AI6102: Machine Learning Methodologies & Applications

L10: Clustering

Sinno Jialin Pan

Nanyang Technological University, Singapore

Homepage: http://www.ntu.edu.sg/home/sinnopan

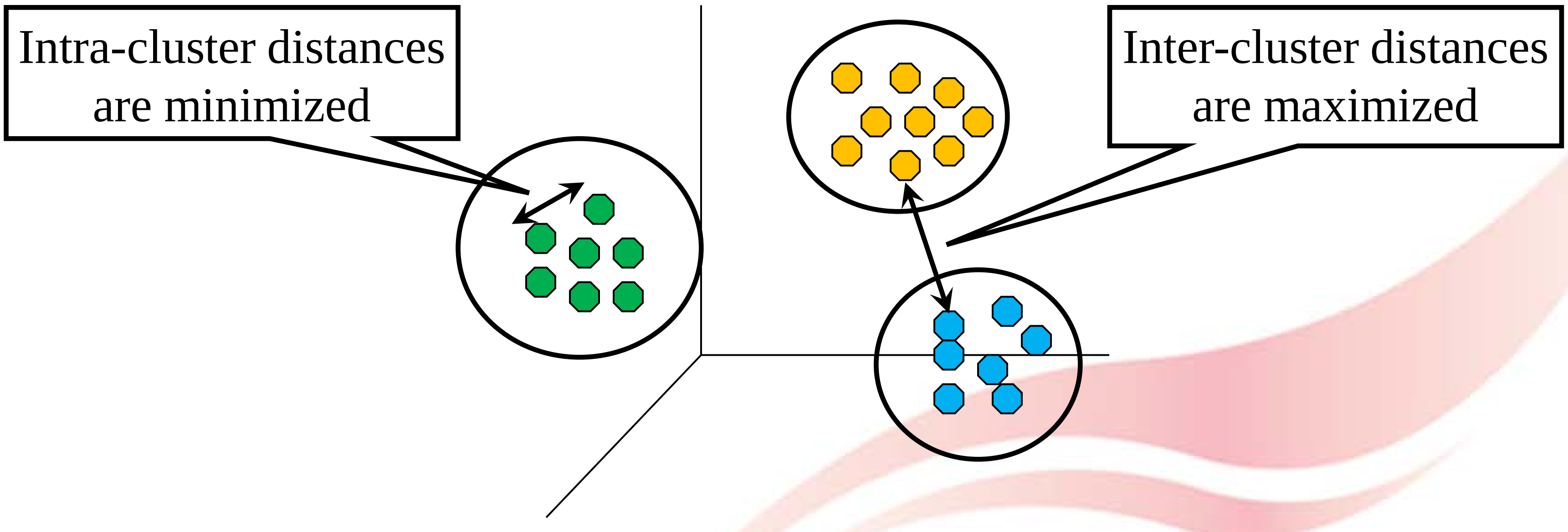
Clustering Defintion

In mathematics

- Given: a set of $\{\boldsymbol{x}_i\}$ for $i = 1, \ldots, N$, where $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \ldots, x_{im}]$ is m-dimensional vector of numerical values
- Goal: to learn a model to automatically assign each input data instance to a group
  - $g: \boldsymbol{x}_i \rightarrow z_i$, where z_i is the index of a group

Clustering Illustrative Example

- Finding groups of data points such that the data instances in a group are
  - similar to one another
  - different from the data instances in other groups



Clustering: User Segmentation

Common hyper-parameter of most clustering algorithms

Suppose we want to cluster potential customers into $\boxed{3}$ groups, and advertise a different loaning plan to different groups

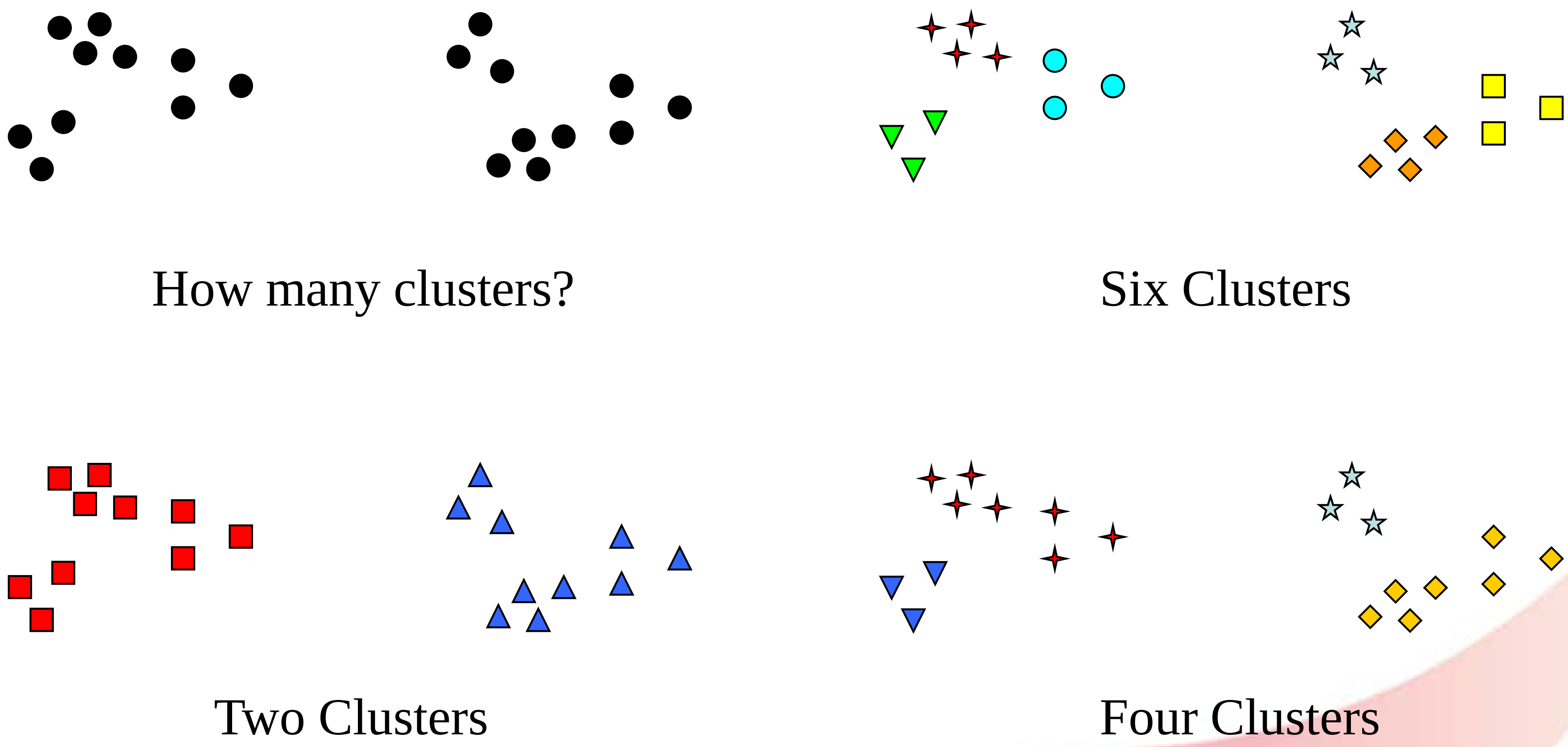
ID	Gender	Profession	Income	Saving
1	F	Engineer	60k	200k
2	M	Student	10k	20k
…	…	…	…	…
10	M	Student	8k	5k

	X_1	X_2	…	X_{m-1}	X_m
$\boldsymbol{x}_1$	1	0	…	60	200
$\boldsymbol{x}_2$	0	1	…	10	20
	…	…	…	…	…
$\boldsymbol{x}_{10}$	0	1	…	8	5

$$g: \boldsymbol{x} \rightarrow z$$

	X_1	X_2	…	X_{m-1}	X_m	Z
$\boldsymbol{x}_1$	1	0	…	60	200	1
$\boldsymbol{x}_2$	0	1	…	10	20	3
	…	…	…	…	…	
$\boldsymbol{x}_{10}$	0	1	…	8	5	1

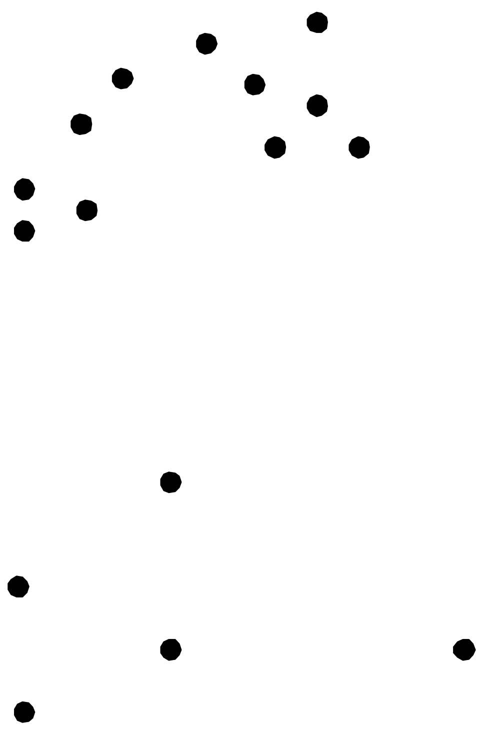
Ambiguity of Clusters



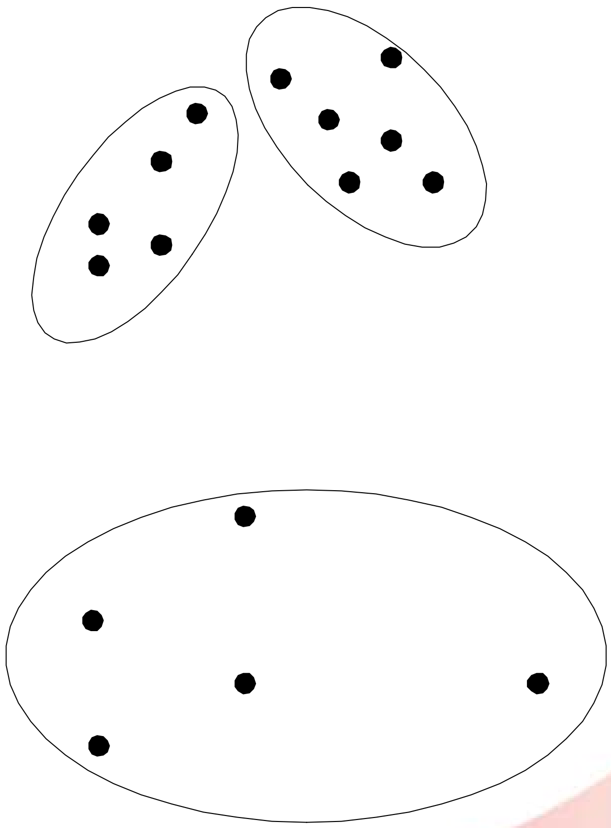
Types of Clustering

- A clustering is a set of clusters
- Partitional Clustering
  - Divide data instances into **non-overlapping** clusters such that each data instance is in exactly one cluster
- Hierarchical clustering
  - A set of **nested** clusters organized as a hierarchical tree

Partitional Clustering

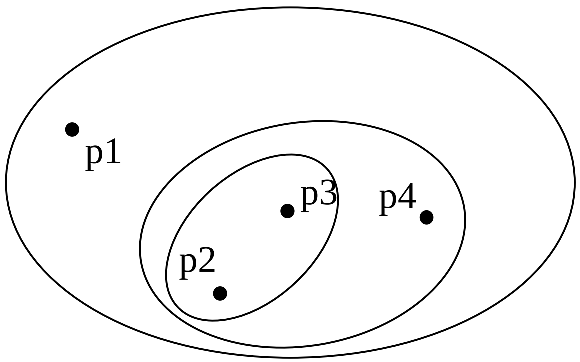


Original data instances

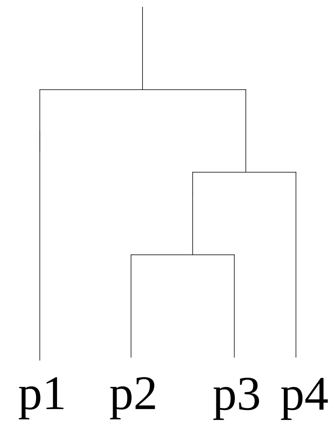


A Partitional Clustering

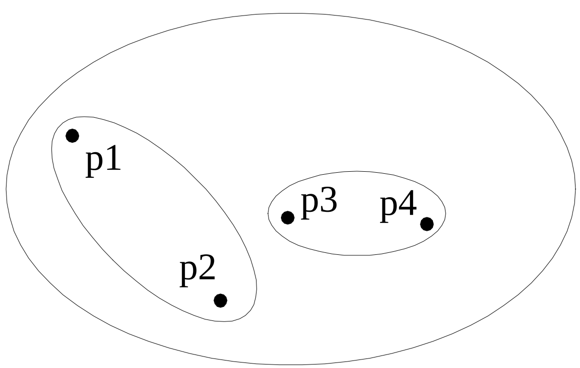
Hierarchical Clustering



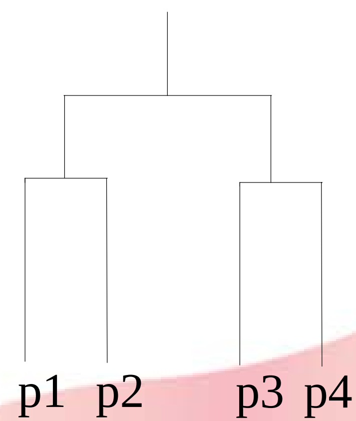
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

Other Distinctions

- Probabilistic versus non-probabilistic
  - In probabilistic clustering, a data instance belongs to every cluster with a probability
  - The sum of the probabilities equals 1
- Partial versus complete
  - In partial clustering, only some of the data instances are clustered

Cluster Algorithms

- K-means and its variants
- Hierarchical clustering

K-means Clustering

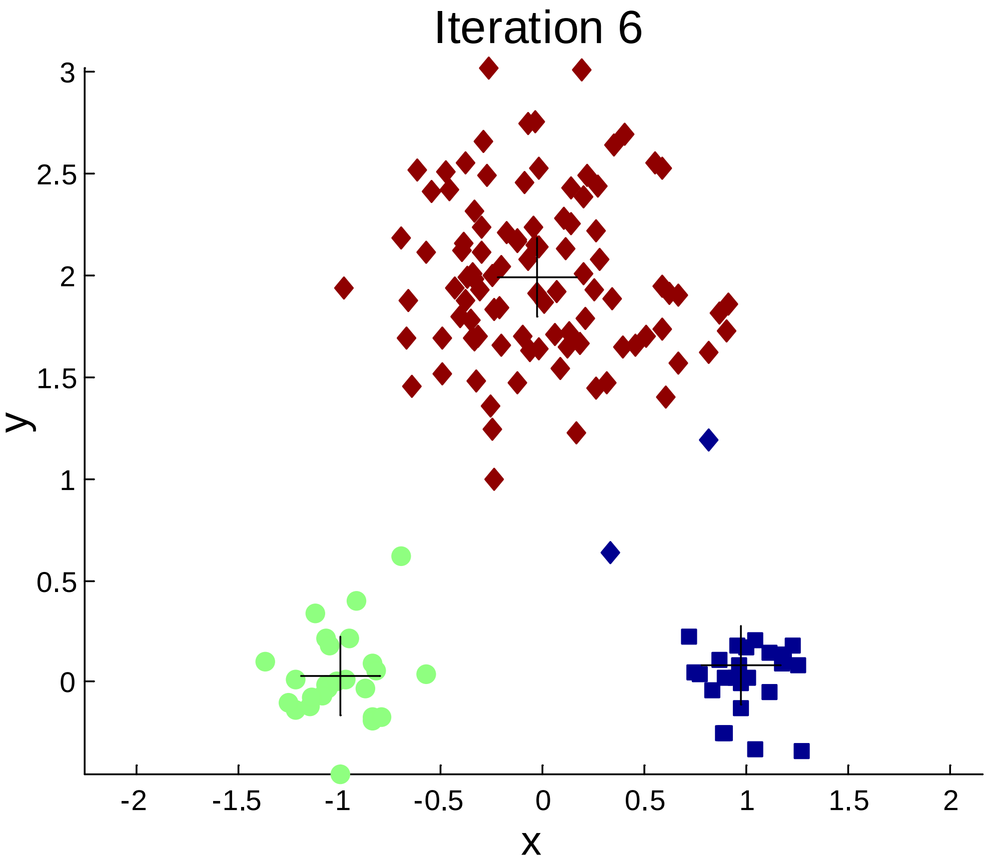
- Partitional clustering approach
- Each cluster is associated with a centroid (center point)
- Each data instance is assigned to the cluster with the closest centroid
- Number of clusters, K, must be specified
- Basic algorithm:

1. Select K data instances as the initial centroids
2. **Repeat**
3. Form K clusters by assigning all data instances to the corresponding closest centroid
4. Recompute the centroid of each cluster
5. **Until** The centroids do not change

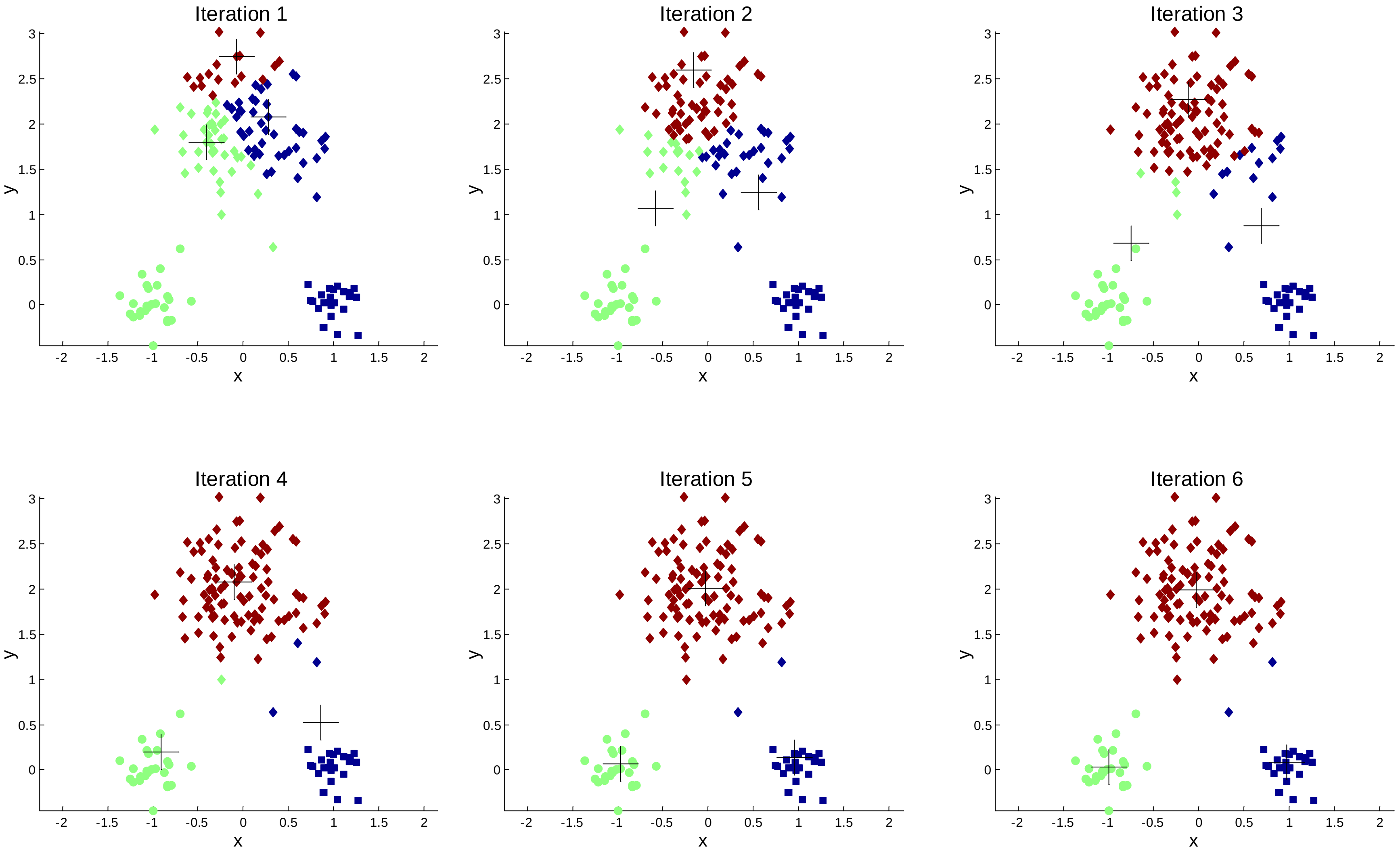
K-means Clustering (cont.)

- Initial centroids are often chosen randomly
- The centroid is (typically) the mean of the data instances in the cluster
- 'Closeness' is measured by a proximity
  - E.g., Euclidean distance
- K-means will converge for common distance measures like Euclidean distance
  - In practice, it converges in the first few iterations
  - Often the stopping condition is changed to 'Until relatively few points change clusters'

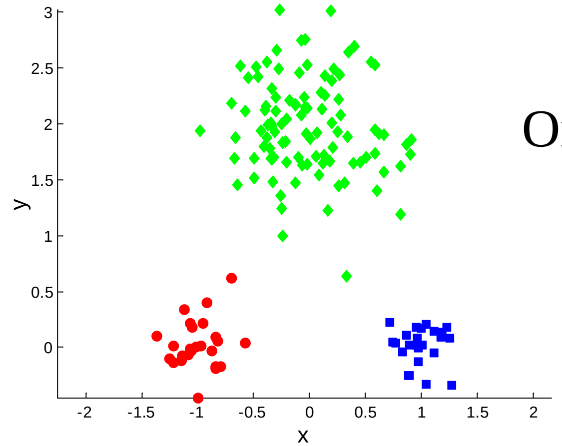
Illustrative Example



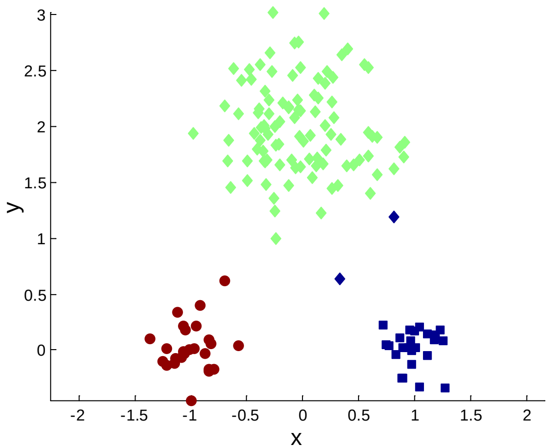
Illustrative Example (cont.)



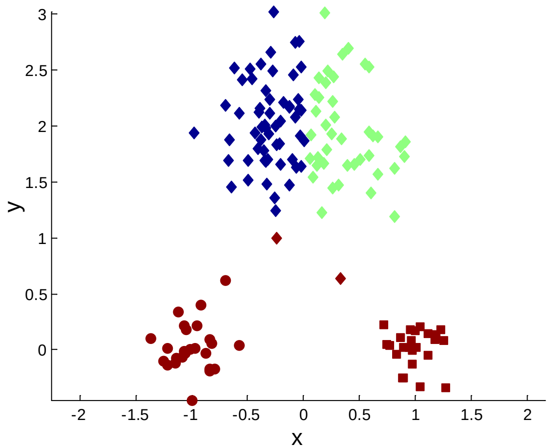
Compare K-Means Clusterings



Original Points



Optimal Clustering



Sub-optimal Clustering

Evaluation

- A most common measure is Sum of Squared Error (SSE)
  - For each data point, the "error" is the distance to the nearest cluster that is represented by a centroid
  - To get SSE, we square these errors and sum them

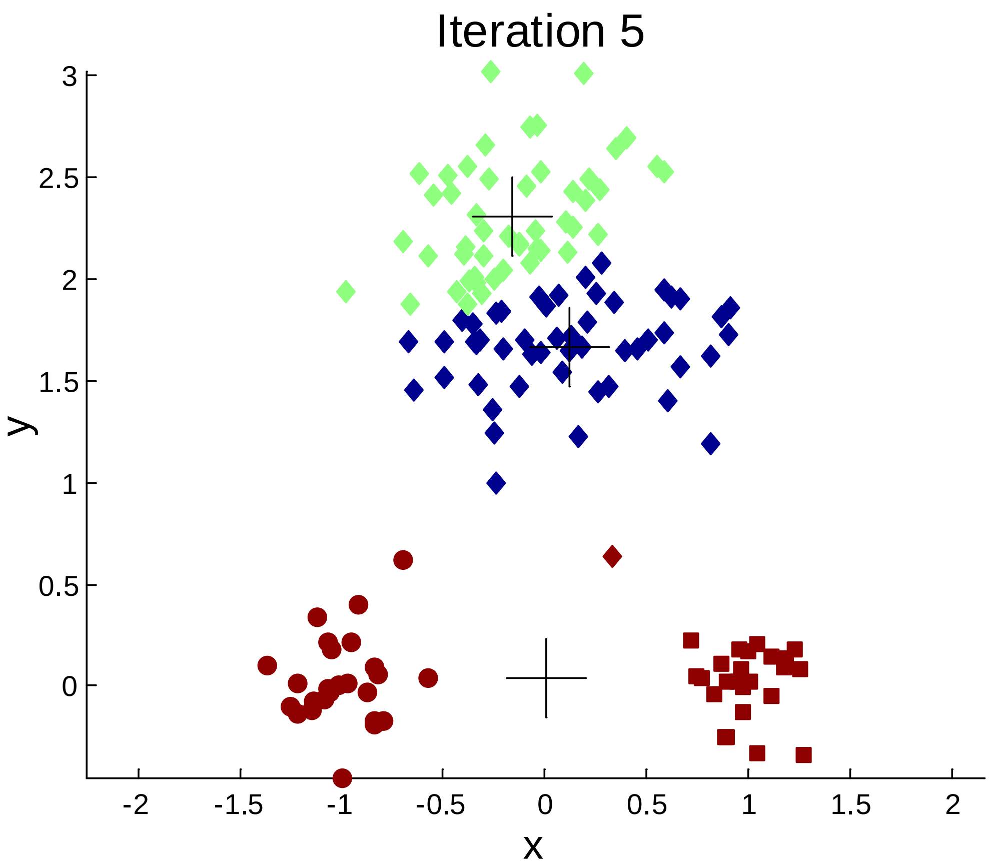
Total SSE $\quad \text{SSE} = \sum_{i=1}^{K} \boxed{\sum_{x \in C_i} \text{dist}(\boxed{\boldsymbol{c}_i}, \boldsymbol{x})^2}$

Centroid of the cluster C_i

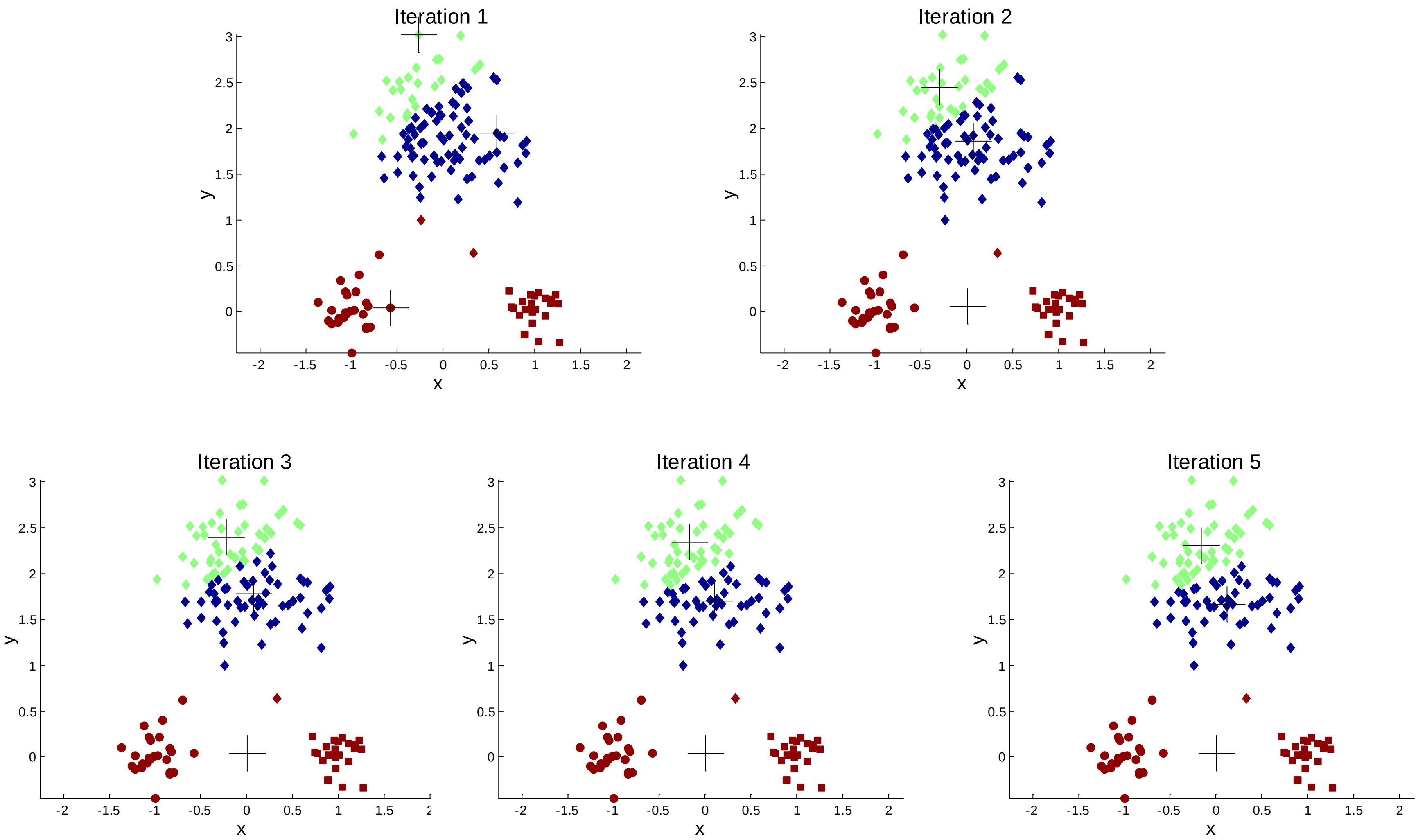
$$\boxed{\text{Cluster SSE for } C_i = \sum_{\boldsymbol{x} \in C_i} \text{dist}(\boldsymbol{c}_i, \boldsymbol{x})^2}$$

  - Given two different runs of K-means, we choose the one with the smallest Total SSE

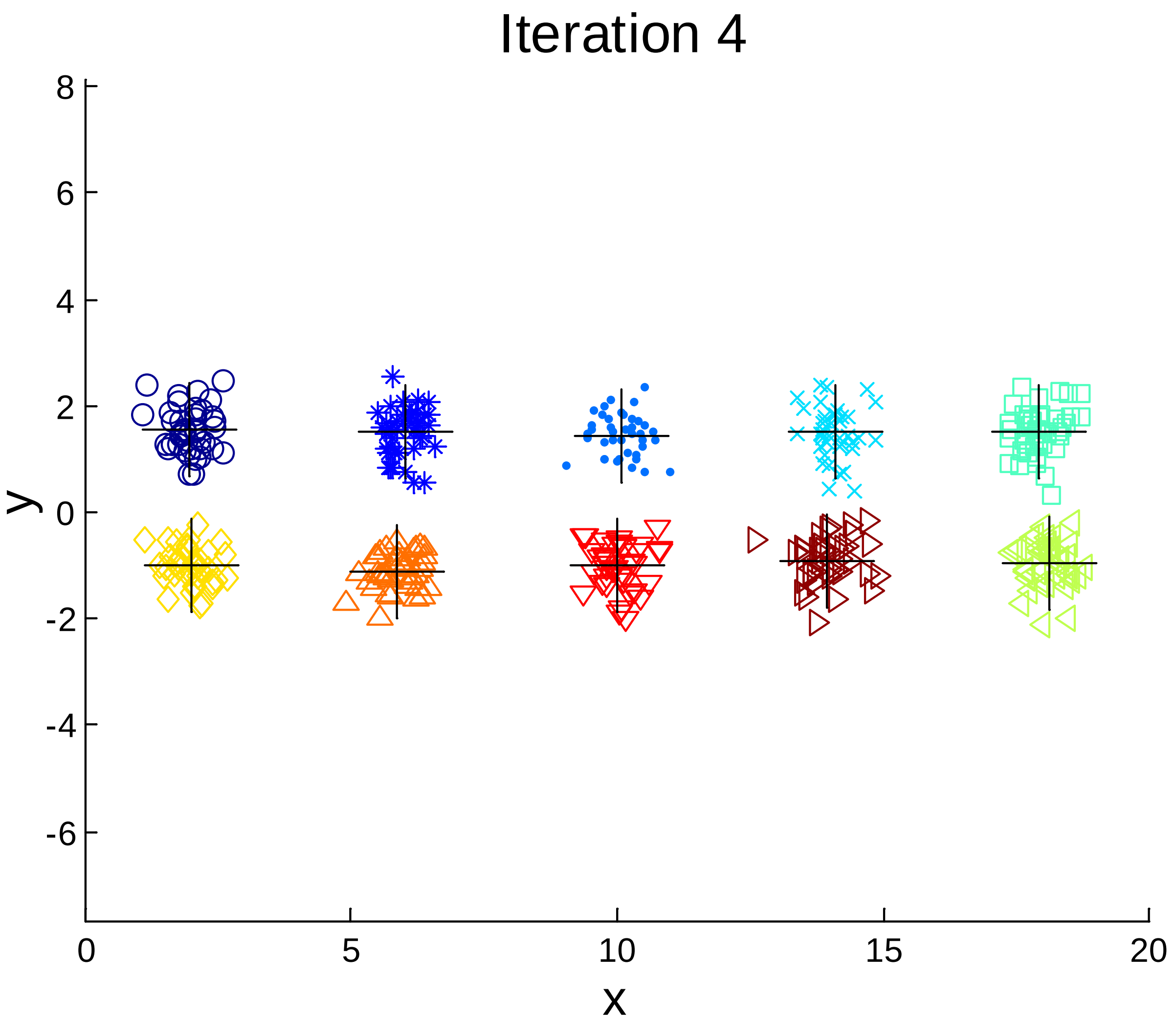
Importance of Initial Centroids



Importance of Initial Centroids (cont.)

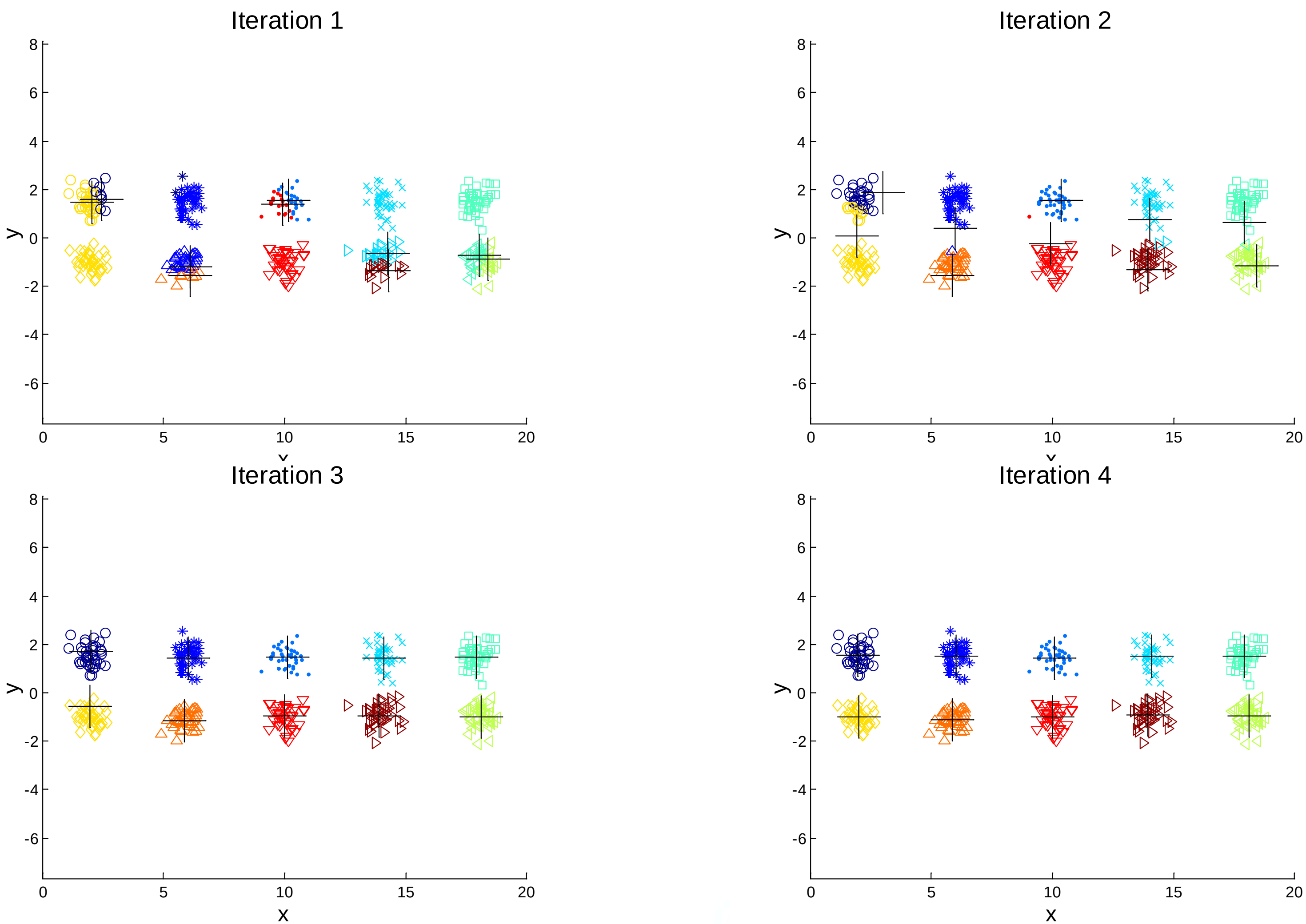


Another Example



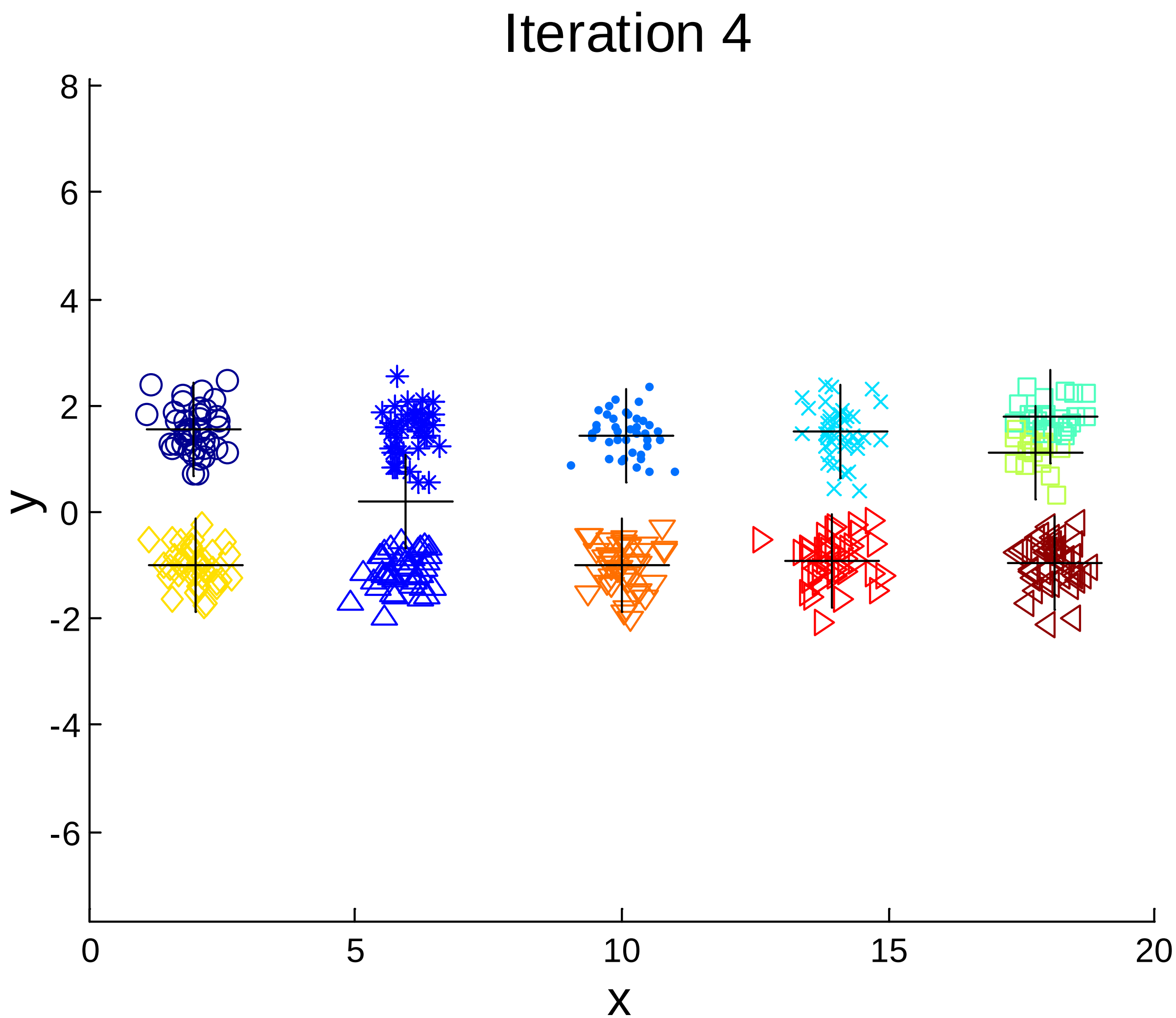
Starting with two initial centroids in one cluster of each pair of clusters

Another Example (cont.)



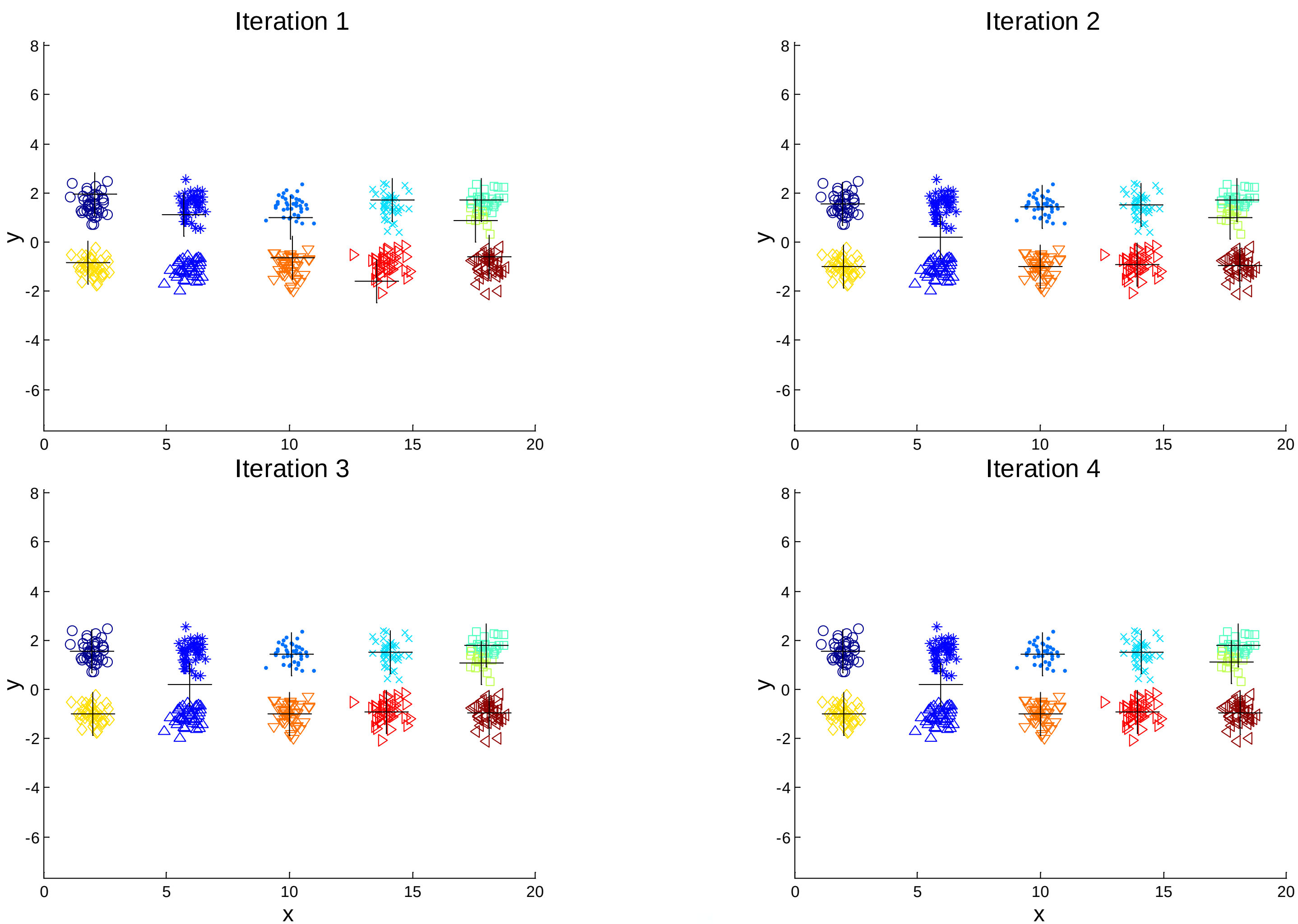
Starting with two initial centroids in one cluster of each pair of clusters

Another Example (cont.)



Starting with some pairs of clusters having three initial centroids, while some have only one

Another Example (cont.)



Starting with some pairs of clusters having three initial centroids, while some have only one

Potential Solutions

- Multiple runs
  - Choose the best one based on SSE
- Bisecting K-means
  - A variant of K-means

Bisecting K-Means

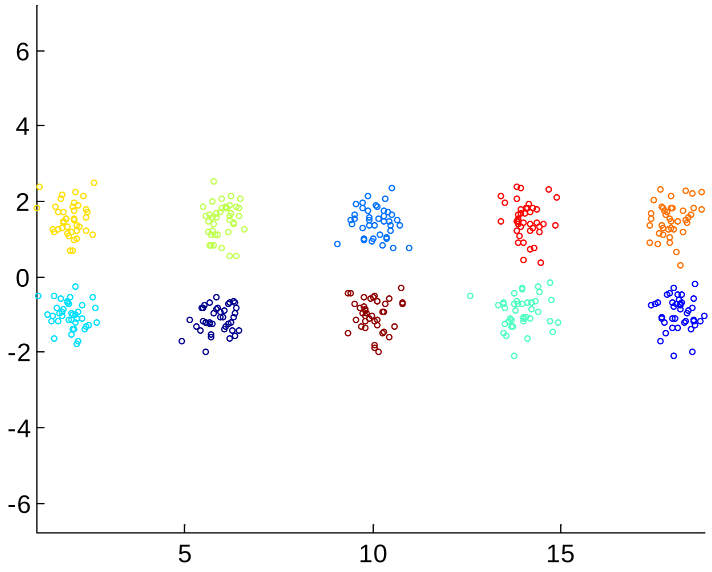
- Basic algorithm:
  1. Initialize the list of one cluster that contains all points
  2. **Repeat**
  3. Select a cluster from the list of clusters
  4. **For** $i = 1$ to T **do**
  5. Bisect the selected cluster using basic K-means
  6. **End**
  7. Add the two clusters from the bisection with lowest SSE over the T runs to the list of clusters
  8. **Until** the list of clusters contain K clusters

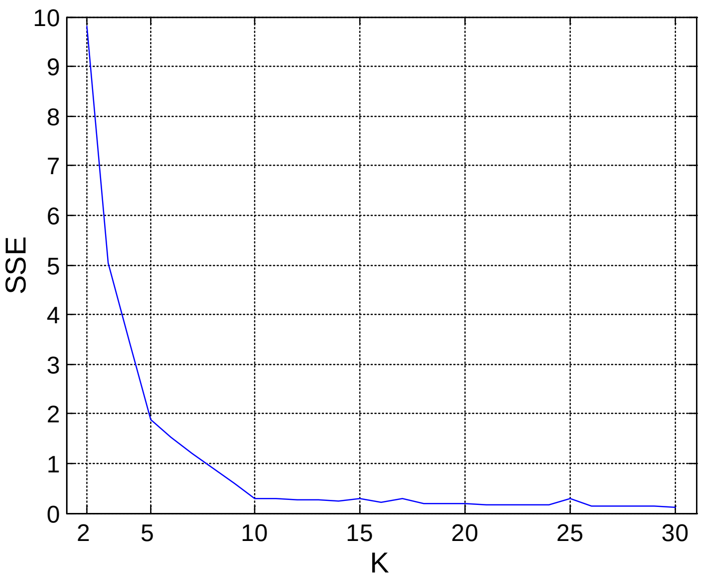
Empty Clusters Issue

- Basic *K*-means algorithm can yield empty clusters
- Several strategies to choose a replacement centroid
  - Choose the data instance that contributes most to SSE
    $$\arg\max_{\boldsymbol{x}} \operatorname{dist}(\boldsymbol{c}, \boldsymbol{x}),$$
    where $\boldsymbol{c}$ is the corresponding centroid of the cluster to which $\boldsymbol{x}$ is assigned
  - Randomly choose a data instance from the cluster with the highest Cluster SSE
  - If there are several empty clusters, the above can be repeated several times

Estimation of K

- SSE can be used to estimate the number of clusters

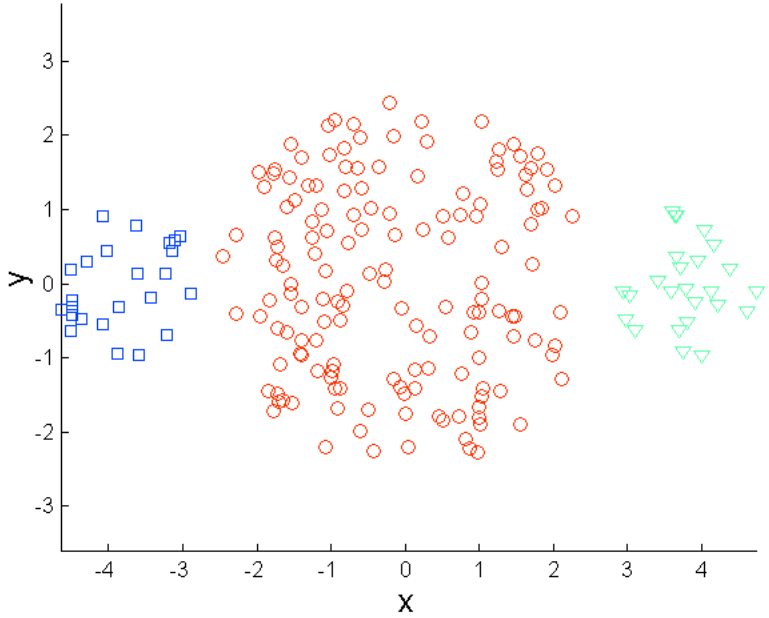




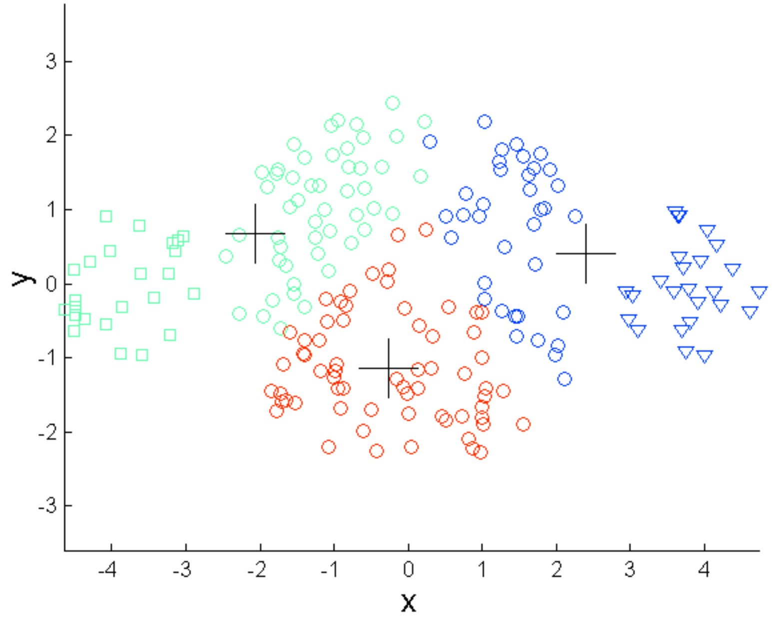
Limitations of K-means

- K-means has problems when clusters are of
  - Different sizes
  - Different densities
  - Non-globular shapes
- K-means also has problems when dataset contains outliers

Clusters Are of Different Sizes

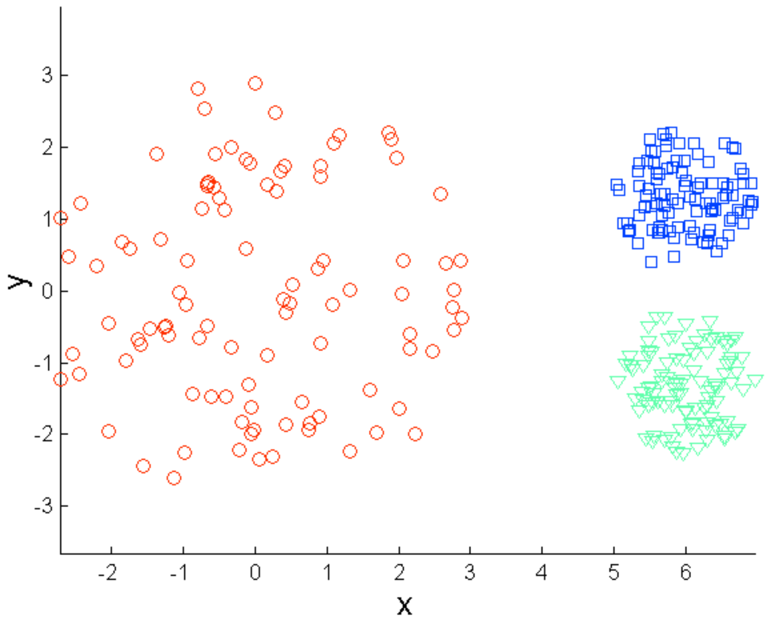


Original Points

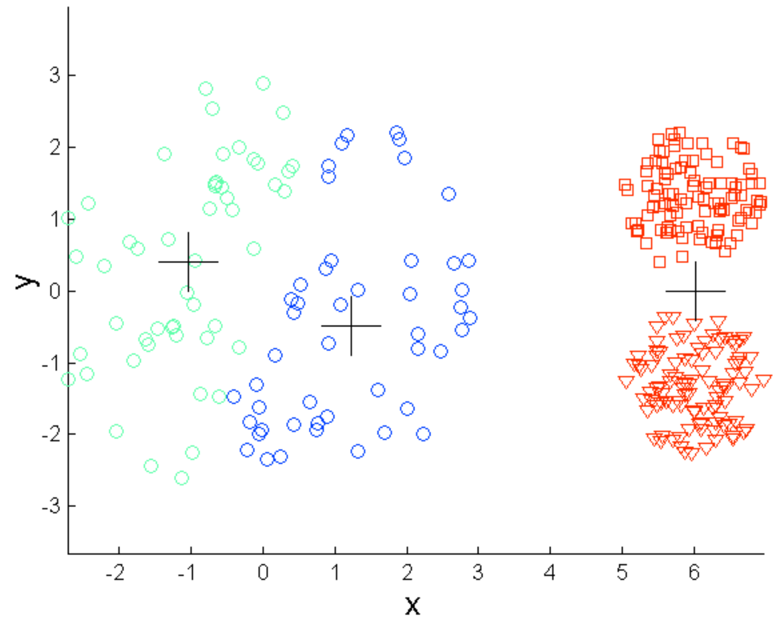


K-means (3 Clusters)

Clusters Are of Different Densities

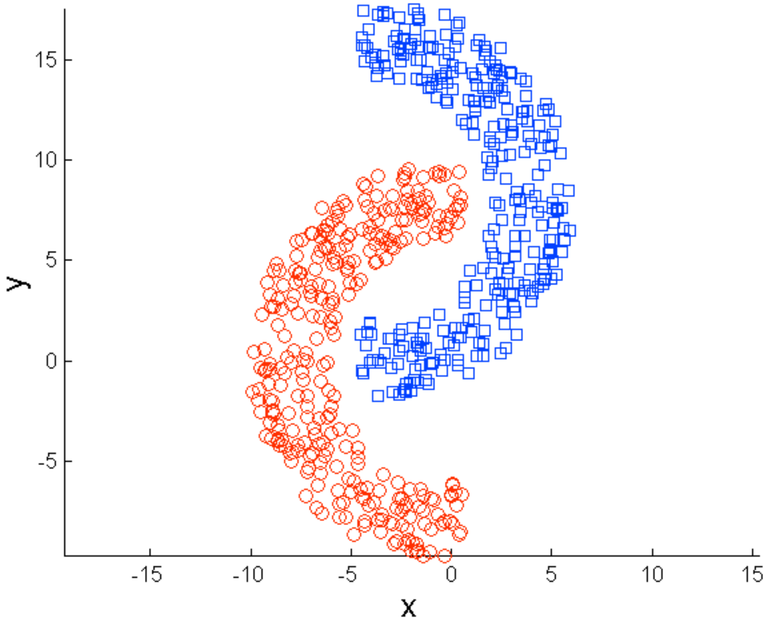


Original Points

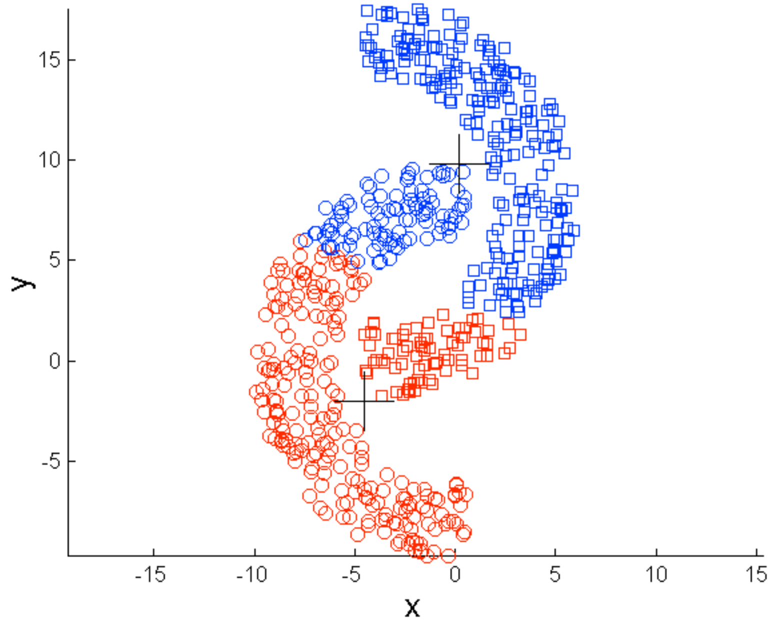


K-means (3 Clusters)

Clusters Are Non-Globular Shapes



Original Points



K-means (2 Clusters)

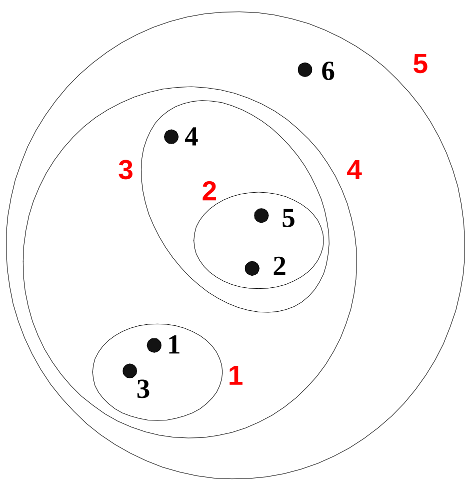
Pre- and Post- Processing

- Pre-processing
  - Normalize the data
  - Eliminate outliers
- Post-processing
  - Eliminate small clusters that may represent outliers
  - Split ‘loose’ clusters, i.e., clusters with relatively high SSE
  - Merge clusters that are ‘close’ and that have relatively low SSE

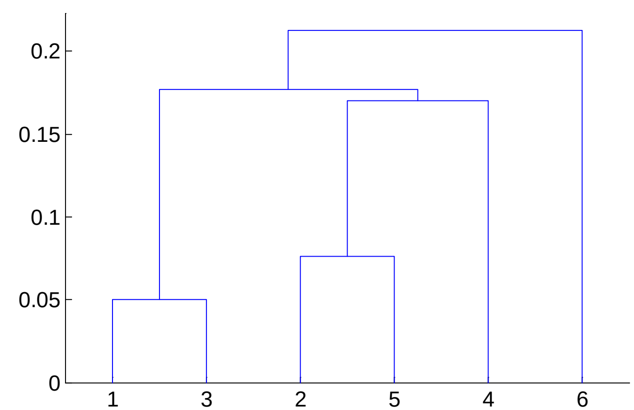
Implementation:
https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

Hierarchical Clustering

- Produce a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - A tree like diagram that records the sequences of merges or splits



Nested cluster diagram



Dendrogram

Strengths of Hierarchical Clustering

- Do not have to assume any particular number of clusters
  - Any desired number of clusters can be obtained by ‘cutting’ the dendrogram at the proper level
- They may correspond to meaningful taxonomies
  - Examples in document organization, biological sciences

Approaches

- Two main types of hierarchical clustering
  - Agglomerative:
    - Start with the data instances as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or K clusters) left
  - Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains one data instance (or there are K clusters)
- Use a proximity matrix (similarity or distance) to merge or split one cluster at a time

Agglomerative Clustering

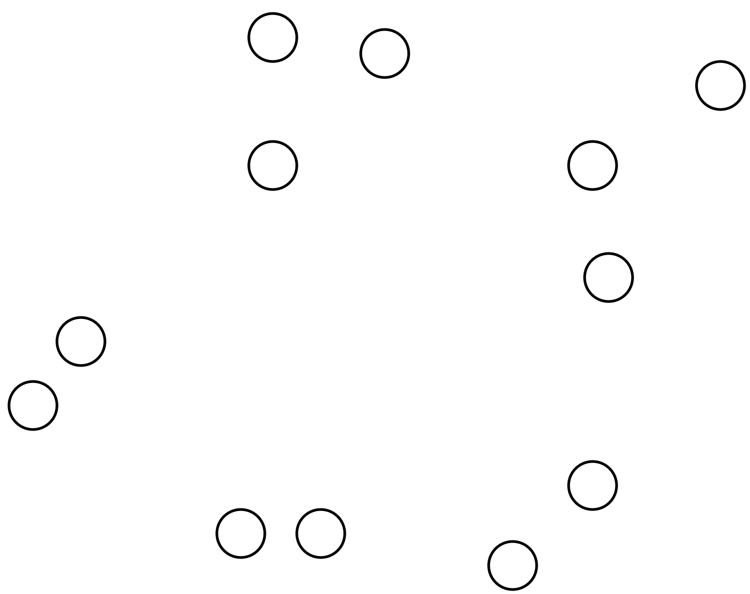
- Basic algorithm:
  1. Compute the $\boxed{\text{proximity}}$ matrix
  2. Let each data instance be a cluster
  3. **Repeat**
  4. Merge the two $\boxed{\text{closest}}$ clusters
  5. Update the proximity matrix
  6. **Until** only a single cluster remains
- Key operation is to compute the proximity of two clusters
  – Different approaches to defining the proximity between clusters lead to different clustering results

distance or similarity

smallest distance or largest similarity

Initial State

- Start with clusters of individual data instances and a proximity matrix between data instances

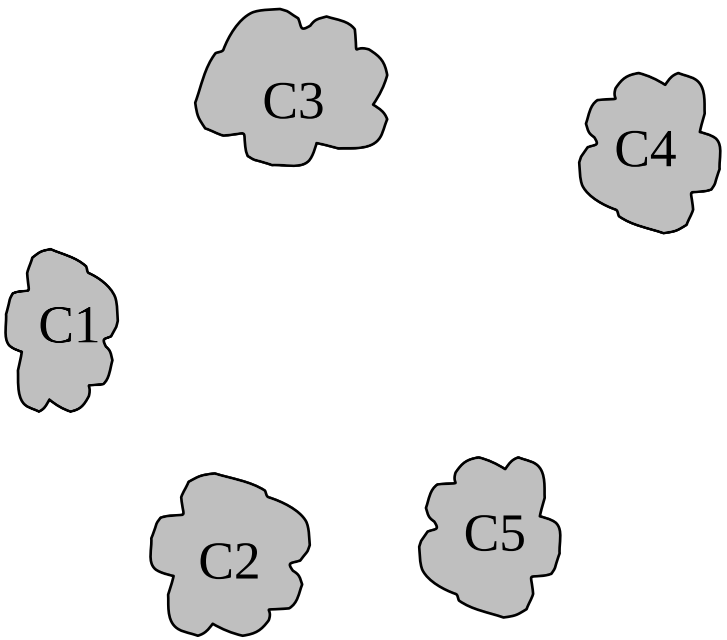


	P1	P2	P3	P4	P5	. . .
P1						
P2						
P3						
P4						
P5						
. . .						

Proximity Matrix

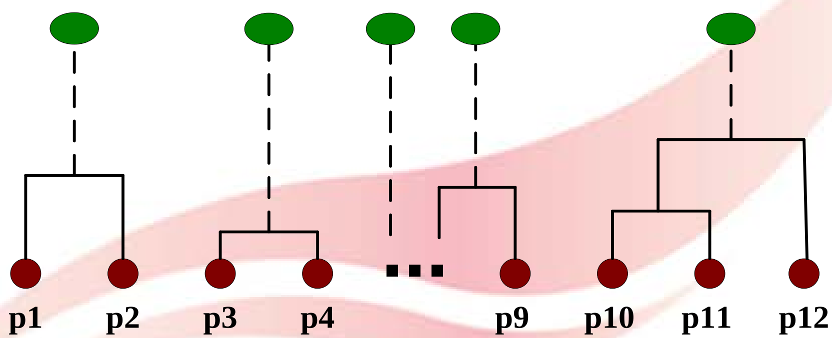
Intermediate State

- After some merging steps, we have some clusters



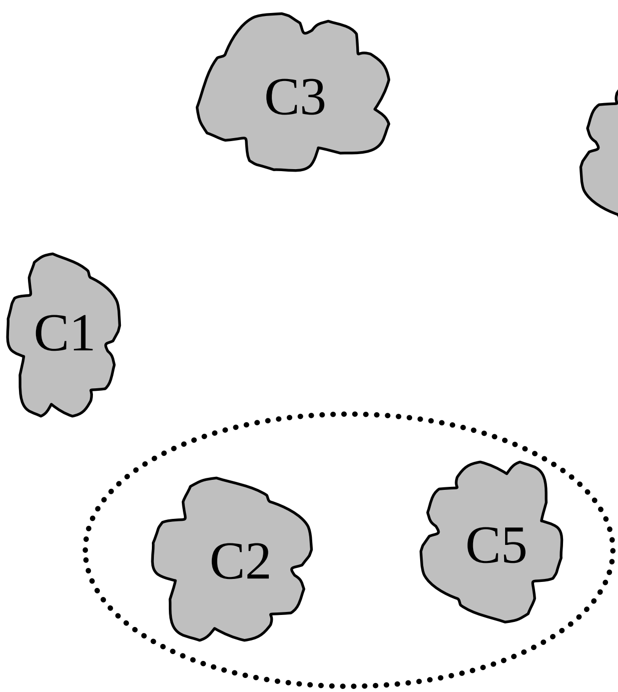
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

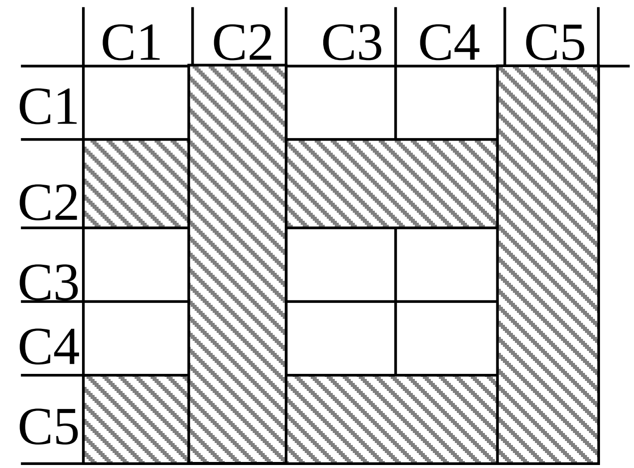
Proximity Matrix



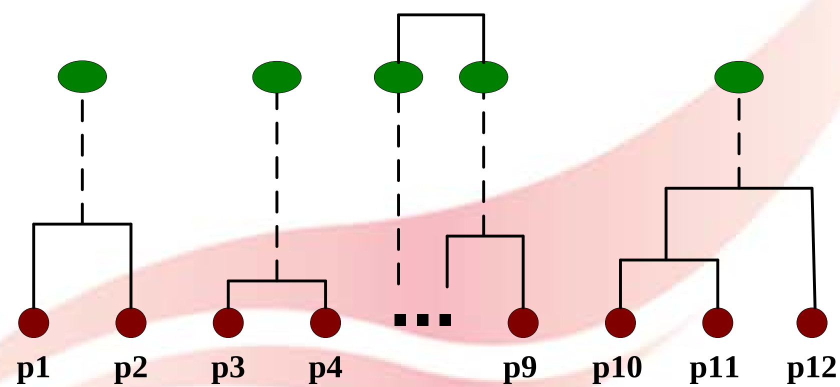
Before Merging

- We want to merge the two closest clusters (C2 and C5) and update the proximity matrix



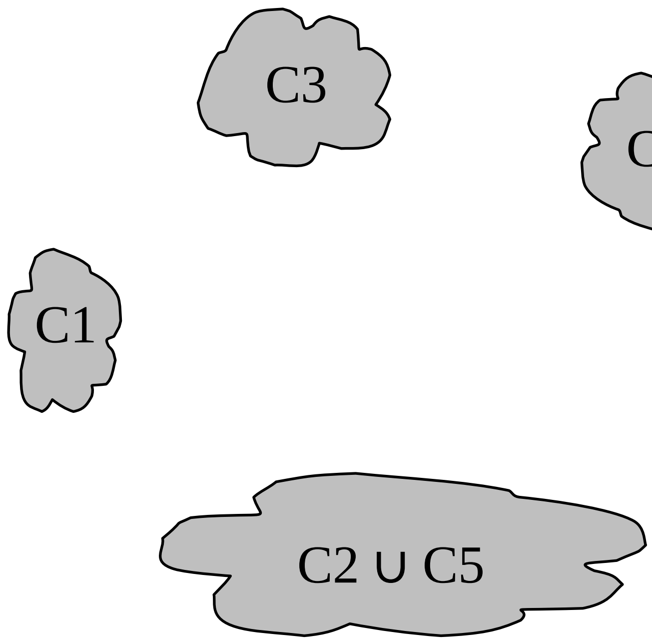


Proximity Matrix



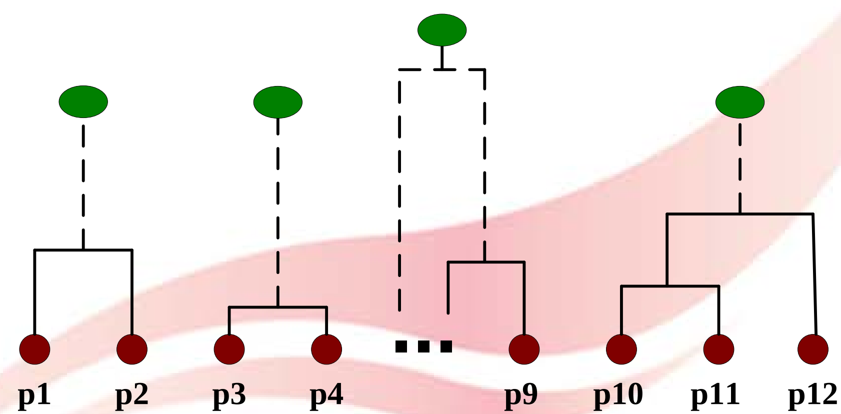
After Merging

- How do we update the proximity matrix?



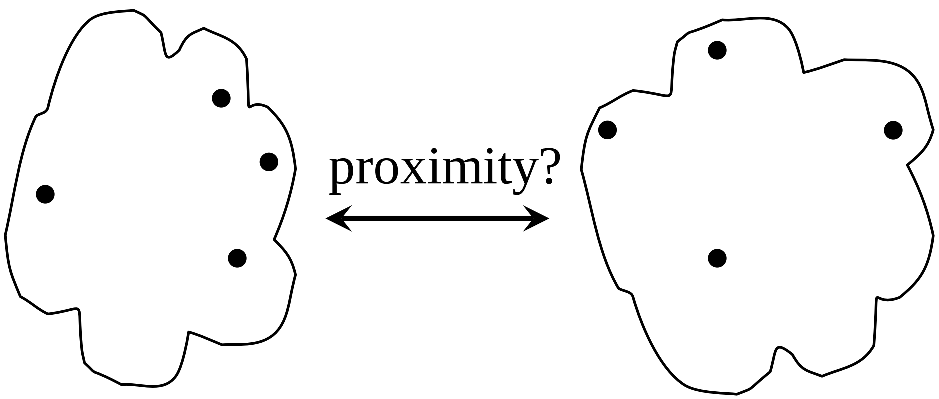
	C1	C2 ∪ C5	C3	C4
C1		?		
C2 ∪ C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix



Inter-Cluster Proximity

- MIN or Single Link
- MAX or Complete Link
- Group Average

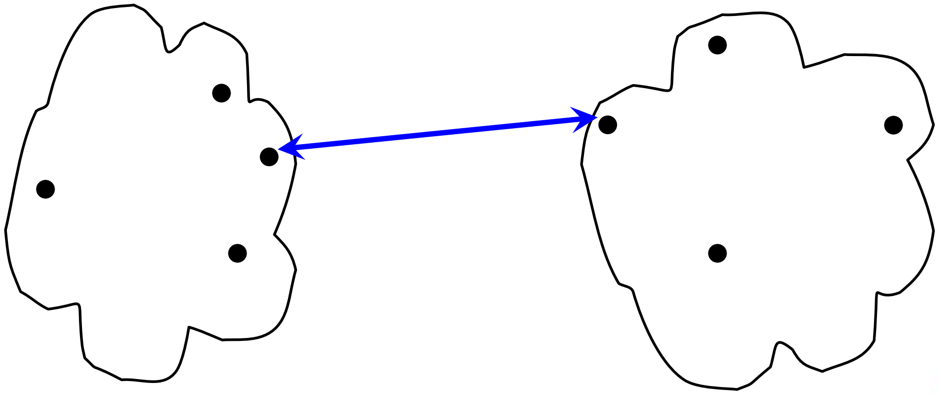


	P1	P2	P3	P4	P5	. . .
P1						
P2						
P3						
P4						
P5						
. . .						

Proximity Matrix

MIN or Single Link

- Defines cluster proximity as the proximity between the closest two points that are in different clusters
  - the shortest edge (single link) between two nodes in different subsets (using graph terms)

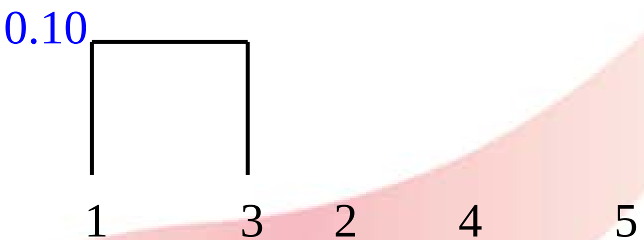


MIN or Single Link (cont.)

- MIN or Single Link: Distance of two clusters is based on the two most closest points in the different clusters

	P1	P2	P3	P4	P5
P1	0.00	0.90	0.10	0.65	0.20
P2	0.90	0.00	0.70	0.60	0.50
P3	0.10	0.70	0.00	0.40	0.30
P4	0.65	0.60	0.40	0.00	0.80
P5	0.20	0.50	0.30	0.80	0.00

Distance matrix

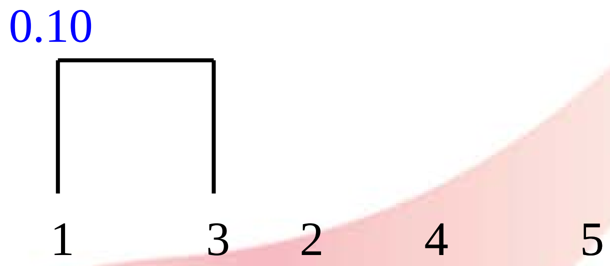


MIN or Single Link (cont.)

- MIN or Single Link: Distance of two clusters is based on the two most closest points in the different clusters

	P1∪P3	P2	P4	P5
P1∪P3	0.00	0.70	0.40	0.20
P2	0.70	0.00	0.60	0.50
P4	0.40	0.60	0.00	0.80
P5	0.20	0.50	0.80	0.00

Distance matrix

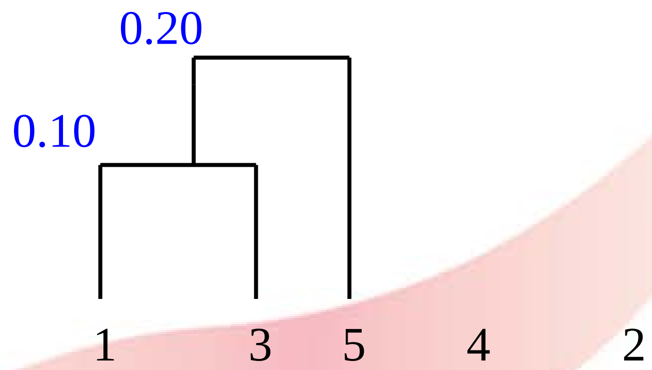


MIN or Single Link (cont.)

- MIN or Single Link: Distance of two clusters is based on the two most closest points in the different clusters

	P1∪P3	P2	P4	P5
P1∪P3	0.00	0.70	0.40	0.20
P2	0.70	0.00	0.60	0.50
P4	0.40	0.60	0.00	0.80
P5	0.20	0.50	0.80	0.00

Distance matrix

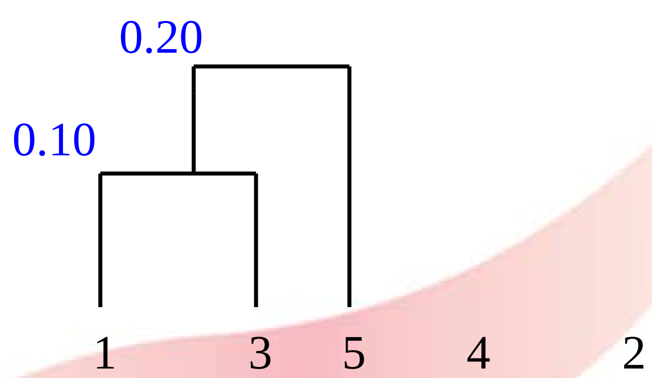


MIN or Single Link (cont.)

- MIN or Single Link: Distance of two clusters is based on the two most closest points in the different clusters

	P1∪P3∪P5	P2	P4
P1∪P3∪P5	0.00	0.50	0.40
P2	0.50	0.00	0.60
P4	0.40	0.60	0.00

Distance matrix

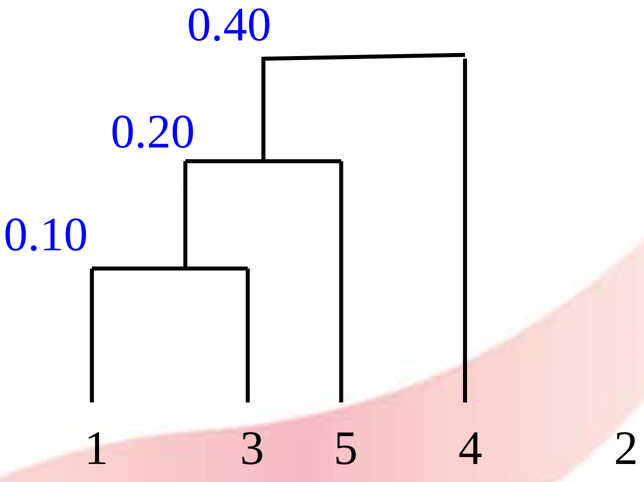


MIN or Single Link (cont.)

- MIN or Single Link: Distance of two clusters is based on the two most closest points in the different clusters

	P1∪P3∪P5	P2	P4
P1∪P3∪P5	0.00	0.50	0.40
P2	0.50	0.00	0.60
P4	0.40	0.60	0.00

Distance matrix

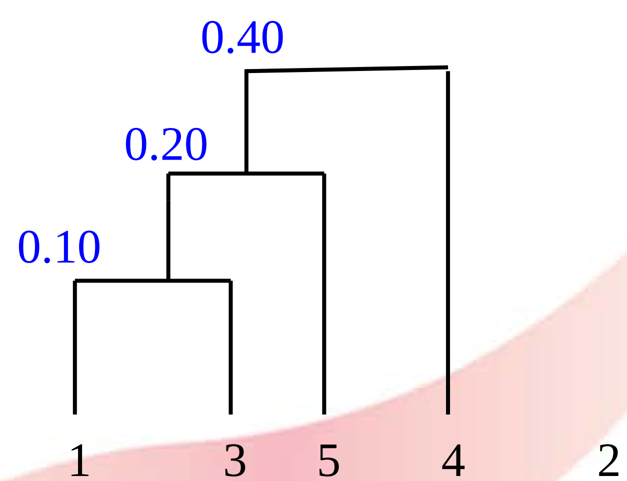


MIN or Single Link (cont.)

- MIN or Single Link: Distance of two clusters is based on the two most closest points in the different clusters

	P1∪P3∪P5∪P4	P2
P1∪P3∪P5∪P4	0.00	0.50
P2	0.50	0.00

Distance matrix

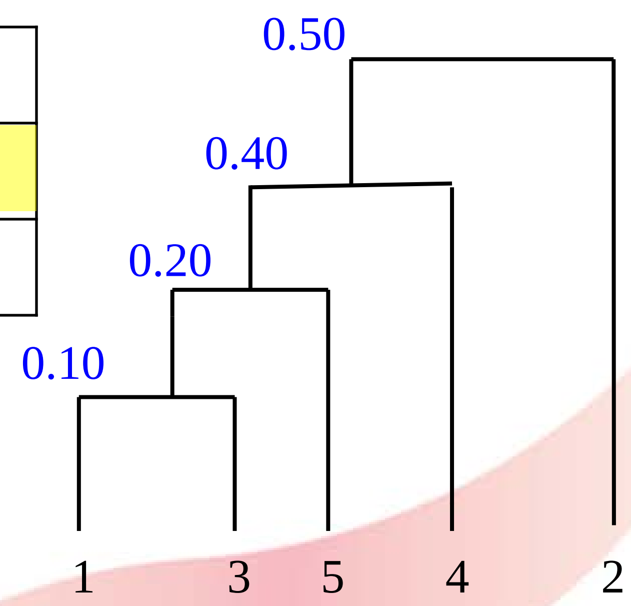


MIN or Single Link (cont.)

- MIN or Single Link: Distance of two clusters is based on the two most closest points in the different clusters

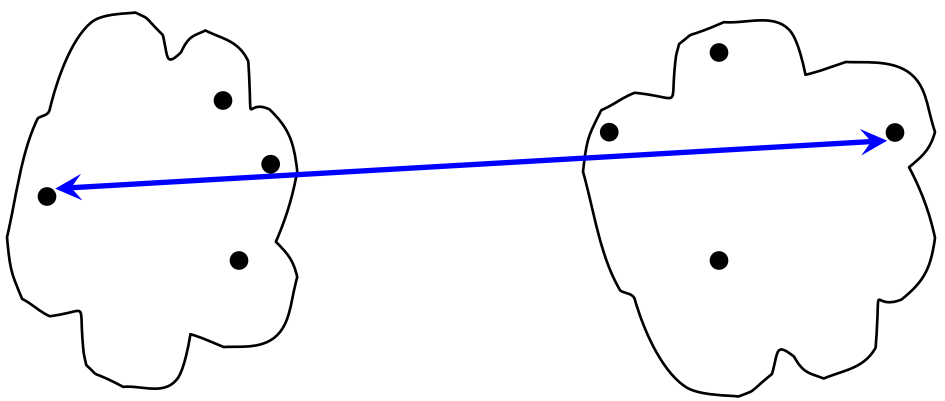
	P1∪P3∪P5∪P4	P2
P1∪P3∪P5∪P4	0.00	0.50
P2	0.50	0.00

Distance matrix



MAX or Complete Link

- Defines cluster proximity as the proximity between the farthest two points that are in different clusters
  - the longest edge (complete link) between two nodes in different subsets (using graph terms)



MAX or Complete Link (cont.)

- MAX or Complete Link: Distance of two clusters is based on the two farthest points in the different clusters

	P1	P2	P3	P4	P5
P1	0.00	0.90	0.10	0.65	0.20
P2	0.90	0.00	0.70	0.60	0.50
P3	0.10	0.70	0.00	0.40	0.30
P4	0.65	0.60	0.40	0.00	0.80
P5	0.20	0.50	0.30	0.80	0.00

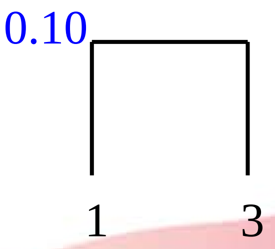
Distance matrix

MAX or Complete Link (cont.)

- MAX or Complete Link: Distance of two clusters is based on the two farthest points in the different clusters

	P1	P2	P3	P4	P5
P1	0.00	0.90	0.10	0.65	0.20
P2	0.90	0.00	0.70	0.60	0.50
P3	0.10	0.70	0.00	0.40	0.30
P4	0.65	0.60	0.40	0.00	0.80
P5	0.20	0.50	0.30	0.80	0.00

Distance matrix

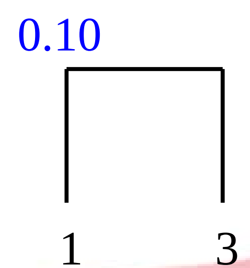


MAX or Complete Link (cont.)

- MAX or Complete Link: Distance of two clusters is based on the two farthest points in the different clusters

	P1∪P3	P2	P4	P5
P1∪P3	0.00	0.90	0.65	0.30
P2	0.90	0.00	0.60	0.50
P4	0.65	0.60	0.00	0.80
P5	0.30	0.50	0.80	0.00

Distance matrix

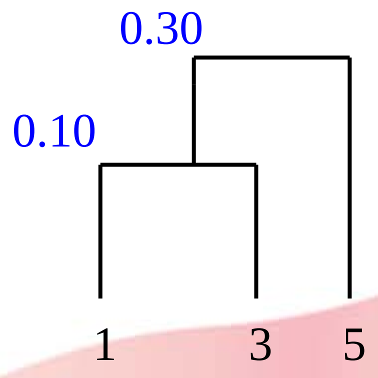


MAX or Complete Link (cont.)

- MAX or Complete Link: Distance of two clusters is based on the farthest points in the different clusters

	P1∪P3	P2	P4	P5
P1∪P3	0.00	0.90	0.65	0.30
P2	0.90	0.00	0.60	0.50
P4	0.65	0.60	0.00	0.80
P5	0.30	0.50	0.80	0.00

Distance matrix

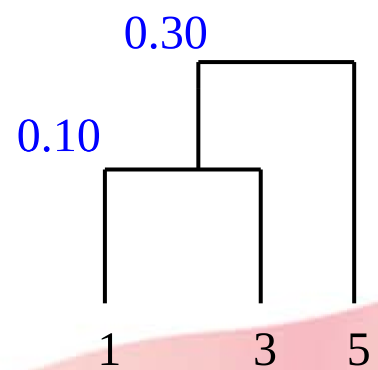


MAX or Complete Link (cont.)

- MAX or Complete Link: Distance of two clusters is based on the two farthest points in the different clusters

	P1∪P3∪P5	P2	P4
P1∪P3∪P5	0.00	0.90	0.80
P2	0.90	0.00	0.60
P4	0.80	0.60	0.00

Distance matrix

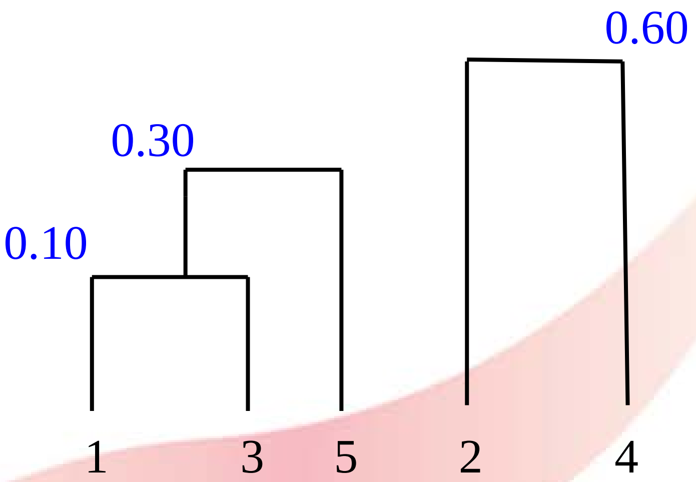


MAX or Complete Link (cont.)

- MAX or Complete Link: Distance of two clusters is based on the two farthest points in the different clusters

	P1∪P3∪P5	P2	P4
P1∪P3∪P5	0.00	0.90	0.80
P2	0.90	0.00	0.60
P4	0.80	0.60	0.00

Distance matrix

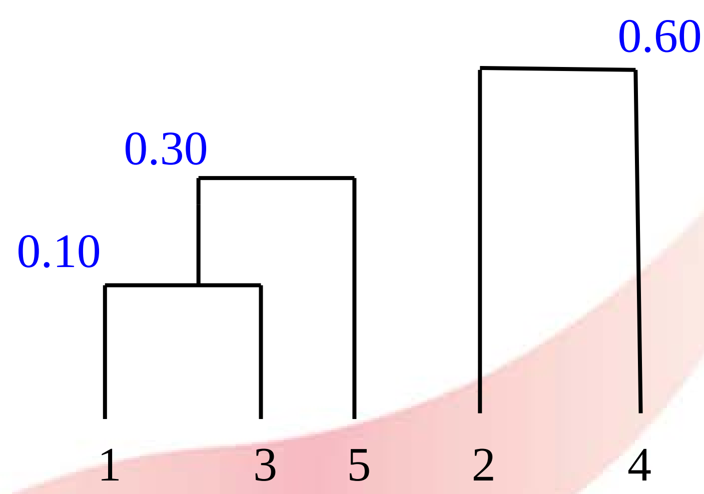


MAX or Complete Link (cont.)

- MAX or Complete Link: Distance of two clusters is based on the two farthest points in the different clusters

	P1∪P3∪P5	P2∪P4
P1∪P3∪P5	0.00	0.90
P2∪P4	0.90	0.00

Distance matrix

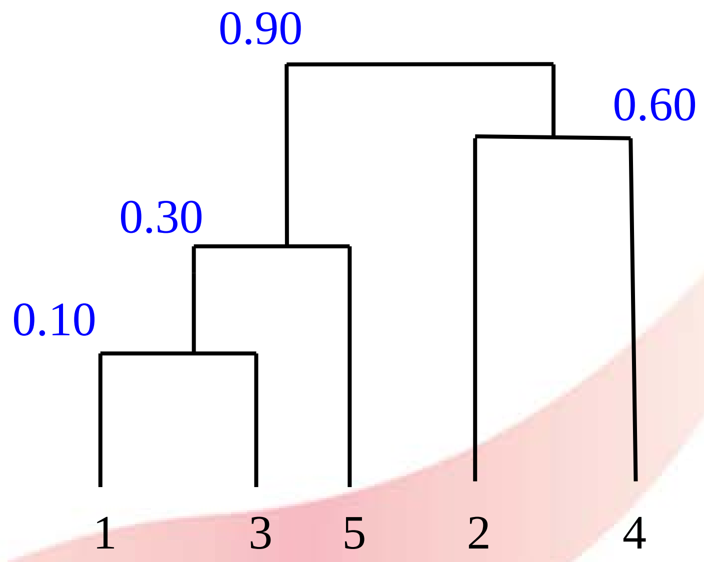


MAX or Complete Link (cont.)

- MAX or Complete Link: Distance of two clusters is based on the two farthest points in the different clusters

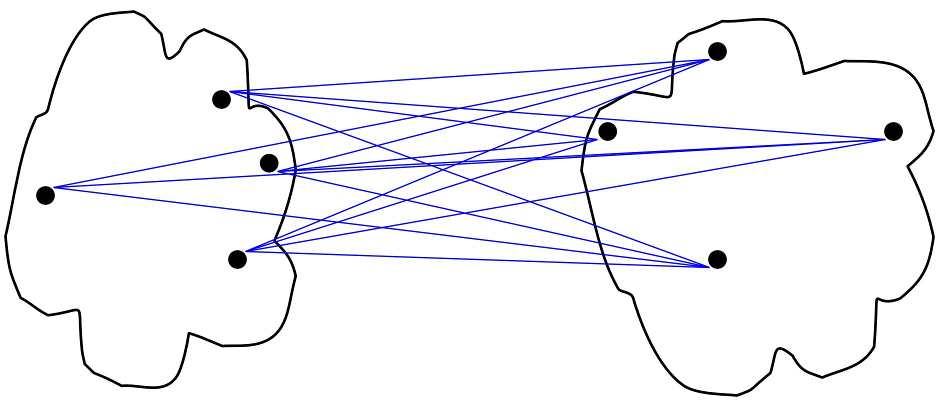
	P1∪P3∪P5	P2∪P4
P1∪P3∪P5	0.00	0.90
P2∪P4	0.90	0.00

Distance matrix



Group Average

- Defines cluster proximity as the average pairwise proximities of all pairs of points from different clusters
  - average length of edges between nodes in different subsets (using graph terms)



Group Average (cont.)

- Proximity of two clusters is the average of pairwise proximity between points in the two clusters

$$\text{Proximity}(C_i, C_j) = \frac{\sum_{\boldsymbol{x}_i \in C_i, \boldsymbol{x}_j \in C_j} \text{Proximity}(\boldsymbol{x}_i, \boldsymbol{x}_j)}{|C_i| \times |C_j|}$$

- Need to use average connectivity for scalability since total proximity favors large clusters

Limitations

- Once a decision is made to combine two clusters, it cannot be undone
- Sensitivity to noise and outliers

Implementation

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html#sklearn.cluster.AgglomerativeClustering

Implementation of other clustering algorithms

https://scikit-learn.org/stable/modules/classes.html#module-sklearn.cluster

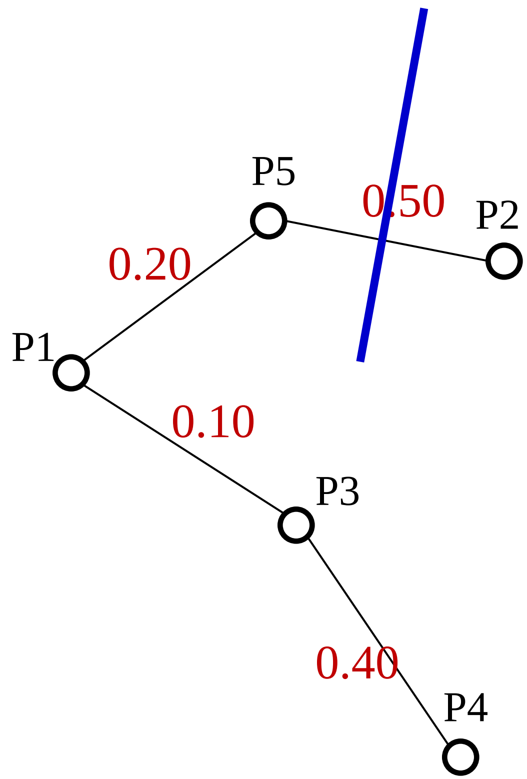
Divisive Hierarchical Clustering

- Basic algorithm:
  1. Compute a minimum spanning tree for the proximity graph
  2. **Repeat**
  3. Create a new cluster by breaking the link corresponding to the largest distance (smallest similarity)
  4. **Until** only singleton clusters remain
- Minimum Spanning Tree (MST)
  - Start with a tree that consists of any data instance
  - In successive steps, look for the closest pair of points $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ such that one point $(\boldsymbol{x}_i)$ is in the current tree but the other $(\boldsymbol{x}_j)$ is not
  - Add $(\boldsymbol{x}_j)$ to the tree and put an edge between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$

An Example

The distance matrix between 5 points

	P1	P2	P3	P4	P5
P1	0.00	0.90	0.10	0.65	0.20
P2	0.90	0.00	0.70	0.60	0.50
P3	0.10	0.70	0.00	0.40	0.30
P4	0.65	0.60	0.40	0.00	0.80
P5	0.20	0.50	0.30	0.80	0.00



Suppose $K = 2$, and P3 is chosen at the beginning for constructing the MST

Thank you!