CZ4041/CE4041: Machine Learning

Lesson 10: Clustering

Kelly KE

School of Computer Science and Engineering,
NTU, Singapore

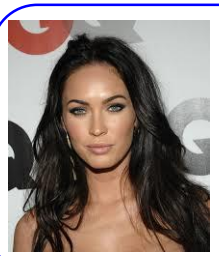
Acknowledgements: some figures are adapted from the lecture notes of the book "Introduction to Data Mining" (Chap. 8). Slides are modified from the version prepared by Dr. Sinno Pan.

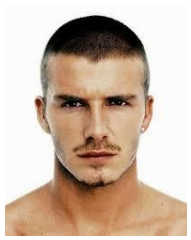
Supervised Learning

- The examples presented to computers are pairs of inputs and the corresponding outputs, the goal is to "learn" a mapping or model from inputs to labels

Labeled training data

Inputs: Face images






Outputs: Female or Male

Female Female Male Male

label = f(input)

Unsupervised Learning

- The examples presented to computers are a set of inputs without any outputs, the goal is to "learn" an intrinsic structure of the examples, e.g., clusters of examples, density of the examples

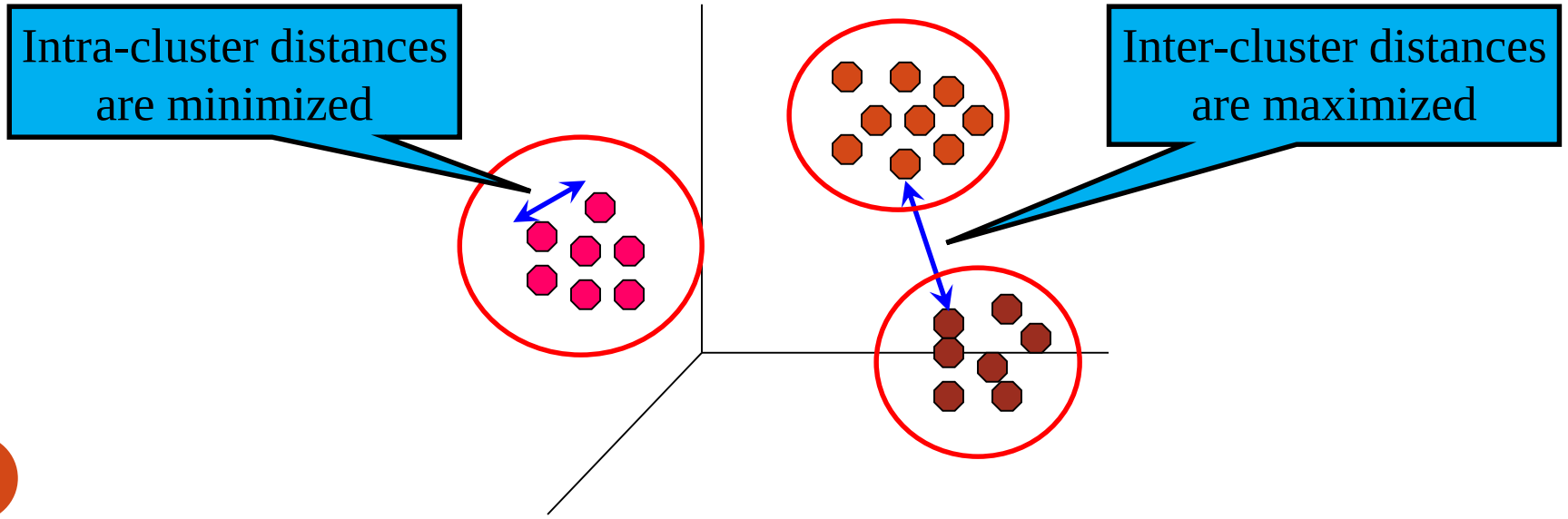
Unlabeled training data

Inputs: Face images

Groups of similar faces

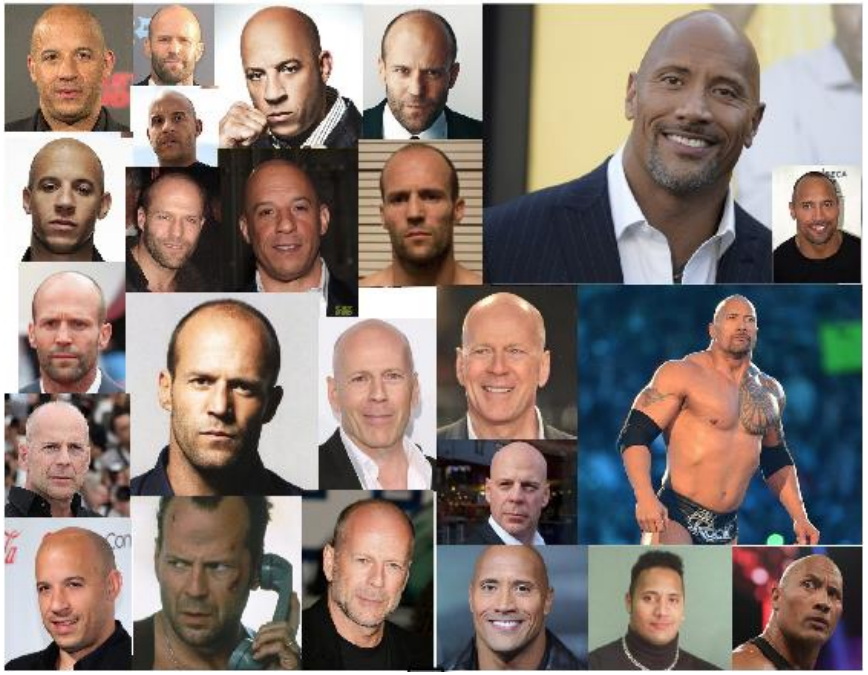
What is Cluster Analysis?

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



An example of clustering

Grouping faces for different persons



Images are copied from http://blog.dlib.net/2017/02/high-quality-face-recognition-with-deep.html

Another example of clustering

Customer segmentation

Based on customers' profiles:
education background, occupation, shopping behaviours, etc.

What is not Cluster Analysis?

- Supervised classification
  - Have class label information
- Simple segmentation
  - Dividing customers into different groups alphabetically, by last name

Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters

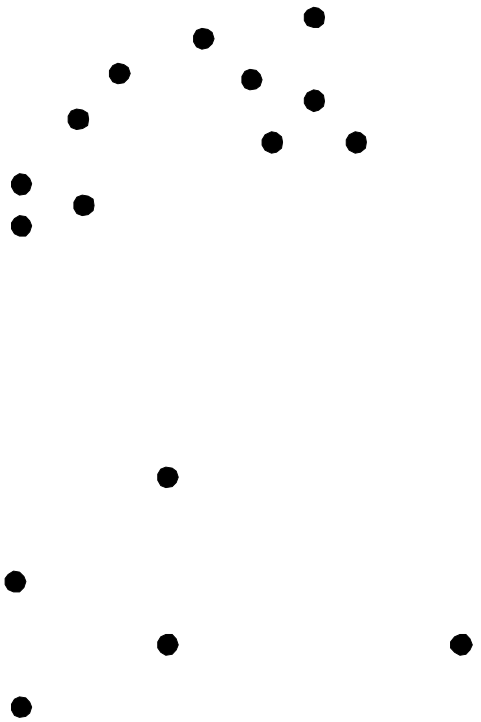
Two Clusters

Four Clusters

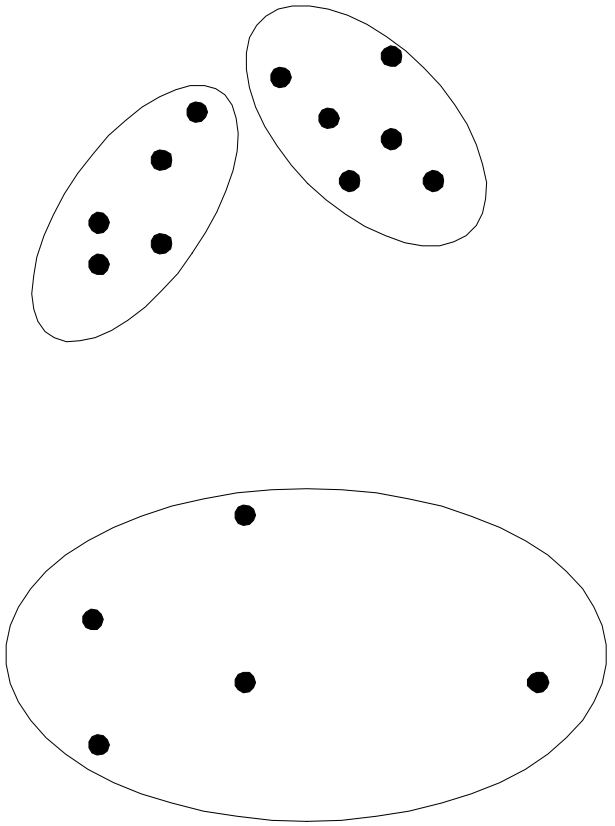
Types of Clusterings

- A clustering is a set of clusters
- Important distinction between hierarchical and partitional sets of clusters
- Partitional clustering
  - Divide data instances into **non-overlapping** subsets (clusters) so that each data instance is in exactly one subset
- Hierarchical clustering
  - A set of **nested** clusters organized as a hierarchical tree

Partitional Clustering

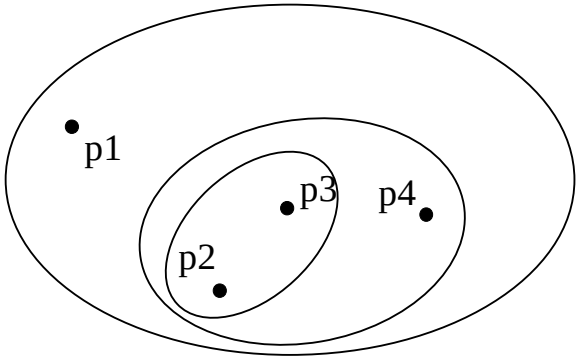


Original instances

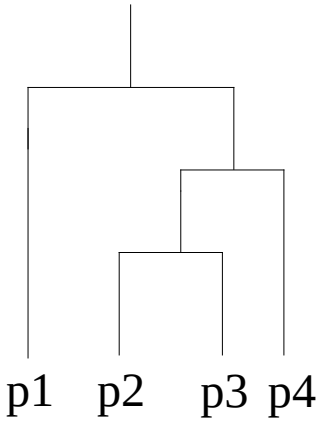


A Partitional Clustering

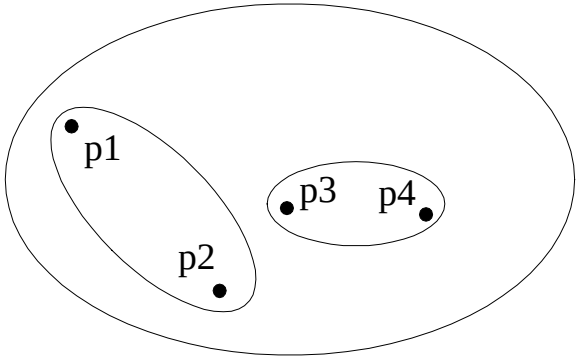
Hierarchical Clustering



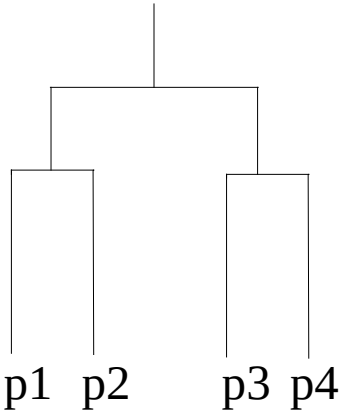
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

Other Distinctions Between Clusterings

- Exclusive versus non-exclusive
  - In non-exclusive clustering, instances may belong to multiple clusters
- Fuzzy versus non-fuzzy
  - In fuzzy clustering, a point belongs to every cluster with some weight between 0 and 1
  - Weights must sum to 1
- Partial versus complete
  - In partial clustering, only some of the instances are clustered

Clustering Algorithms

- K-means and its variants
- Hierarchical clustering

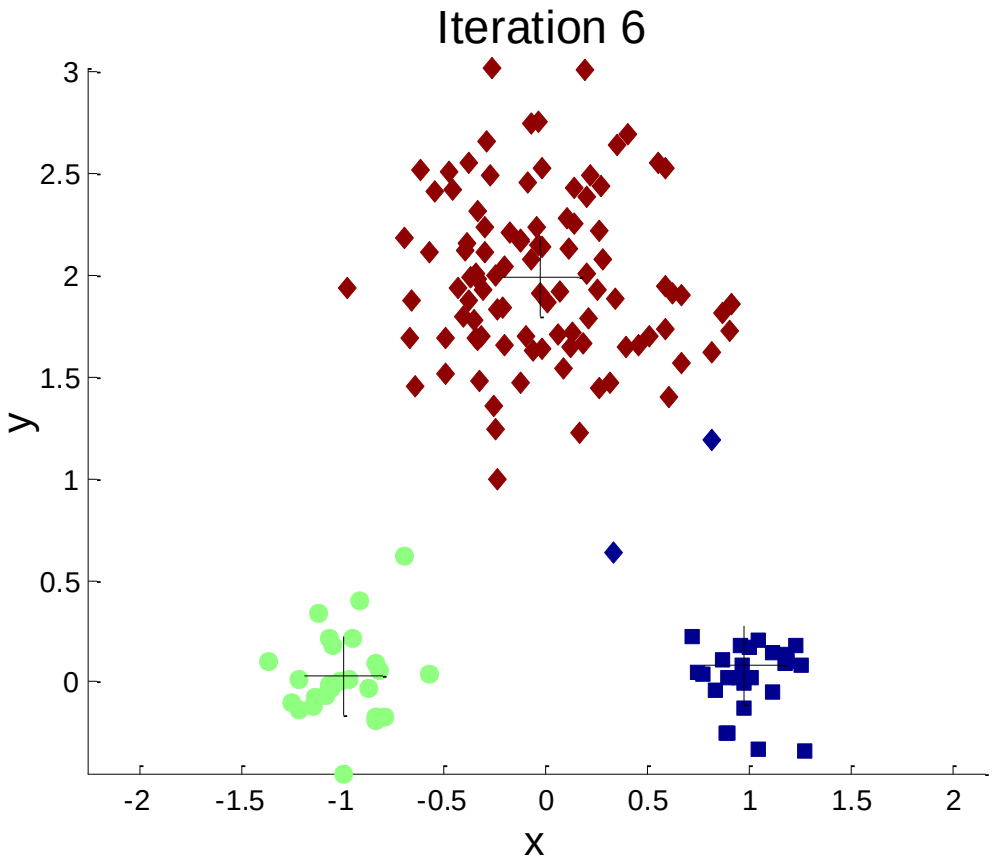
K-means Clustering

- Partitional clustering approach
- Each cluster is associated with a centroid (center point)
- Each point 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 closest centroid
  4. Recompute the centroid of each cluster
  5. **Until** The centroids do not change

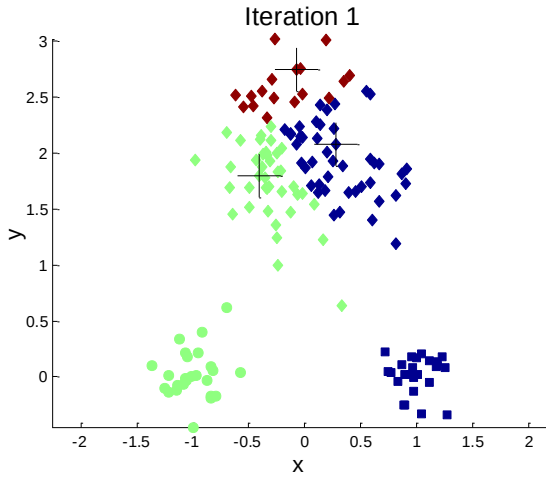
K-means Clustering – Details

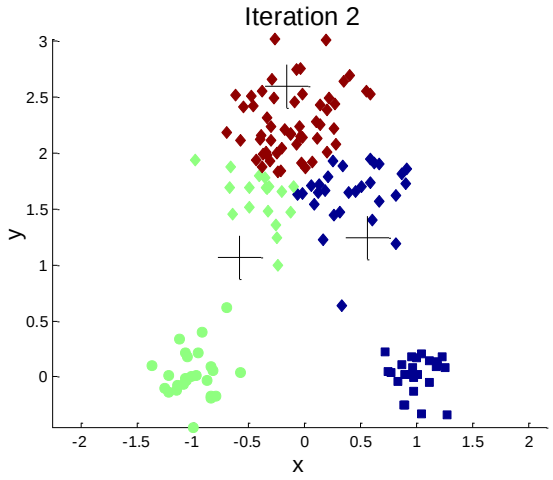
- Initial centroids are often chosen randomly
- The centroid is (typically) the mean of the data instances in the cluster
- 'Closeness' is measured by a proximity
  - Distance: Euclidean distance, etc
  - Similarity: cosine similarity, correlation, etc
- *K*-means will converge for common similarity measures mentioned above
  - In practice, it converges in the first few iterations
  - Often the stopping condition is changed to 'Until relatively few instances change clusters'

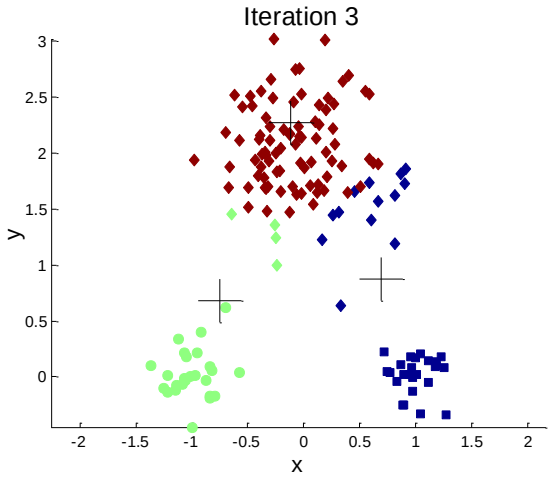
K-means Illustration

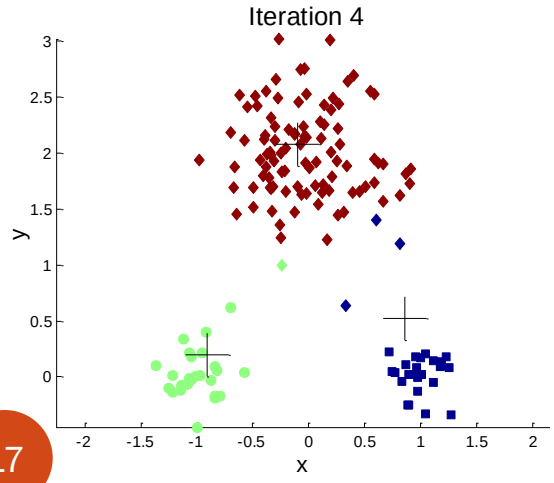


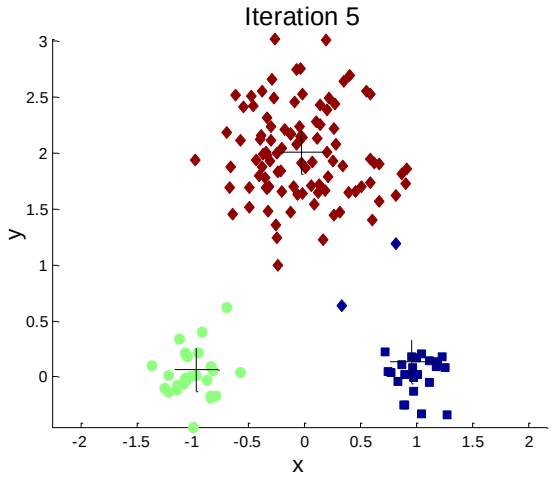
K-means Illustration (cont.)

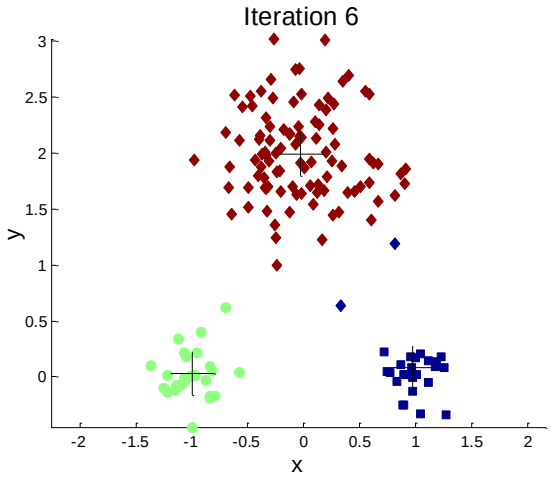




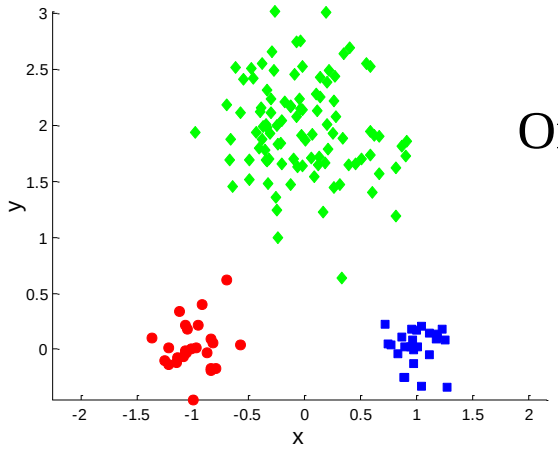




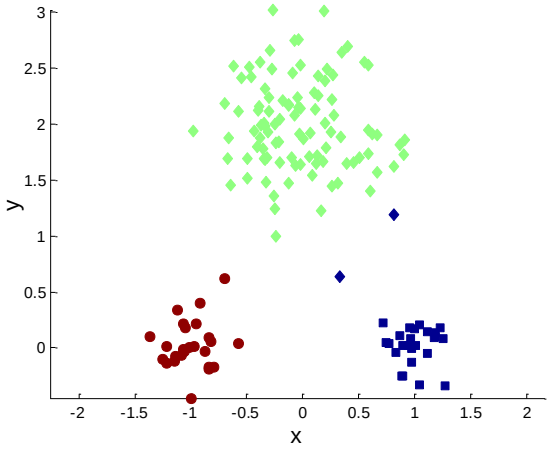




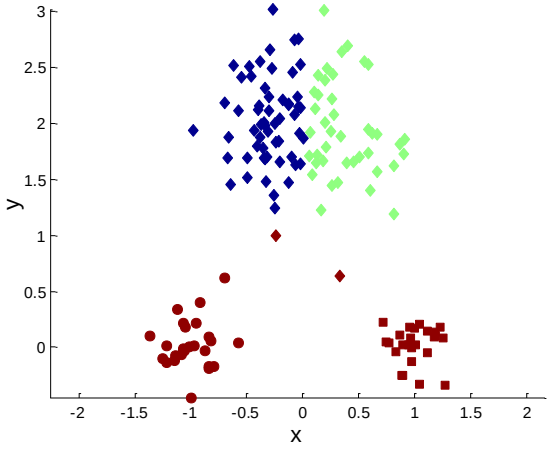
Two different *K*-means Clusterings



Original Points



Optimal Clustering



Sub-optimal Clustering

Evaluating *K*-means Clusters

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

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

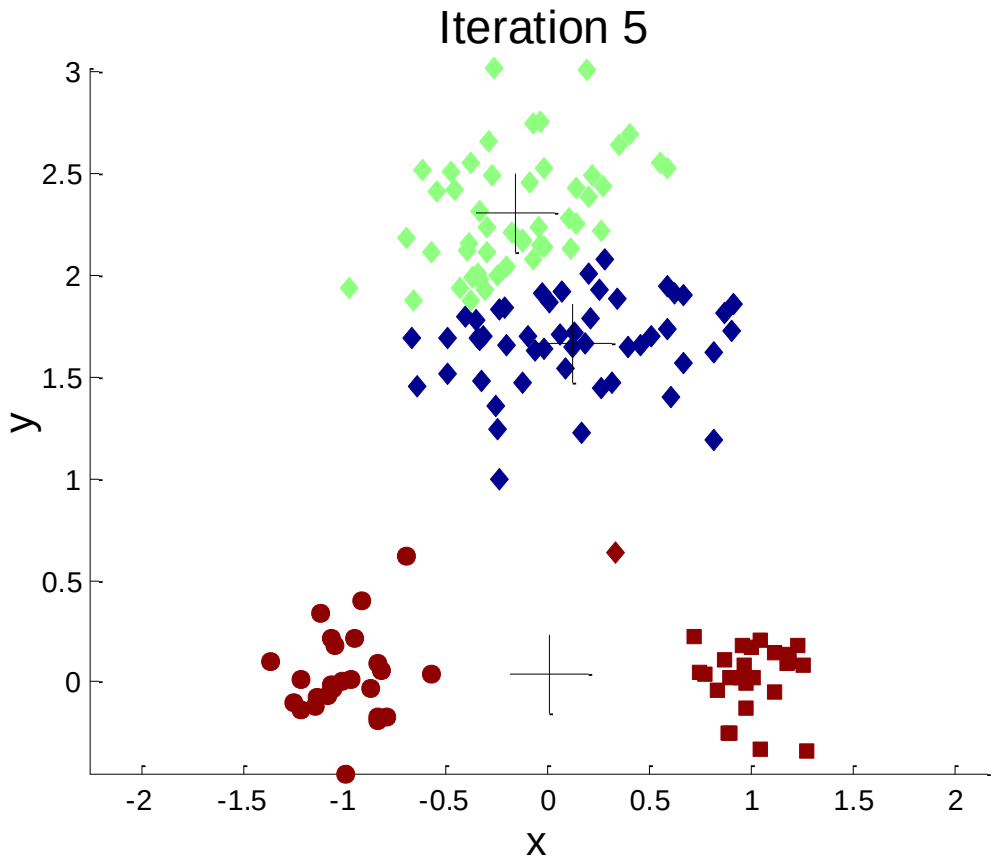
Centroid of the cluster C_i

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

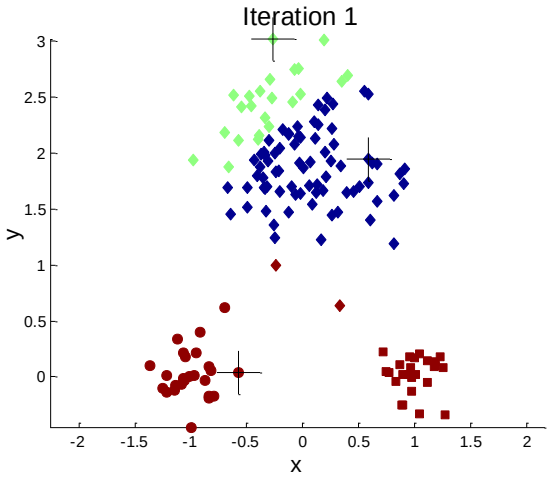
Evaluating *K*-means Clusters

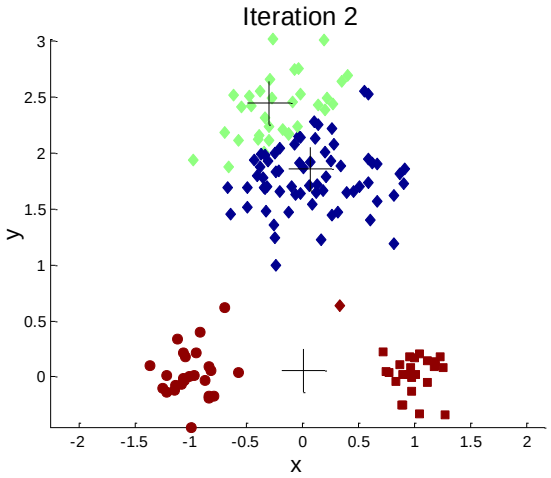
- Given two different runs of *K*-means, we can choose the one with the smallest Total SSE
- One easy way to reduce SSE is to increase *K*, the number of clusters
  - However, a good clustering with smaller *K* can have a lower SSE than a poor clustering with higher *K*

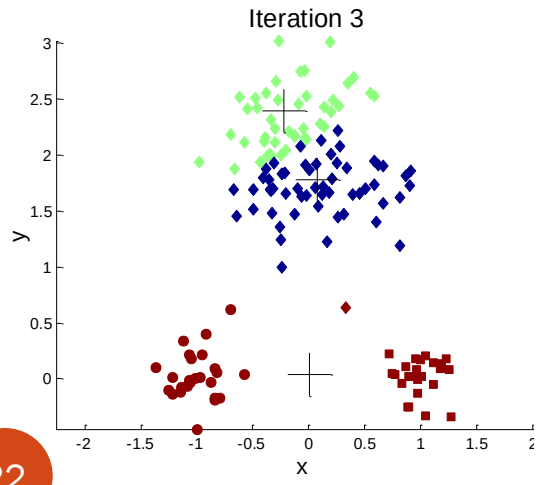
Importance of Initial Centroids

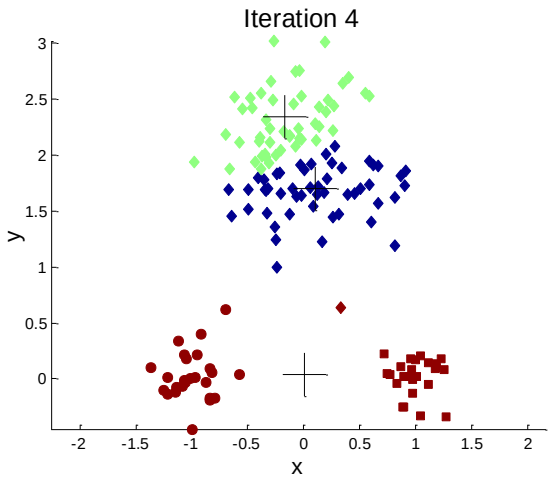


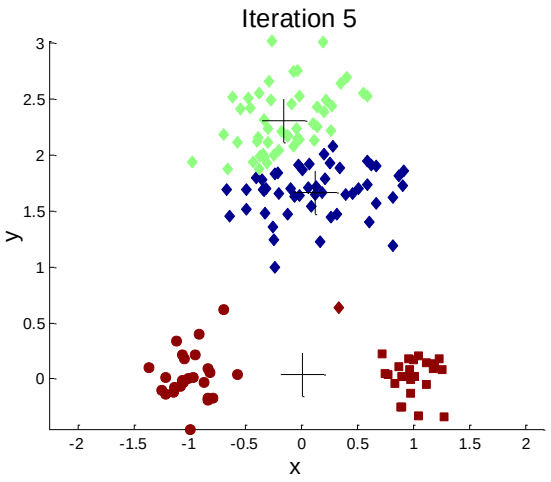
Importance of Initial Centroids (cont.)



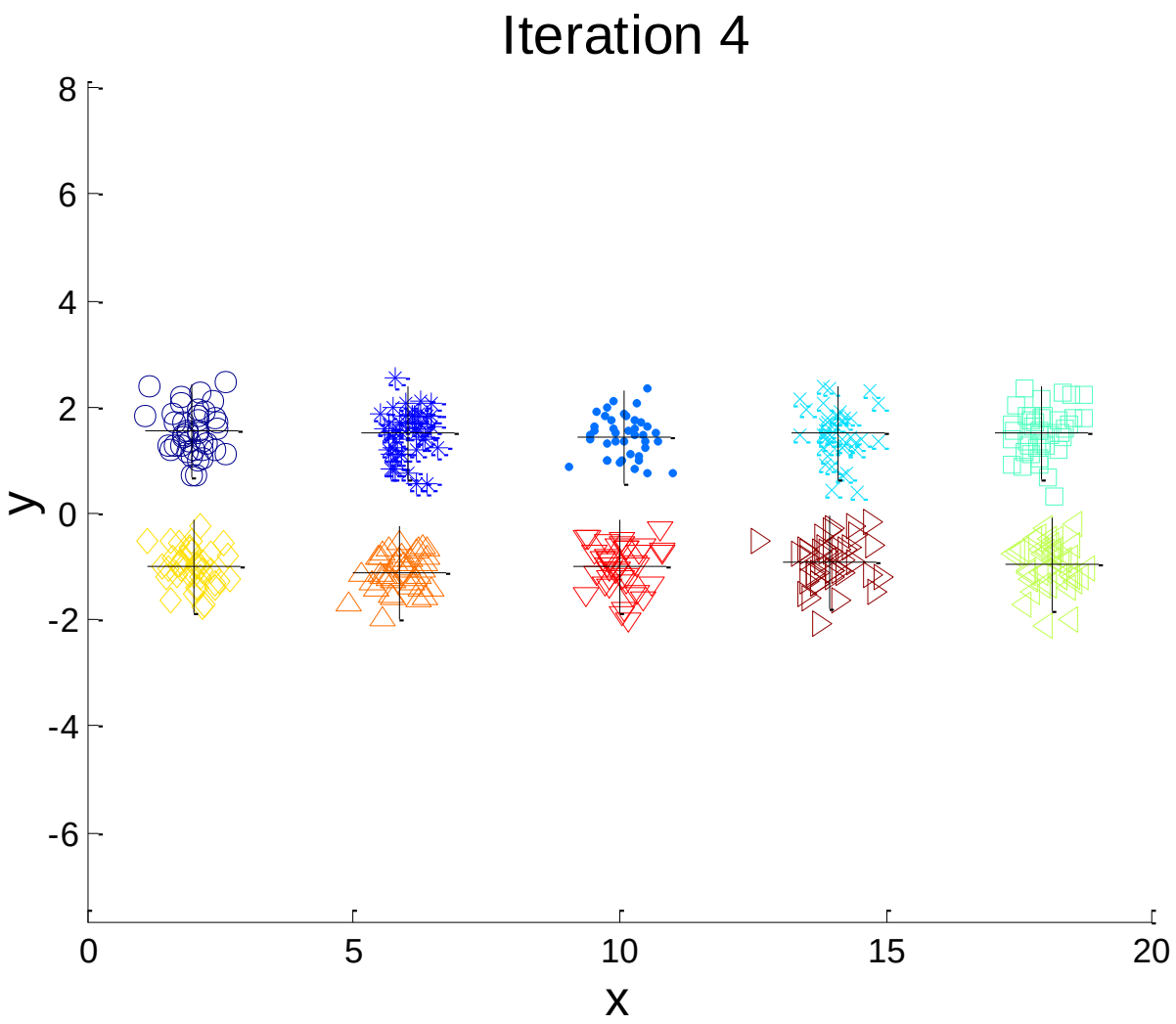






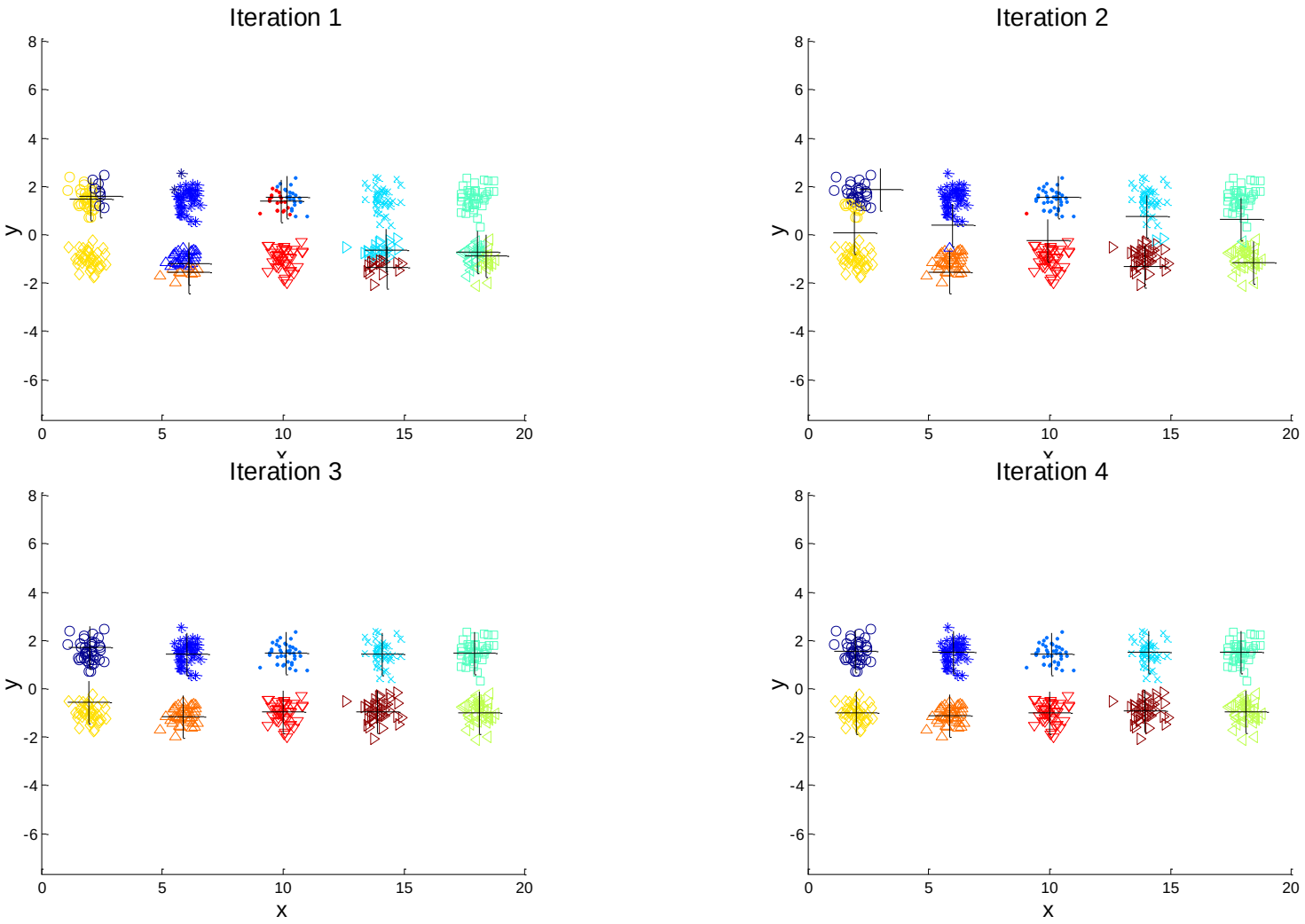


10 Clusters Example



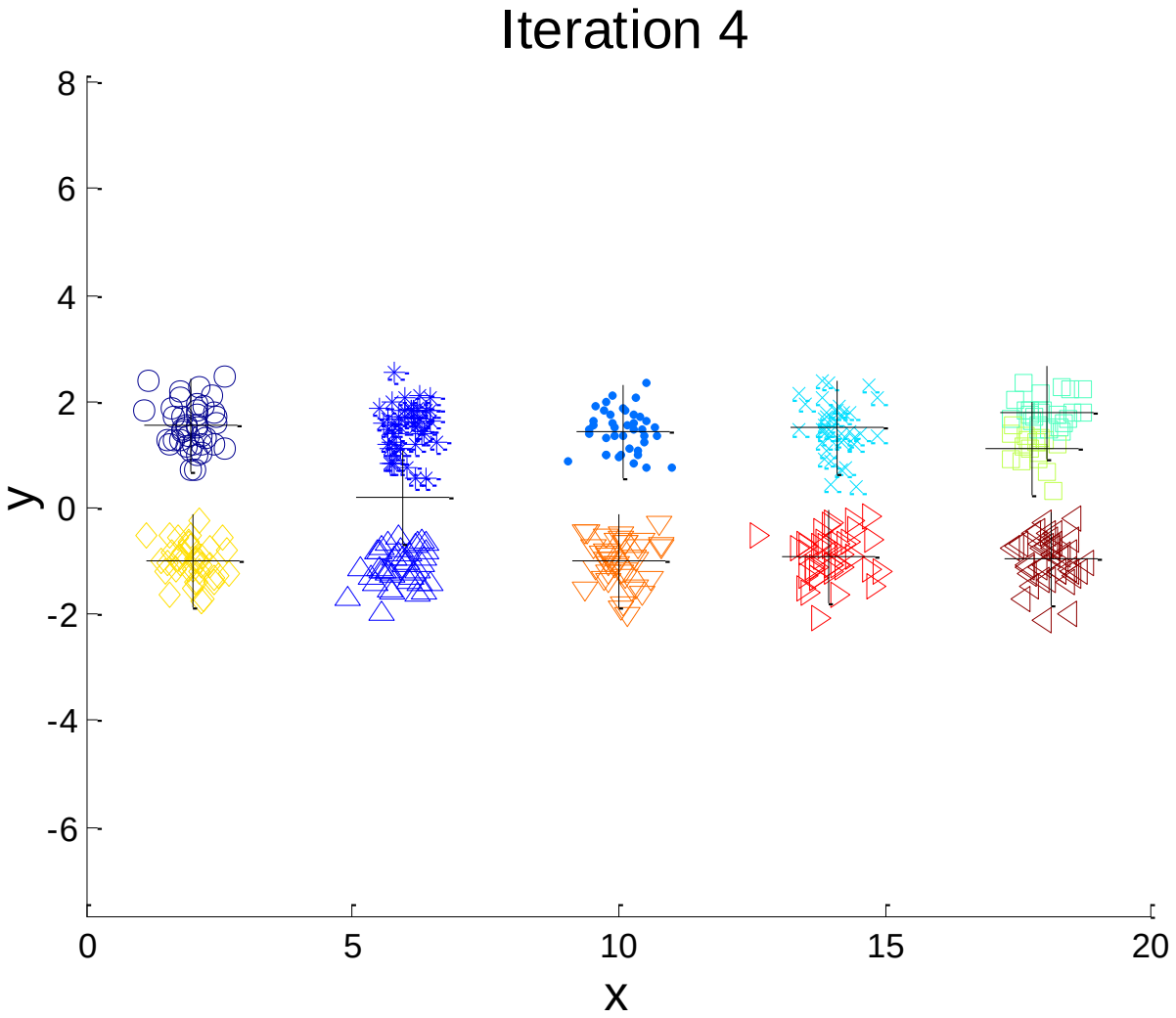
Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example (cont.)



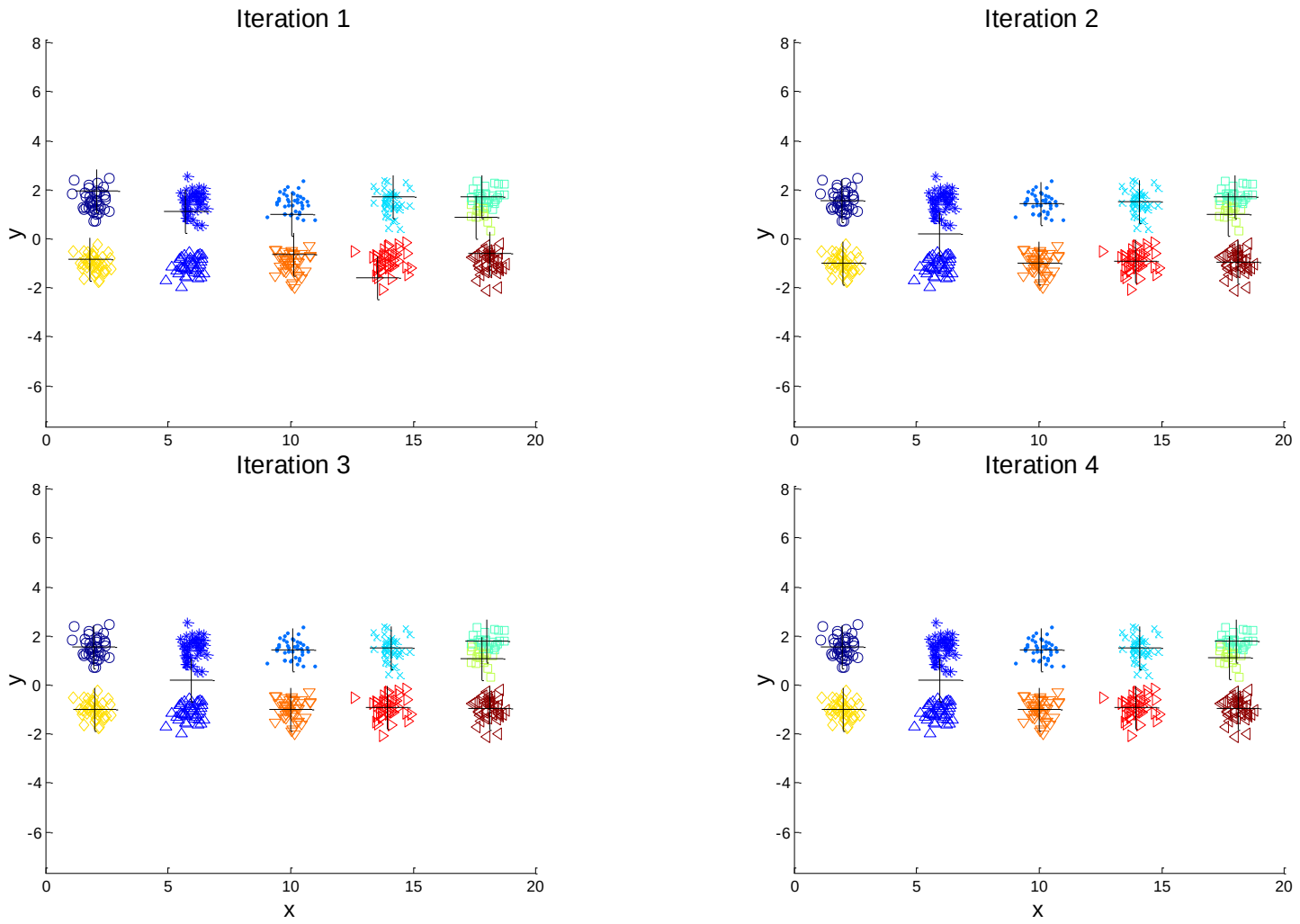
Starting with two initial centroids in one cluster of each pair of clusters

10 Clusters Example (cont.)



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

10 Clusters Example (cont.)



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

Initial Centroids Issue

- Multiple runs
- Postprocessing
  - Decompose a cluster with high Cluster SSE,
  - Merge clusters with low Cluster SSE, which are close to each other
- Bisecting K-means

Pre-processing 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

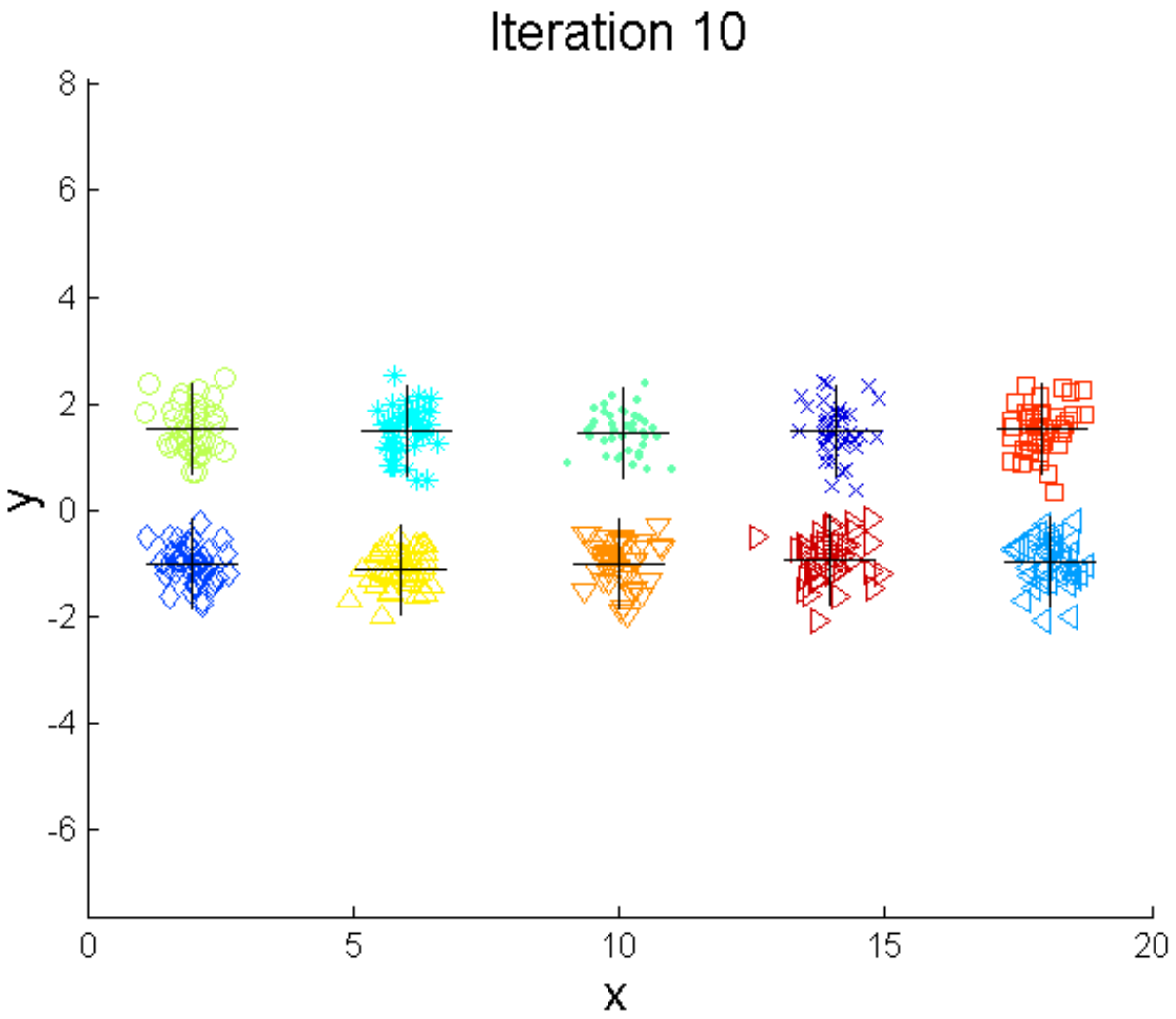
Empty Clusters Issue

- Basic *K*-means algorithm can yield empty clusters
- Several strategies to choose a replacement centroid
  - Choose the point that contributes most to SSE
  - Choose a point from the cluster with the highest Cluster SSE
  - If there are several empty clusters, the above can be repeated several times

Bisecting K-means

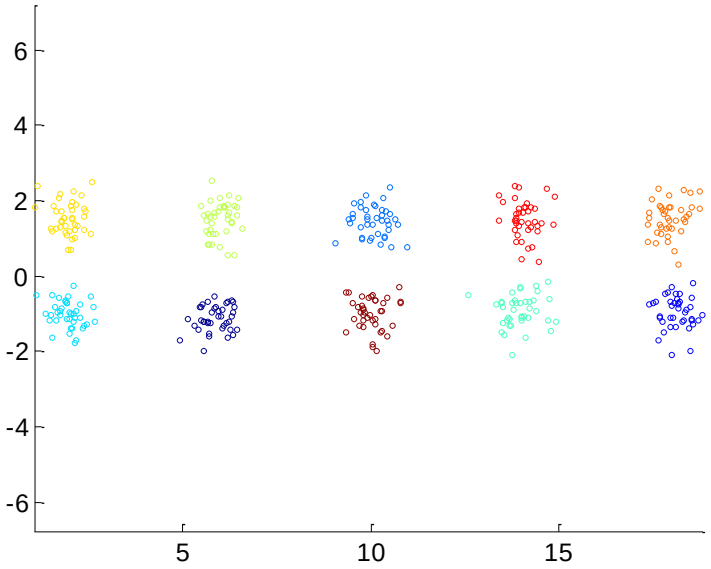
- Basic algorithm:
  1. Initialize the list of clusters to be a single cluster that contains all points
  2. **Repeat**
  3. Select a cluster from the list of clusters
  4. **For** $i = 1$ to T **do**
  5. Bisect the selected cluster using basic K-means
  6. **End**
  7. Add the two clusters from the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

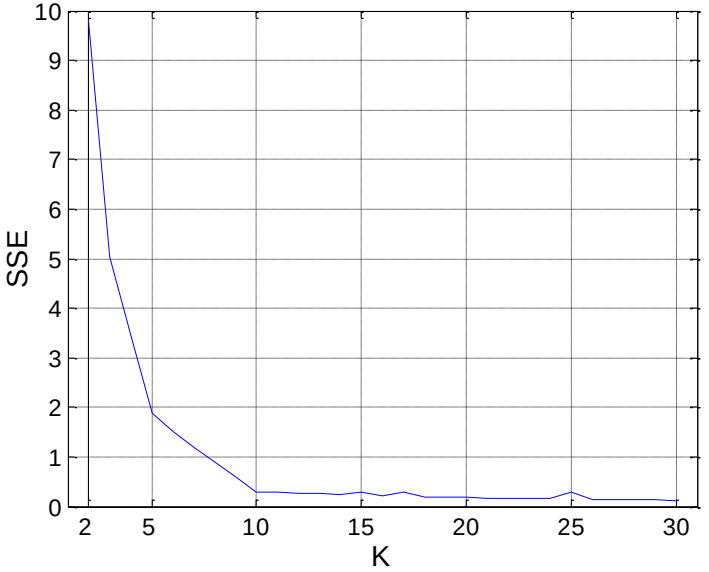
Bisecting *K*-means



Estimation of *K*

- SSE can be used to estimate the number of clusters

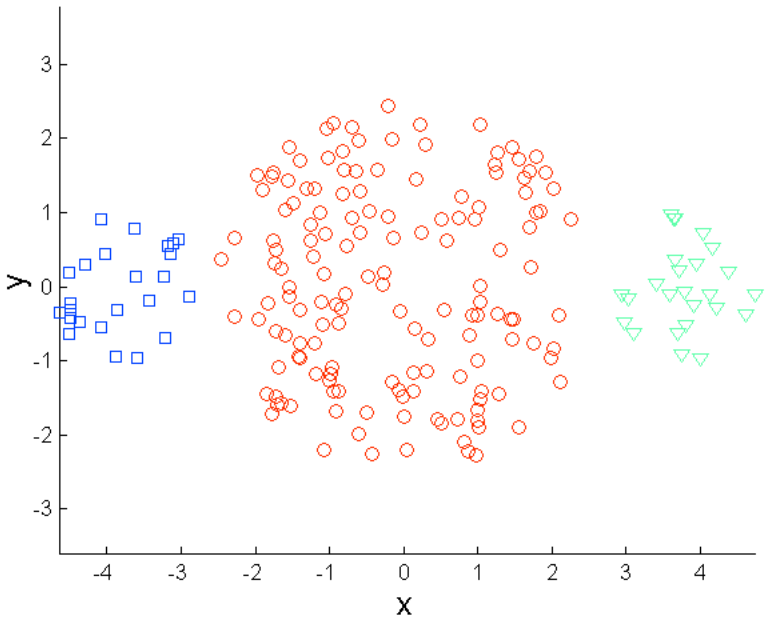




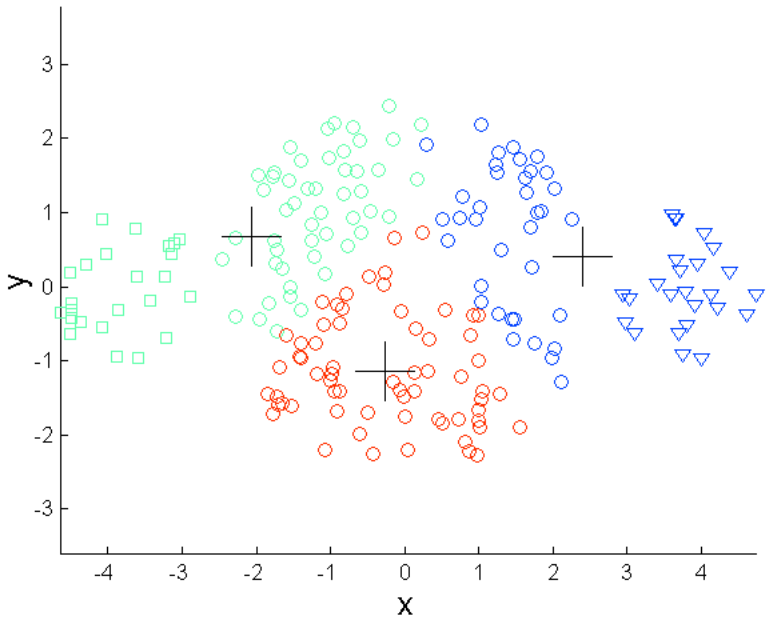
Limitations of *K*-means

- *K*-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes
- *K*-means has problems when the data contains outliers

Limitations of *K*-means: Differing Sizes

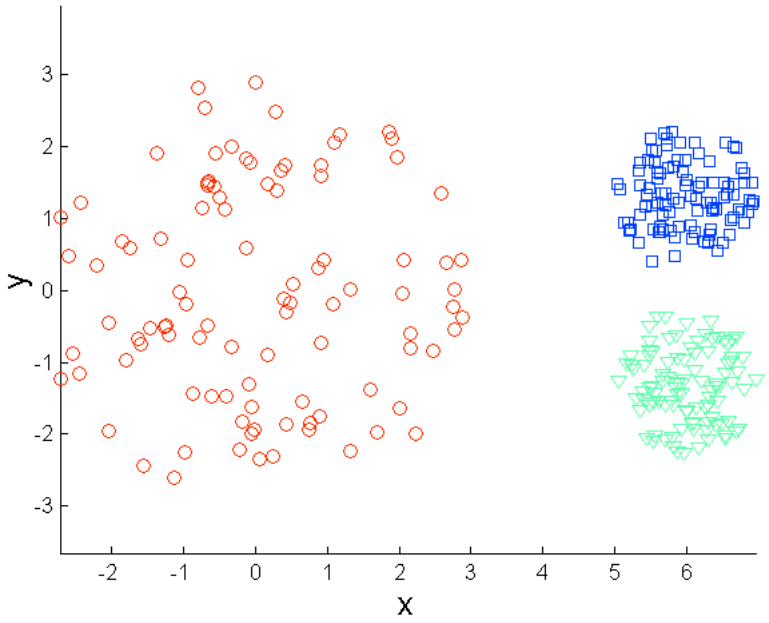


Original Points

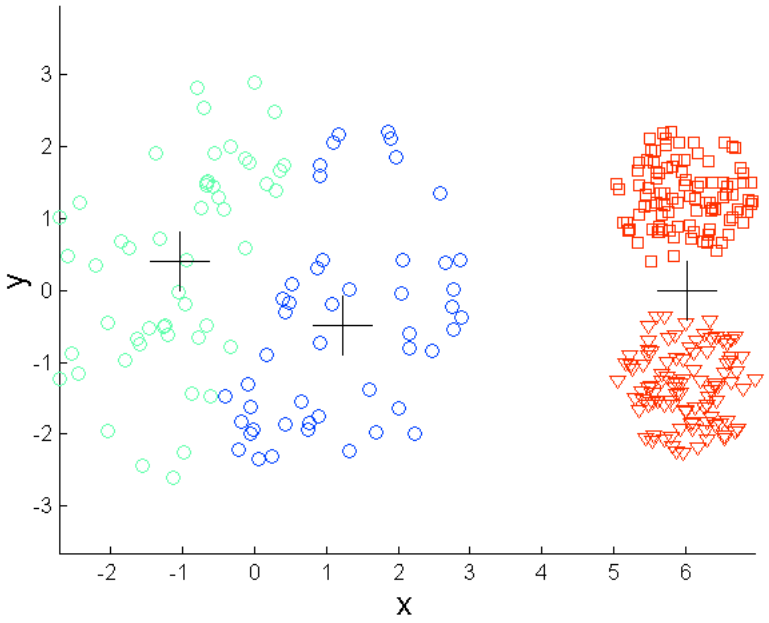


K-means (3 Clusters)

Limitations of *K*-means: Differing Density

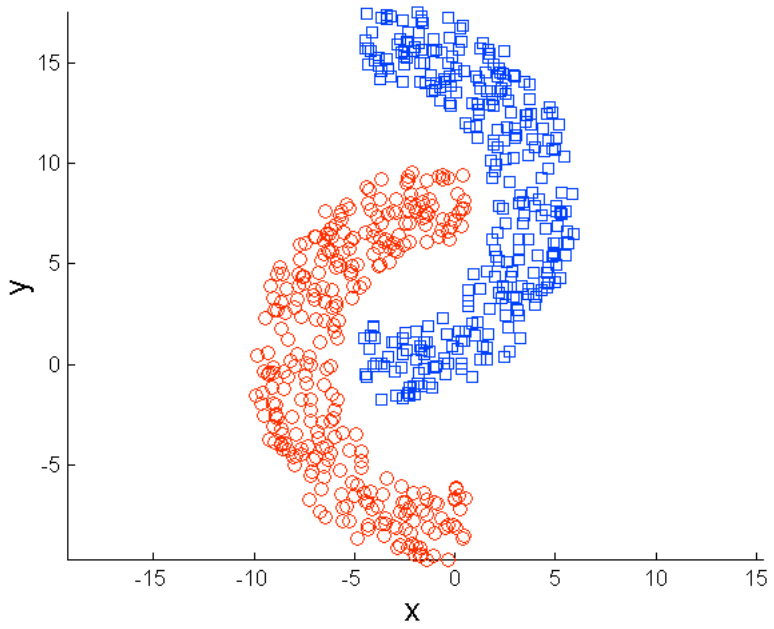


Original Points

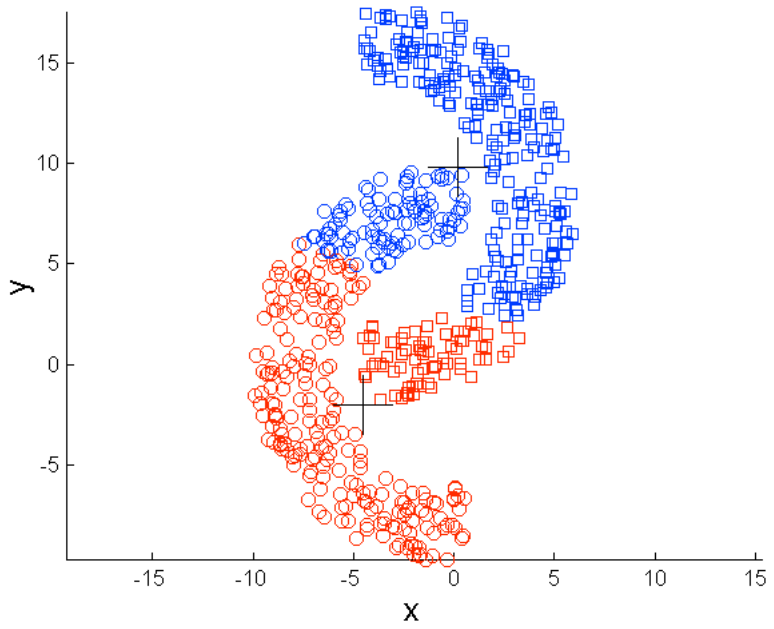


K-means (3 Clusters)

Limitations of *K*-means: Non-globular Shapes

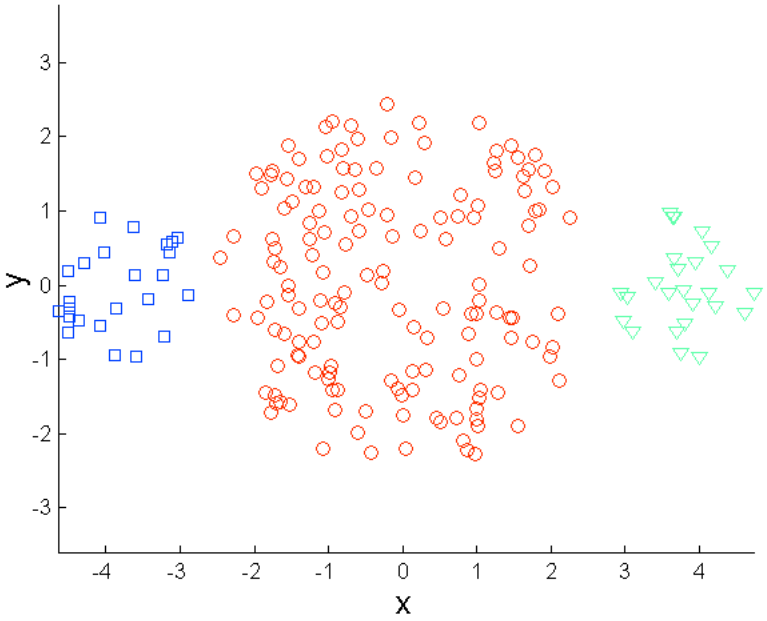


Original Points

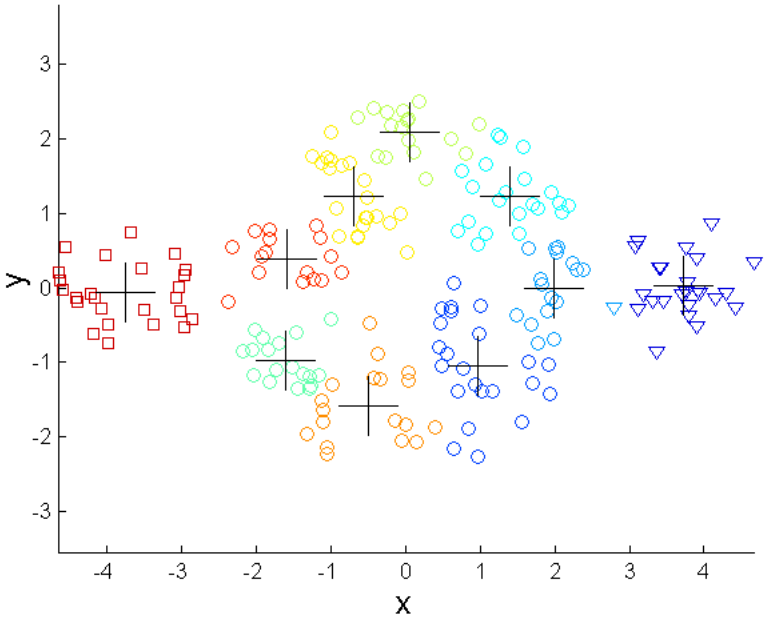


K-means (2 Clusters)

Solution



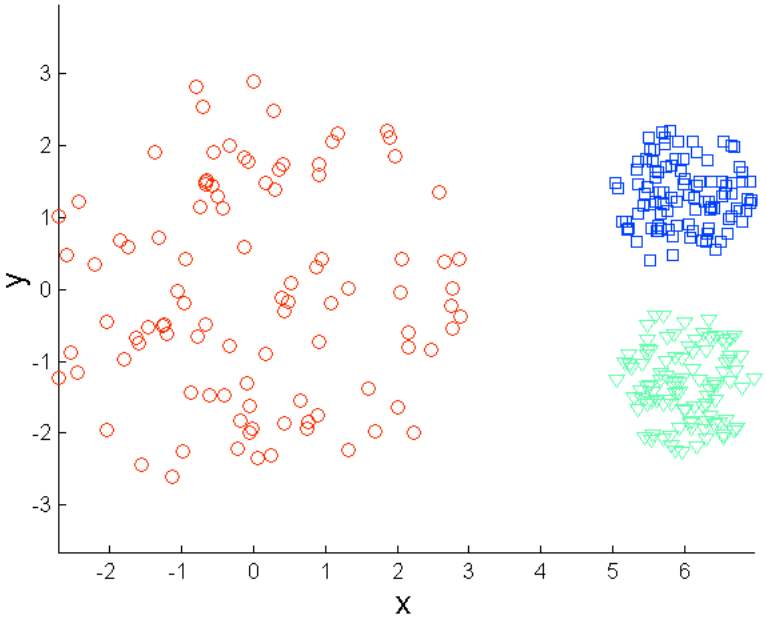
Original Points



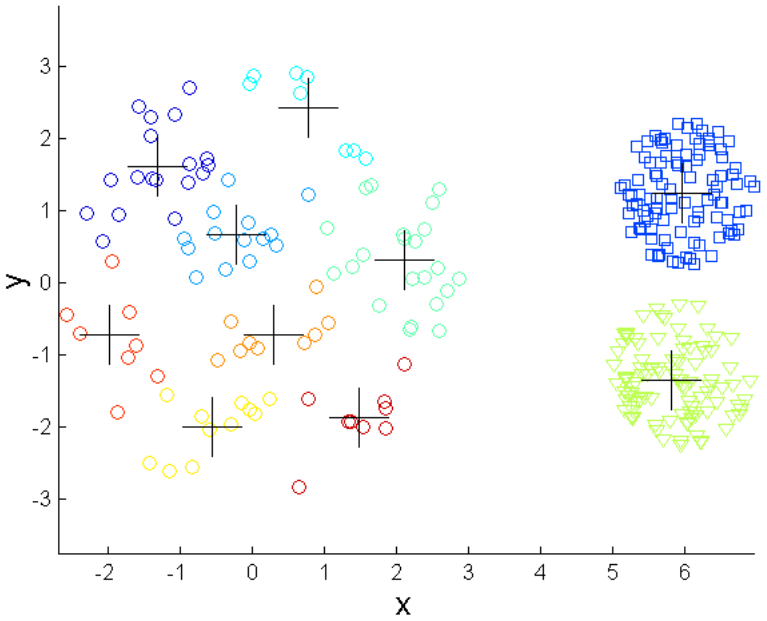
K-means Clusters

One solution is to use many clusters. Find parts of clusters, but need to put together.

Solution (cont.)

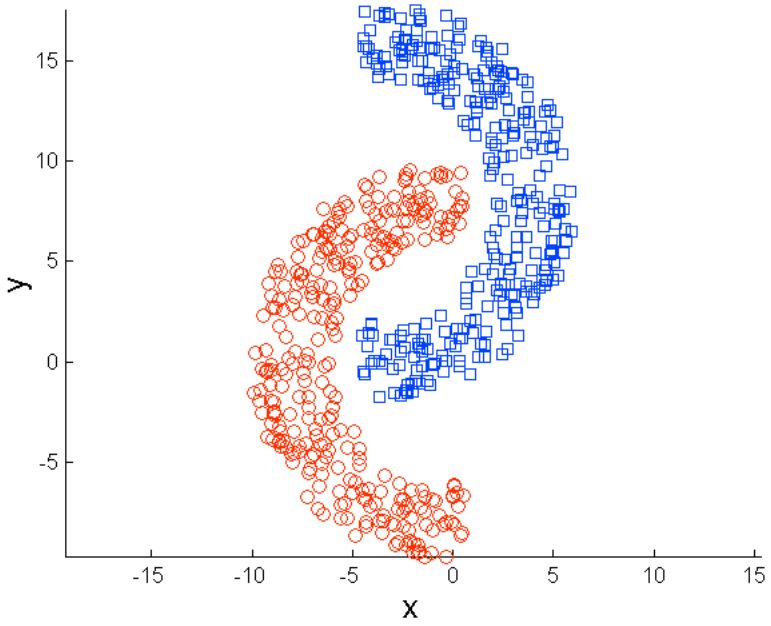


Original Points

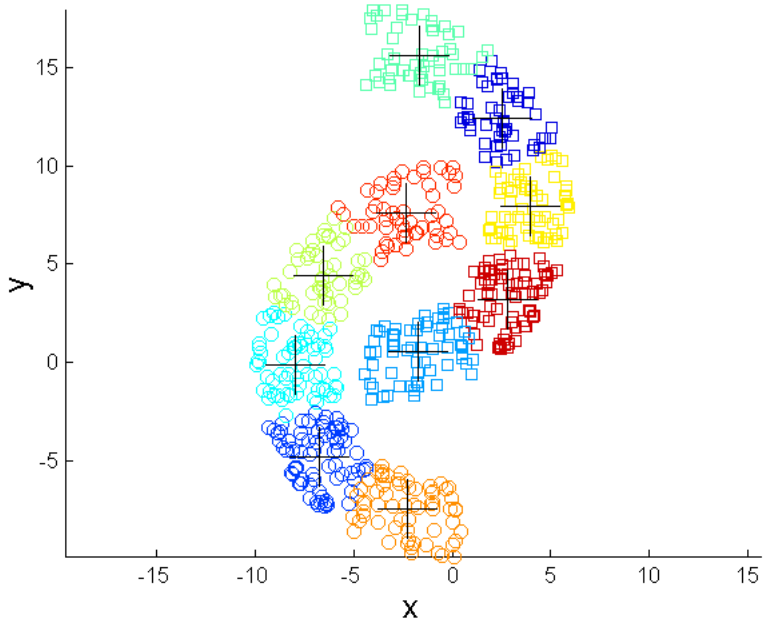


K-means Clusters

Solution (cont.)



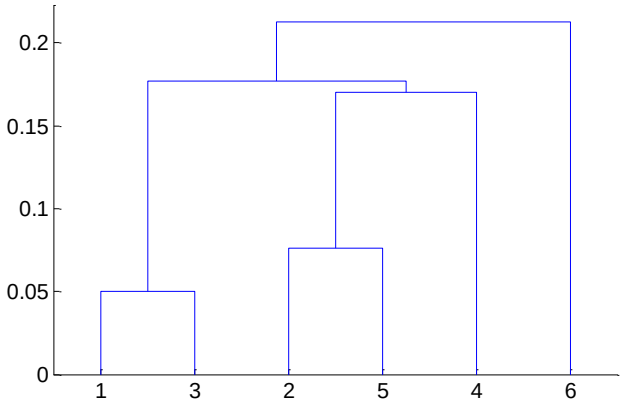
Original Points



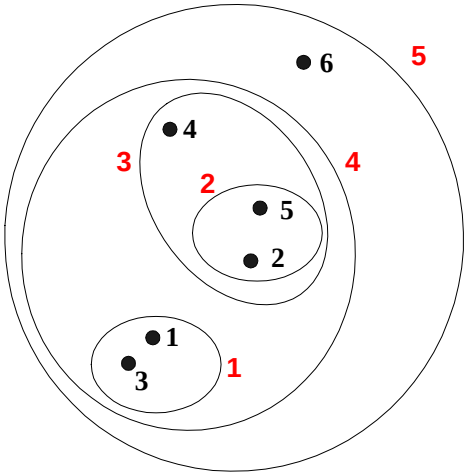
K-means Clusters

Hierarchical Clustering

- Produces 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



Dendrogram



Nested cluster diagram

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 dendogram at the proper level
- They may correspond to meaningful taxonomies
  - Examples in document organization, biological sciences

Hierarchical Clustering (cont.)

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

Agglomerative Clustering Algorithm

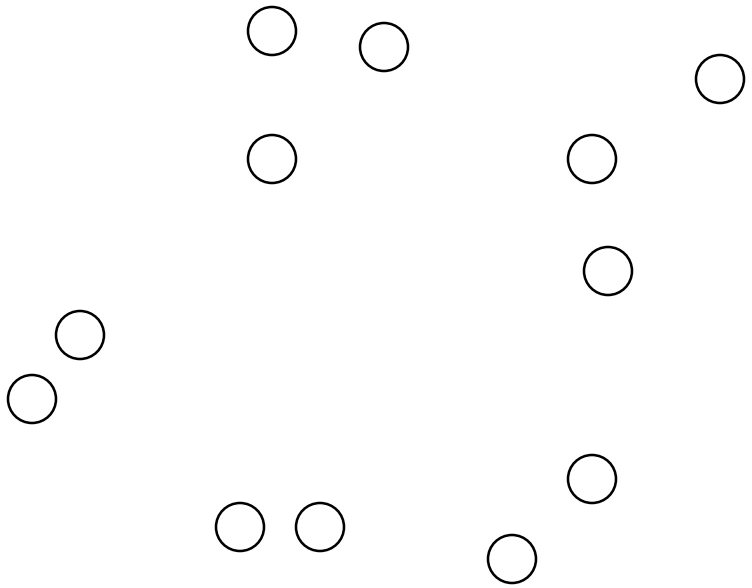
- Basic algorithm:
  1. Compute the proximity matrix
  2. Let each data instance be a cluster
  3. **Repeat**
  4. Merge the two closest clusters
  5. Update the proximity matrix
  6. **Until** only a single cluster remains
- Key operation is to compute the proximity of two clusters
  - Different approaches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

distance or similarity

smallest distance or largest similarity

Starting Situation

- Start with clusters of individual instances and a proximity matrix



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

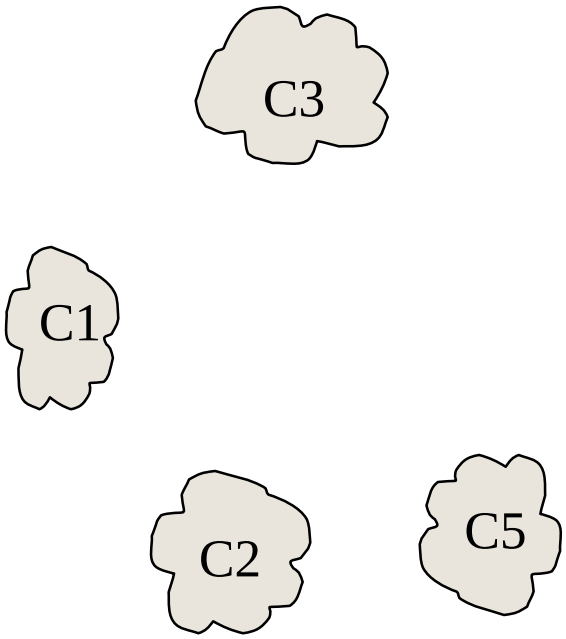
Proximity Matrix

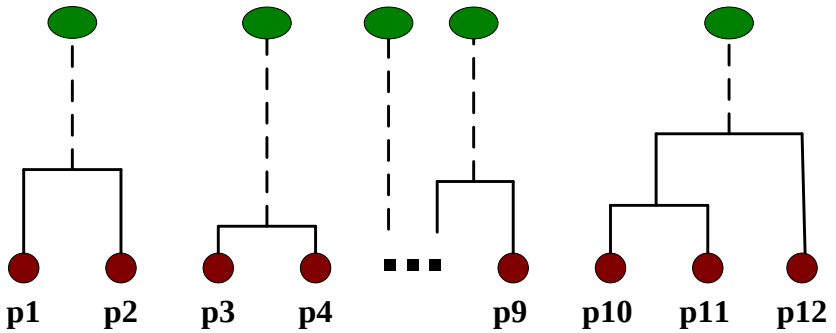
Intermediate Situation

- After some merging steps, we have some clusters

	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



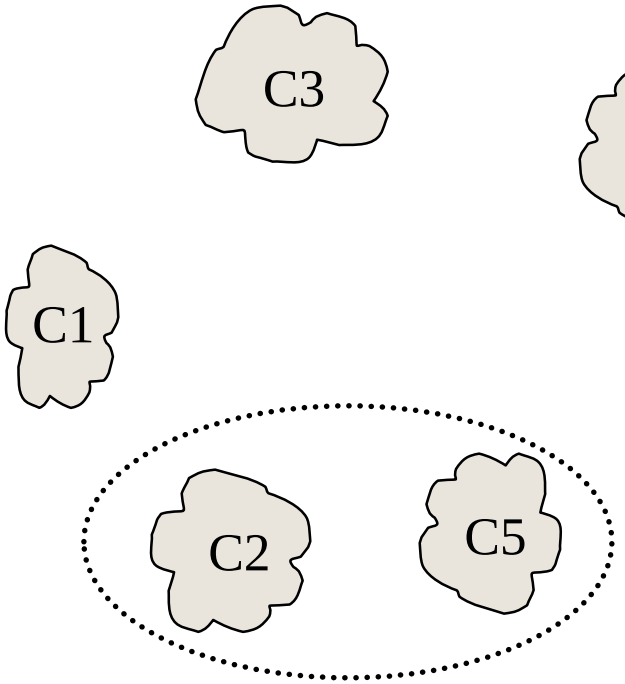


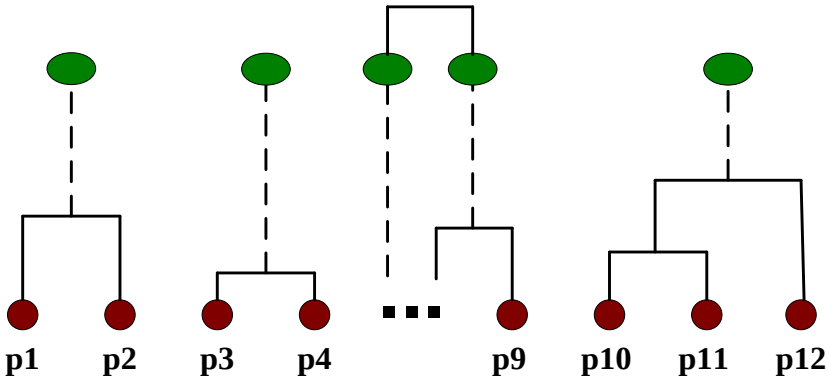
Intermediate Situation…

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

	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

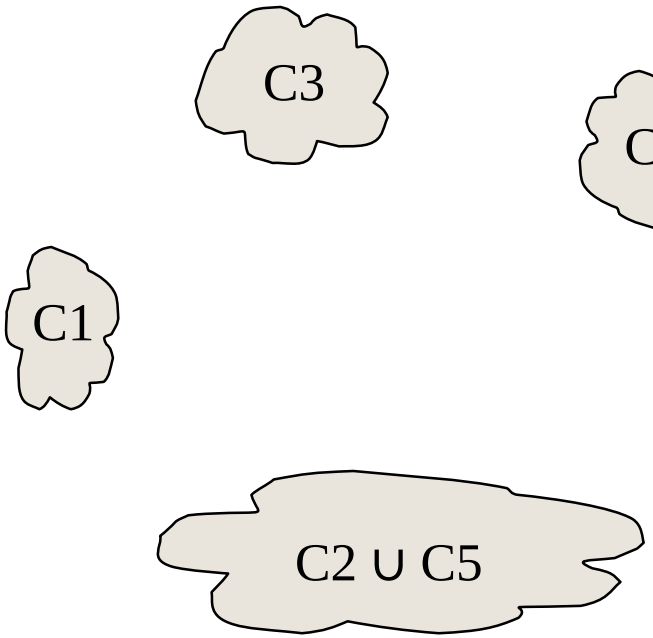
Proximity Matrix





After Merging

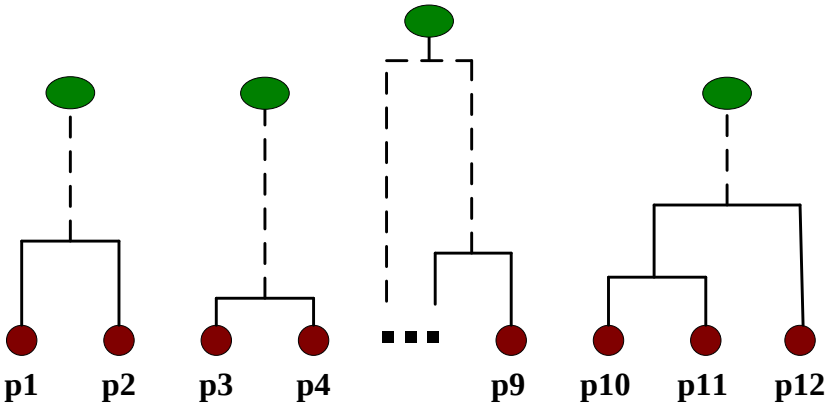
- The question is “How do we update the proximity matrix?”



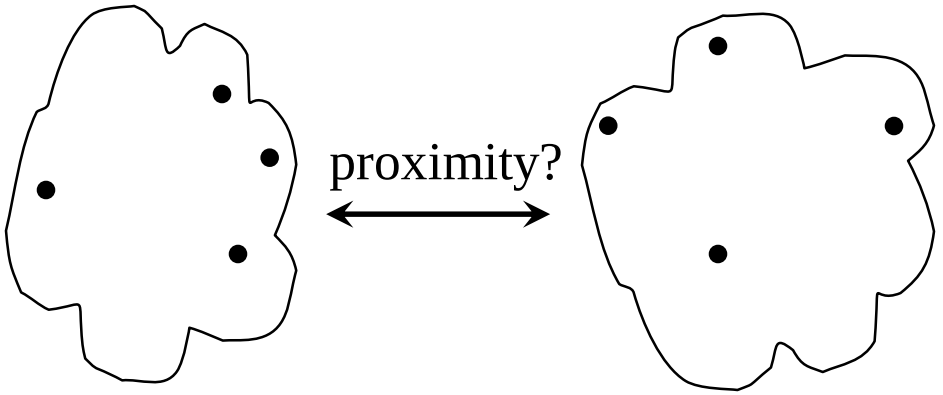
C2 ∪ C5

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

Proximity Matrix



Define Inter-Cluster Proximity

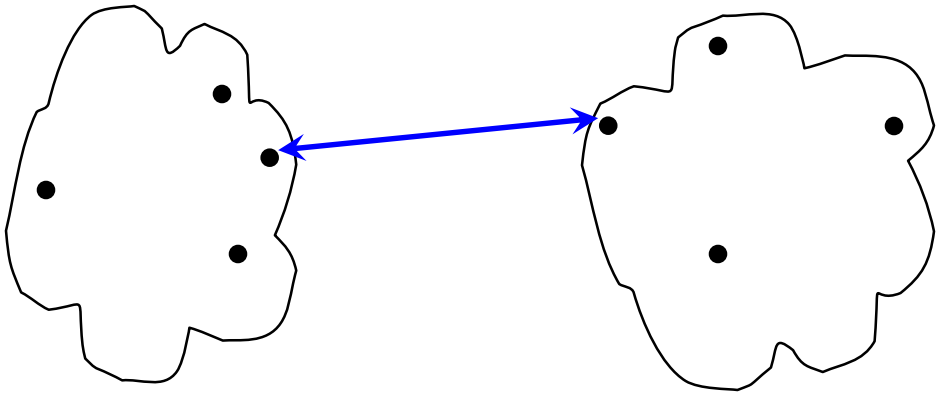


	P1	P2	P3	P4	P5	. . .
P1						
P2						
P3						
P4						
P5						
. . .						

Proximity Matrix

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

Inter-Cluster Similarity (I)



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

Proximity Matrix

MIN or Single Link:

Defines cluster proximity as

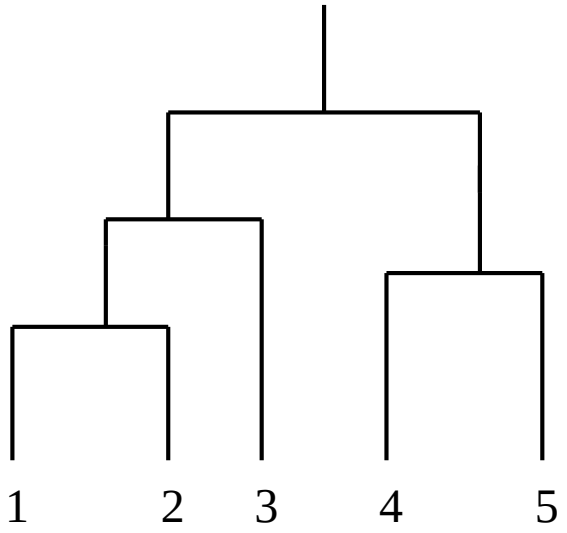
- the proximity between the closest two data points that are in different clusters
- or the shortest edge (single link) between two nodes in different subsets (using graph terms)

MIN or Single Link

- Similarity of two clusters is based on the two closest points (most similar) in the different clusters

	P1	P2	P3	P4	P5
P1	1.00	0.90	0.10	0.65	0.20
P2	0.90	1.00	0.70	0.60	0.50
P3	0.10	0.70	1.00	0.40	0.30
P4	0.65	0.60	0.40	1.00	0.80
P5	0.20	0.50	0.30	0.80	1.00

Similarity matrix

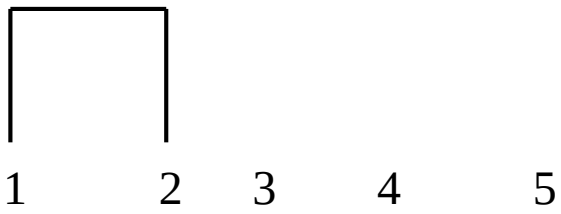


Clustering with MIN

Step 1: Merge the two closest clusters (largest similarity)

	P1	P2	P3	P4	P5
P1	1.00	0.90	0.10	0.65	0.20
P2	0.90	1.00	0.70	0.60	0.50
P3	0.10	0.70	1.00	0.40	0.30
P4	0.65	0.60	0.40	1.00	0.80
P5	0.20	0.50	0.30	0.80	1.00

Similarity matrix

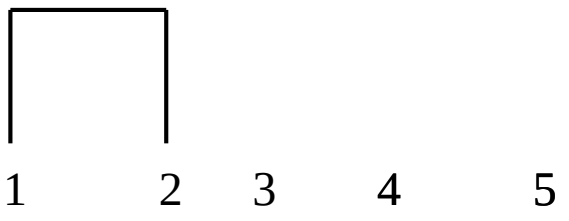


Clustering with MIN (cont.)

- Step 2: Update proximity matrix based on MIN: proximity of two clusters is based on the two closest points in different clusters (largest similarity)

	P1∪P2	P3	P4	P5
P1∪P2	1.00	0.70	0.65	0.50
P3	0.70	1.00	0.40	0.30
P4	0.65	0.40	1.00	0.80
P5	0.50	0.30	0.80	1.00

Similarity matrix

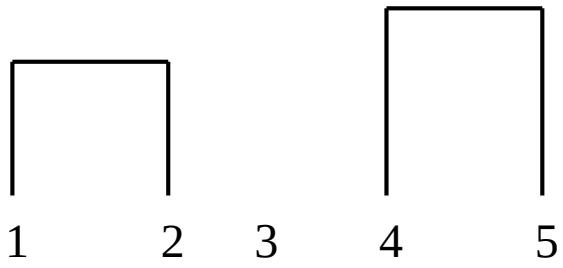


Clustering with MIN (cont.)

Step 1: Merge the two closest clusters (largest similarity)

	P1∪P2	P3	P4	P5
P1∪P2	1.00	0.70	0.65	0.50
P3	0.70	1.00	0.40	0.30
P4	0.65	0.40	1.00	0.80
P5	0.50	0.30	0.80	1.00

Similarity matrix

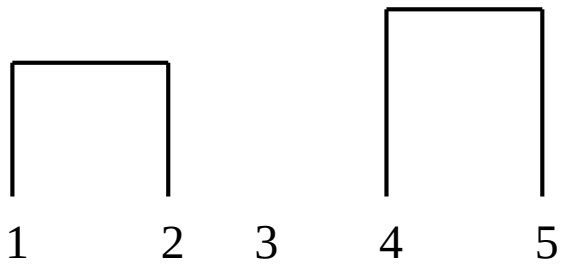


Clustering with MIN (cont.)

- Step 2: Update proximity matrix based on MIN: proximity of two clusters is based on the two closest points in different clusters (largest similarity)

	P1∪P2	P3	P4∪P5	
P1∪P2	1.00	0.70	0.65	
P3	0.70	1.00	0.40	
P4∪P5	0.65	0.40	1.00	
	0.50	0.30	0.80	1.00

Similarity matrix

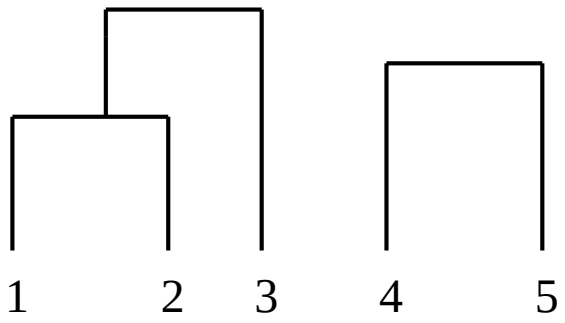


Clustering with MIN (cont.)

Step 1: Merge the two closest clusters (largest similarity)

	P1∪P2	P3	P4∪P5
P1∪P2	1.00	0.70	0.65
P3	0.70	1.00	0.40
P4∪P5	0.65	0.40	1.00

Similarity matrix

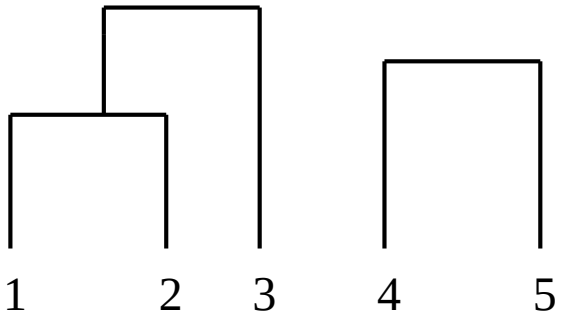


Clustering with MIN (cont.)

- Step 2: Update proximity matrix based on MIN: proximity of two clusters is based on the two closest points in different clusters (largest similarity)

	P1∪P2∪P3	P4∪P5
P1∪P2∪P3	1.00	0.65
P4∪P5	0.65	1.00

Similarity matrix

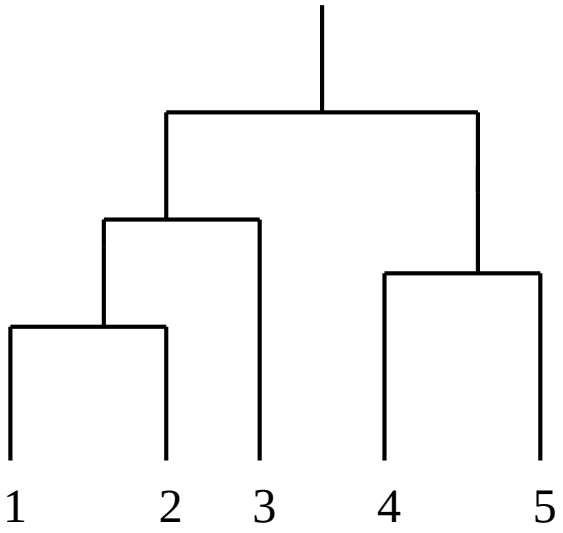


Clustering with MIN (cont.)

Step 1: Merge the two closest clusters (largest similarity)

	P1∪P2∪P3	P4∪P5
P1∪P2∪P3	1.00	0.65
P4∪P5	0.65	1.00

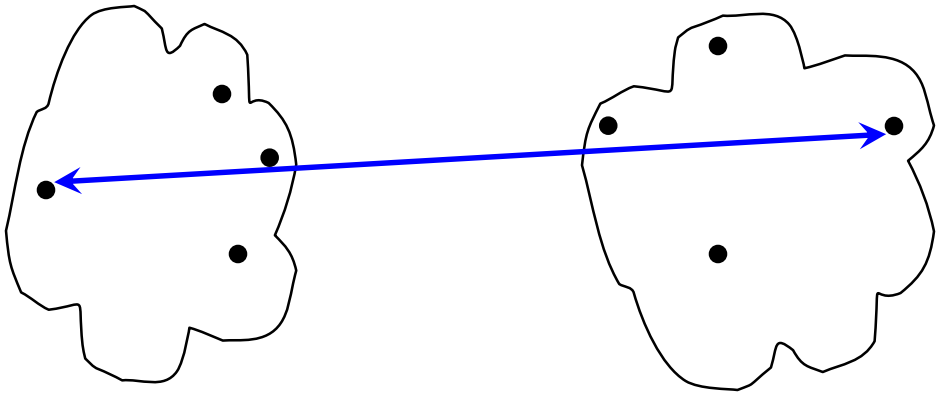
Similarity matrix



What about the proximity matrix is a distance matrix?

Tutorial

Inter-Cluster Similarity II



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

Proximity Matrix

MAX or Complete Link:

Defines cluster proximity as

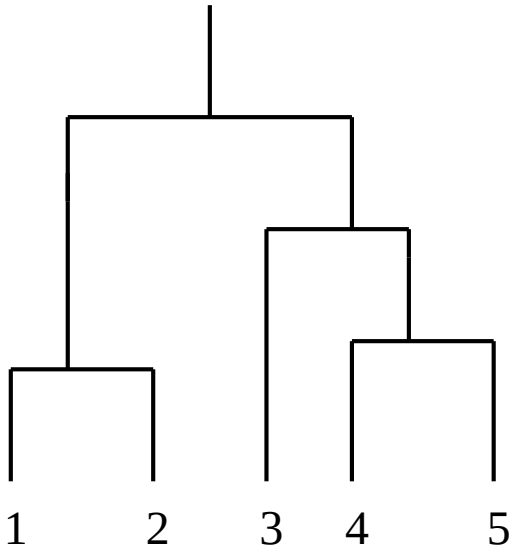
- the proximity between the farthest two points that are in different clusters
- or the longest edge (complete link) between two nodes in different subsets (using graph terms)

MAX or Complete Link

- Similarity of two clusters is based on the two least similar (most distant) points in the different clusters

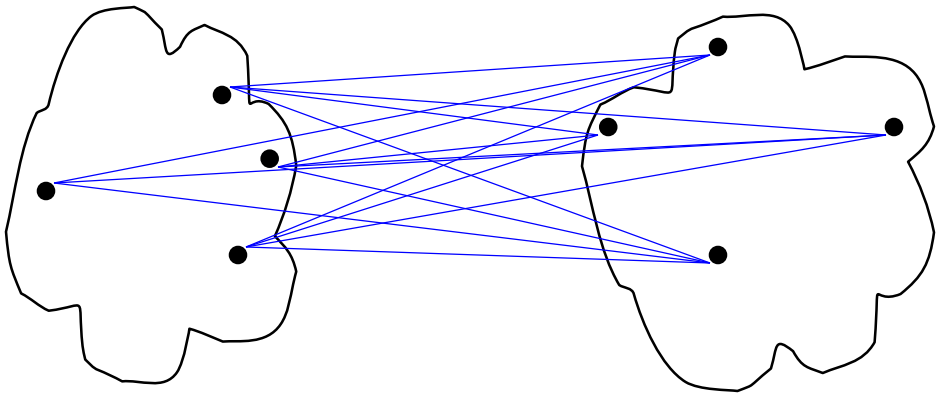
	P1	P2	P3	P4	P5
P1	1.00	0.90	0.10	0.65	0.20
P2	0.90	1.00	0.70	0.60	0.50
P3	0.10	0.70	1.00	0.40	0.30
P4	0.65	0.60	0.40	1.00	0.80
P5	0.20	0.50	0.30	0.80	1.00

Similarity matrix



Tutorial

Inter-Cluster Similarity (III)



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

Proximity Matrix

Group Average:

Defines cluster proximity as

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

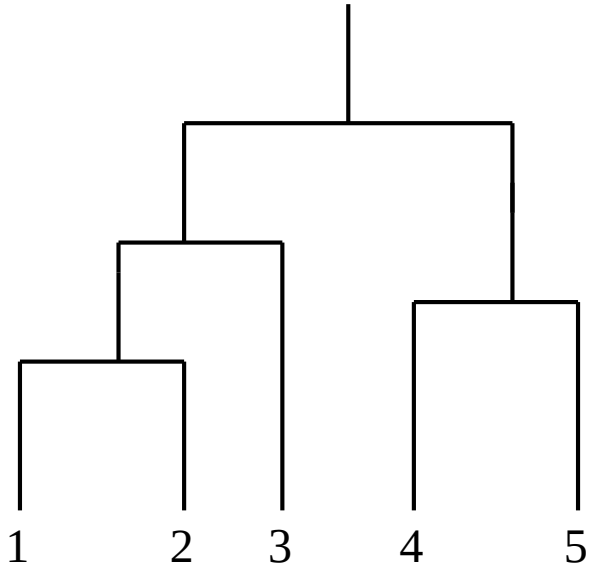
Group Average

- 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

	P1	P2	P3	P4	P5
P1	1.00	0.90	0.10	0.65	0.20
P2	0.90	1.00	0.70	0.60	0.50
P3	0.10	0.70	1.00	0.40	0.30
P4	0.65	0.60	0.40	1.00	0.80
P5	0.20	0.50	0.30	0.80	1.00



Agglomerative Clustering: Limitations

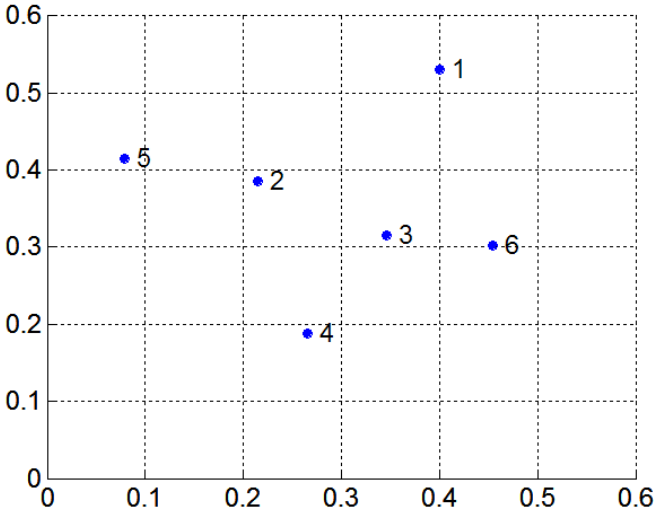
- Once a decision is made to combine two clusters, it cannot be undone
- No objective function is directly minimized
- Different schemes have problems with one or more of the following:
  - Sensitivity to noise and outliers
  - Difficulty in handling different-sized clusters

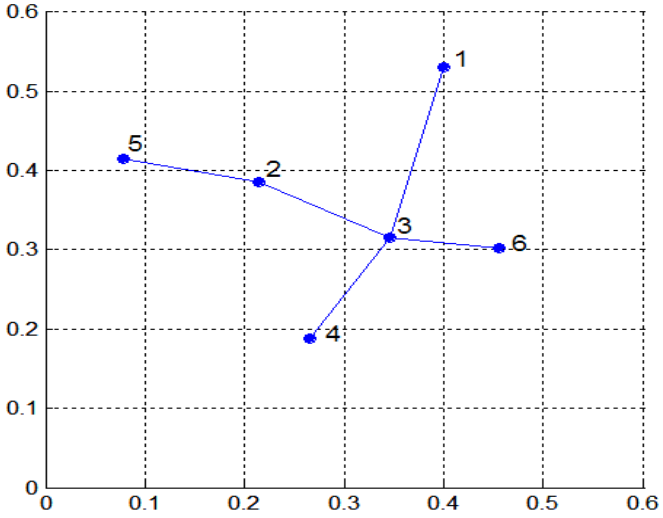
Divisive Hierarchical Clustering

- Basic algorithm:
  1. Generate a minimum spanning tree to connect all data instances as a single cluster
  2. **Repeat**
  3. Create a new cluster by breaking the link corresponding to the largest distance (smallest similarity)
  4. **Until** only singleton clusters remain

Divisive Hierarchical Clustering (cont.)

- Minimum Spanning Tree (MST)
  - Start with a tree that consists of any point
  - 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 $\boldsymbol{x}_j$

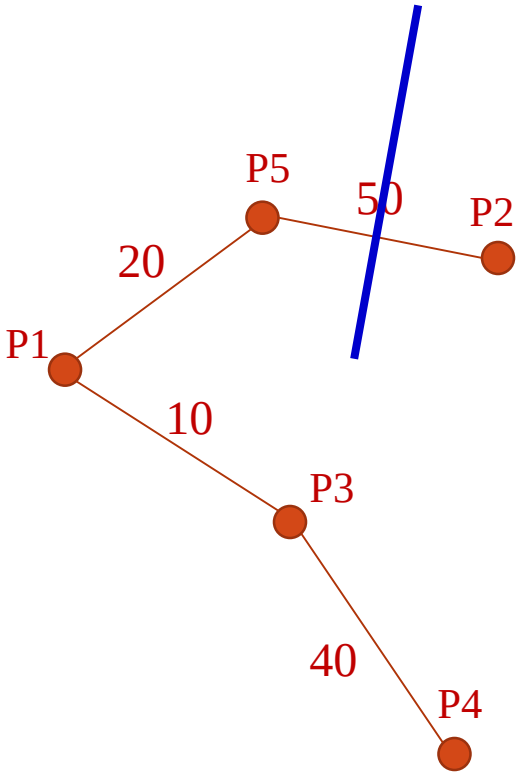




An Example of DHC

The distance matrix between 5 points

	P1	P2	P3	P4	P5
P1	0	90	10	65	20
P2	90	0	70	60	50
P3	10	70	0	40	30
P4	65	60	40	0	80
P5	20	50	30	80	0



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

Thank you!