||_2^2}$$

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{otherwise.} \end{cases}$$

    - M-step: update the prototypes with the data points assigned;

      → determine μ_k with the new r_{nk}

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

$$2 \sum_{n=1}^{N} r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k) = 0$$

For each k,

set the derivative of J to 0 with respect to μ_k

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk} \mathbf{x}_n}{\sum_n r_{nk}}$$

EM for Gaussian mixture

$$r_{ik} \triangleq p(z_i = k|\mathbf{x}_i, \boldsymbol{\theta}) \quad = \quad \frac{p(z_i = k|\boldsymbol{\theta})p(\mathbf{x}_i|z_i = k, \boldsymbol{\theta})}{\sum_{k'=1}^{K} p(z_i = k'|\boldsymbol{\theta})p(\mathbf{x}_i|z_i = k', \boldsymbol{\theta})}$$

→ Responsibility of cluster k for data i

soft assignment vs hard assignment in E step

$$z_i^* = \arg\max_k r_{ik}$$

EM for Gaussian mixture

- E-step: evaluate the responsibilities (assignments)

$$r_{ik} = \frac{\pi_k p(\mathbf{x}_i|\boldsymbol{\theta}_k^{(t-1)})}{\sum_{k'} \pi_{k'} p(\mathbf{x}_i|\boldsymbol{\theta}_{k'}^{(t-1)})}$$

- M-step: re-estimate the means, covariances, and mixing coefficients

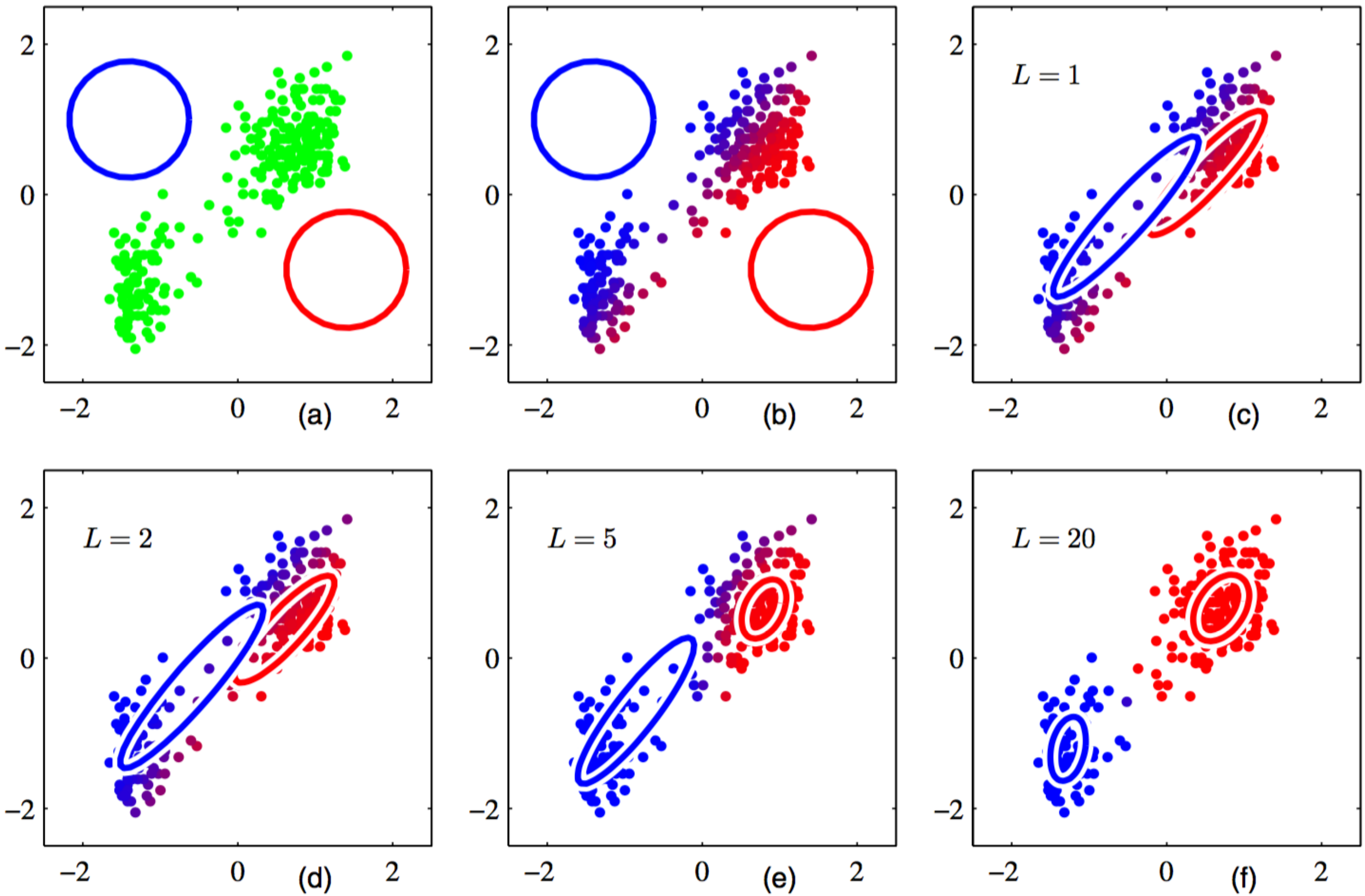
$$\pi_k = \frac{1}{N}\sum_i r_{ik} = \frac{r_k}{N} \longleftarrow \text{Weighted number of data assigned to cluster k}$$

$$\boldsymbol{\mu}_k = \frac{\sum_i r_{ik}\mathbf{x}_i}{r_k}$$

$$\boldsymbol{\Sigma}_k = \frac{\sum_i r_{ik}(\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T}{r_k}$$

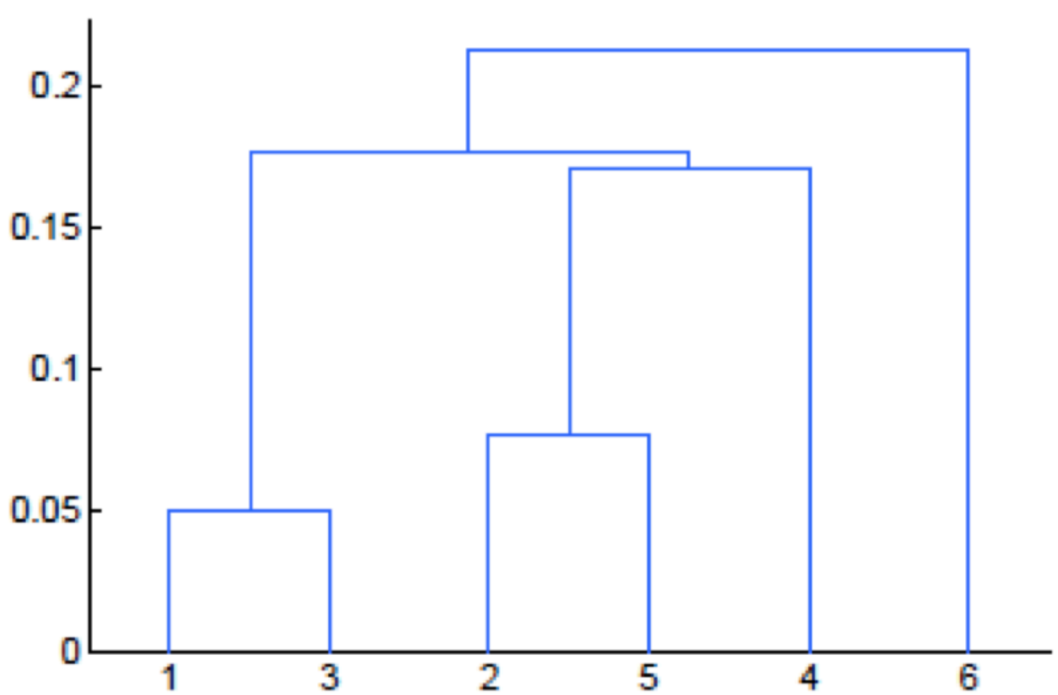
- The mean of cluster k is the weighted average of all data points assigned to cluster k
- The covariance is proportional to the weighted empirical scatter matrix

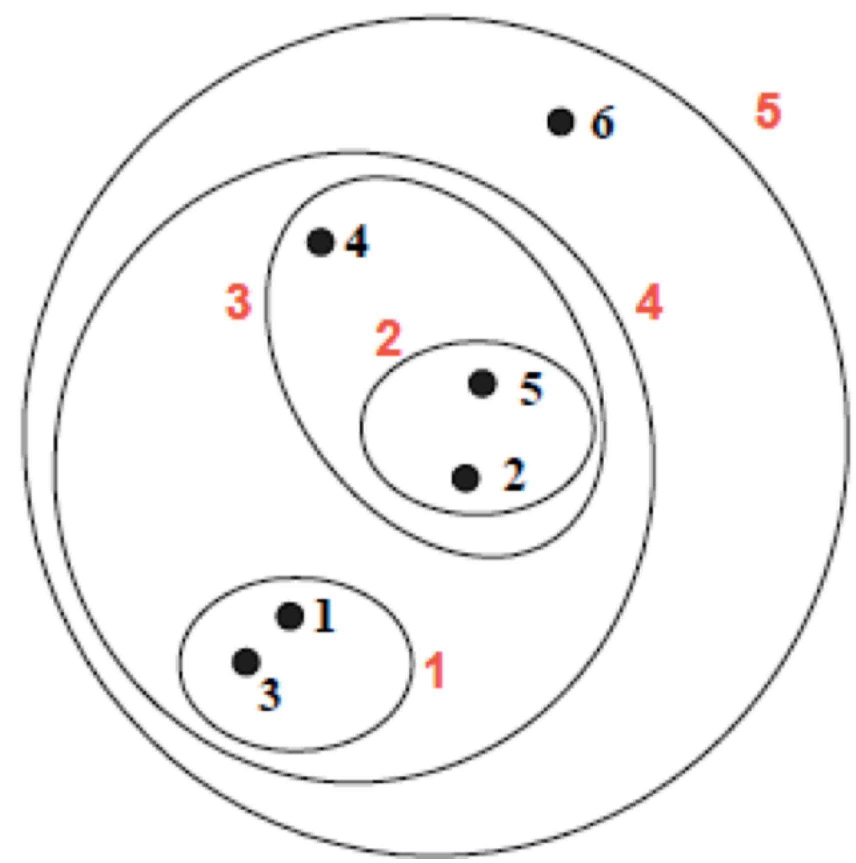
EM for Gaussian mixture



Hierarchical clustering

- use distance matrix
- do not need the number of clusters (=k) as input
- need to decide when to stop
- bottom-up(agglomerative) and top-down(divisive) approaches



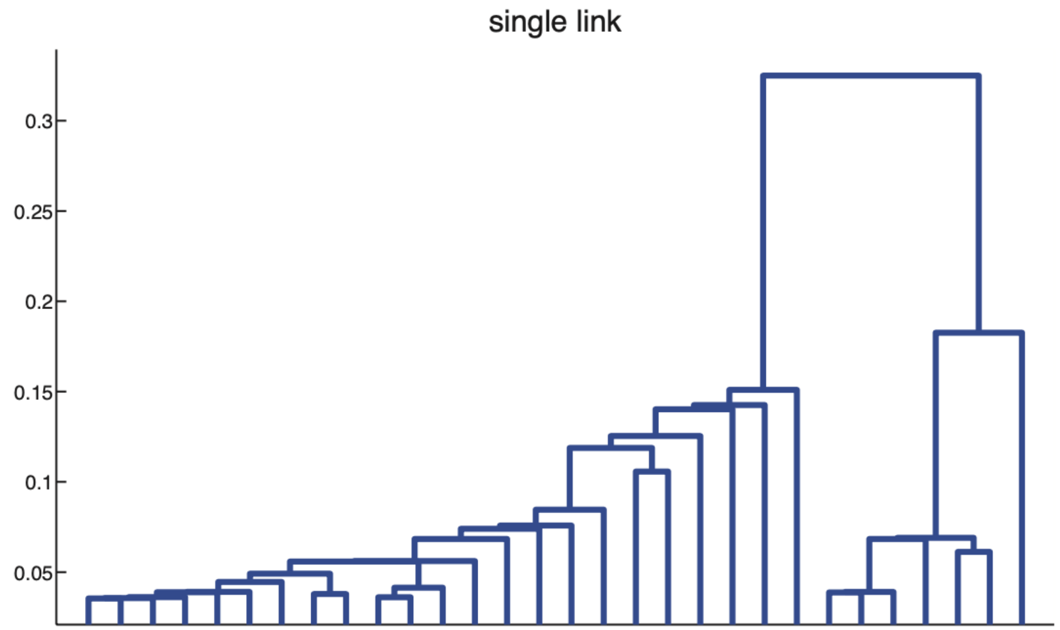


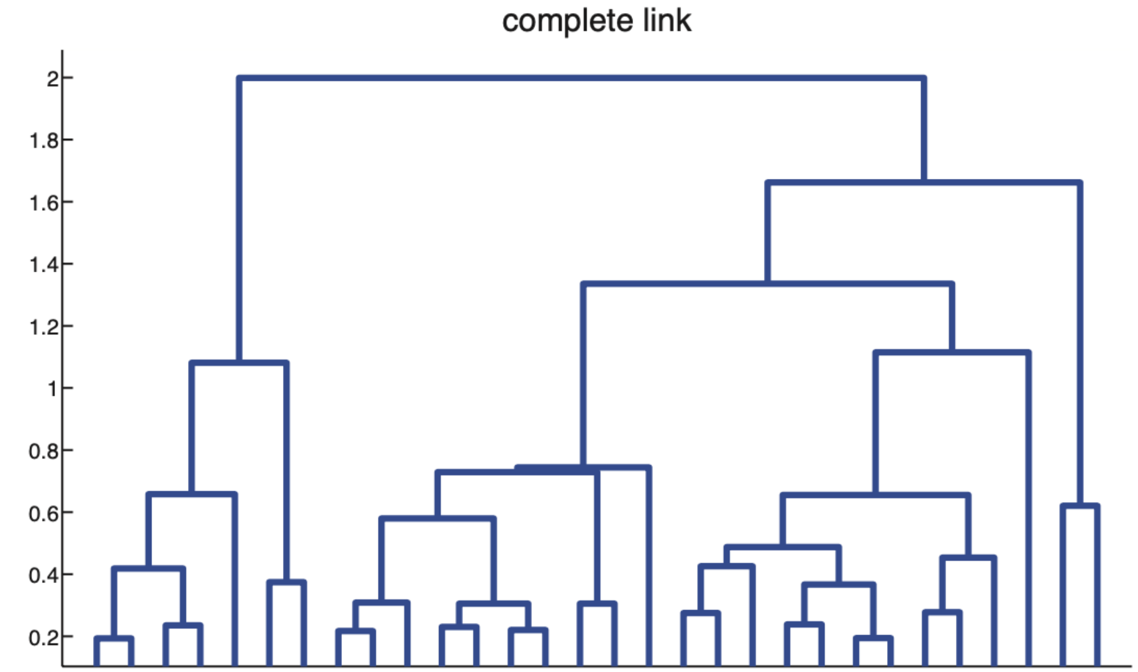
Agglomerative clustering

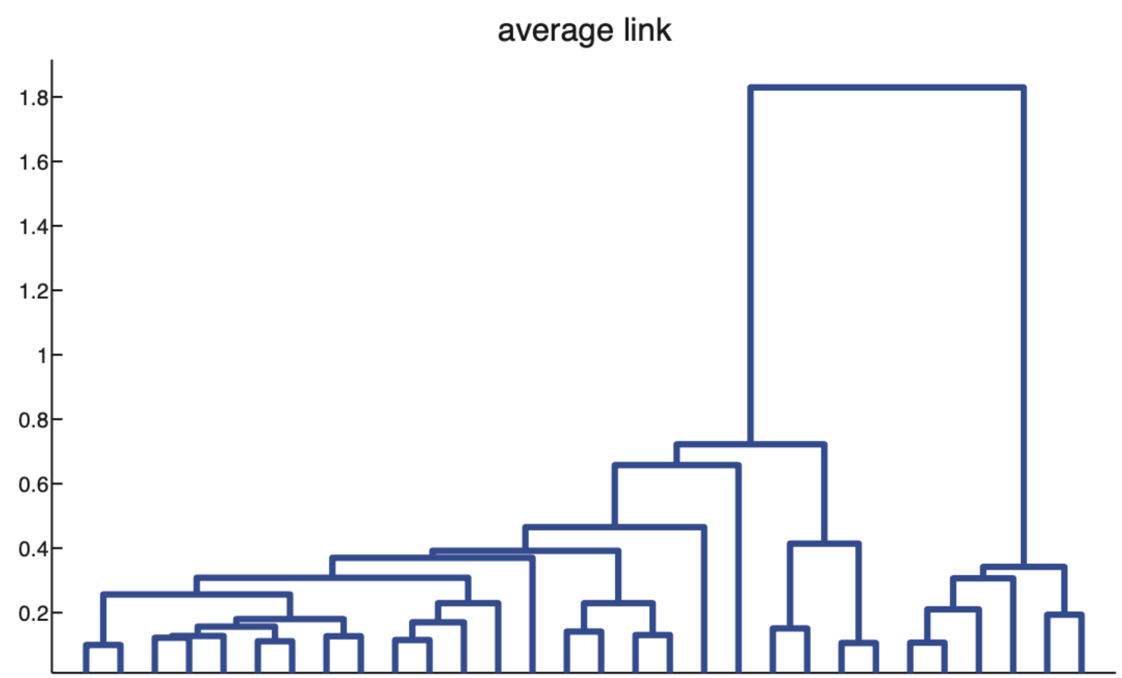
Algorithm

initialize clusters as singletons: **for** $i \leftarrow 1$ **to** n **do** $C_i \leftarrow \{i\}$;
initialize set of clusters available for merging: $S \leftarrow \{1, \ldots, n\}$;
repeat
    Pick 2 most similar clusters to merge: $(j, k) \leftarrow \arg\min_{j,k \in S} d_{j,k}$;
    Create new cluster $C_\ell \leftarrow C_j \cup C_k$;
    Mark j and k as unavailable: $S \leftarrow S \setminus \{j, k\}$;
    if $C_\ell \neq \{1, \ldots, n\}$ **then**
        Mark ℓ as available, $S \leftarrow S \cup \{\ell\}$;
    foreach $i \in S$ **do**
        Update dissimilarity matrix $d(i, \ell)$;
until *no more clusters are available for merging*;

Agglomerative clustering



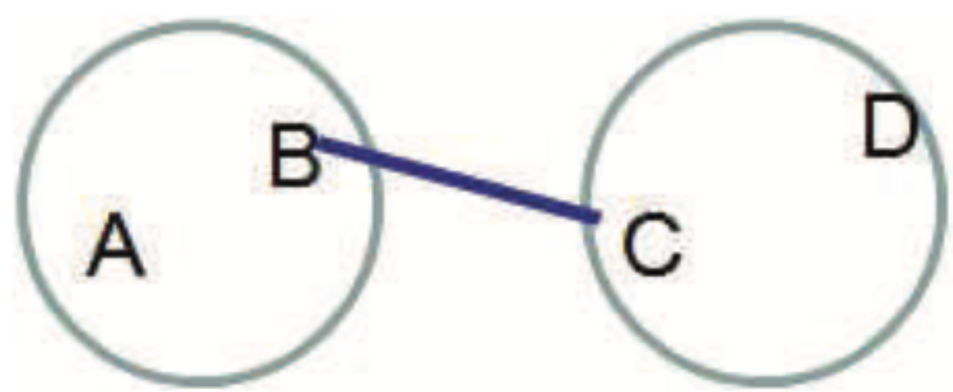




Single link clustering

- Nearest neighbor clustering
- The distance between two groups G and H is defined as the distance between the two closest members of each group

$$d_{SL}(G,H) = \min_{i \in G, i' \in H} d_{i,i'}$$



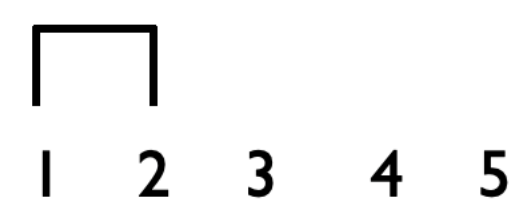
Bottom-up approach

$$
\begin{array}{c|ccccc|}
 & 1 & 2 & 3 & 4 & 5 \\
\hline
1 & 0 & & & & \\
2 & 2 & 0 & & & \\
3 & 6 & 3 & 0 & & \\
4 & 10 & 9 & 7 & 0 & \\
5 & 9 & 8 & 5 & 4 & 0
\end{array}
\longrightarrow
\begin{array}{c|cccc|}
 & 1,2 & 3 & 4 & 5 \\
\hline
1,2 & 0 & & & \\
3 & 3 & 0 & & \\
4 & 9 & 7 & 0 & \\
5 & 8 & 5 & 4 & 0
\end{array}
$$

$$d_{(1,2),3} = \min\{d_{1,3}, d_{2,3}\} = \min\{6, 3\} = 3$$

$$d_{(1,2),4} = \min\{d_{1,4}, d_{2,4}\} = \min\{10, 9\} = 9$$

$$d_{(1,2),5} = \min\{d_{1,5}, d_{2,5}\} = \min\{9, 8\} = 8$$

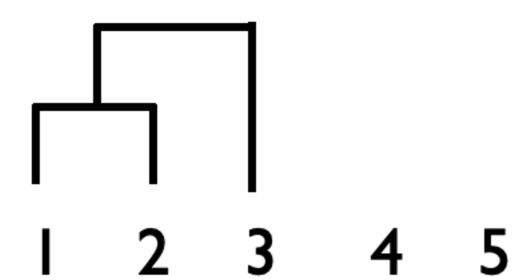


Bottom-up approach

$$\begin{array}{cc} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 6 & 3 & 0 & & \\ 10 & 9 & 7 & 0 & \\ 9 & 8 & 5 & 4 & 0 \end{bmatrix} \end{array} \longrightarrow \begin{array}{cc} & \begin{array}{cccc} 1,2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1,2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & & & \\ 3 & 0 & & \\ 9 & 7 & 0 & \\ 8 & 5 & 4 & 0 \end{bmatrix} \end{array} \longrightarrow \begin{array}{cc} & \begin{array}{ccc} 1,2,3 & 4 & 5 \end{array} \\ \begin{array}{c} 1,2,3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & & \\ 7 & 0 & \\ 5 & 4 & 0 \end{bmatrix} \end{array}$$

$$d_{(1,2,3),4} = \min\{d_{(1,2),4}, d_{3,4}\} = \min\{9, 7\} = 7$$

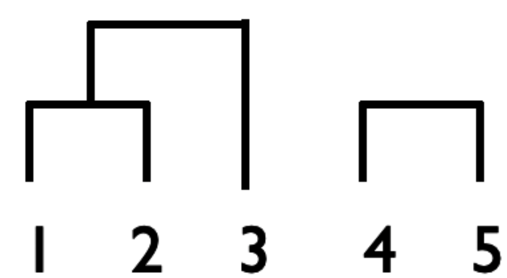
$$d_{(1,2,3),5} = \min\{d_{(1,2),4}, d_{3,4}\} = \min\{8, 5\} = 5$$



Bottom-up approach

$$
\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}
\begin{array}{c}
\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\
\begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 6 & 3 & 0 & & \\ 10 & 9 & 7 & 0 & \\ 9 & 8 & 5 & 4 & 0 \end{bmatrix}
\end{array}
\rightarrow
\begin{array}{c|cccc}
 & 1,2 & 3 & 4 & 5 \\ \hline
1,2 & 0 & & & \\
3 & 3 & 0 & & \\
4 & 9 & 7 & 0 & \\
5 & 8 & 5 & 4 & 0
\end{array}
\rightarrow
\begin{array}{c|ccc}
 & 1,2,3 & 4 & 5 \\ \hline
1,2,3 & 0 & & \\
4 & 7 & 0 & \\
5 & 5 & 4 & 0
\end{array}
\rightarrow
\begin{array}{c|cc}
 & 1,2,3 & 4,5 \\ \hline
1,2,3 & 0 & \\
4,5 & 5 & 0
\end{array}
$$

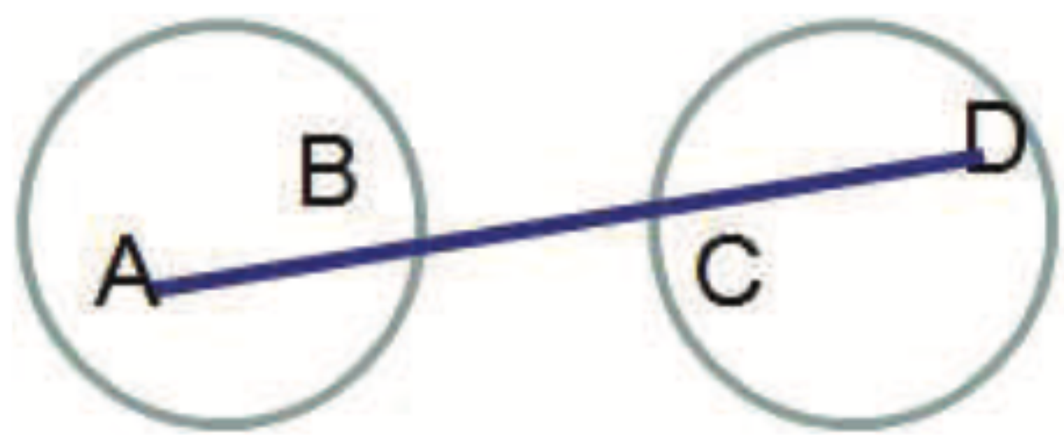
$$d_{(1,2,3),(4,5)} = \min\{d_{(1,2,3),4}, d_{(1,2,3),5}\} = \min\ \{7, 5\} = 5$$



Complete link clustering

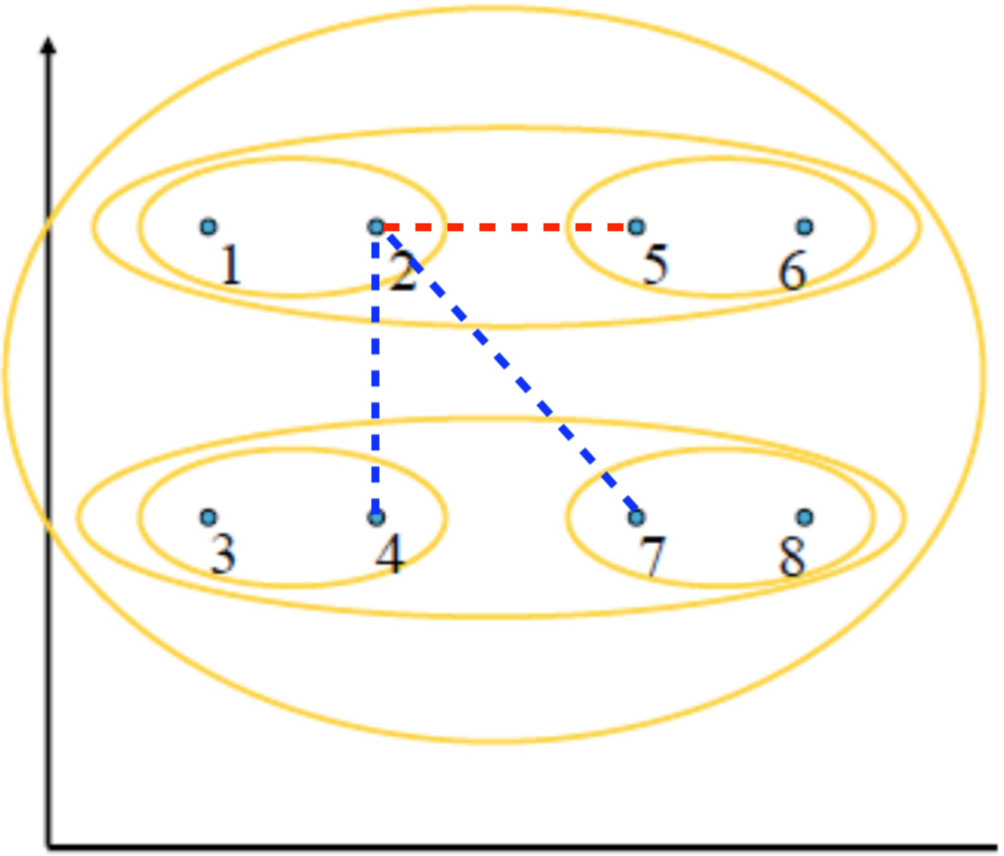
- Furthest neighbor clustering
- The distance between two groups G and H is defined as the distance between the two closest members of each group

$$d_{CL}(G,H) = \max_{i \in G, i' \in H} d_{i,i'}$$

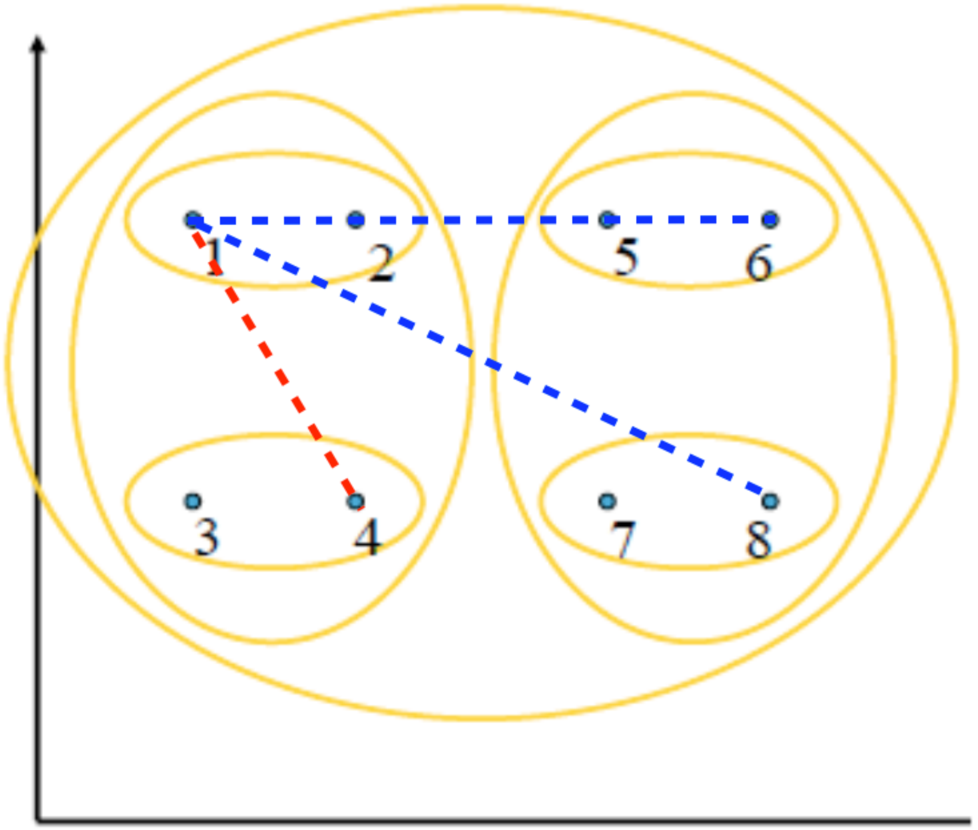


Bottom-up approach

single linkage



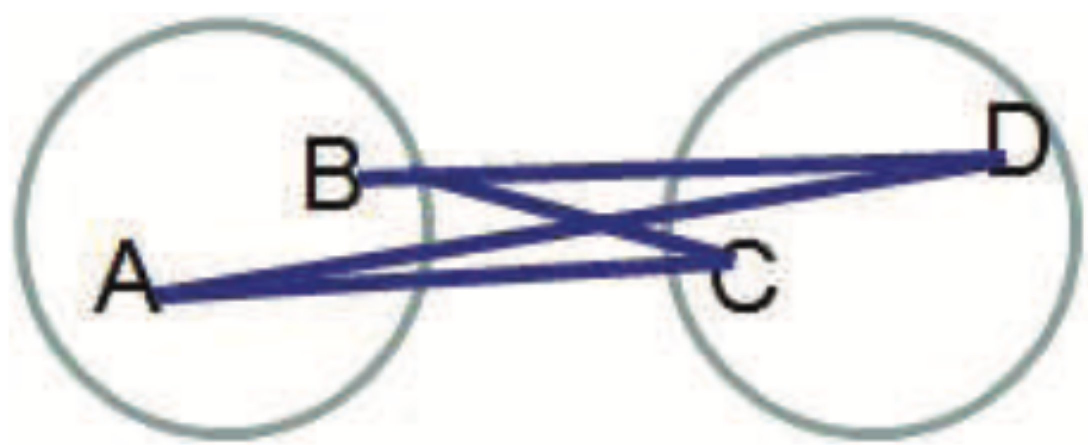
complete linkage



Average link clustering

- Measures the average distance between all pairs

$$d_{avg}(G, H) = \frac{1}{n_G n_H} \sum_{i \in G} \sum_{i' \in H} d_{i,i'}$$



Ward's method

- The distance between two clusters is how much the sum of squares will increase when the clusters are merged
- Keep the growth of this merging cost as small as possible

$$\Delta(A,B) = \sum_{i\in A\cup B} \|\vec{x}_i - \vec{m}_{A\cup B}\|^2 - \sum_{i\in A} \|\vec{x}_i - \vec{m}_A\|^2 - \sum_{i\in B} \|\vec{x}_i - \vec{m}_B\|^2$$

Divisive clustering

Bisecting k-means

- Pick the cluster with the largest diameter and split it using the k–means algorithm with K=2