Business Intelligence

TICS-423
Universidad Adolfo Ibáñez

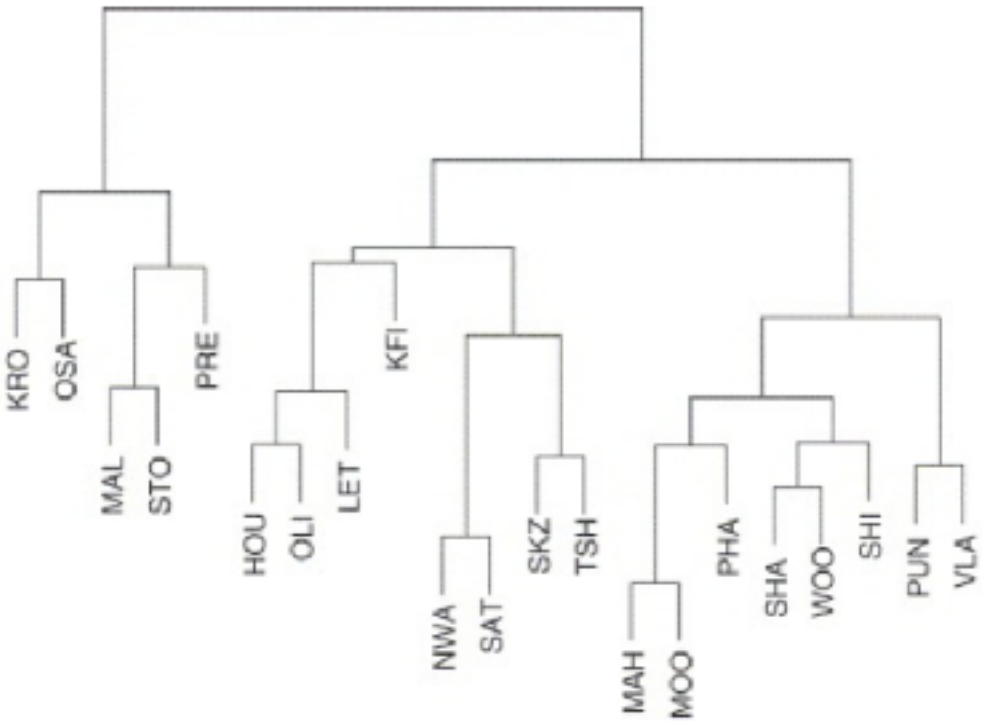
Week 11: 10 October - 14 October, 2016

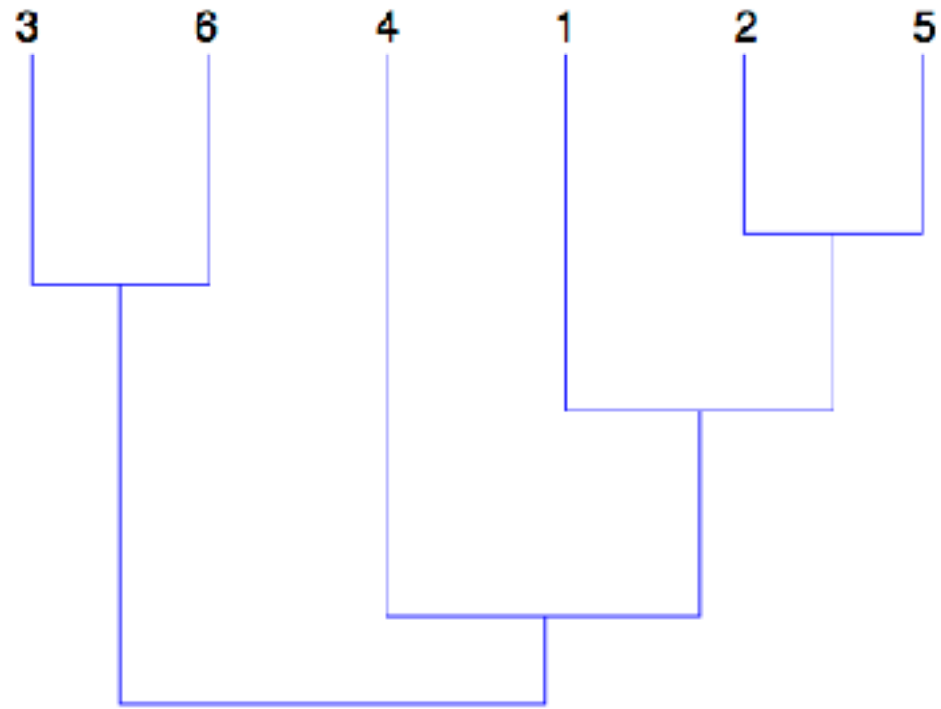
Claudio Diaz
Sebastián Moreno
Gonzalo Ruz

Descriptive modelling
Clustering
Hierarchical methods

DM, clustering, hierarchical methods

- The hierarchical methods can be agglomerative or divisive, in both cases a dendrogram is generated showing the sequences of merges or splits.





Agglomerative

Divisive

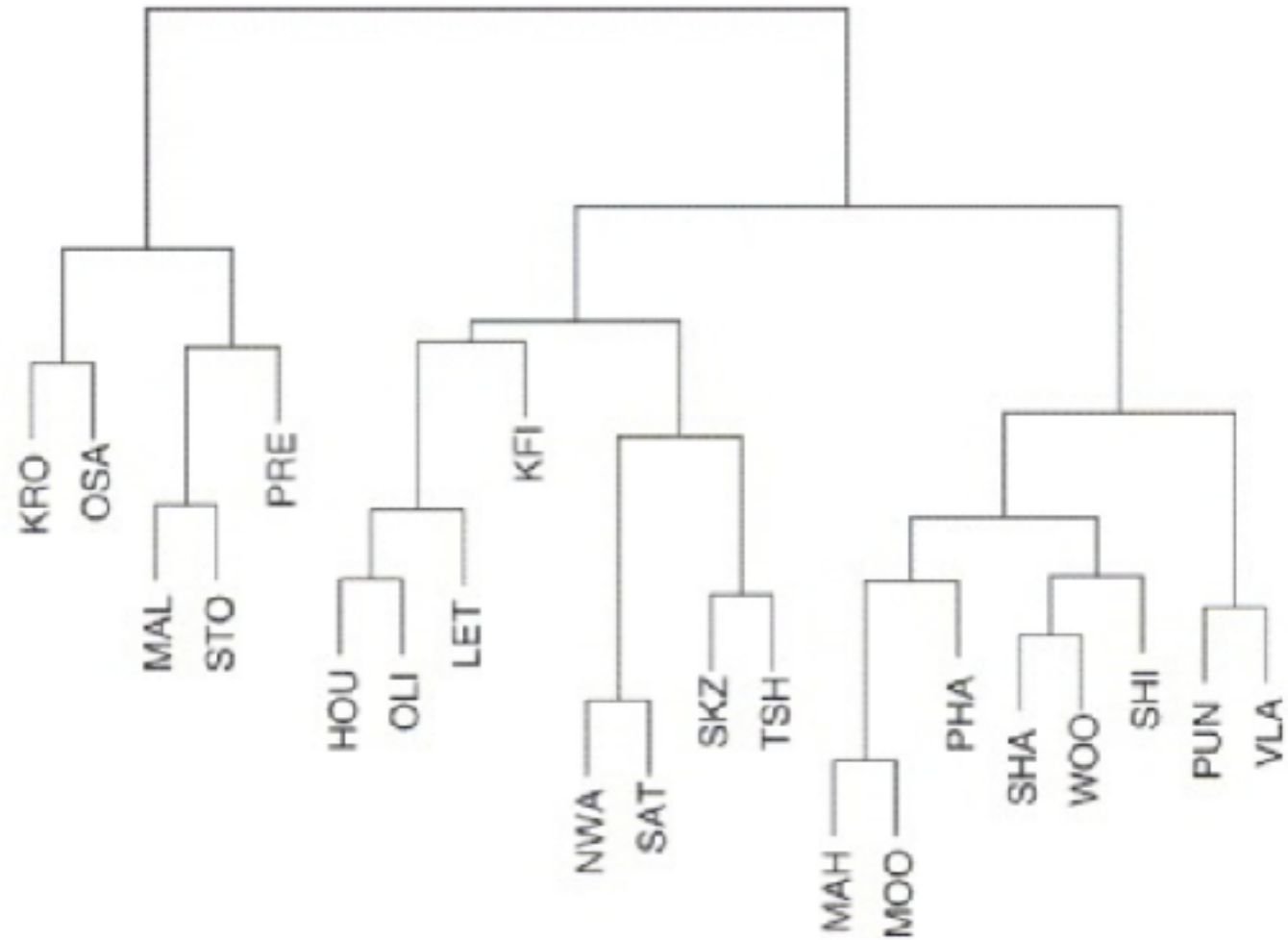
DM, clustering, hierarchical methods

- Task Specification: **Descriptive Modeling**
- Data Representation: **Numerical data**
- Knowledge representation: Agglomerative and divisive clustering
  Model space: hierarchy of groupings from size 1 to n or from n to 1.
- Learning
  - Search algorithm: Greedy, heuristic search successively chooses pair of clusters to merge/split that minimize/maximize distance
  - Scoring function: Distance measure between two clusters (e.g., single link), considers pairwise distances between two sets of nodes
- Output: Dendrograms depicts sequences of merges or splits and height indicates distance

Descriptive modelling
Clustering
Hierarchical methods: **Agglomerative clustering**

DM, clustering, hierarchical methods, agglomerative

- Agglomerative: It starts with the points as individual clusters, then at each step, merge the closest pair of clusters until only one cluster left.



DM, clustering, hierarchical methods, agglomerative

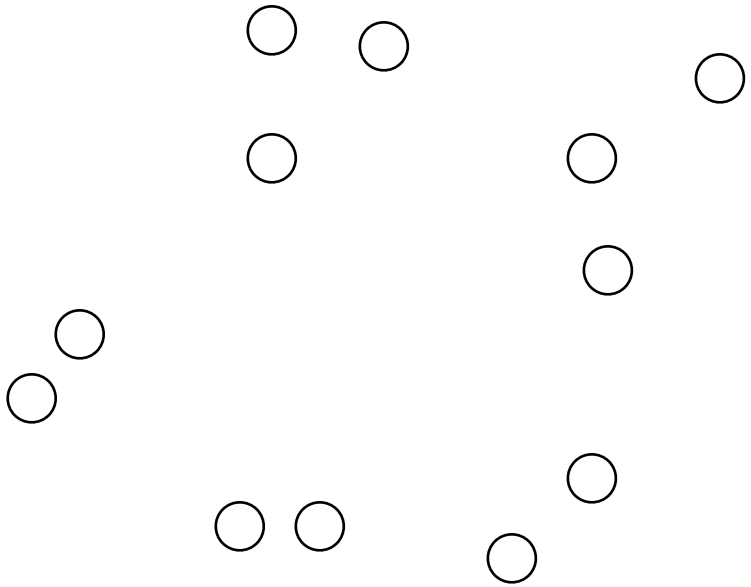
- The basic algorithm for agglomerative is straightforward
  Let each data point be a cluster
  Compute the proximity matrix (distance matrix among each cluster)
  Repeat
    Merge the two closest clusters
    Update the proximity matrix
  Until only a single cluster remains

- Key operation is the computation of the proximity of two clusters

- Different approaches to defining the distance between clusters distinguish the different algorithms

DM, clustering, hierarchical methods, agglomerative

- Start with clusters of individual points and a proximity matrix

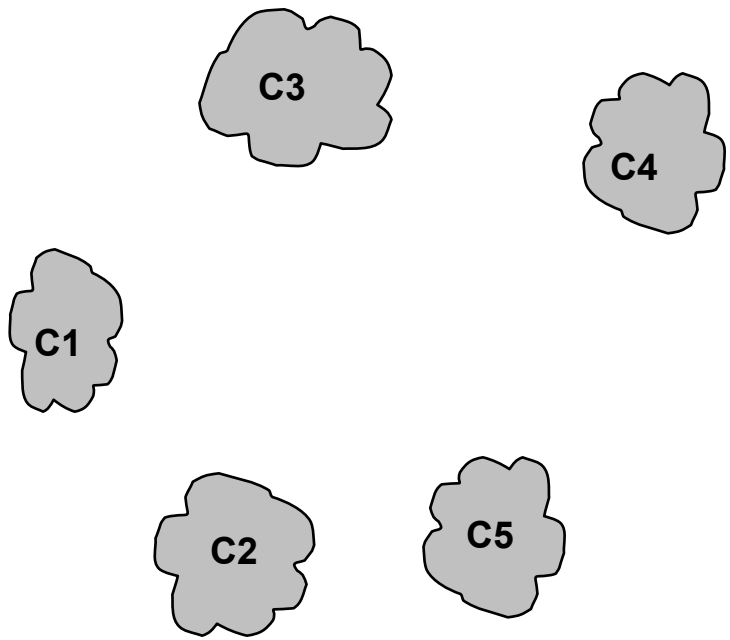


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

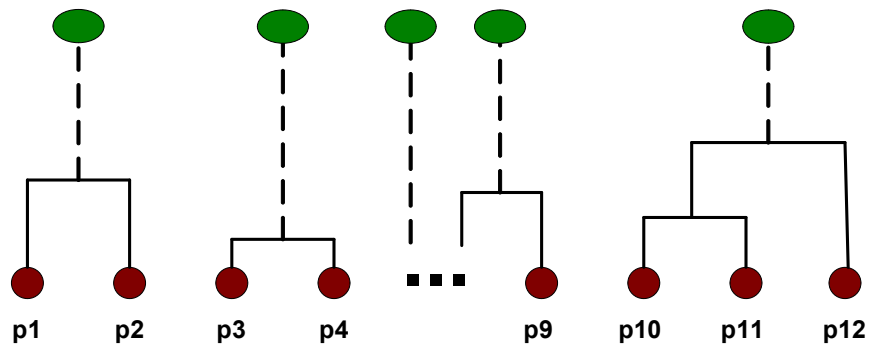
DM, clustering, hierarchical methods, agglomerative

- After some merging steps, we have some clusters



	C1	C2	C3	C4	C5
C1	0,0	0,5	0,6	1,3	1,2
C2		0,0	1,1	1,2	0,4
C3			0,0	0,7	1,1
C4				0,0	0,9
C5					0,0

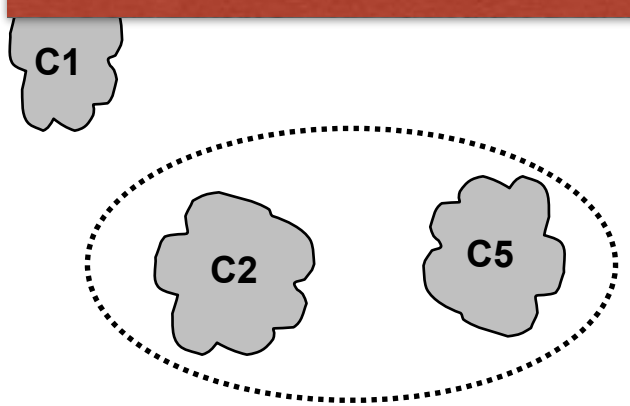
Proximity Matrix

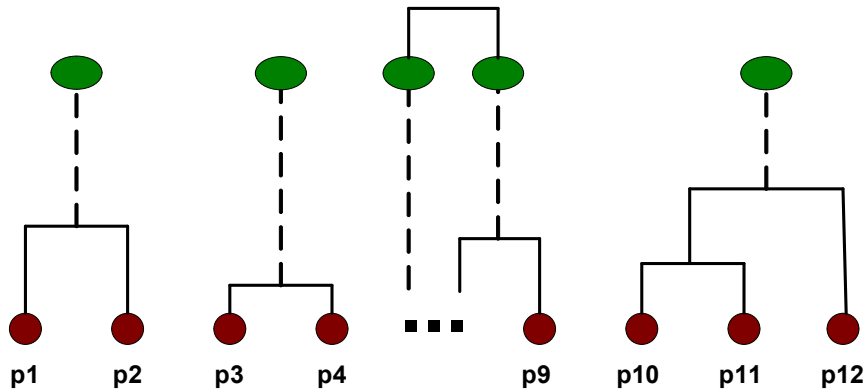


DM, clustering, hierarchical methods, agglomerative

- After some merging steps, we have some clusters
- We merge the two closest clusters
- Update the proximity matrix.

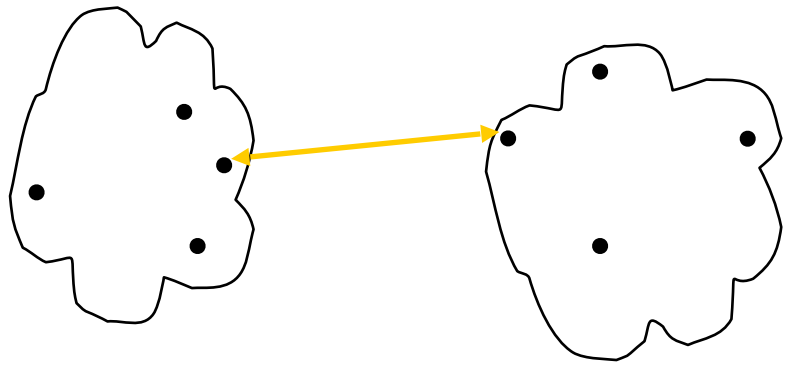
How can we updated the distance between clusters?





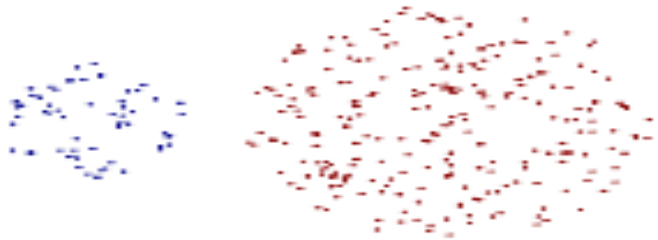
DM, clustering, hierarchical methods, agglomerative

- **Single linkage:** The distance between clusters is based on the two most similar (closest) points in the different clusters

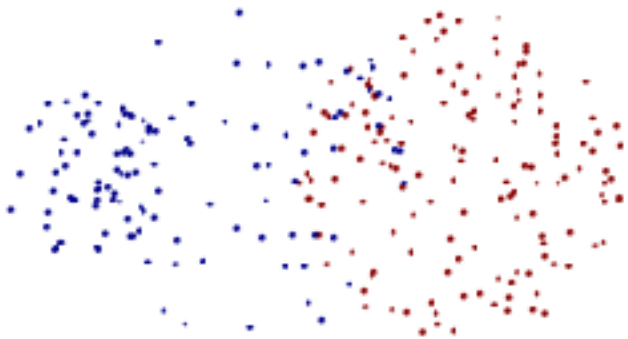


$$D(C_i, C_j) = \mathbf{min}\{d(x, y) | \mathbf{x} \in C_i, \mathbf{y} \in C_j\}$$

- **Strengths:**
  Produces long thin clusters

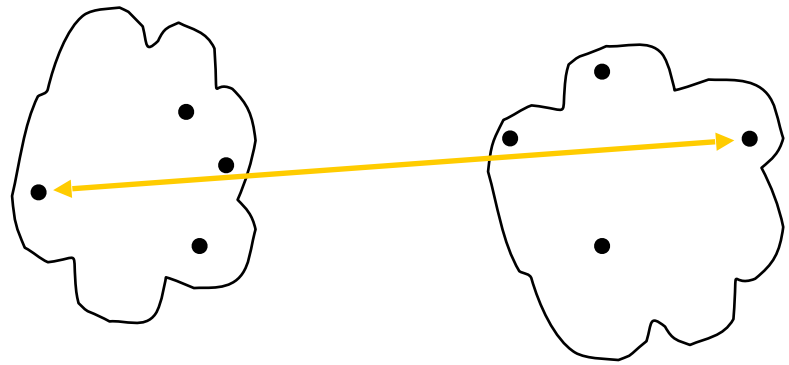


- **Limitations:**
  Sensitive to outliers



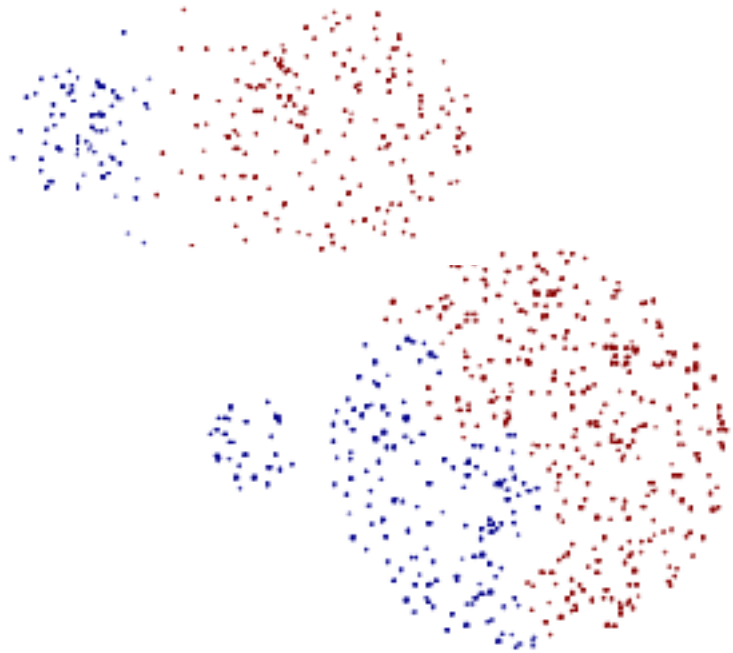
DM, clustering, hierarchical methods, agglomerative

- **Complete linkage:** The distance between clusters is based on the two most dissimilar (most distant) points in the different clusters



$$D(C_i, C_j) = \mathbf{max}\{d(x, y)|\mathbf{x} \in C_i, \mathbf{y} \in C_j\}$$

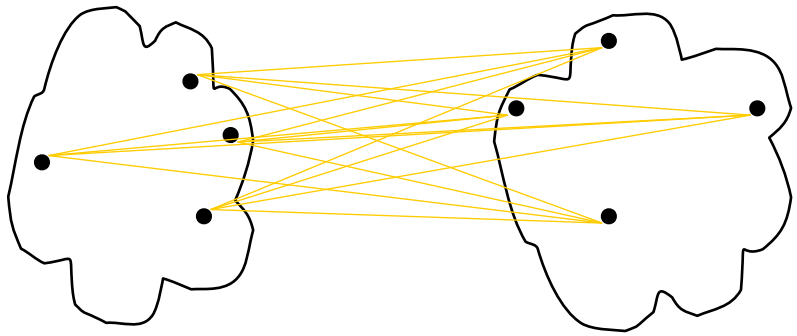
- **Strengths:**
  Less susceptible to outliers



- **Limitations:**
  Tends to break large clusters
  Biased towards globular clusters

DM, clustering, hierarchical methods, agglomerative

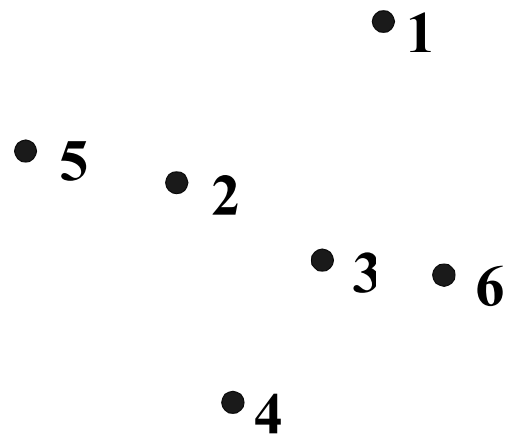
- **Average linkage:** The distance between clusters is based on the average distance among all points in the different clusters



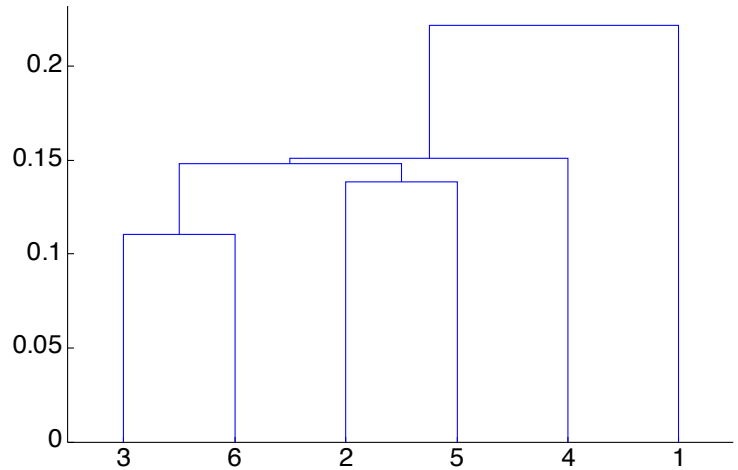
$$D(C_i, C_j) = \mathbf{avg}\{d(x, y) | \mathbf{x} \in C_i, \mathbf{y} \in C_j\}$$

- Compromise between Single and Complete Linkage
- **Strengths:**
  Less susceptible to outliers
- **Limitations:**
  Biased towards globular clusters

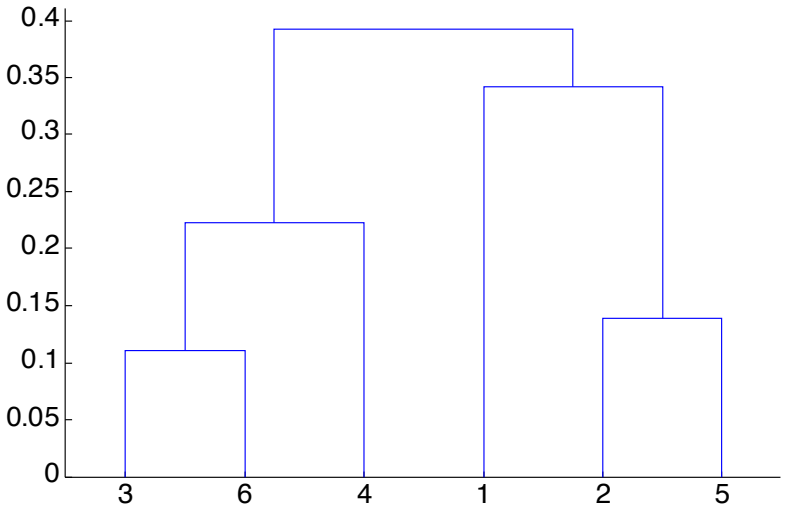
DM, clustering, hierarchical methods, agglomerative



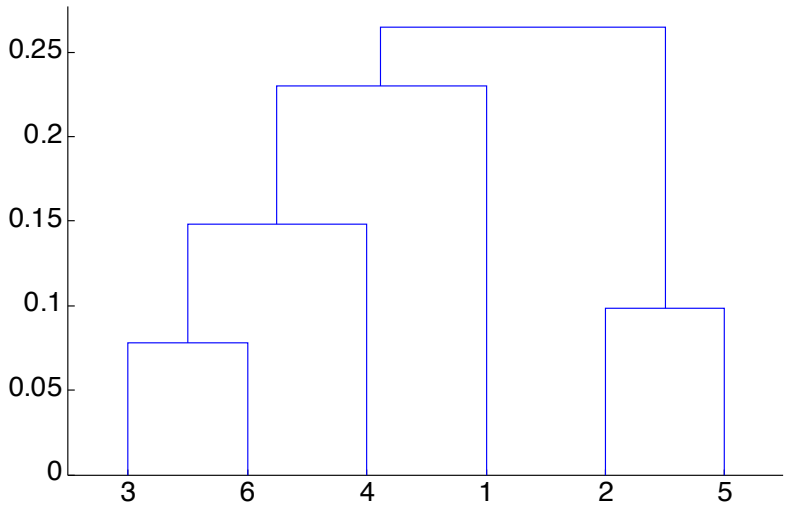
Original data



Single linkage



Complete linkage



Average linkage

DM, clustering, HM, agglomerative, problems

- The algorithm is too expensive $O(n^3)$
  There are n steps, to join clusters, and at each step we calculate the proximity matrix $O(n^2)$.

- Once a decision is made to combine two clusters, it cannot be undone

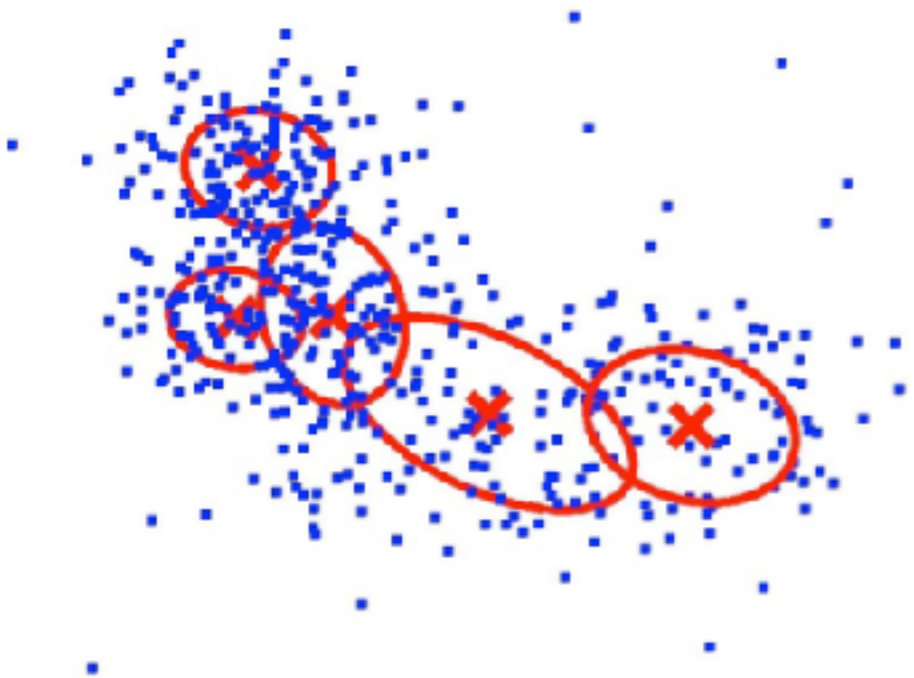
- No objective function is directly minimized

- Different schemes have different problems

Descriptive modelling
Clustering
Probabilistic methods

DM, clustering, probabilistic methods

- Probabilistic methods provides full distributional description for each component, generating soft clusters. i.e. given the model, each point has a K-component vector of membership probabilities.



$$f(x) = \sum_{k=1}^{K} w_k f_k(x;\theta)$$

probability of observing x

likelihood of point belonging to cluster k

likelihood of x being generated from cluster k

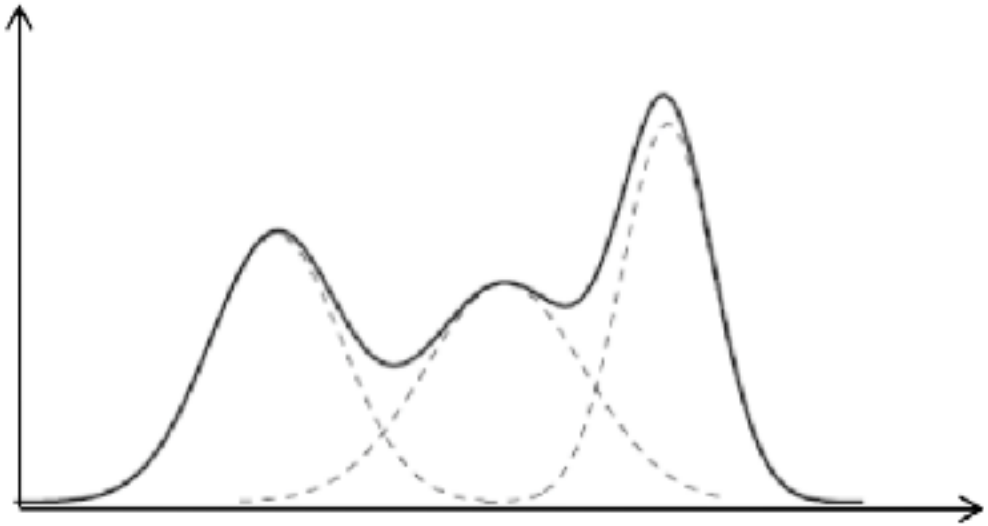
DM, clustering, probabilistic methods

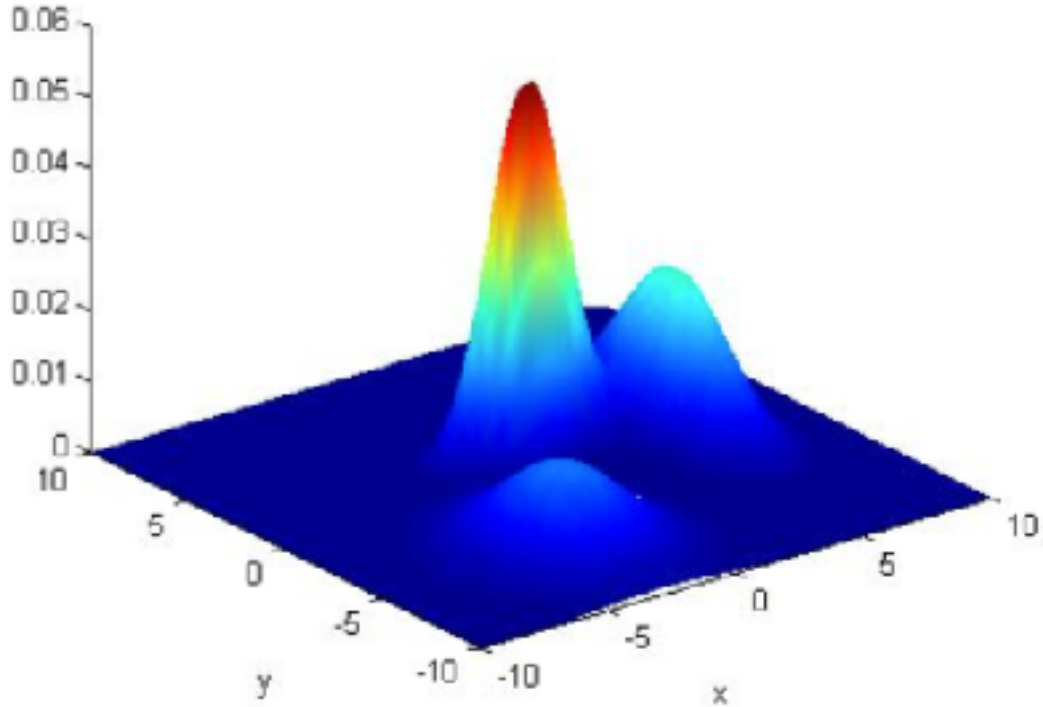
- Task Specification: **Descriptive Modeling**
- Data Representation: **Numerical data**
- Knowledge representation: Parametric model
  parameters = mixture coefficient and component parameters
- Learning
  - Search algorithm: Different methods, among them Expectation maximization, which iteratively find parameters that maximize likelihood and predicts cluster memberships
  - Scoring function: Likelihood
- Output: A nxK matrix of membership probabilities (n for the points, K for the number of clusters).

Descriptive modelling
Clustering
Probabilistic methods: **Gaussian mixture model**

DM, clustering, probabilistic methods, GMM

- A Gaussian Mixture Model (GMM) assumes that the data was generated from a mixture of K multi-dimensional Gaussians, where each component has parameters: $N_k(\mu_k, \sum_k)$





$$p(x) = \sum_{k=1}^{K} p(k)p(x|k) = \sum_{k=1}^{K} p(k)p(x|x \sim N(\mu_k, \Sigma_k)) = \sum_{k=1}^{K} \pi_k N(\mu_k, \Sigma_k)$$

DM, clustering, GMM, multivariate gaussian

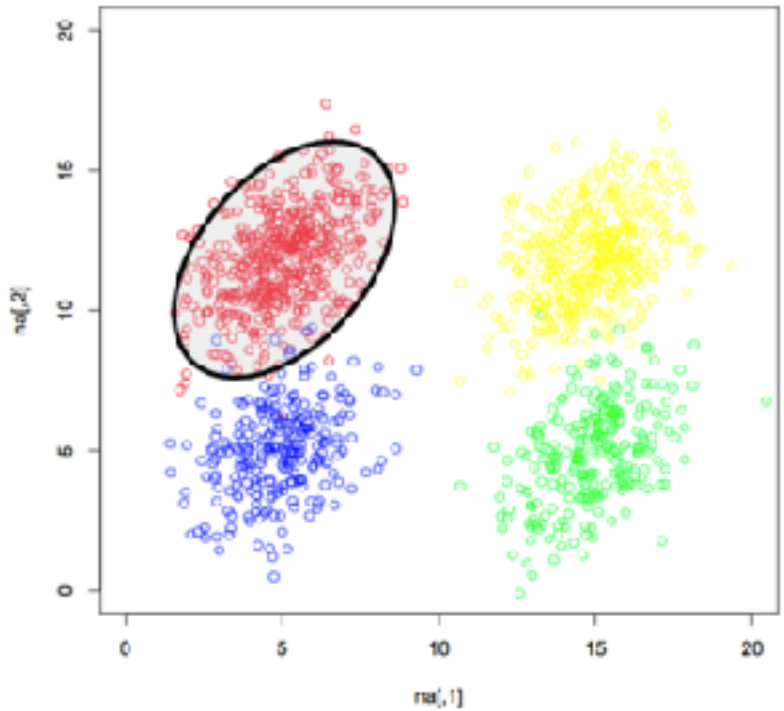
- A multi-dimensional Gaussians, for data with p dimensions is specified as follows

$$x \sim N(\mu, \Sigma)$$

where

$$\mu = (E[X_1], \ldots, E[X_p])$$

$$\Sigma = \begin{bmatrix} Var(X_1) & \cdots & Cov(X_1, X_p) \\ \cdots & \cdots & \cdots \\ Cov(X_1, X_p) & \cdots & Var(X_p) \end{bmatrix}$$



$$p(\mathbf{x}) = p(x_1, \cdots, x_p) = \frac{1}{\sqrt{(2\pi)^p |\Sigma|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu)\right)$$

DM, clustering, GMM, learning

- How can we learn the different parameters?

$$p(x) = \sum_{k=1}^{K} \pi_k N(\mu_k, \Sigma_k)$$

Mixing coefficients

Component means and covariance matrix

DM, clustering, GMM, learning

- How can we learn the different parameters?

$$p(x) = \sum_{k=1}^{K} \pi_k N(\mu_k, \Sigma_k)$$

- The log likelihood takes the following form, where there is no closed form solution for the MLE

$$\begin{aligned} \ln P(D|\pi, \mu, \Sigma) &= \sum_{i=1}^{N} \ln p(x_i|M) \\ &= \sum_{i=1}^{N} \ln \left[\sum_{k=1}^{K} \pi_k N(\mu_k, \Sigma_k)) \right] \end{aligned}$$

DM, clustering, GMM, learning

- If we assume that given the data point x_i, we know the value of a new variable Z, where $z_{ij}=1$ implies that x_i was generated by the normal distribution j.

Z	C_1	C_2	…	C_K
x_1	0	1		0
x_2	0	0		1
…				
x_n	0	0		0

$$\Rightarrow \begin{cases} p(z_{ik}=1) = \pi_k \Rightarrow P(\mathbf{z}_i) = \prod_{i=1}^{K} \pi_k^{z_{ik}} \\ p(\mathbf{x}_i|z_{ik}=1) = N(\mathbf{x}_i|\mu_k, \Sigma_k) \\ p(\mathbf{x}_i, \mathbf{z}_i) = \prod_{i=1}^{K} \pi_k^{z_{ik}} N(\mathbf{x}_i|\mu_k, \Sigma_k)^{z_{ik}} \end{cases}$$

$$p(\mathbf{x}) = \sum_{k=1}^{K} p(\mathbf{z})p(\mathbf{x}|\mathbf{z}) = \sum_{k=1}^{K} \pi_k N(\mathbf{x}|\mu_k, \Sigma_k)$$

- We arrive to the same expression than before, but we a new variable.
- However, now we can also work with the join distribution p(**x**,**z**)

DM, clustering, GMM, learning

- Considering that

$$\ln p(\mathbf{x}|\pi,\mu,\Sigma) = \ln \left[\sum_{k=1}^{K} p(\mathbf{x},\mathbf{z}|\pi,\mu,\Sigma) \right]$$

- We can calculate the log likelihood of p(**X**,**Z**)

Unfortunately, Z is unknown

$$\ln p(D,\mathbf{Z}|\pi,\mu,\Sigma) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}(\ln \pi_k + \ln N(\mathbf{x}_n|\mu_k,\Sigma_k))$$

- Leading to the following MLE solutions

$$\pi_k = \frac{1}{N} z_{nk} \qquad N(\mu_k,\Sigma_k)$$

Each normal distribution can be calculated using only the points belonging to the cluster k.

DM, clustering, GMM, learning

- We can use the Expectation-Maximization algorithm, which instead of maximise the log-likelihood, it maximizes the expected log likelihood with respect to **Z**

$$E_Z[\ln p(D, \mathbf{Z}|\pi, \mu, \Sigma)] = \sum_{n=1}^{N}\sum_{k=1}^{K} E[z_{nk}](\ln \pi_k + \ln N(\mathbf{x}_n|\mu_k, \Sigma_k))$$

$$E[z_{nk}] = \frac{\pi_k N(\mathbf{x}_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(\mathbf{x}_n|\mu_j, \Sigma_j)}$$

- $E[z_{nk}]$ is the ratio of the weighted probability of being generated by $N(\boldsymbol{\mu}_k, \sum_k)$ with respect to the K possible normal distributions.

DM, clustering, GMM, learning

- Algorithm

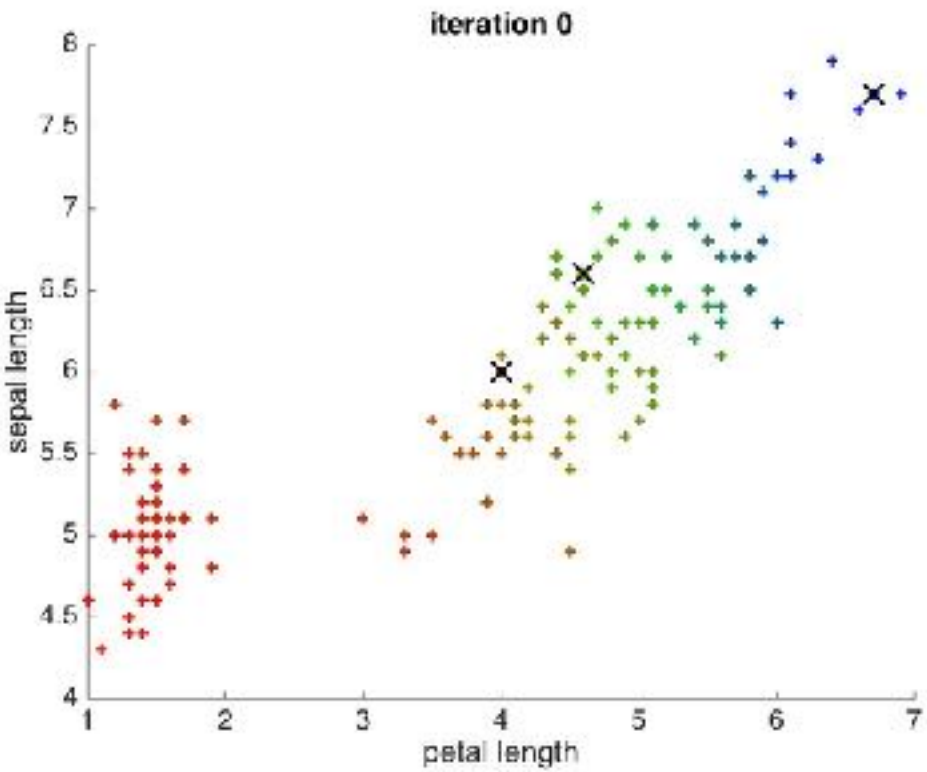
1. Choose an initial setting for all parameters $\boldsymbol{\pi}$ $\boldsymbol{\mu}_k$ $\sum_k$
2. **E-step:** estimate all $E[z_{nk}] = \dfrac{\pi_k N(\mathbf{x}_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(\mathbf{x}_n|\mu_j, \Sigma_j)}$
3. **M-step:** maximize $E_Z[\ln p(D, \mathbf{Z}|\pi, \mu, \Sigma)] = \sum_{n=1}^{N}\sum_{k=1}^{K} E[z_{nk}](\ln \pi_k + \ln N(\mathbf{x}_n|\mu_k, \Sigma_k))$

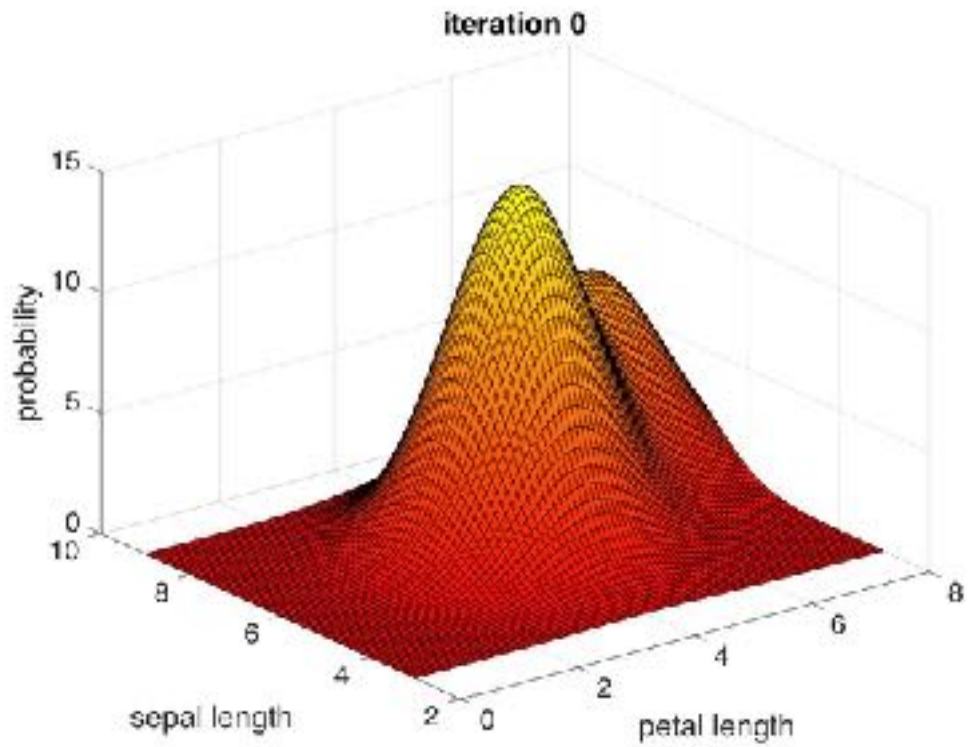
$$\mu_k = \frac{1}{N_k}\sum_{i=1}^{N} E[z_{ik}]\mathbf{x}_i \qquad N_k = \sum_{i=1}^{N} E[z_{ik}]$$

$$\Sigma_k = \frac{1}{N_k}\sum_{i=1}^{N} E[z_{ik}](\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^T \qquad \pi_k = \frac{N_k}{N}$$

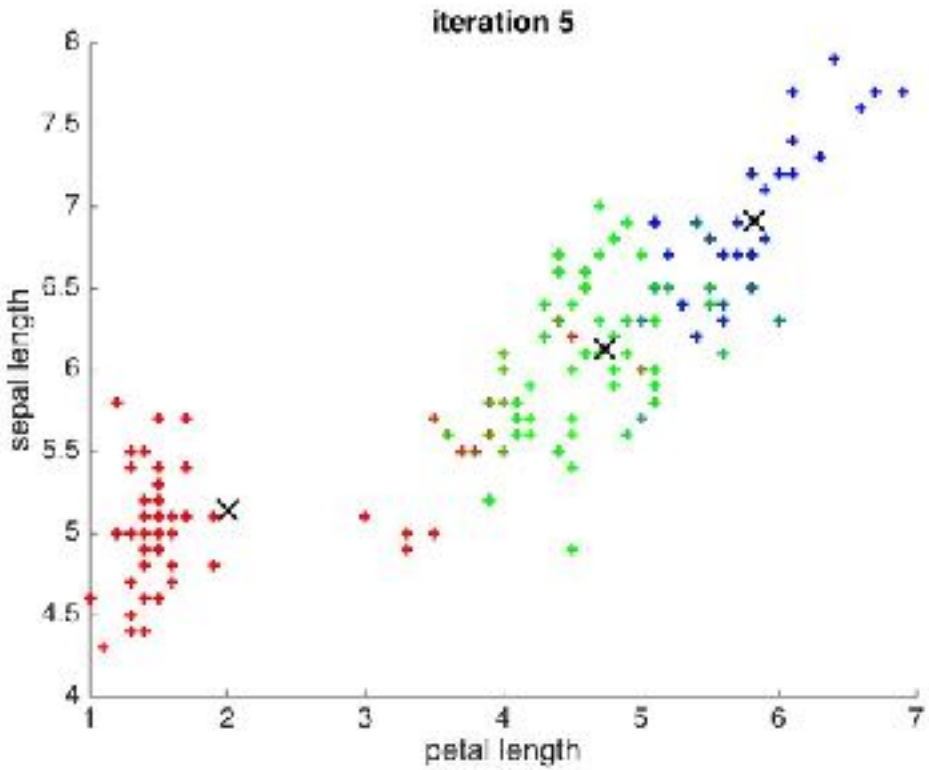
4. Check for convergence of the parameters, if convergence is not satisfied go to **step 2**.

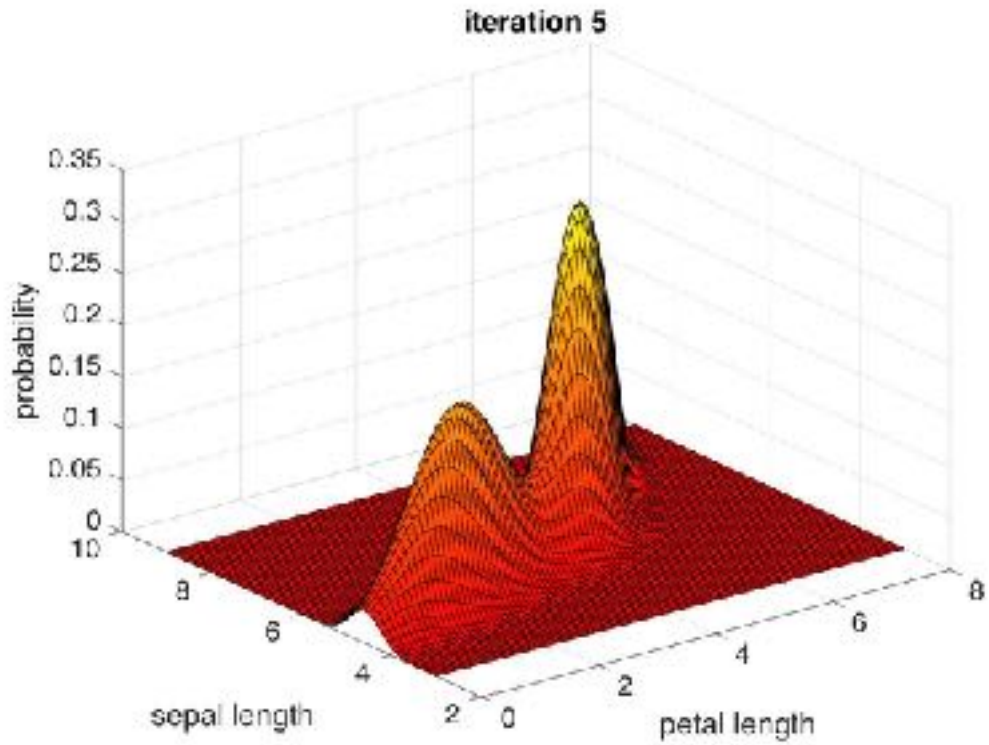
DM, clustering, GMM, petal dataset example



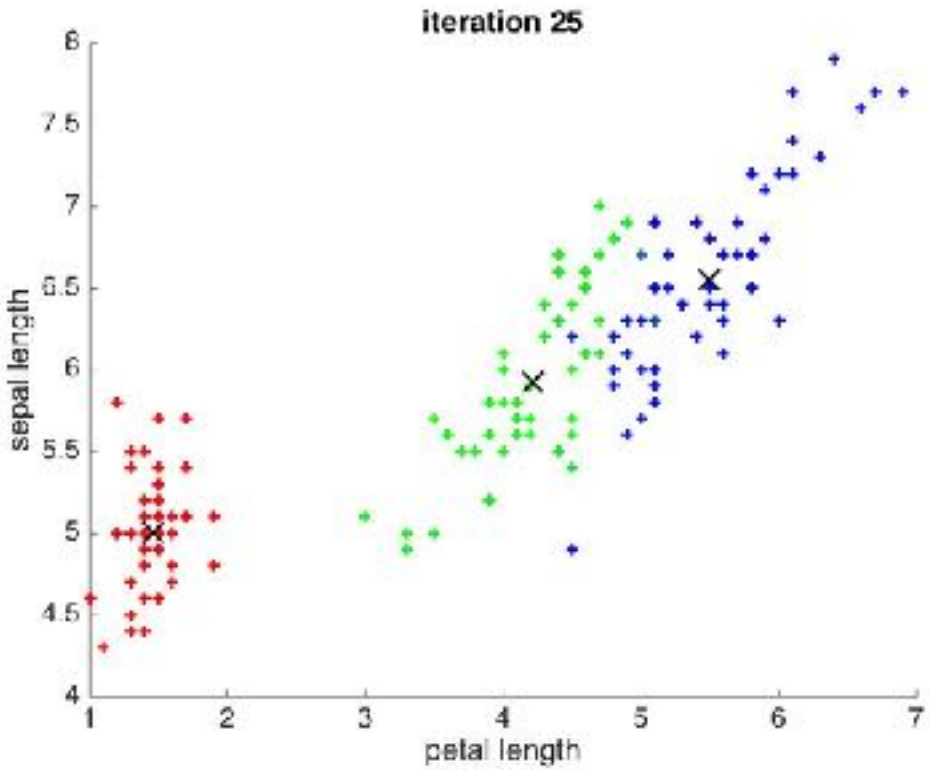


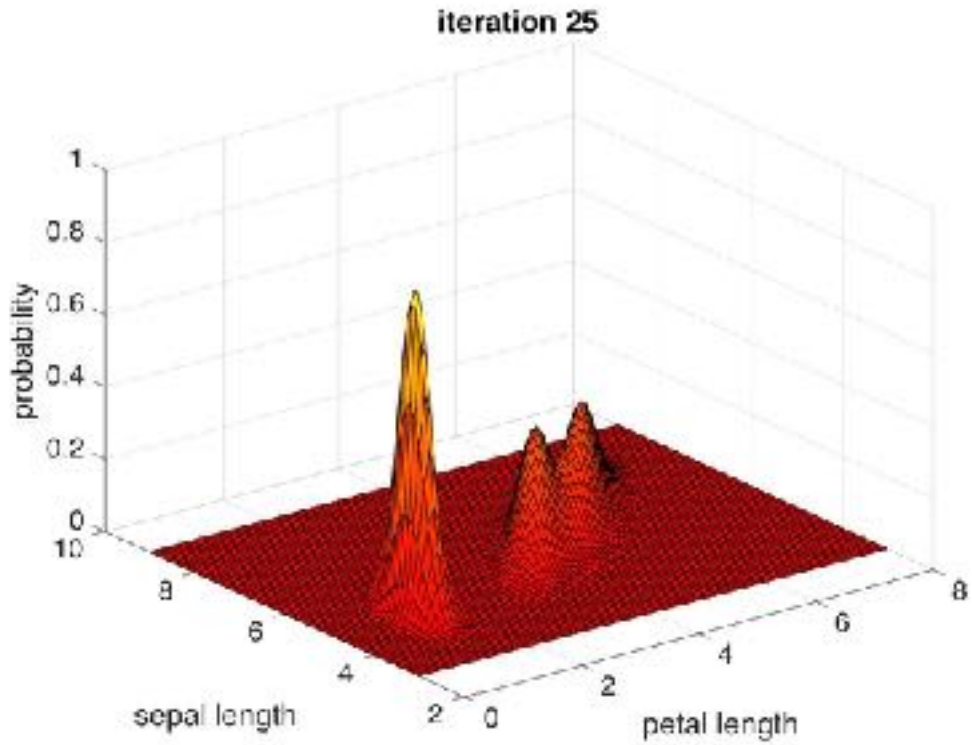
DM, clustering, GMM, petal dataset example





DM, clustering, GMM, petal dataset example

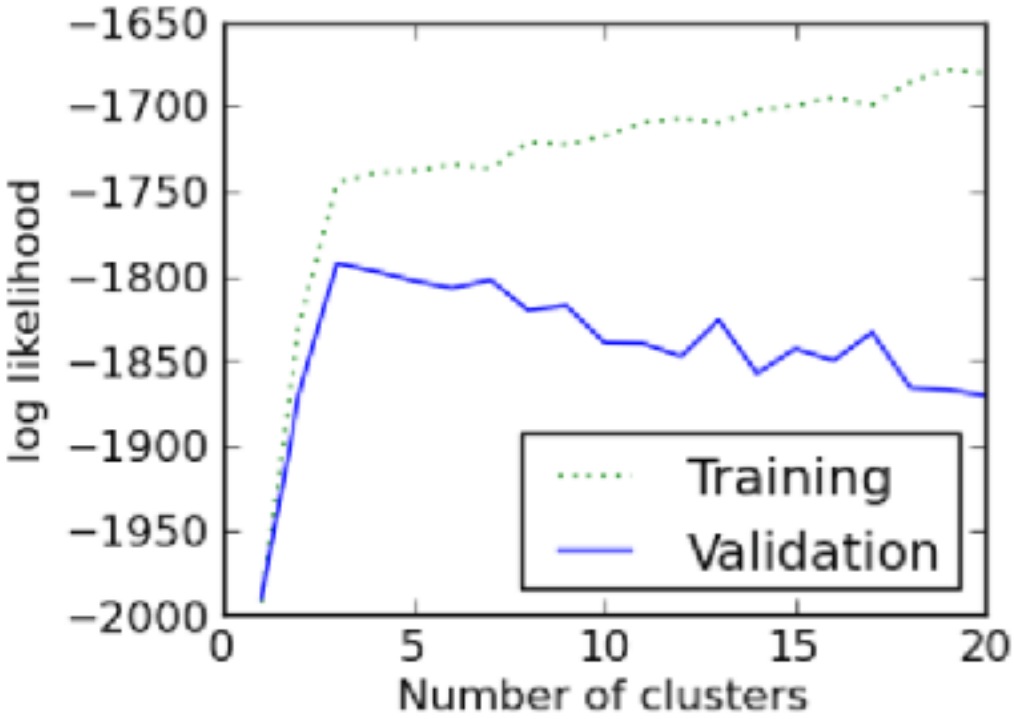


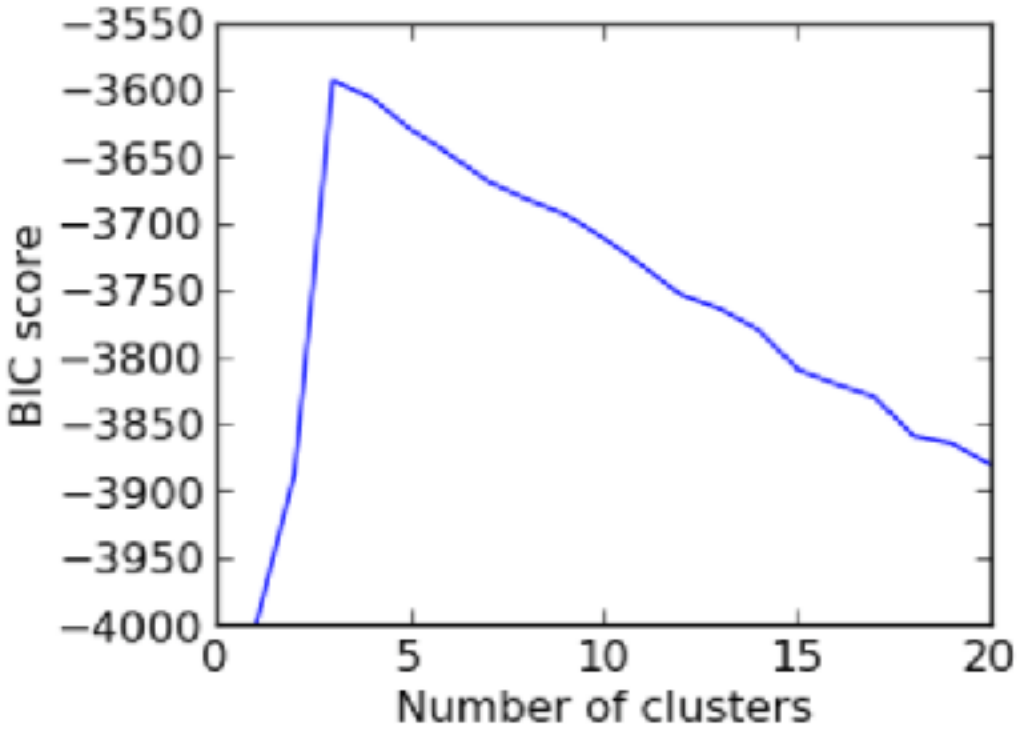


DM, clustering, GMM, selecting K

- How to select K?
  - If we choose K to maximize likelihood, when K increases the value of the maximum likelihood cannot decrease. Thus more complex models will always improve likelihood.
  - It is necessary to penalize the complexity of the model. We need a balance between between how well the model fits and the data and the simplicity of the model
- Score(θ,M) = error(M) + penalty(M)
  Penalty may depend on the number of parameters in the model (p) and the number of data points (n).
  Error is generally based on likelihood of the data given the model (L).
- AIC Akaike information criterion $\text{Score}_{AIC} = -2 \log L + 2p$
- BIC: Bayesian information criterion $\text{Score}_{BIC} = -2 \log L + p \log n$

DM, clustering, GMM, selecting K

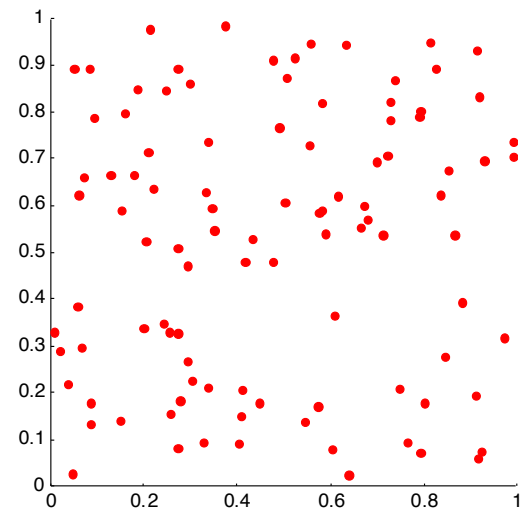




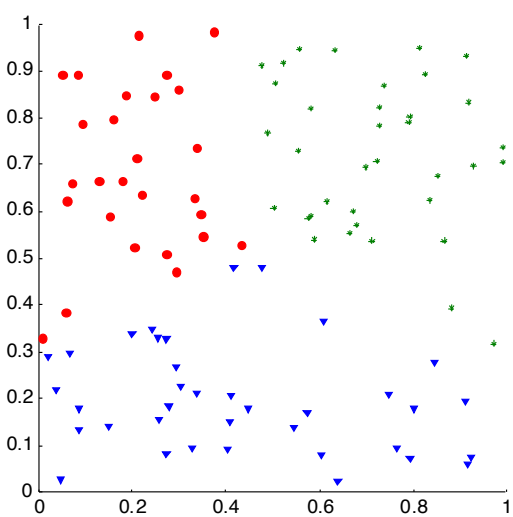
Descriptive modelling
Clustering
Evaluation

Descriptive modeling, clustering, evaluation

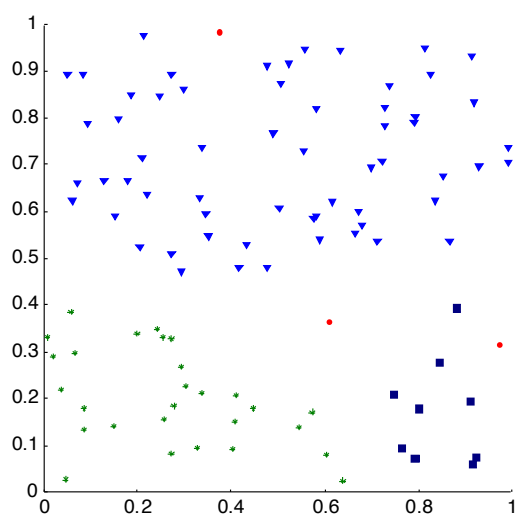
- How to evaluate the “goodness” of the resulting clusters?
- This will help us to:
  - avoid finding patterns in noise
  - compare clustering algorithms
  - compare two sets of clusters
  - compare two clusters



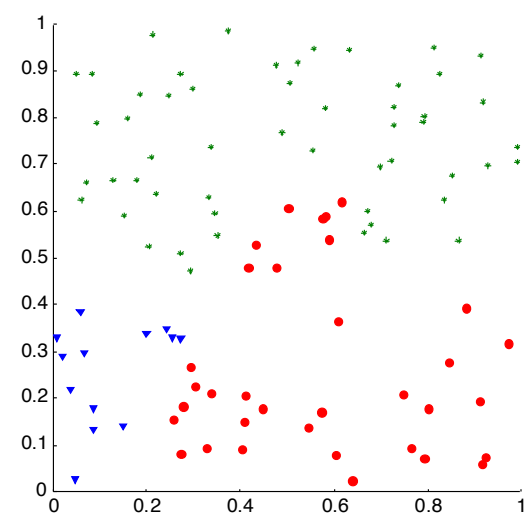
Random Points



K-means



DBSCAN



Complete Link

Descriptive modeling, clustering, evaluation

- Cluster evaluation approaches:
  - Determine the **clustering tendency** of the data, i.e., distinguish whether non-random structure actually exists in the data.
  - Evaluate the clusters using known **class labels (supervised)**.
  - Evaluate how well the **clusters "fit"** the data **(unsupervised)**.
  - Determine which of two different clustering results is better.
  - Determine the "correct" number of clusters.

DM, clustering, evaluation, clustering tendency

- **clustering tendency**: evaluate whether a dataset has clusters before clustering.
- Most common approach (for low-dimensional Euclidean data)
  - Use a statistical test for spatial randomness
  - Hopkins statistic: sample 20 points from dataset, generate 20 random points in same space

$$H = \frac{\sum_{i=1}^{p} w_i}{\sum_{i=1}^{p} u_i + \sum_{i=1}^{p} w_i}$$

w_i: distance from random point to nearest neighbour in data

u_i: distance from sample point to nearest neighbour in data

  - Values near 0.5 indicate random data, 1.0 indicates highly clustered, and 0.0 indicates uniformly distributed.

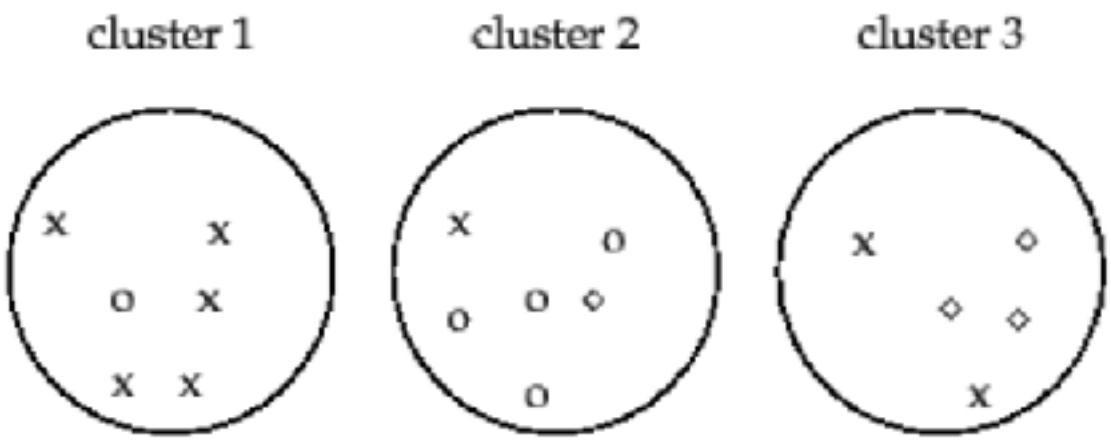
DM, clustering, evaluation, supervised

- **Supervised evaluation:** measures the extent to which clusters match external class label values.
- If you have class labels why cluster?
  - Usually labels come from small hand-labeled dataset for evaluation; but have remaining large dataset to cluster automatically.
  - May want to assess how close clusterings correspond to classes but still allow for more variation in the clusters.
- The classification-oriented evaluation are based on the class labels: **Purity, Entropy, Normalised mutual information gain, Precision, Recall, Accuracy**
- The similarity-oriented evaluation are based on premise that any pair of objects in the same cluster should have the same class and vice versa: **Correlation, Rand, Jaccard.**

DM, clustering, evaluation, supervised, purity

- **Purity:** measure which cluster (G) contain objects of a particular class (C). Higher purity implies better clusters.
  - The purity of each cluster i in G is determined by the number of examples

$$purity(C, G) = \frac{1}{N} \sum_{i=1}^{K} \max_{j} N_j \in G_i$$



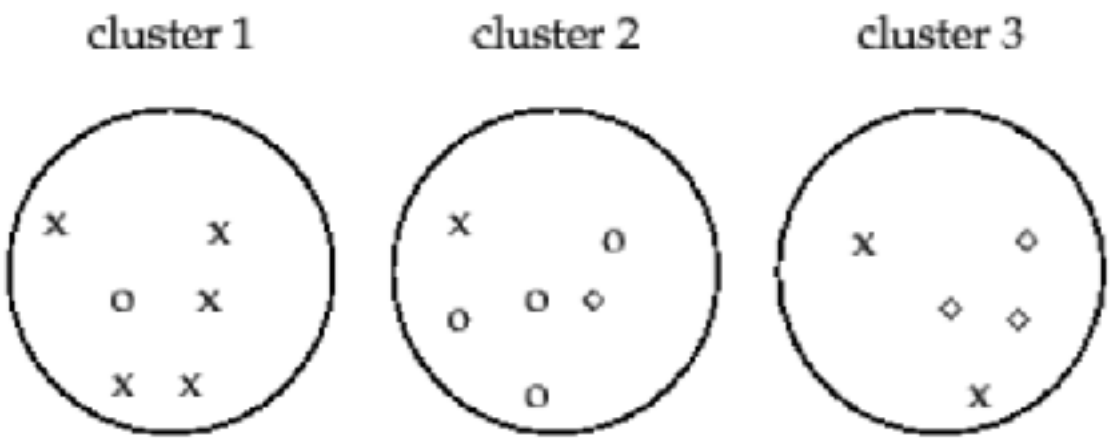
	N_1	N_2	N_3
Cluster 1 G_1	5	1	0
Cluster 2 G_2	1	4	1
Cluster 3 G_3	2	0	3

$$purity(C, G) = \frac{1}{17}(5 + 4 + 3) \approx 0.7$$

DM, clustering, evaluation, supervised, entropy

- **Entropy:** measure the degree to which each cluster (G) consists of objects of a single class (C). Lower entropy implies better clusters.
  - For each cluster i compute the probability of class j (within the cluster)

$$entropy(C,G) = \sum_{i=1}^{K} - \sum_{j=1}^{C} p_{ij} \ln(p_{ij})$$



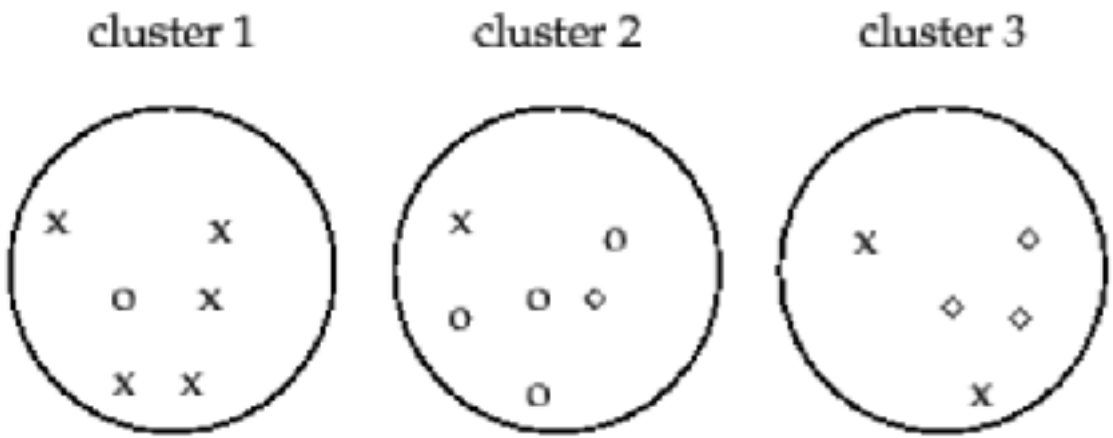
probability	C_1	C_2	C_3
Cluster 1 G_1	5/6	1/6	0
Cluster 2 G_2	1/6	4/6	1/6
Cluster 3 G_3	2/5	0	3/5

$$entropy(C,G) = -\left[\frac{5}{6}\ln\left(\frac{5}{6}\right) + \frac{1}{6}\ln\left(\frac{1}{6}\right)\right] - \left[\frac{1}{6}\ln\left(\frac{1}{6}\right) + \frac{4}{6}\ln\left(\frac{4}{6}\right) + \frac{1}{6}\ln\left(\frac{1}{6}\right)\right] - \left[\frac{2}{5}\ln\left(\frac{2}{5}\right) + \frac{3}{5}\ln\left(\frac{3}{5}\right)\right]$$

DM, clustering, evaluation, supervised, NMI

- **Normalized mutual information gain:** Measures the amount of information by which our knowledge about the classes (C) increases when we are told what the clusters (G) are. Lower values implies better clusters.

$$NMI(C,G) = \frac{I(C,G)}{H(C)+H(G)} = \frac{\sum_c \sum_g p(c,g) \ln \frac{p(c,g)}{p(c)p(g)}}{-\sum_c p(c) \ln p(c) - \sum_g p(g) \ln p(g)}$$



probability	C_1	C_2	C_3
Cluster 1 G_1	5/6	1/6	0
Cluster 2 G_2	1/6	4/6	1/6
Cluster 3 G_3	2/5	0	3/5

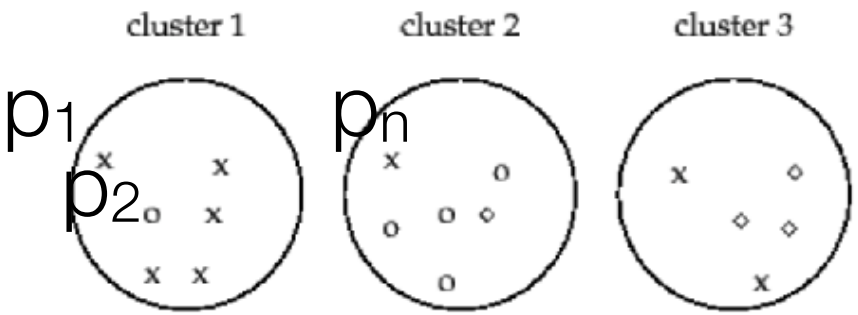
$$H(C) = -\left[\frac{8}{17}\ln\left(\frac{8}{17}\right) + \frac{5}{17}\ln\left(\frac{5}{17}\right) + \frac{4}{17}\ln\left(\frac{4}{17}\right)\right] \approx 1.06$$

$$H(G) = -\left[\frac{6}{17}\ln\left(\frac{6}{17}\right) + \frac{6}{17}\ln\left(\frac{6}{17}\right) + \frac{5}{17}\ln\left(\frac{5}{17}\right)\right] \approx 1.10$$

$$NMI(C,G) = \frac{4.50}{1.06+1.10} \approx 2.08$$

DM, clustering, evaluation, supervised, similarity

- Based on premise that any pair of objects in the same cluster should have the same class and vice versa
- Construct the “ideal” similarity matrix based on cluster membership
  - Entry i,j is 1 if i and j are in the same cluster, 0 otherwise
- Construct the “ideal” similarity matrix based on class values
  - Entry i,j is 1 if i and j are in the same class, 0 otherwise
- Use measure that compares the two ideal similarity matrices



cluster	p_1	p_2	…	p_n
p_1	1	1		0
p_2	1	1		0
…				
p_n	0	0		1

class	p_1	p_2	…	p_n
p_1	1	0		1
p_2	0	1		0
…				
p_n	1	0		1

DM, clustering, evaluation, supervised, similarity

- Measures of binary similarity between two ideal matrices
  - f_{00} = # pairs of objects having diff class and diff cluster
  - f_{01} = # pairs of objects having diff class and same cluster
  - f_{10} = # pairs of objects having same class and diff cluster
  - f_{11} = # pairs of objects having same class and same cluster

$$rand = \frac{f_{00} + f_{11}}{f_{00} + f_{01} + f_{10} + f_{11}} \qquad Jaccard = \frac{f_{11}}{f_{01} + f_{10} + f_{11}}$$

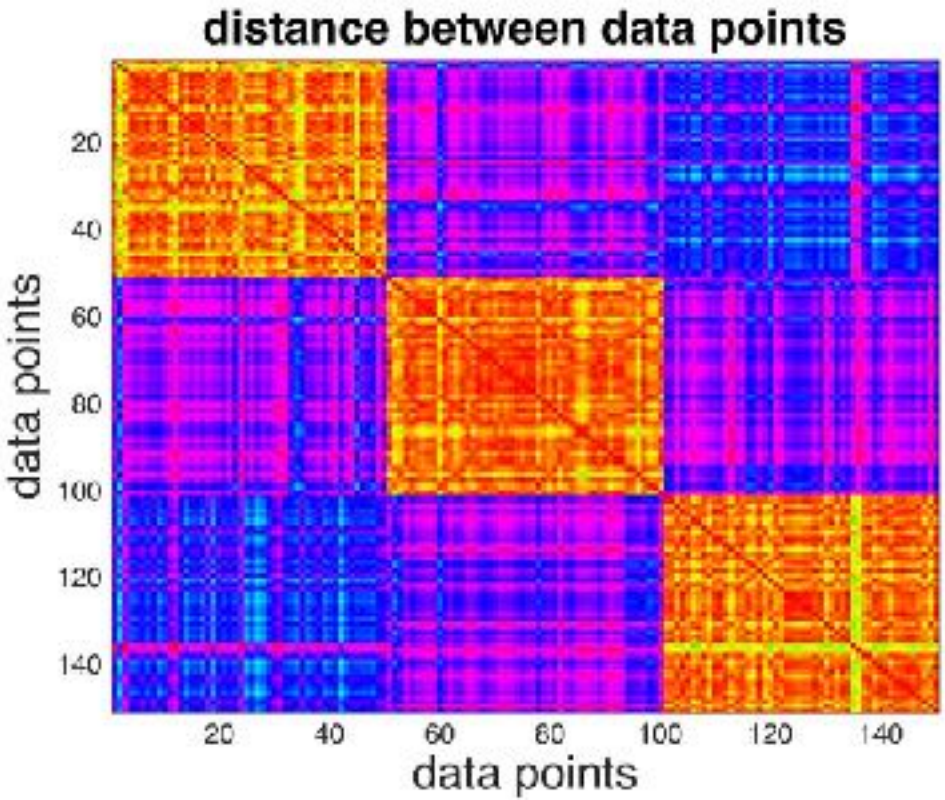
$$\left.\begin{aligned} f_{00} &= 144 \\ f_{01} &= 40 \\ f_{10} &= 48 \\ f_{11} &= 57 \end{aligned}\right\} \quad \begin{aligned} rand &\approx 0.70 \\ Jaccard &\approx 0.39 \end{aligned}$$

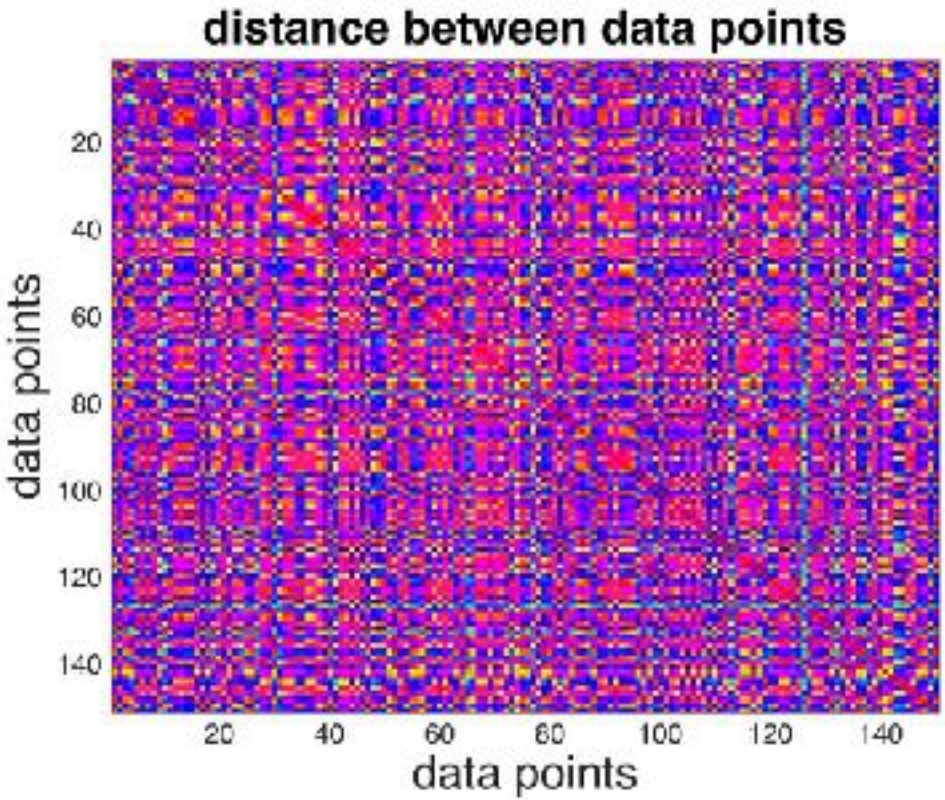
DM, clustering, evaluation, unsupervised

- **Unsupervised evaluation:** measures goodness of fit without class labels.
- There are different methods to evaluate unsupervised clustering:
  - Visual inspection based on the proximity matrix
  - Correlation between similarity and clustering results
  - Internal measures: **Cohesion**, **Separation**, and **Silhouette coefficient**.

DM, clustering, evaluation, unsupervised, visual

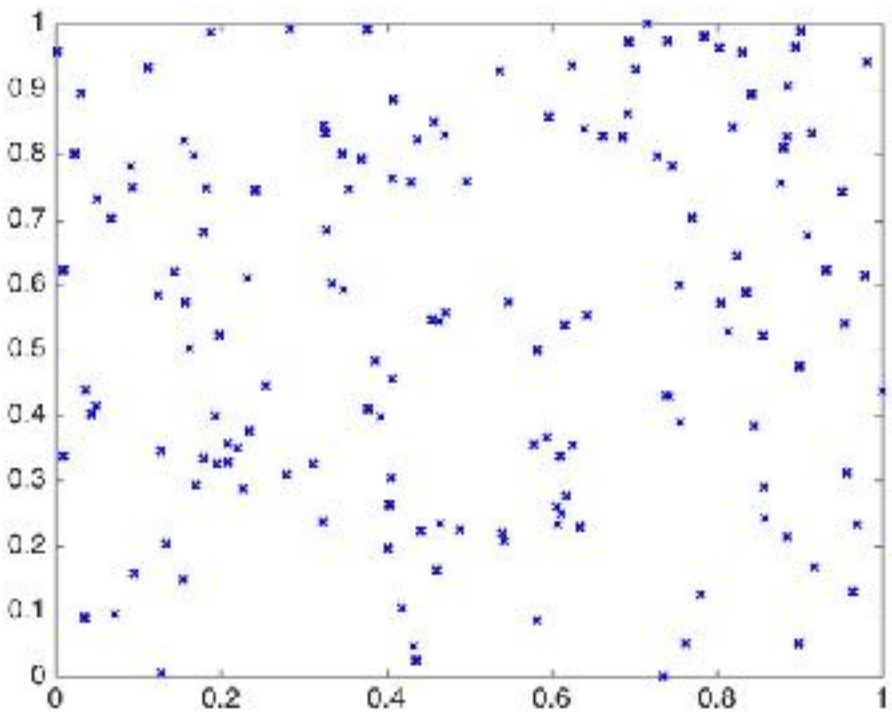
- Calculate, the proximity matrix between points.
- Order the proximity matrix based on cluster labels.
- Create the distance matrix
- Visually inspect (good clusterings exhibit clear block pattern)

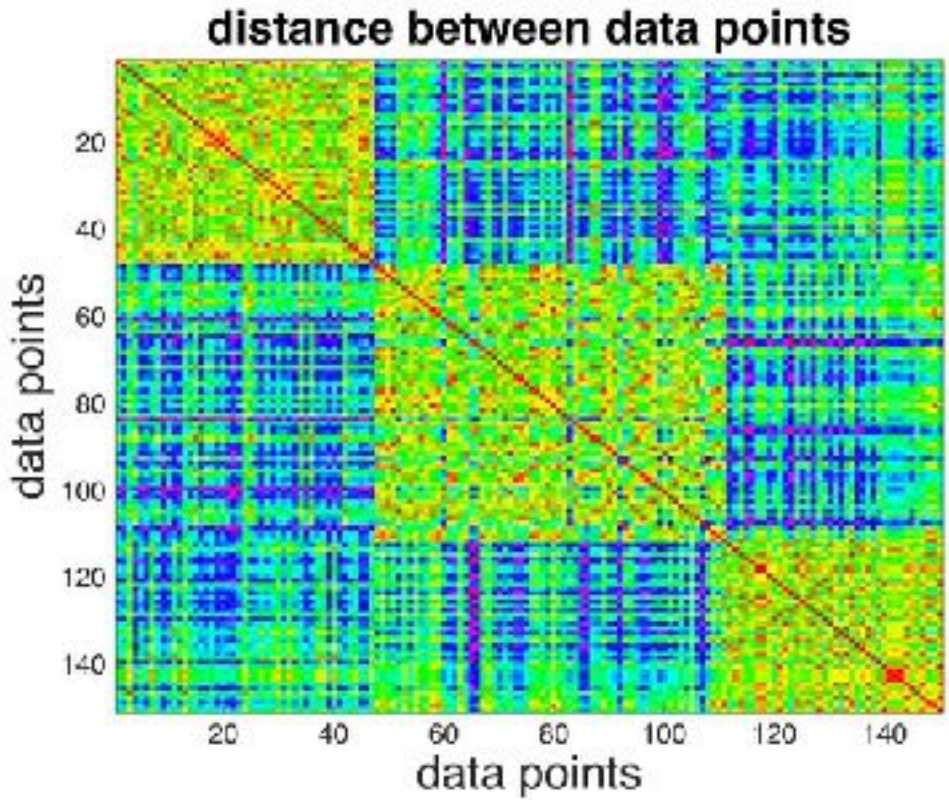




DM, clustering, evaluation, unsupervised, visual

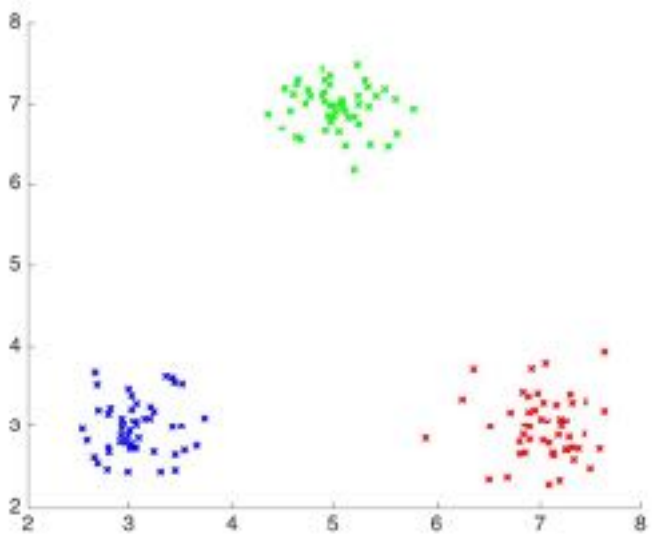
- Calculate, the proximity matrix between points.
- Order the proximity matrix based on cluster labels.
- Create the distance matrix
- Visually inspect (good clusterings exhibit clear block pattern)



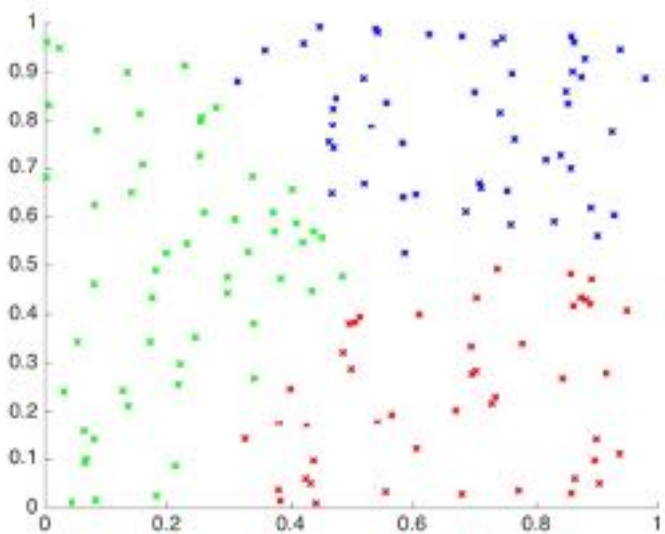


DM, clustering, evaluation, unsupervised, correlation

- Construct the **initial similarity** matrix among all points s(i,j)=1/(1+d(i,j))
- Construct the **“ideal” similarity** matrix based on cluster membership: entry i,j is 1 if i and j are in the same cluster, 0 otherwise.
- Compute the correlation between the initial similarity matrix and the “ideal” similarity matrix (X and Y axis are the initial/ideal similarity respectively).
- High correlation indicates that points in same cluster are close to each other



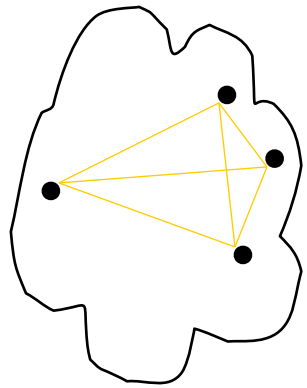
corr=0.95



corr=0.59

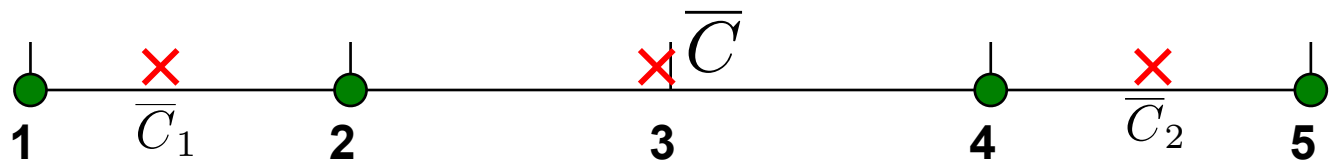
DM, clustering, evaluation, unsupervised, Internal measures

- **Cohesion**: Measures how closely related the objects are within each cluster.
- **Sum of squared errors (SSE)** is the sum of the squared distance of a point to its cluster centroid.



cohesion

$$SSE_{total} = \sum_{k=1}^{K} \sum_{\mathbf{x} \in C_k} (\mathbf{x} - \overline{C}_i)^2$$



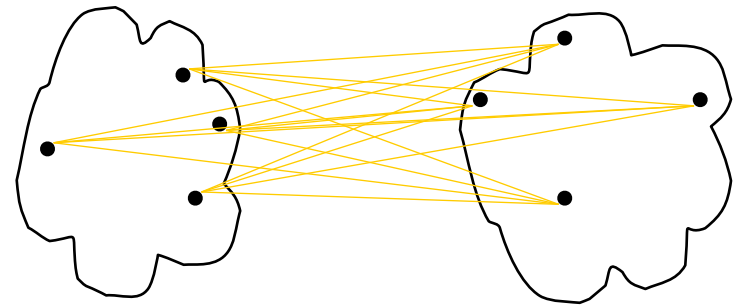
$$\text{K=1} \Rightarrow SSE_{total} = (1-3)^2 + (2-3)^2 + (4-3)^2 + (5-3)^2 = 10$$

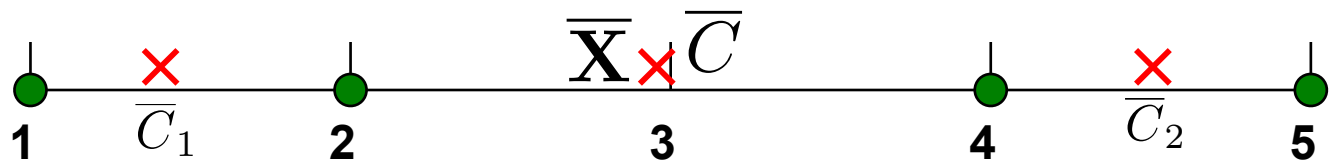
$$\text{K=2} \Rightarrow SSE_{total} = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$$

DM, clustering, evaluation, unsupervised, Internal measures

- **Separation**: Measures how distinct a cluster is from the other clusters.
- **Between group sum of squares (SSB)** is the sum of the squared distance of a cluster centroid, to the overall mean (mean of all the data points).

$$SSB_{total} = \sum_{k=1}^{K} |C_i|(\overline{C}_i - \overline{\mathbf{X}})^2$$



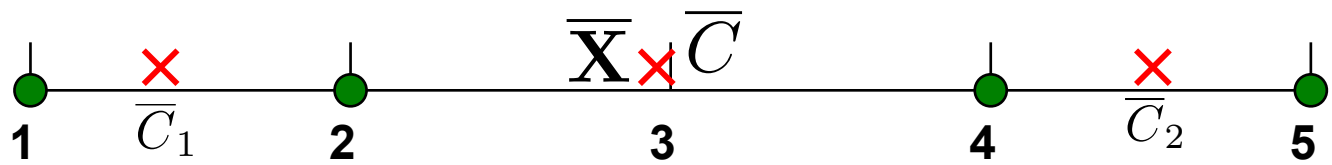


$$\text{K=1} \Rightarrow SSB_{total} = 4 * (3-3)^2 = 0$$

$$\text{K=2} \Rightarrow SSB_{total} = 2 * (1.5-3)^2 + 2 * (4.5-3)^2 = 9$$

DM, clustering, evaluation, unsupervised, Internal measures

- **Separation and cohesion**: The sum of SSE_{total} and SSB_{total} is equal to the total sum of squared error (distance of each point to overall mean).
- Thus minimizing cohesion is equivalent to maximizing separation.



$$\text{K=1} \Rightarrow SSE_{total} = (1-3)^2 + (2-3)^2 + (4-3)^2 + (5-3)^2 = 10$$

$$\text{K=1} \Rightarrow SSB_{total} = 4 * (3-3)^2 = 0$$

$$\text{K=2} \Rightarrow SSE_{total} = (1-1.5)^2 + (2-1.5)^2 + (4-4.5)^2 + (5-4.5)^2 = 1$$

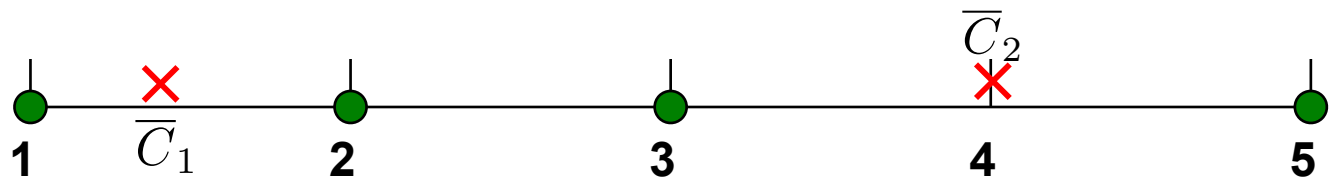
$$\text{K=2} \Rightarrow SSB_{total} = 2 * (1.5-3)^2 + 2 * (4.5-3)^2 = 9$$

DM, clustering, evaluation, unsupervised, Internal measures

- The **Silhouette coefficient** combines both cohesion and separation. Typically it varies between -1 and 1, where values closer to 1 implies better clustering.
- For an individual point i:
  - Calculate a_i, the average distance of i to points in same cluster
  - Calculate b_{ij} the average distance of point i to all points in cluster j.
  - Calculate b_i, the minimum b_{ij} such as point i does not belong to cluster j.
  - The Silhouette coefficient for point i is $S_i = (b_i - a_i) / \max(a_i, b_i)$
- A negative value implies that point i, is closer to another cluster instead of its cluster. If a_i close to 0 (low cohesion), then S_i is close 1.
- The silhouette coefficient of a cluster is the average of the silhouette coefficients of points belonging to the cluster.
- An overall silhouette coefficient is the average silhouette coefficient of all points.

DM, clustering, evaluation, unsupervised, Internal measures

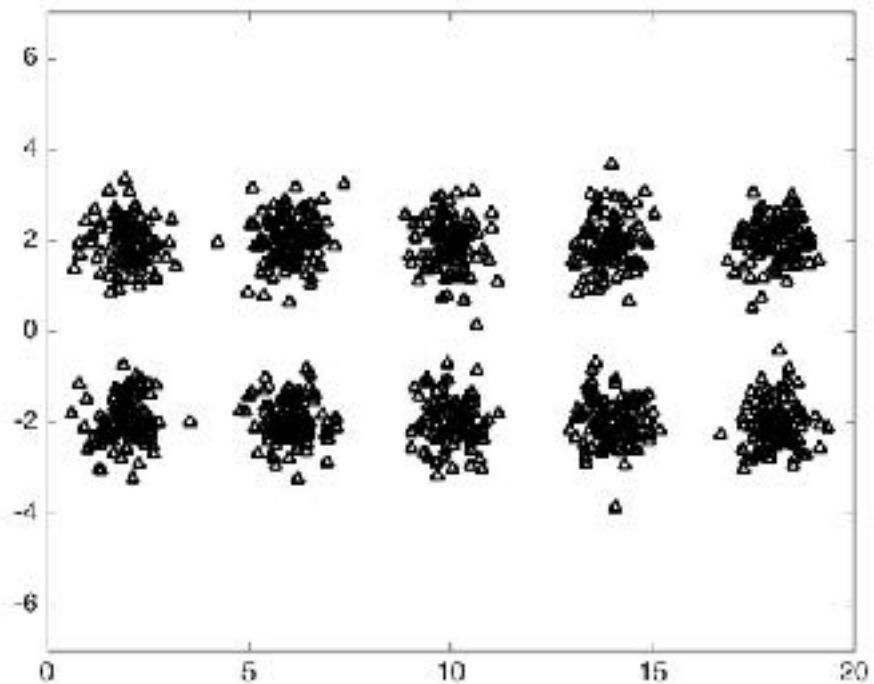
- a_i, the average distance of i to points in same cluster
  Calculate b_{ij} the average distance of point i to all points in cluster j.
  Calculate b_i, the minimum b_{ij} such as point i does not belong to cluster j.
  The Silhouette coefficient for point i is $S_i = (b_i - a_i) / \max(a_i, b_i)$

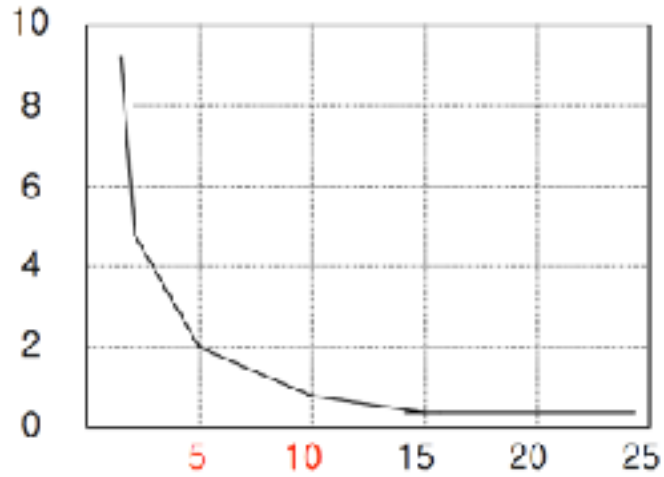


	a_i	b_{i1}	b_{i2}	b_i	s_i
1	1,0	—	3,0	3,0	2/3
2	1,0	—	2,0	2,0	1/2
3	2,0	1,5	—	1,5	-0.5/2
5	2,0	3,5	—	3,5	1.5/2

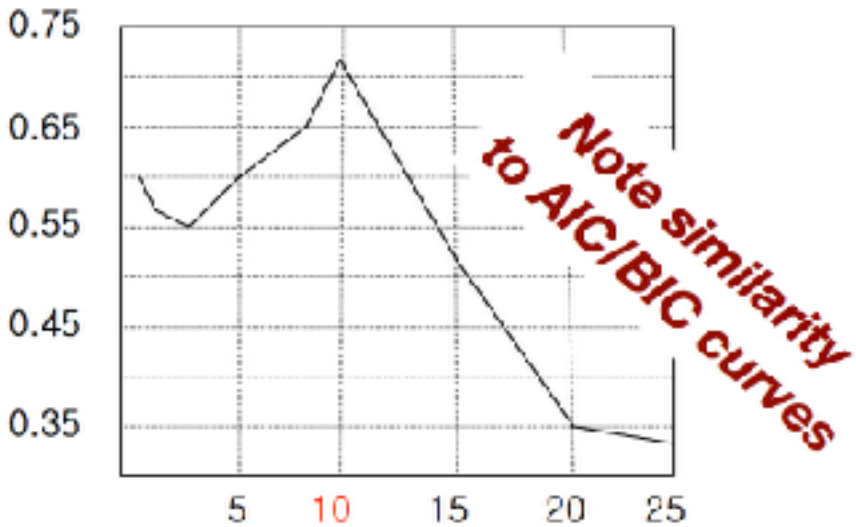
DM, clustering, evaluation, determining K

- To determine the best value of K, evaluate a specific measure over a range of K (SSE_{total}, AIC, BIC), look for peak, dip, or knee in evaluation measure.





SSE



Silhouette

DM, clustering, evaluation, Assessing significance

- How do we know a score is “good”?
- How do we know that a difference between two algorithms is significant?
- This is the same problem we had for predictive models
- Need a sampling distribution to compare to
  - Can generate random data in same space and compute empirical sampling distribution.
  - Can partition data to get multiple folds for evaluation.

Descriptive modeling, clustering

"The validation of clustering structures is the most difficult and frustrating part of cluster analysis. Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage."

Algorithms for Clustering Data, Jain and Dubes