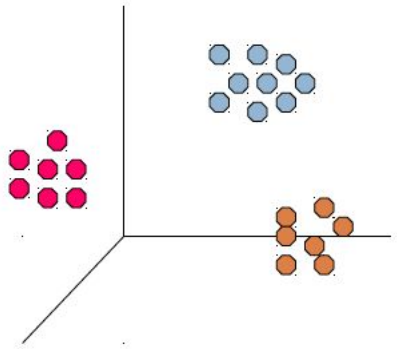
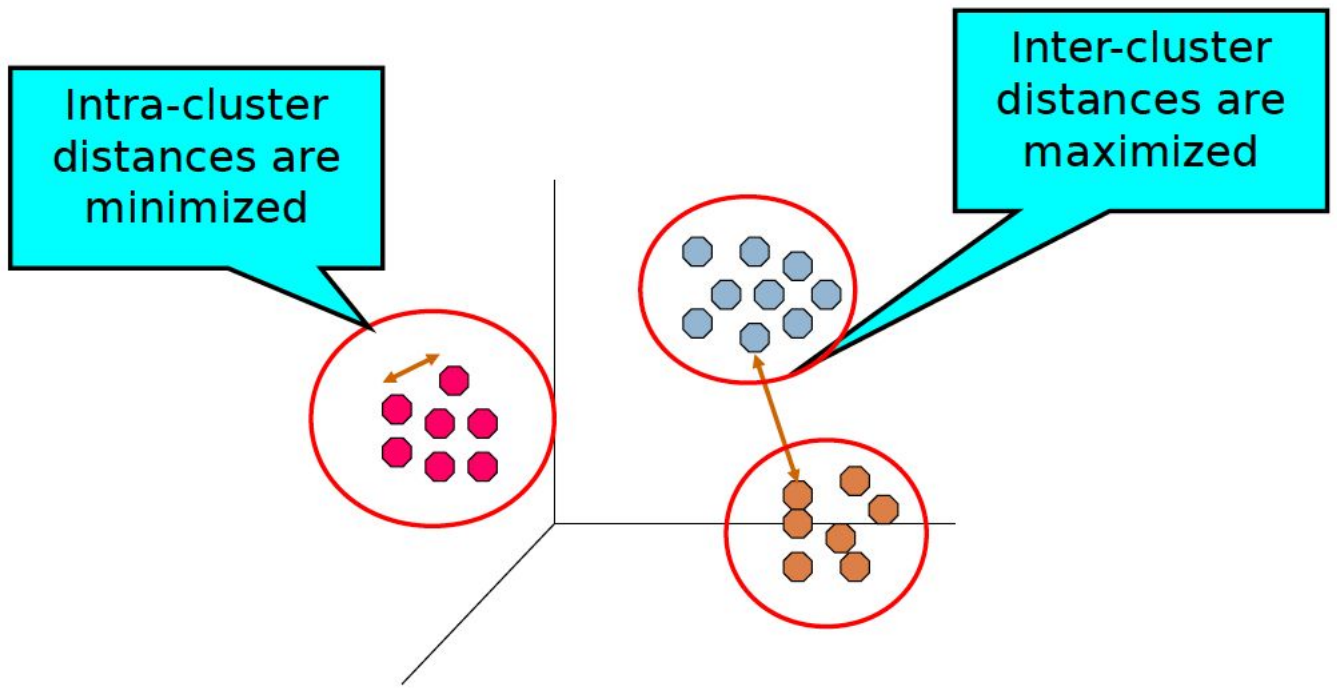
Clustering

Unsupervised Learning Algorithms

What is Cluster Analysis?

- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups.




Intra-cluster distances are minimized
Inter-cluster distances are maximized

Applications of Clustering

- It’s a key intermediate step for other machine learning tasks
  - Outlier detection: Outliers are those “far away” from any cluster.
- Data summarization, compression, Reduction
- Collaborative filtering, recommender systems
  - Find like minded users or similar products
- Multimedia data analysis, biological data analysis, social network analysis
  - Clustering images,

What is not Cluster Analysis?

- Clustering is not Classification
  - Classification is supervised learning algorithm, where training dataset has class label information
  - Clustering is unsupervised learning algorithm, where training data does not contain label information.

Notion of a Cluster can be Ambiguous

How many clusters?

Notion of a Cluster can be Ambiguous

How many clusters?

Six Clusters

Notion of a Cluster can be Ambiguous

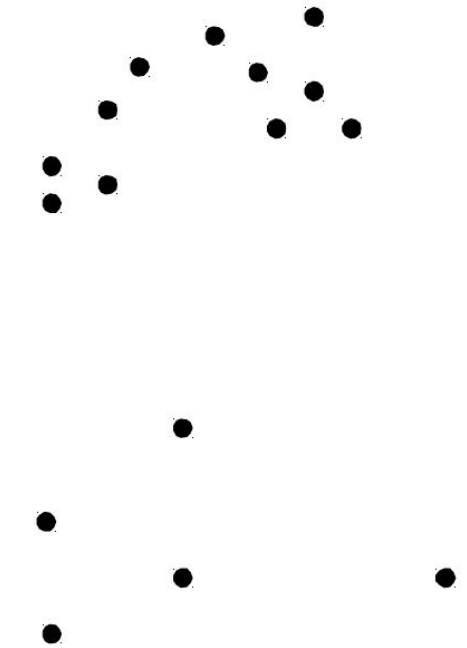
Two Clusters

Four Clusters

Types of Clustering

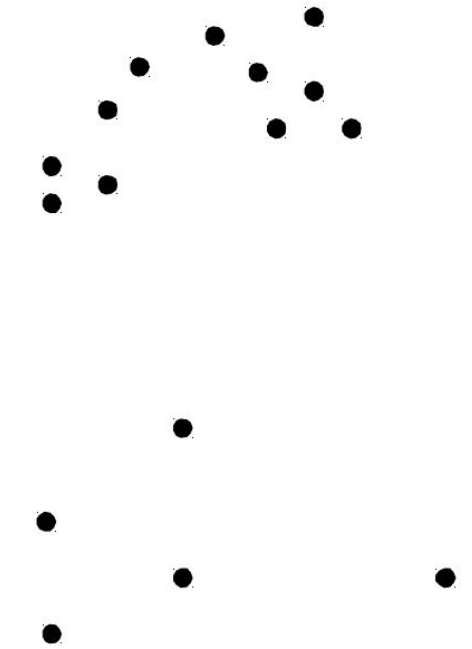
- Partitional Clustering
  - A division data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset
- Hierarchical clustering
  - A set of nested clusters organized as a hierarchical tree

Partitional Clustering – non overlapping set of clusters

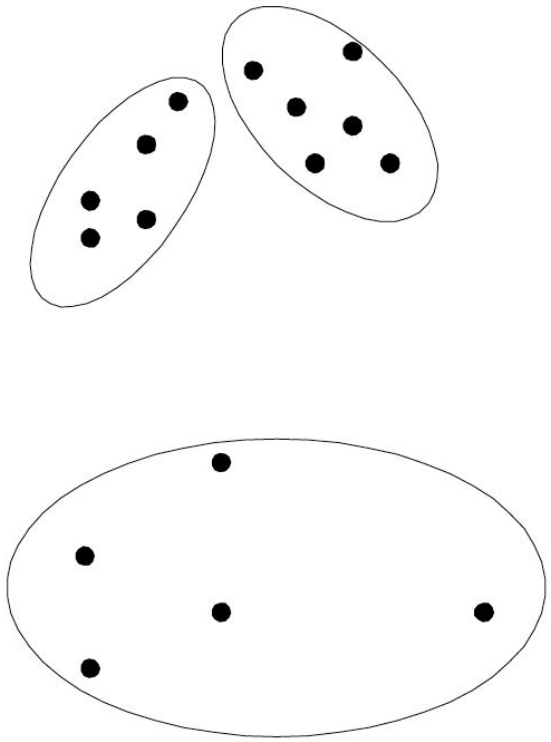


Original Points

Partitional Clustering – non overlapping set of clusters

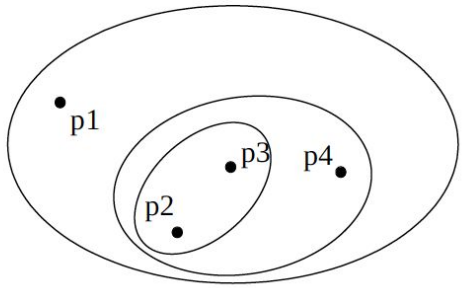


Original Points

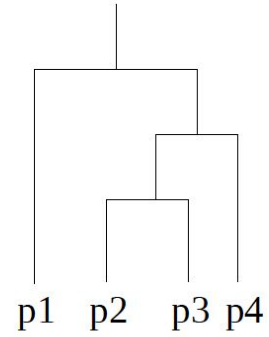


A Partitional Clustering

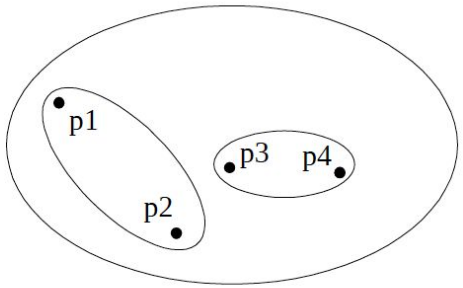
Hierarchical Clustering – may have overlapping set of clusters



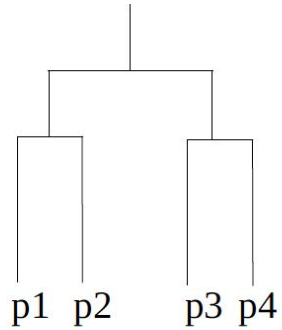
Traditional Hierarchical Clustering



Traditional Dendrogram



Non-traditional Hierarchical Clustering



Non-traditional Dendrogram

Types of Attributes by Measurement Scale

- Categorical (Qualitative) Attribute
  - Nominal
    - Examples: ID numbers, eye color, zip codes
  - Ordinal
    - Examples: rankings (e.g., taste of potato chips on a scale from 1-10), grades, height in {tall, medium, short}
- Numeric (Quantitative) Attribute
  - Interval
    - Examples: calendar dates, temperatures in Celsius or Fahrenheit.
  - Ratio
    - Examples: temperature in Kelvin, length, time, counts

Type of Attributes by Number of Values

- Discrete Attribute
  - Has only a finite or countably infinite set of values
  - Examples: zip codes, counts, or the set of words in a collection of documents
  - Often represented as integer variables.
  - Note: binary attributes are a special case of discrete attributes
- Continuous Attribute
  - Has real numbers as attribute values
  - Examples: temperature, height, or weight.
  - Practically, real values can only be measured and represented using a finite number of digits.
  - Continuous attributes are typically represented as floating-point variables.

Types of Attributes

- By measure scale
  - Categorical (Qualitative) Attribute
    - Nominal
    - Ordinal
  - Numeric (Quantitative) Attribute
    - Interval
    - Ratio
- By number of values
  - Discrete Attribute
  - Continuous Attribute

Similarity and Dissimilarity

- Similarity
  - Numerical measure of how alike two data objects are.
  - Is higher when objects are more alike.
  - Often falls in the range [0,1]
- Dissimilarity
  - Numerical measure of how different are two data objects
  - Lower when objects are more alike
  - Minimum dissimilarity is often 0
  - Upper limit varies

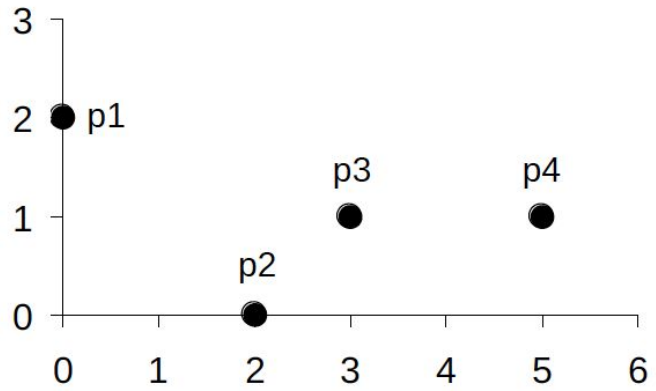
Dissimilarity between Data Objects: Euclidean Distance

- Euclidean Distance

$$dist = \sqrt{\sum_{k=1}^{n} (p_k - q_k)^2}$$

- Where n is the number of dimensions (attributes) and p_k and q_k are, respectively, the k^{th} attributes (components) or data objects p and q.
- Standardization is necessary, if scales differ.

Euclidean Distance



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Distance Matrix

Similarity Between Binary Vectors

- Common situation is that objects, p and q, have only binary attributes
- Compute similarities using the following quantities
  - M01 = the number of attributes where p was 0 and q was 1
  - M10 = the number of attributes where p was 1 and q was 0
  - M00 = the number of attributes where p was 0 and q was 0
  - M11 = the number of attributes where p was 1 and q was 1
- Simple Matching and Jaccard Coefficients
  - SMC = number of matches / number of attributes
    = (M11 + M00) / (M01 + M10 + M11 + M00)
  - J = number of 11 matches / number of not-both-zero attributes values
    = (M11) / (M01 + M10 + M11)

SMC example

p = 1 0 0 0 0 0 0 0 0 0

q = 0 0 0 0 0 0 1 0 0 1

M01 = 2 (the number of attributes where p was 0 and q was 1)

M10 = 1 (the number of attributes where p was 1 and q was 0)

M00 = 7 (the number of attributes where p was 0 and q was 0)

M11 = 0 (the number of attributes where p was 1 and q was 1)

SMC = (M11 + M00)/(M01 + M10 + M11 + M00) = (0+7) / (2+1+0+7) = 0.7

Clustering Algorithms

- K-means and its variants
- Hierarchical clustering
- EM

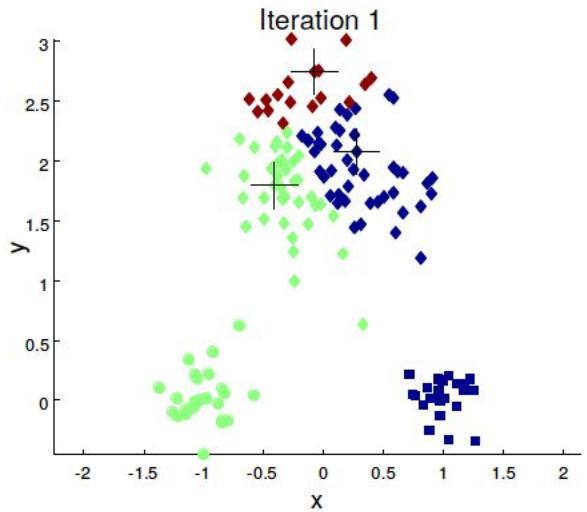
K-means Clustering

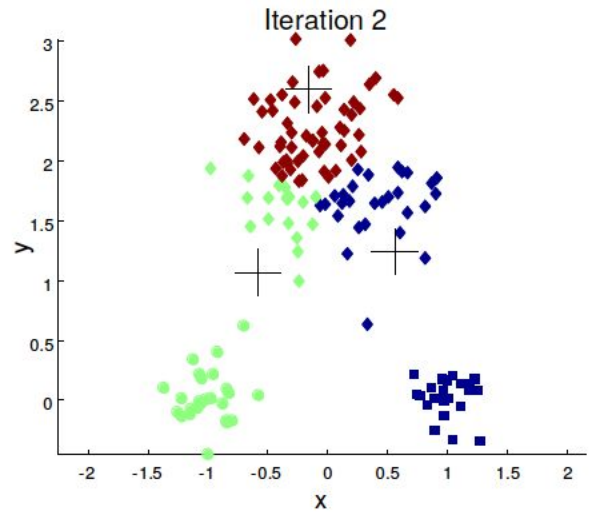
- Partitional clustering approach
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid
- Number of clusters, K, must be specified
- The basic algorithm is very simple

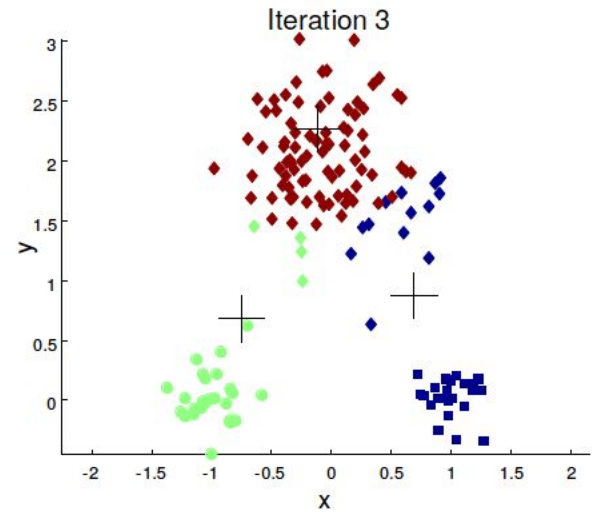
1: Select K points as the initial centroids.
2: **repeat**
3: Form K clusters by assigning all points to the closest centroid.
4: Recompute the centroid of each cluster.
5: **until** The centroids don't change

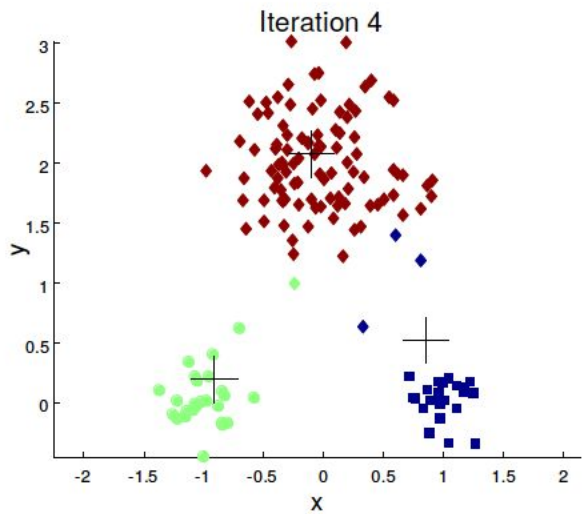
K-means Clustering – Details

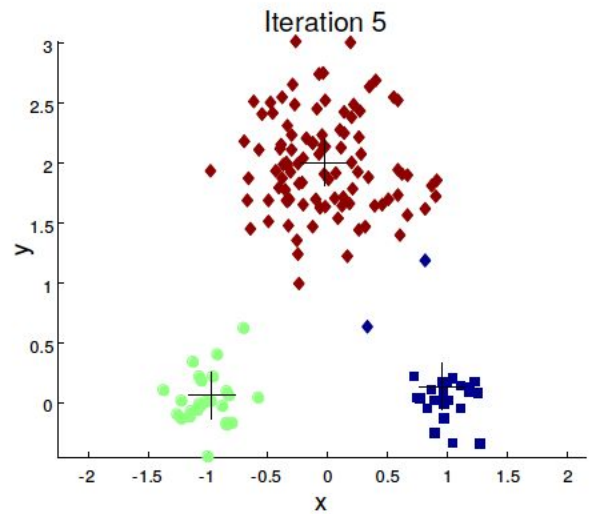
- Initial centroids are often chosen randomly.
  - Clusters produced vary from one run to another.
- The centroid mi is (typically) the mean of the points in the cluster.
- “Closeness” is measured by Euclidean distance, cosine similarity, etc.
- K-means will converge for common similarity measures mentioned above.
- Most of the convergence happens in the first few iterations.
  - Often the stopping condition is changed to ‘Until relatively few points change clusters’

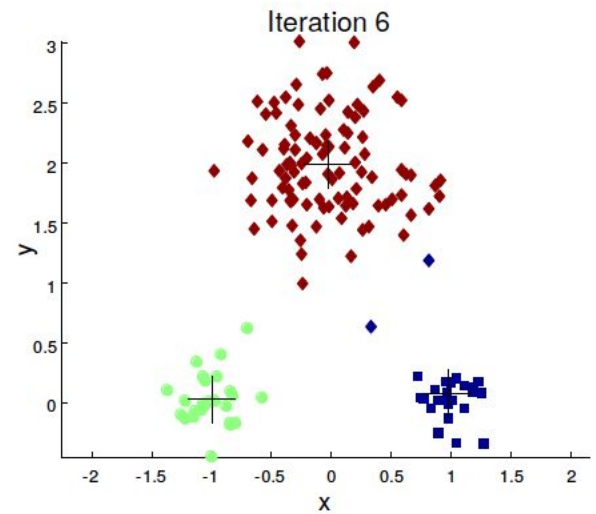

Iteration 1
y
x
3
2.5
2
1.5
1
0.5
0
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2


Iteration 2
y
x
3
2.5
2
1.5
1
0.5
0
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

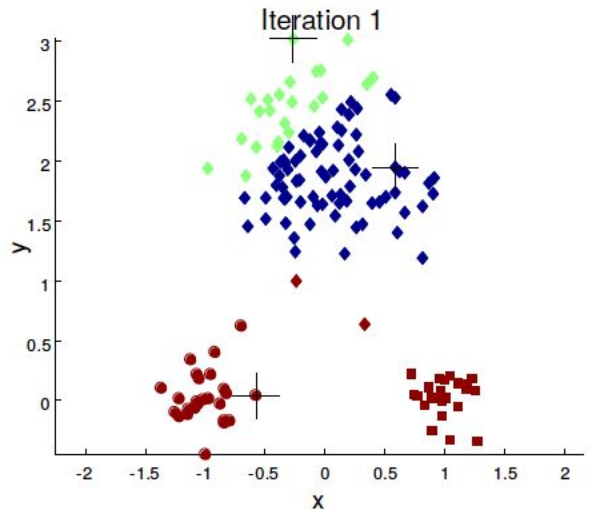

Iteration 3
y
x
3
2.5
2
1.5
1
0.5
0
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

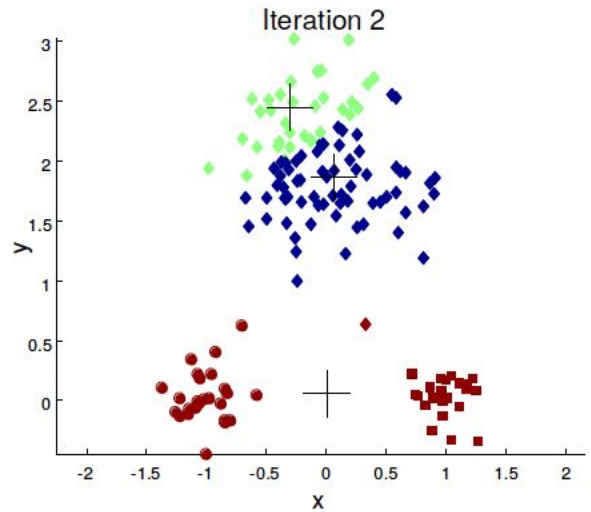

Iteration 4
y
x
3
2.5
2
1.5
1
0.5
0
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

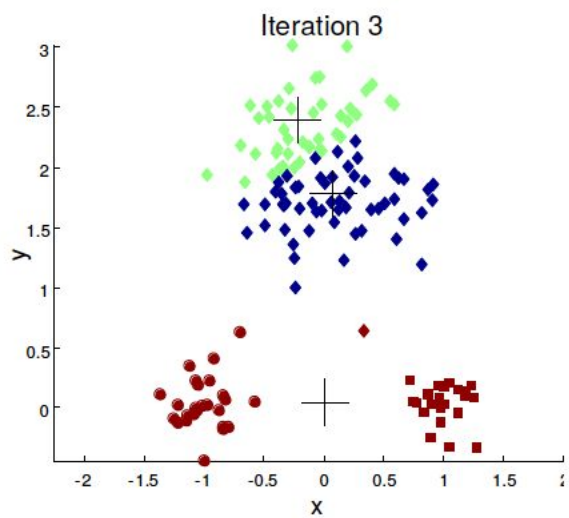

Iteration 5
y
x
3
2.5
2
1.5
1
0.5
0
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

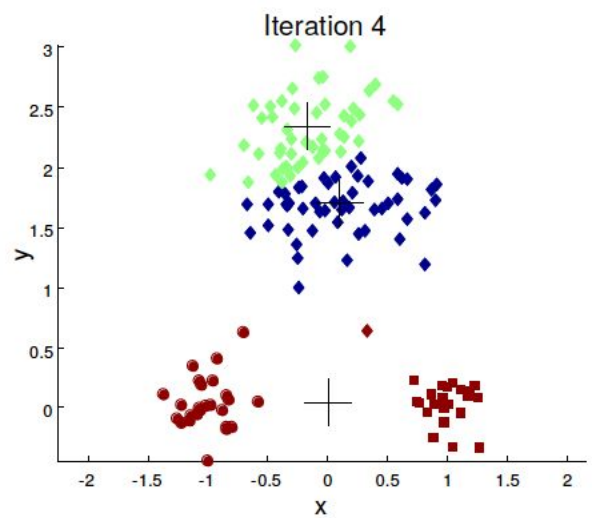

Iteration 6
y
x
3
2.5
2
1.5
1
0.5
0
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

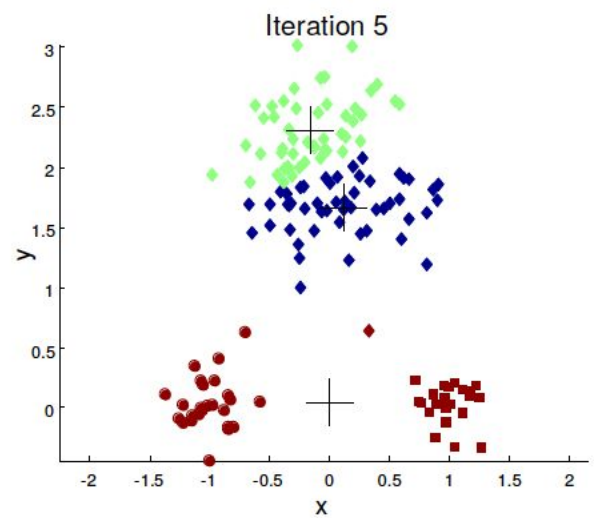
Importance of Choosing Initial Centroids


Iteration 1
x
y


Iteration 2
x
y


Iteration 3
x
y


Iteration 4
x
y


Iteration 5
x
y

Evaluating K-means Clusters

- Most common measure is Sum of Squared Error (SSE)
  - For each point, the error is the distance to the nearest cluster
  - To get SSE, we square these errors and sum them.

$$SSE = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

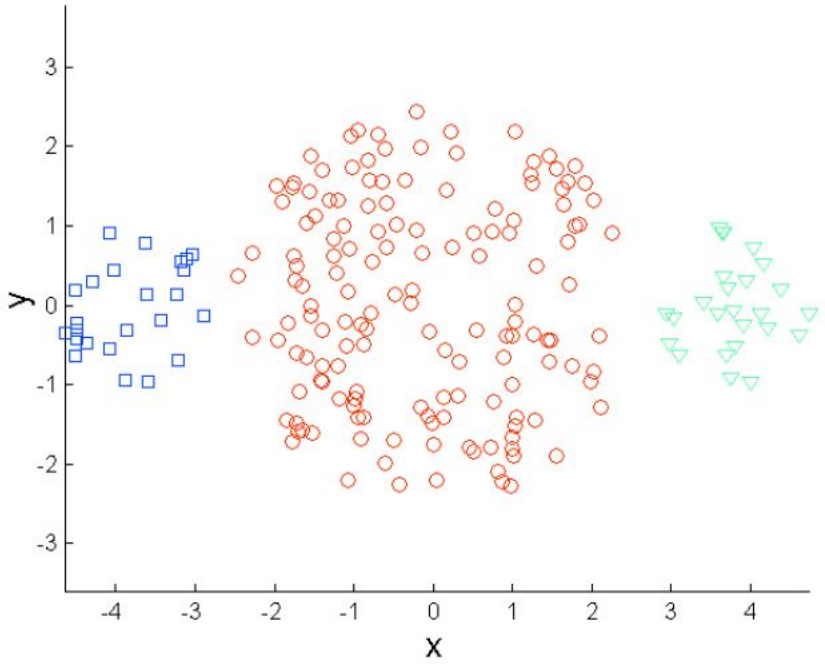
x is a data point in cluster C_i and mi is the centroid of cluster C_i

  - Given two clusterings, we can choose the one with the smallest error
    - the centroids of the clustering with smaller error are a better representation of points.

Limitations of K-means

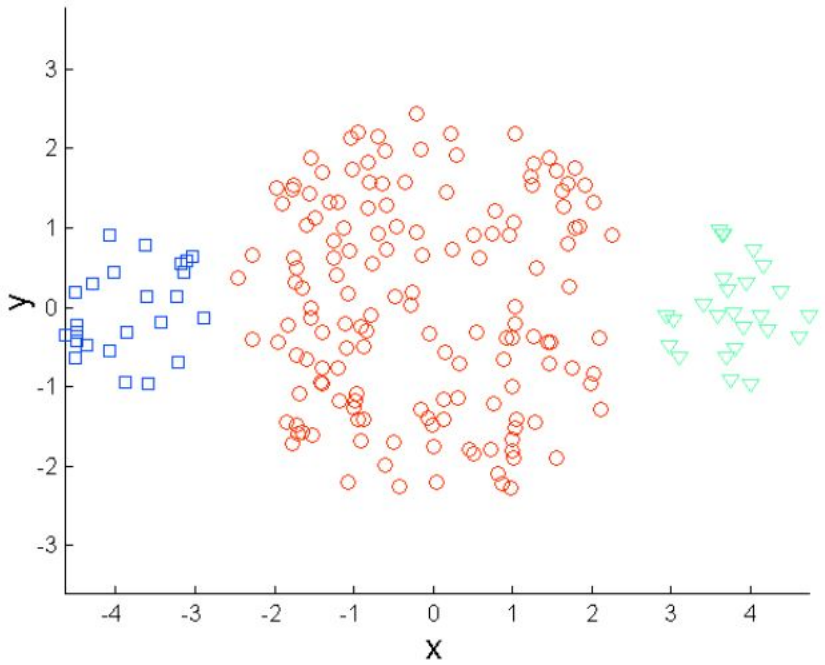
- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes
- K-means has problems when the data contains outliers.

Limitations of K-means: Differing Sizes

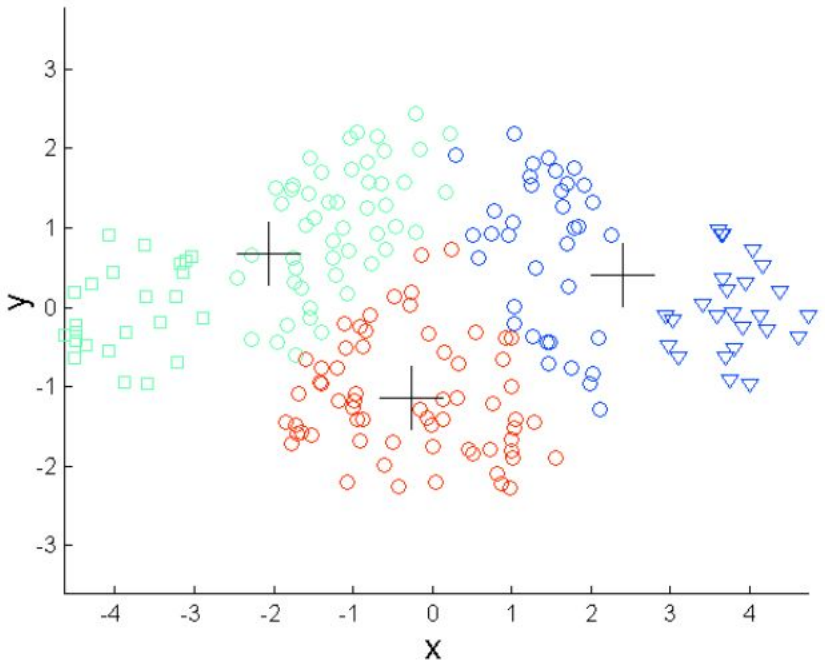


Original Points

Limitations of K-means: Differing Sizes

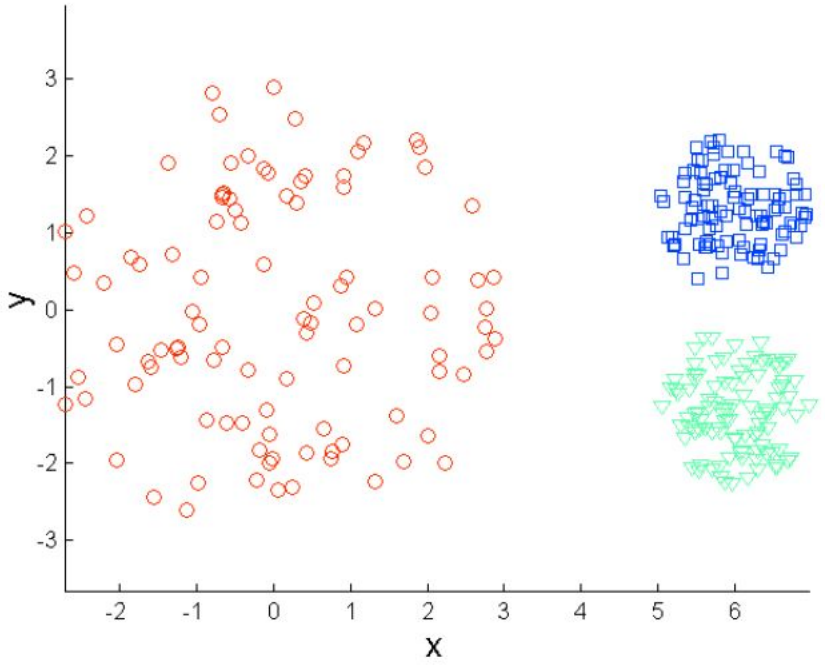


Original Points



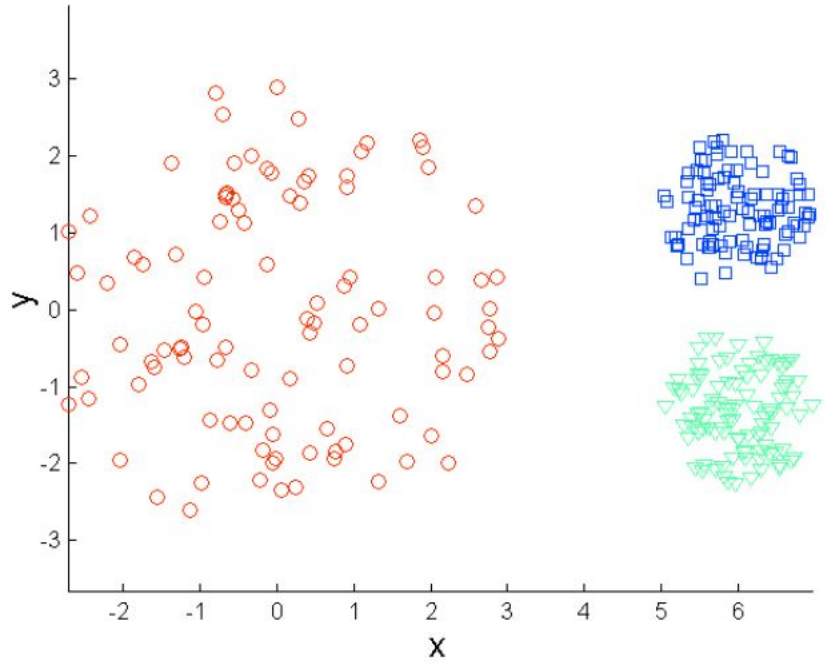
K-means (3 Clusters)

Limitations of K-means: Differing Density

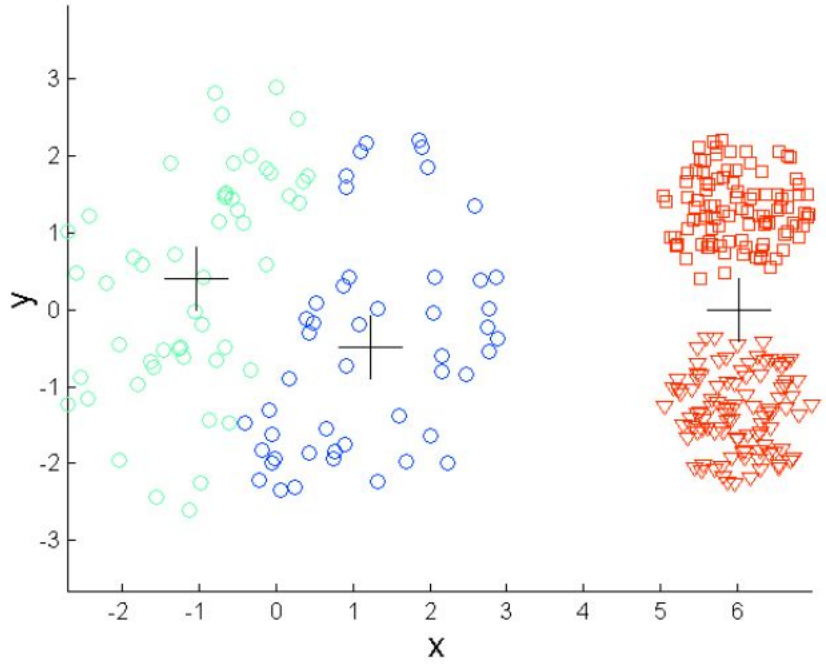


Original Points

Limitations of K-means: Differing Density

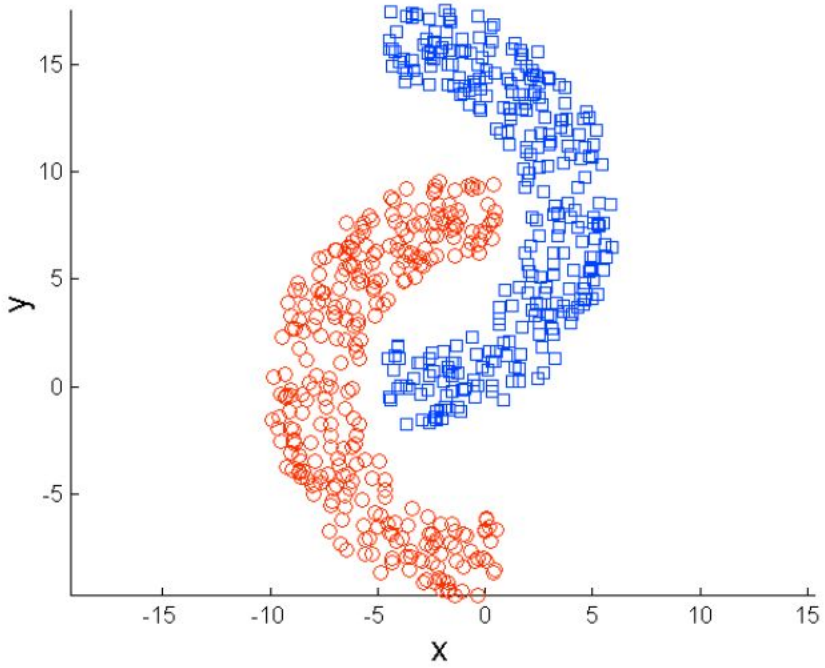


Original Points



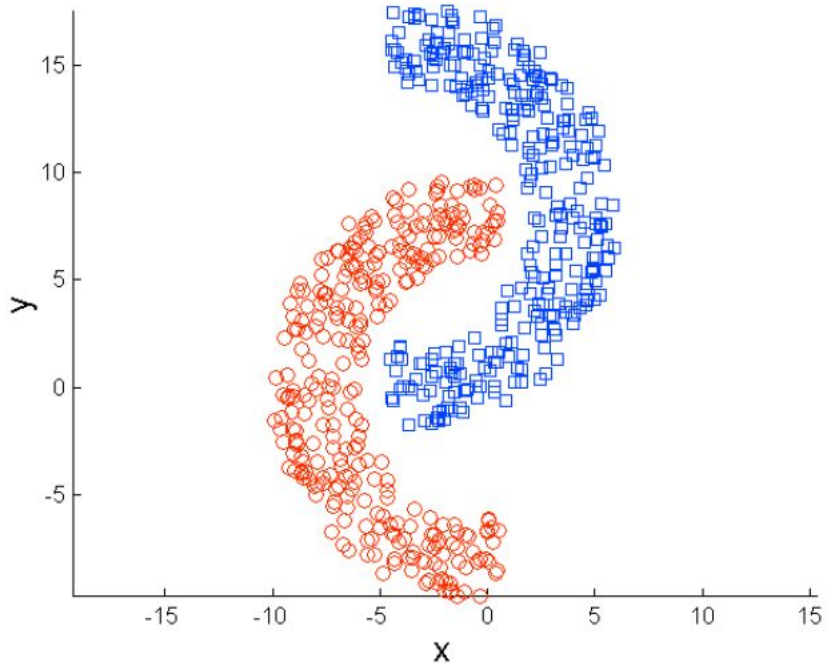
K-means (3 Clusters)

Limitations of K-means: Non-globular Shapes

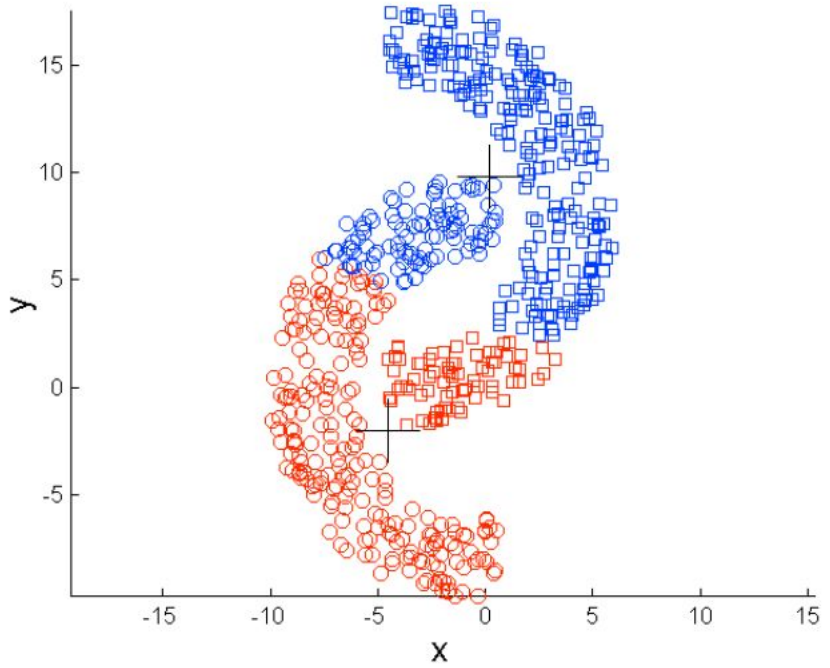


Original Points

Limitations of K-means: Non-globular Shapes



Original Points

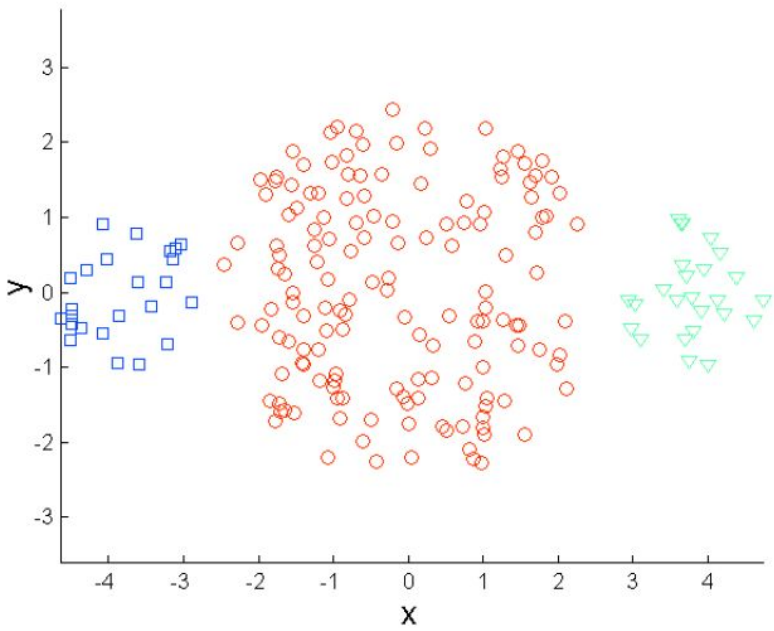


K-means (2 Clusters)

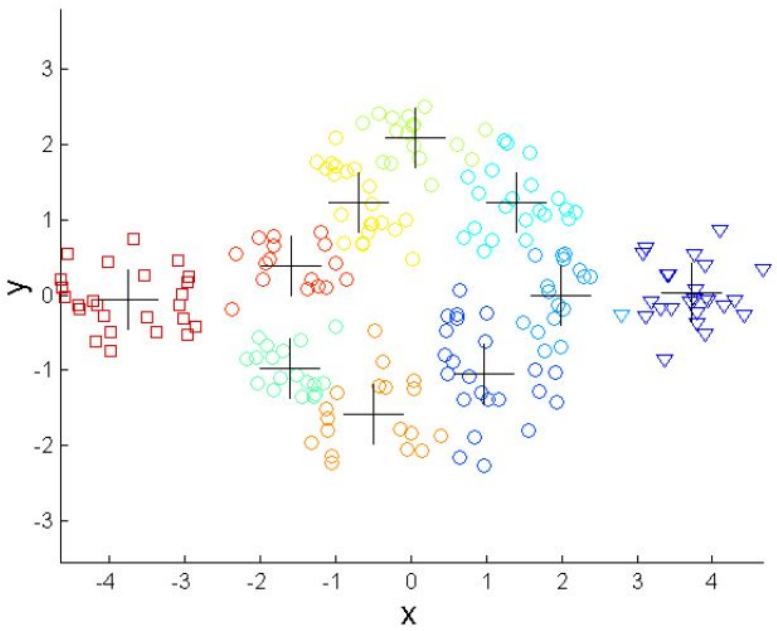
Overcoming K-means Limitations

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

Overcoming K-means Limitations

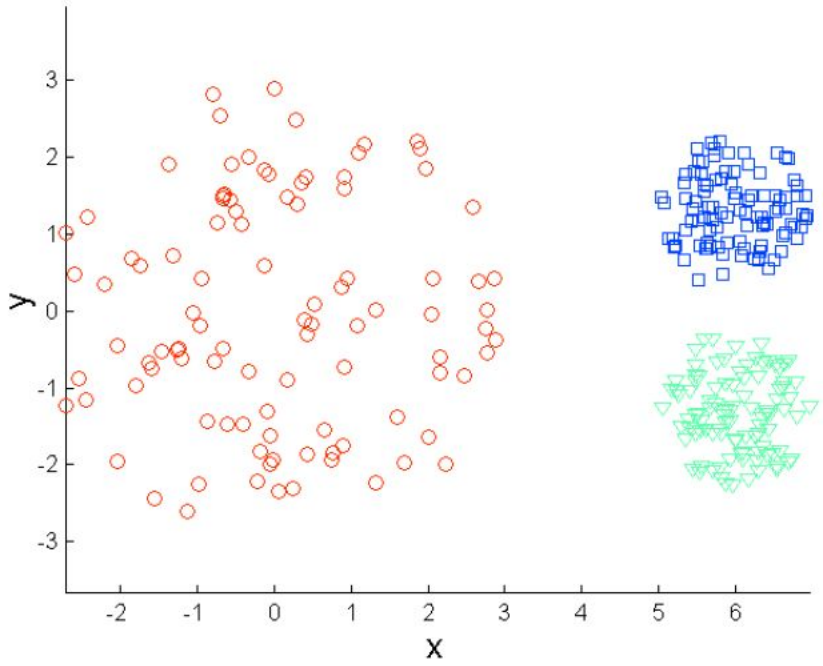


Original Points

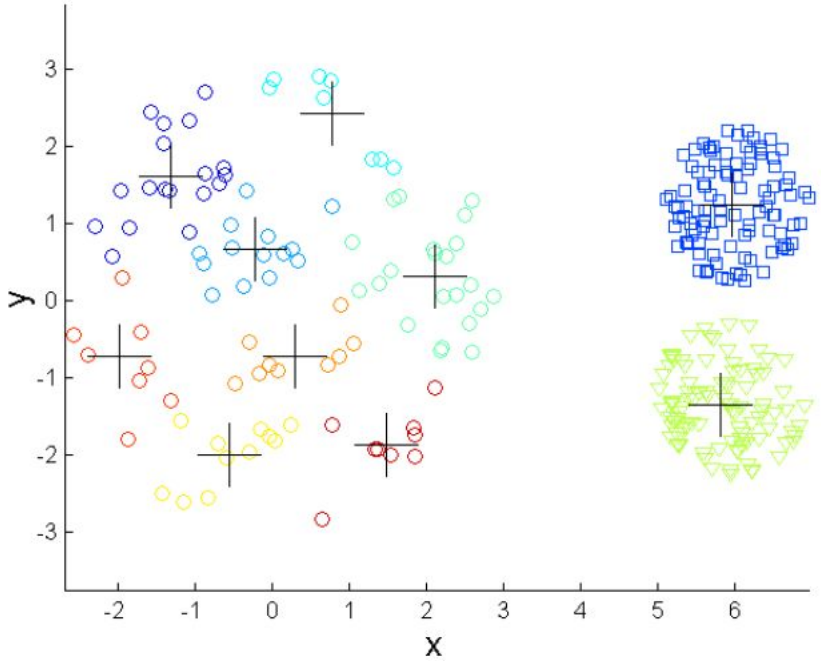


K-means Clusters

Overcoming K-means Limitations

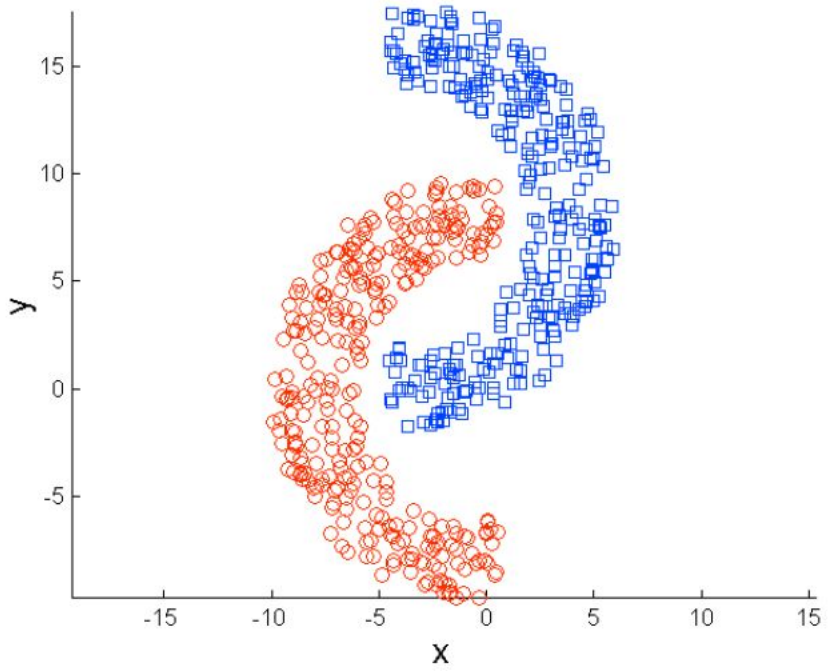


Original Points

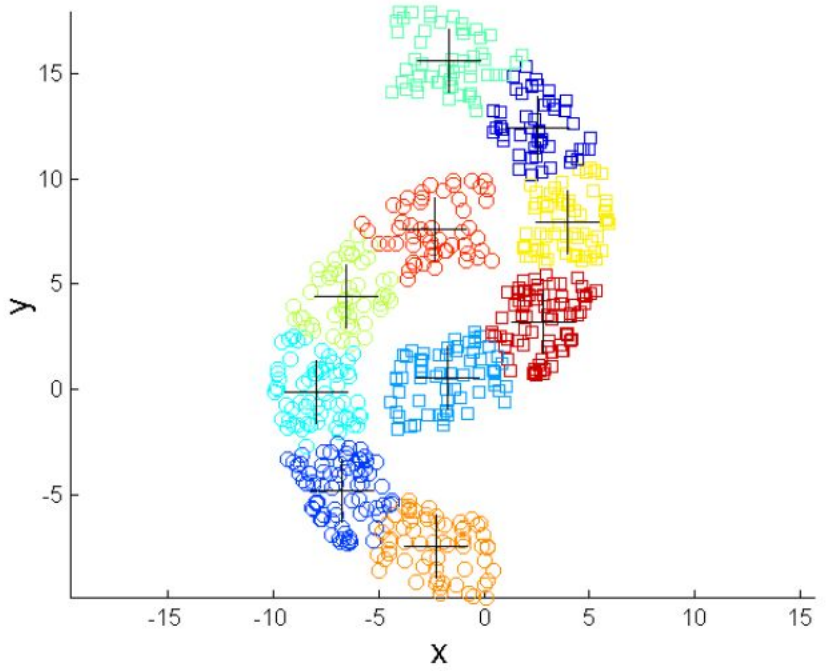


K-means Clusters

Overcoming K-means Limitations



Original Points

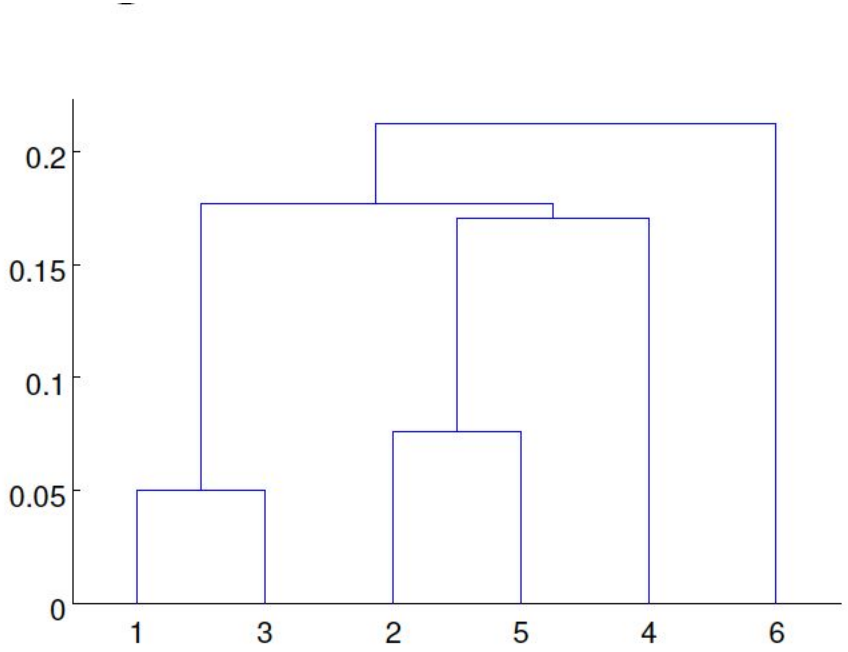


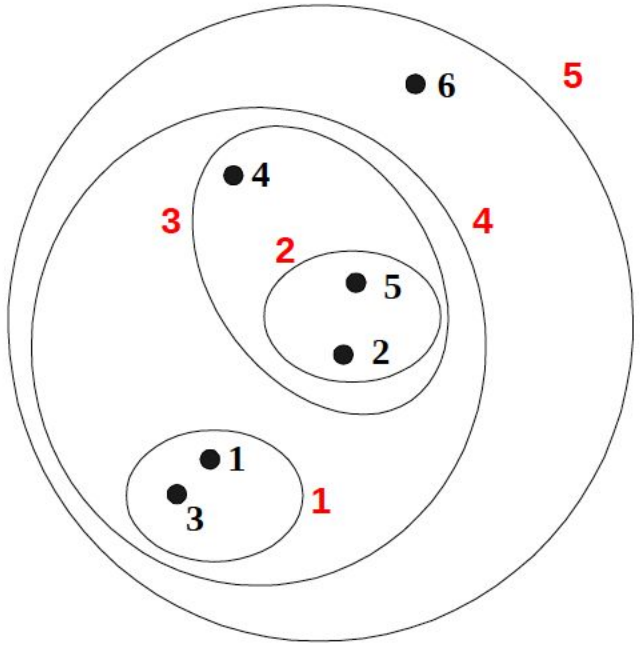
K-means Clusters

Hierarchical Clustering

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





Strengths of Hierarchical Clustering

- No assumptions on the number of clusters
  - Any desired number of clusters can be obtained by ‘cutting’ the dendogram at the proper level
- Hierarchical clusterings may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., phylogeny reconstruction, etc), web (e.g., product catalogs) etc

Hierarchical Clustering

- Two main types of hierarchical clustering
  - Agglomerative:
    - Start with the points 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 a point (or there are *k* clusters)
- Traditional hierarchical algorithms use a similarity or distance matrix
  - Merge or split one cluster at a time

Complexity of hierarchical clustering

- Distance matrix is used for deciding which clusters to merge/split
- At least quadratic in the number of data points
- Not usable for large datasets

Agglomerative clustering algorithm

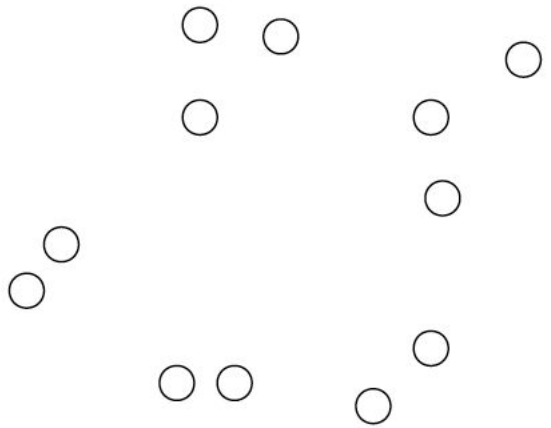
- Most popular hierarchical clustering technique
- Basic algorithm

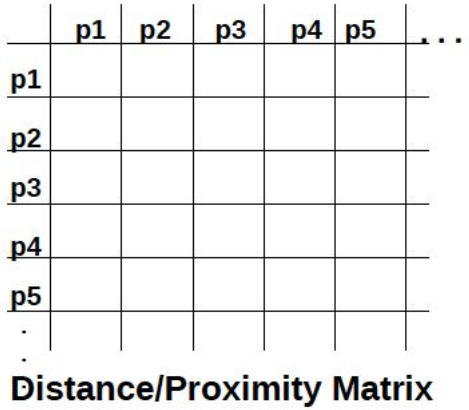
```
1. Compute the distance matrix between the input data points
2. Let each data point be a cluster
3. Repeat
4. Merge the two closest clusters
5. Update the distance matrix
6. Until only a single cluster remains
```

- Key operation is the computation of the distance between two clusters
  - Different definitions of the distance between clusters lead to different algorithms

Input/ Initial setting

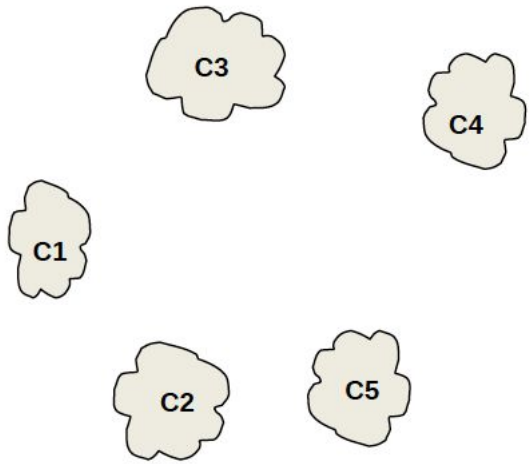
- Start with clusters of individual points and a distance/proximity matrix





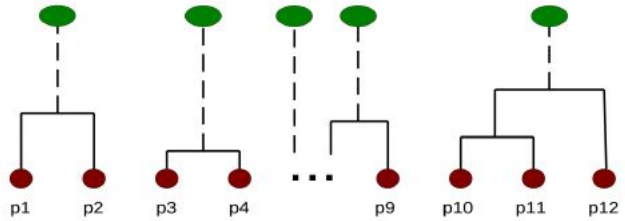
Intermediate State

- After some merging steps, we have some clusters



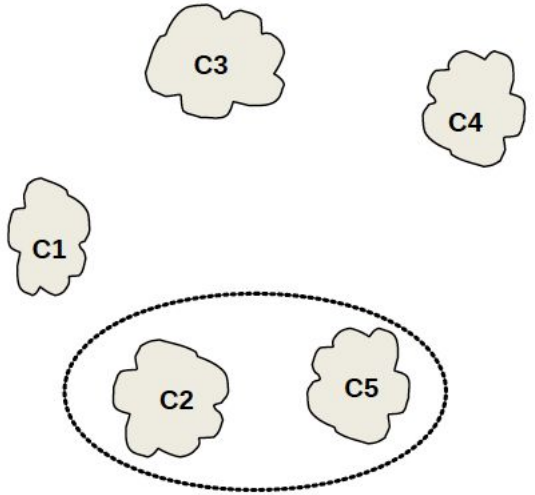
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Distance/Proximity Matrix



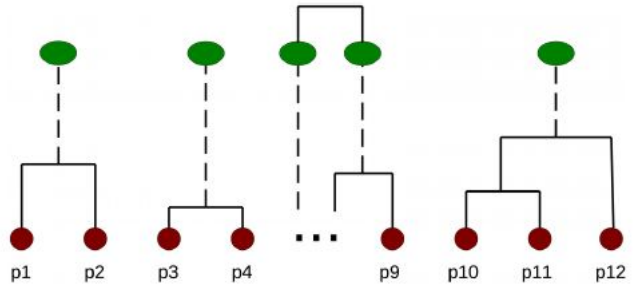
Intermediate State

- Merge the two closest clusters (C2 and C5) and update the distance matrix.



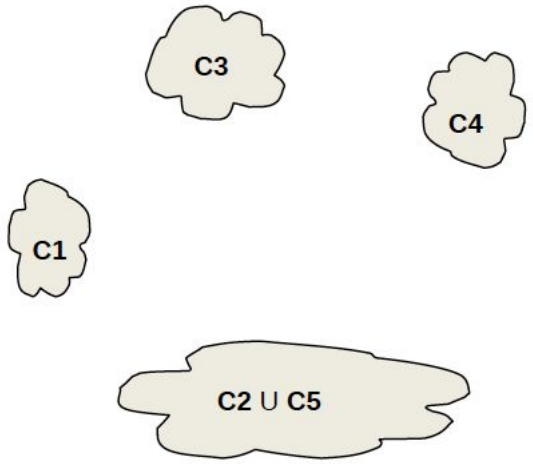
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Distance/Proximity Matrix

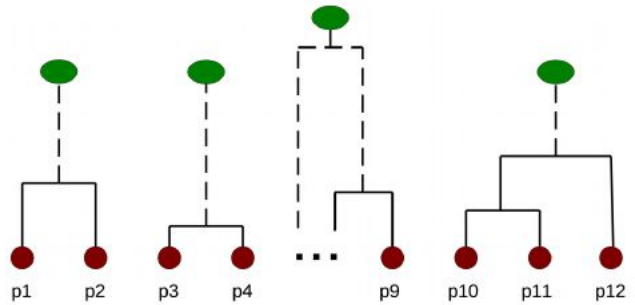


After Merging

- "How do we update the distance matrix?"



	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

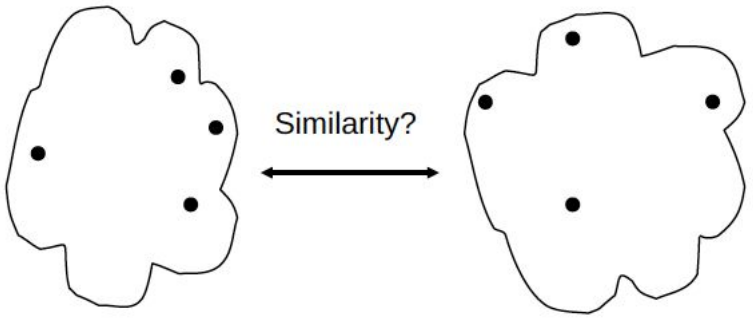


Distance between two clusters

- Each cluster is a set of points
- How do we define distance between two sets of points
  - Lots of alternatives
  - Not an easy task

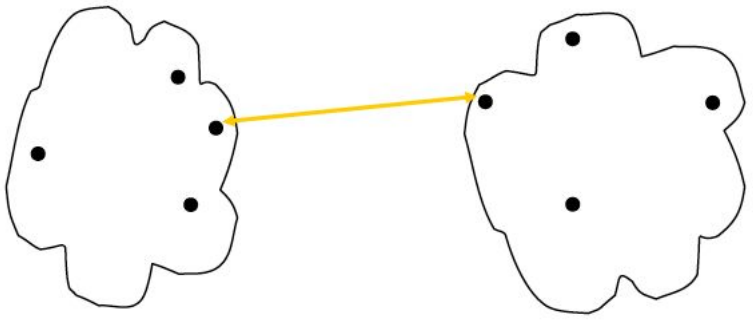
How to Define Inter-Cluster Similarity

- MIN
- MAX
- Group Average
- Distance Between Centroids



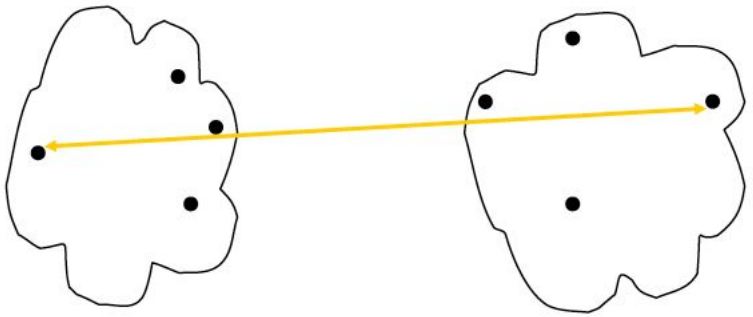
How to Define Inter-Cluster Similarity

- MIN
- MAX
- Group Average
- Distance Between Centroids



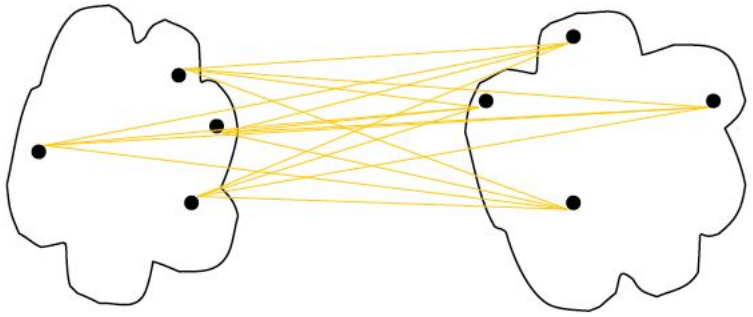
How to Define Inter-Cluster Similarity

- MIN
- MAX
- Group Average
- Distance Between Centroids



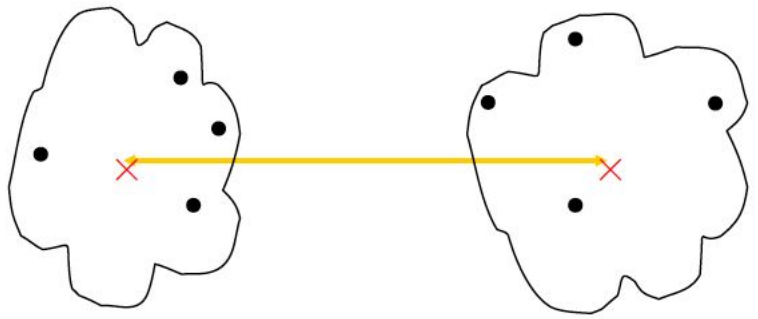
How to Define Inter-Cluster Similarity

- MIN
- MAX
- Group Average
- Distance Between Centroids



How to Define Inter-Cluster Similarity

- MIN
- MAX
- Group Average
- Distance Between Centroids



Distance between two clusters

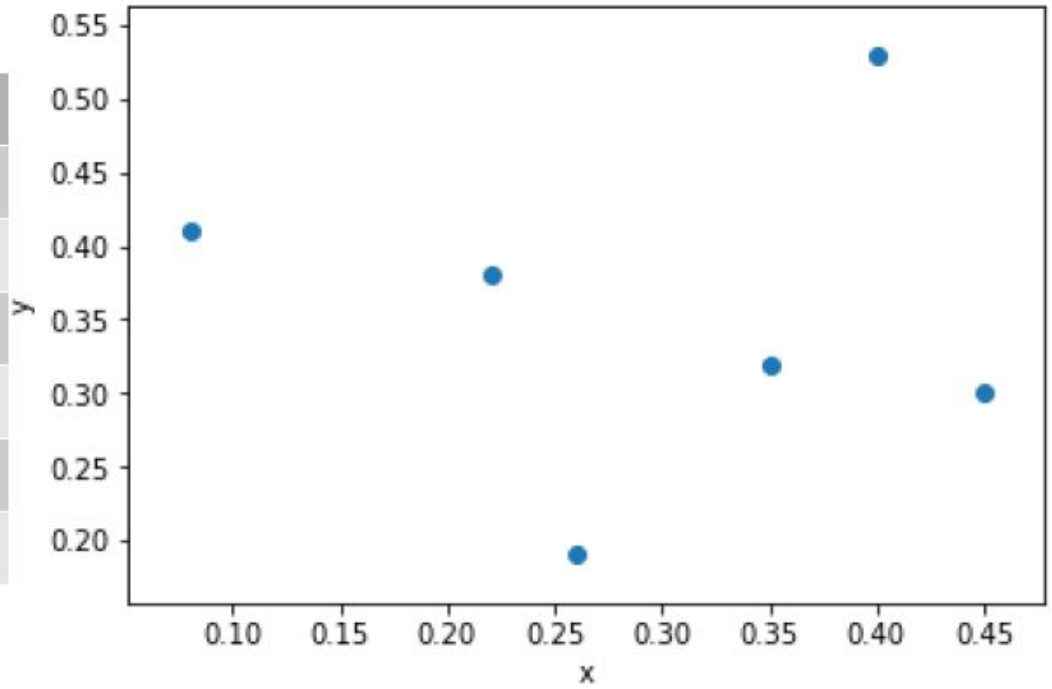
- Single-link distance between clusters C_i and C_j is the minimum distance between any object in C_i and any object in C_j
- The distance is defined by the two most similar objects

$$D_{sl}(C_i, C_j) = \min_{x,y} \{ d(x, y) | x \in C_i, y \in C_j \}$$

Example

- 6 points: {p1, p2, p3, p4, p5, p6}

	x	y
p1	0.40	0.53
p2	0.22	0.38
p3	0.35	0.32
p4	0.26	0.19
p5	0.08	0.41
p6	0.45	0.30



Example

- Euclidean distance matrix of the 6 points

	p1	p2	p3	p4	p5	P6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2	0.24	0.00	0.15	0.20	0.14	0.25
p3	0.22	0.15	0.00	0.15	0.28	0.11
p4	0.37	0.20	0.15	0.00	0.29	0.22
p5	0.34	0.14	0.28	0.29	0.00	0.39
P6	0.23	0.25	0.11	0.22	0.39	0.00

Example

- Euclidean distance matrix of the 6 points

	p1	p2	p3	p4	p5	P6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2		0.00	0.15	0.20	0.14	0.25
p3			0.00	0.15	0.28	0.11
p4				0.00	0.29	0.22
p5					0.00	0.39
P6						0.00

d(pi, pj) = d(pj, pi) : Symmetry property of distance measures

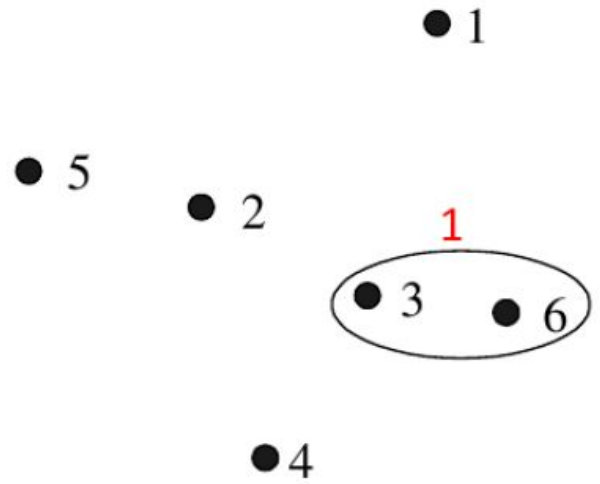
Example

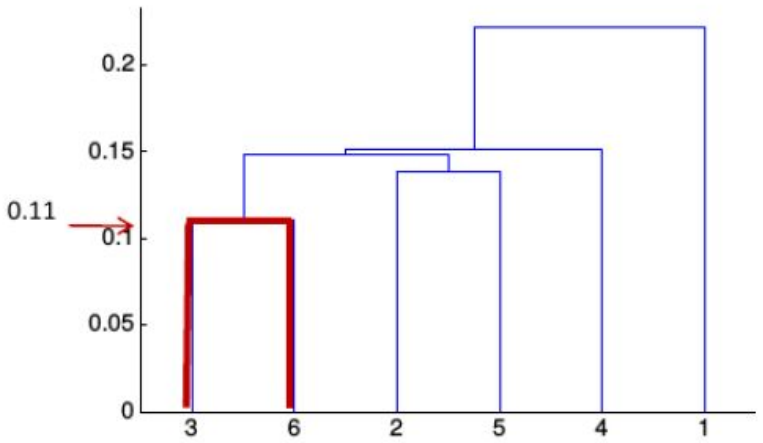
- Euclidean distance matrix of the 6 points

	p1	p2	p3	p4	p5	P6
p1	0.00	0.24	0.22	0.37	0.34	0.23
p2		0.00	0.15	0.20	0.14	0.25
p3			0.00	0.15	0.28	0.11
p4				0.00	0.29	0.22
p5					0.00	0.39
P6						0.00

d(pi, pj) = d(pj, pi) : Symmetry property of distance measures

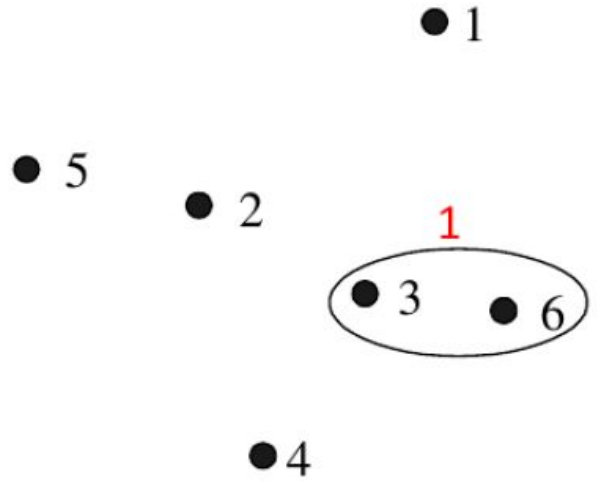
Single-link (MIN) clustering: example

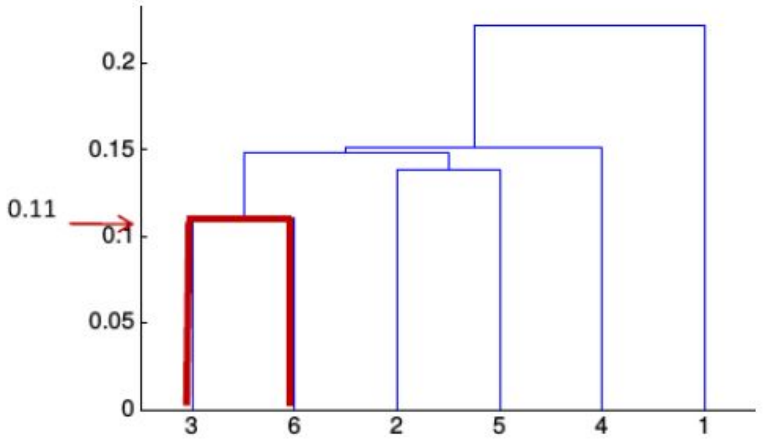




	p1	p2	{p3,p6}	p4	p5
p1	0	0.24	?	0.34	0.23
p2		0	?	0.20	0.14
{p3,p6}			0.11	?	?
p4				0	0.29
p5					0

Single-link (MIN) clustering: example

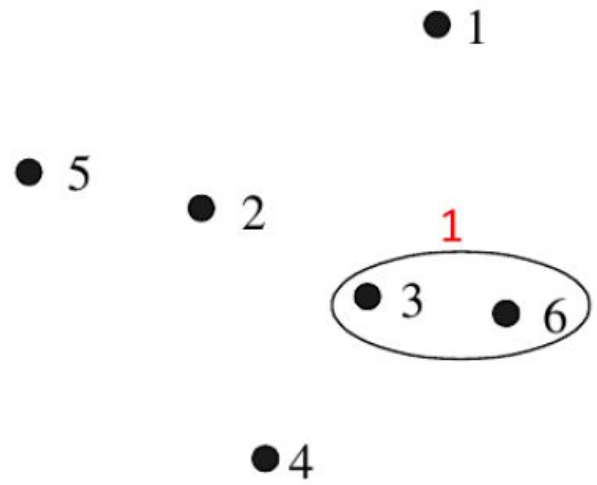


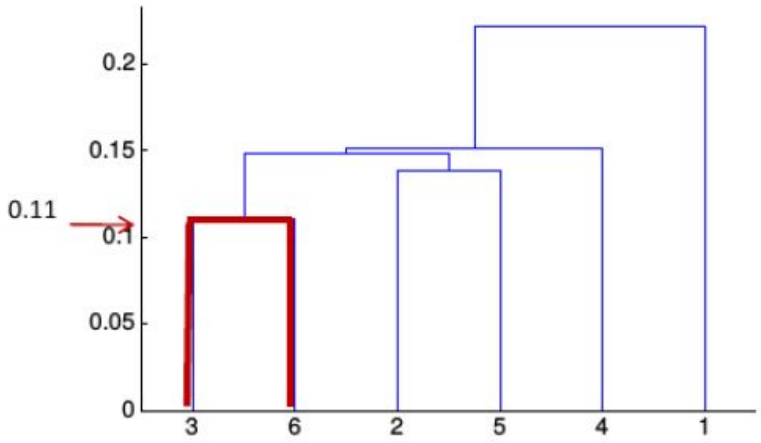


Dist({3,6},{2}) = min(dist(3,2), dist(6,2))
min(0.15, 0.25) = 0.15
Dist({3,6}, {5}) = min(dist(3,5), dist(6,5)) =0.28
Dist ({3,6}, {4}) = min(dist(3,4), dist(6,4)) = 0.15
Dist({3,6}, {1}) = min(dist(3,1), dist(6,1)) = 0.22

	p1	p2	{p3,p6}	p4	p5
p1	0	0.24	0.22	0.34	0.23
p2		0	0.15	0.20	0.14
{p3,p6}			0.11	0.15	0.28
p4				0	0.29
p5					0

Single-link (MIN) clustering: example

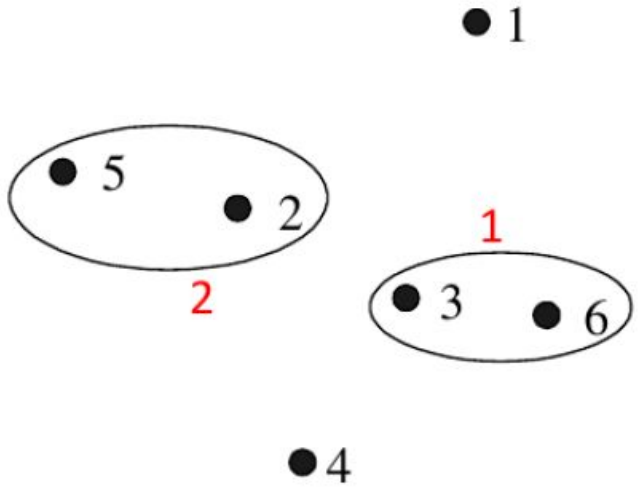


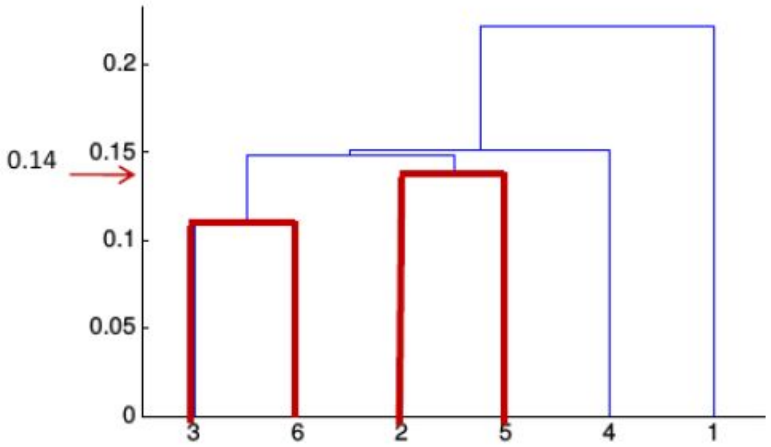


Dist({3,6},{2}) = min(dist(3,2), dist(6,2))
min(0.15, 0.25) = 0.15
Dist({3,6}, {5}) = min(dist(3,5), dist(6,5)) =0.28
Dist ({3,6}, {4}) = min(dist(3,4), dist(6,4)) = 0.15
Dist({3,6}, {1}) = min(dist(3,1), dist(6,1)) = 0.22

	p1	p2	{p3,p6}	p4	p5
p1	0	0.24	0.22	0.34	0.23
p2		0	0.15	0.20	0.14
{p3,p6}			0.11	0.15	0.28
p4				0	0.29
p5					0

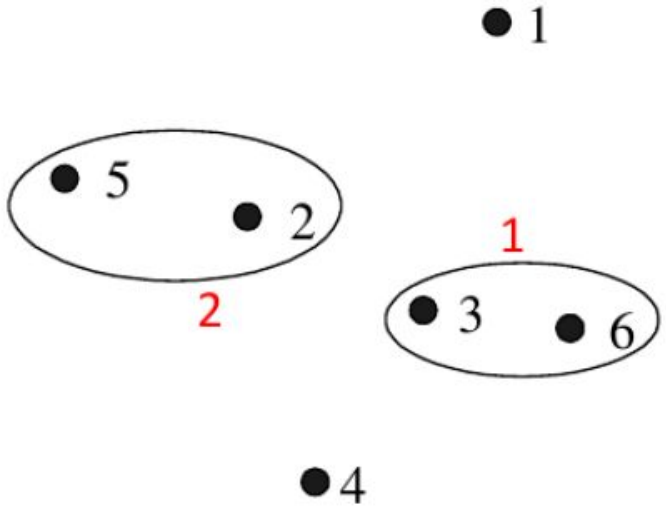
Single-link (MIN) clustering: example

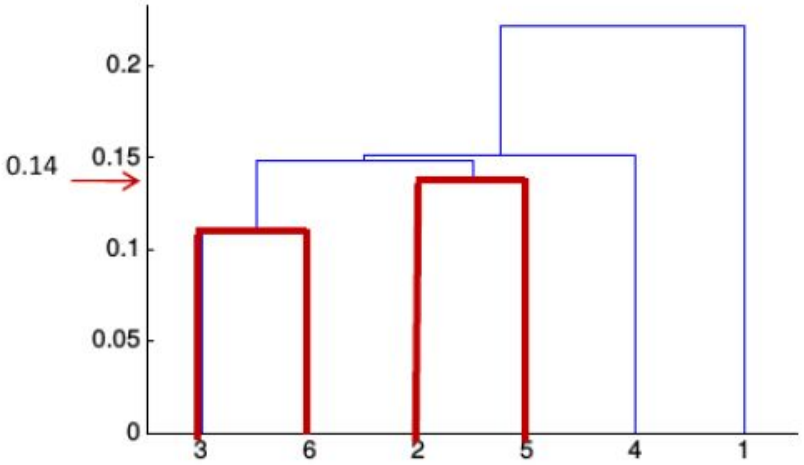




	p1	{p2,p5}	{p3,p6}	p4
p1	0	?	0.22	0.34
{p2,p5}		0.14	?	?
{p3,p6}			0.11	0.15
p4				0

Single-link (MIN) clustering: example





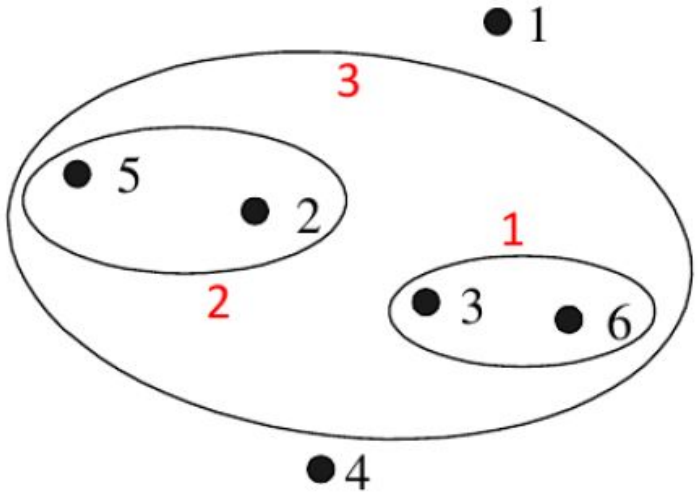
dist(p1, {p2,p5}) = min(dist(p1,p2),dist(p1,p5))
= min(0.24,0.23) = 0.23

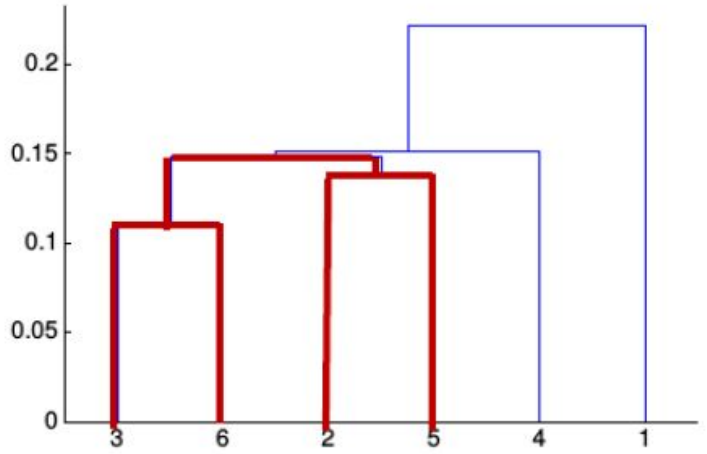
dist({p2,p5},{p3,p6}) =
= min(dist(p2,p3), dist(p2,p6), dist(p5,p3),
dist(p5,p6))
= min(0.15,0.25,0.28,0.39) = 0.15

dist({p2,p5},p4) = min(dist(p2,p4), dist(p5,p4))
= min(0.20, 0.29) = 0.20

	p1	{p2,p5}	{p3,p6}	p4
p1	0	0.23	0.22	0.34
{p2,p5}		0.14	0.15	0.20
{p3,p6}			0.11	0.15
p4				0

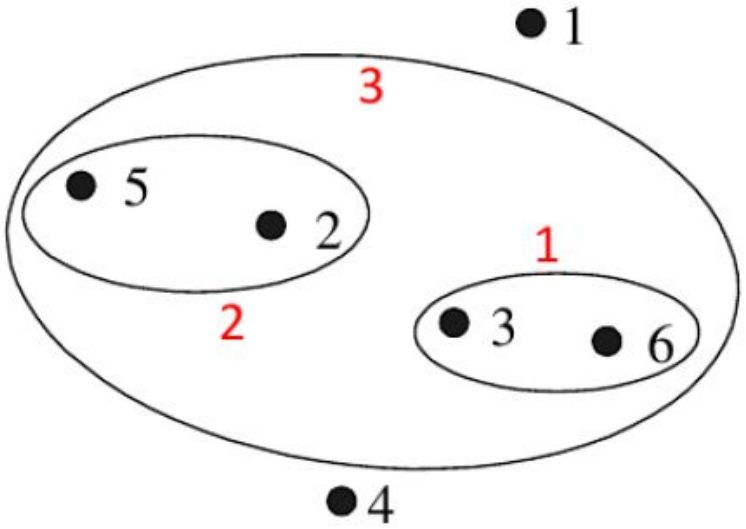
Single-link (MIN) clustering: example

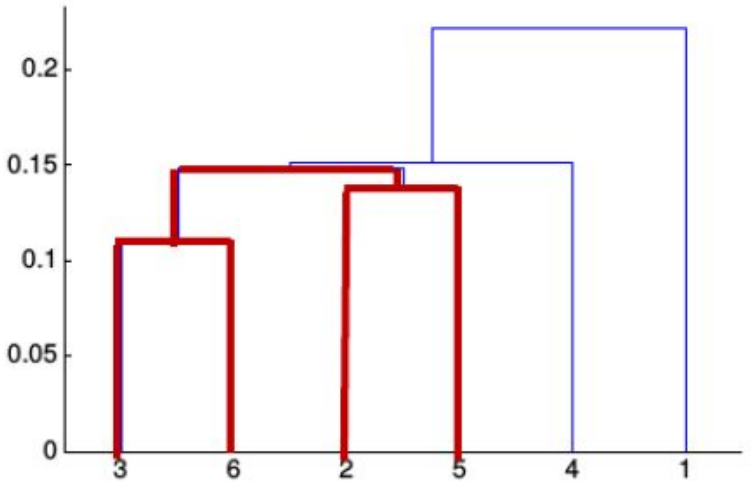




	p1	{p2,p5}	{p3,p6}	p4
p1	0	0.23	0.22	0.34
{p2,p5}		0.14	0.15	0.20
{p3,p6}			0.11	0.15
p4				0

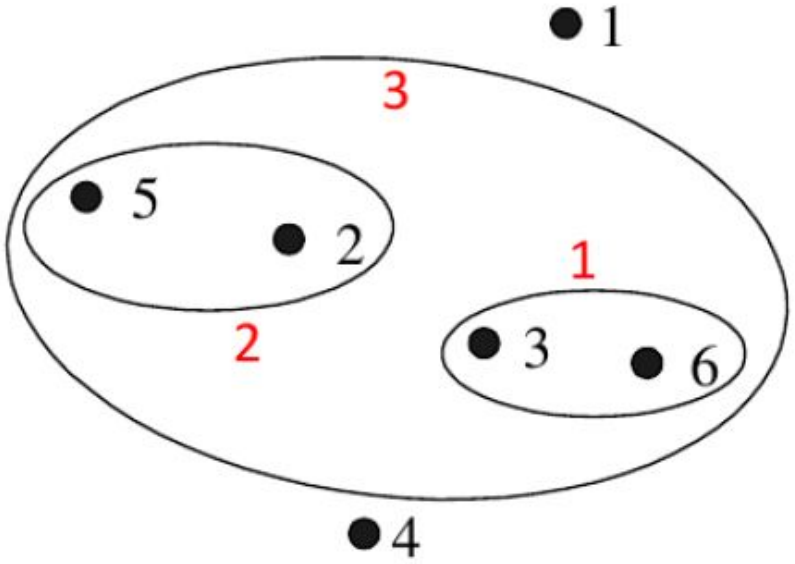
Single-link (MIN) clustering: example

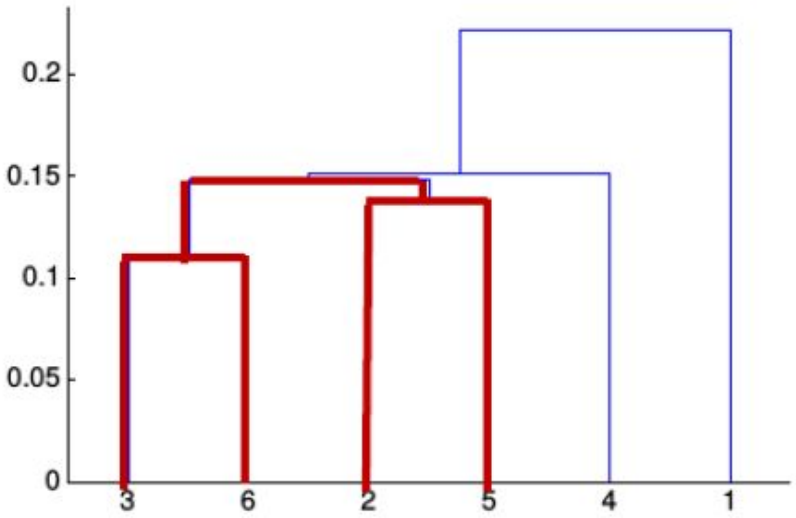




	p1	{p2,p3,p5p6}	p4
p1	0	?	0.34
{p2,p3,p5,p6}		0.15	?
p4			0

Single-link (MIN) clustering: example



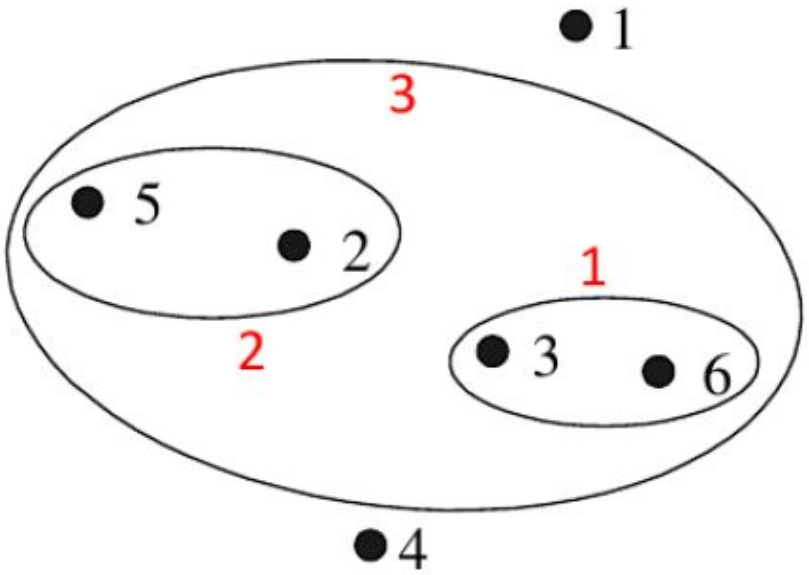


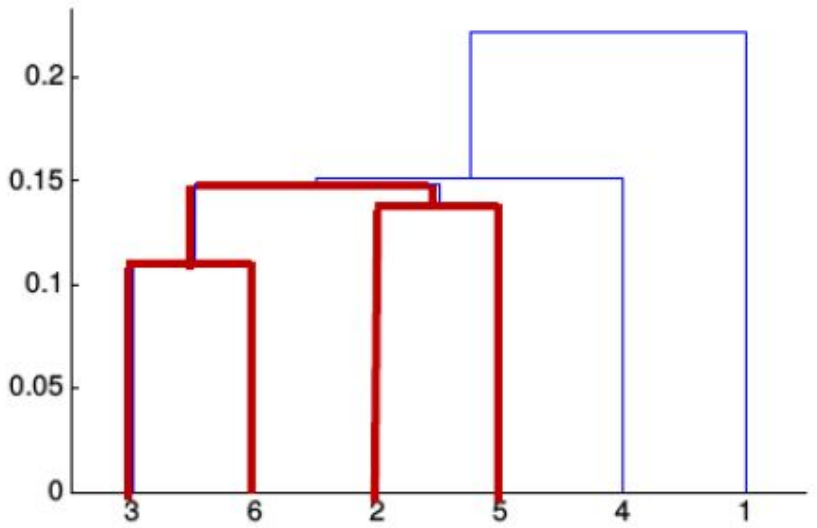
dist(p1, {p2,p3,p5,p6}) = min{ dist(p1,p2), dist(p1,p3),
dist(p1,p5), dist(p1,p6)}
= min(0.24, 0.22, 0.34, 0.23) = 0.22

dist({p2,p3,p5,p6}, p4) = min(0.20, 0.15, 0.29,0.22) = 0.15

	p1	{p2,p3,p5p6}	p4
p1	0	0.22	0.34
{p2,p3,p5,p6}		0.15	0.15
p4			0

Single-link (MIN) clustering: example



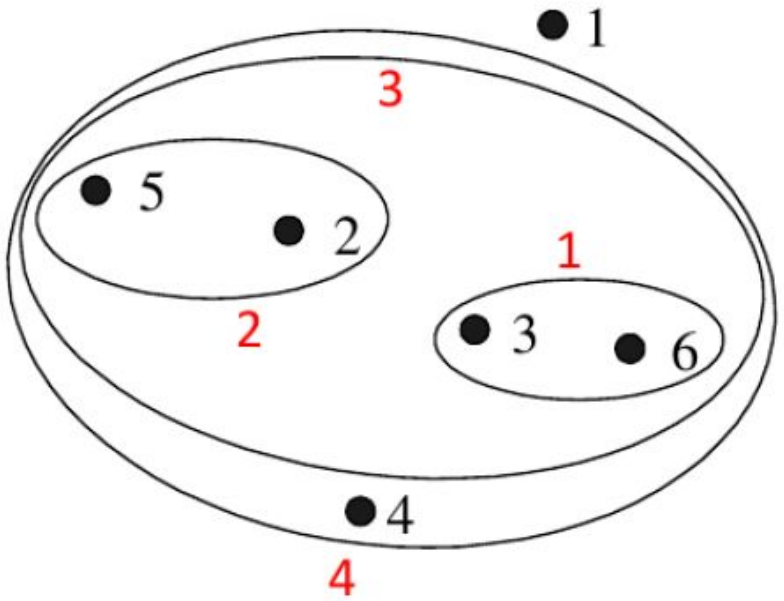


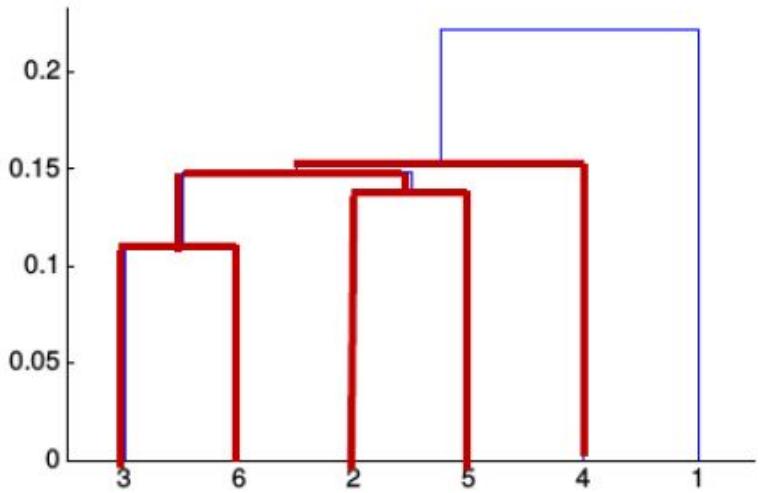
dist(p1, {p2,p3,p5,p6}) = min{ dist(p1,p2), dist(p1,p3),
dist(p1,p5), dist(p1,p6)}
= min(0.24, 0.22, 0.34, 0.23) = 0.22

dist({p2,p3,p5,p6}, p4) = min(0.20, 0.15, 0.29,0.22) = 0.15

	p1	{p2,p3,p5p6}	p4
p1	0	0.22	0.34
{p2,p3,p5,p6}		0.15	0.15
p4			0

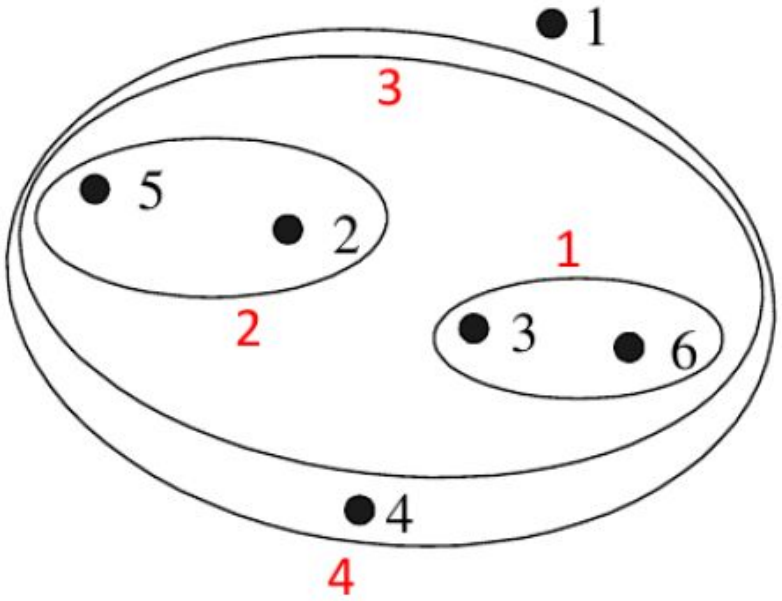
Single-link (MIN) clustering: example

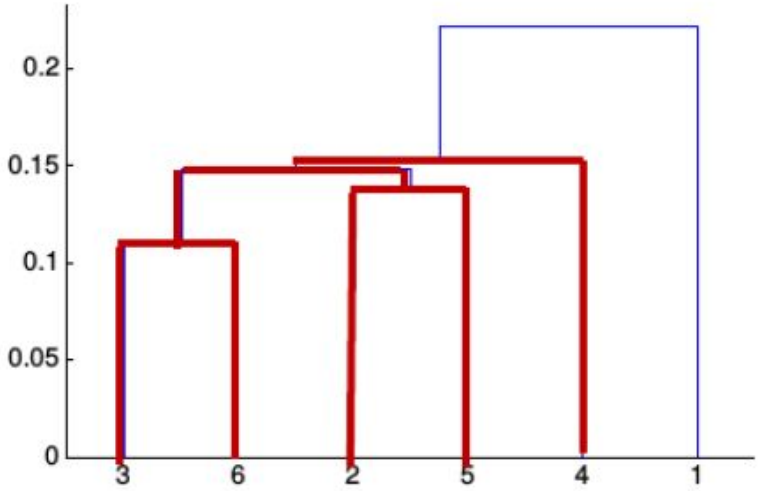




	p1	{p2,p3,p4,p5 p6}
p1	0	?
{p2,p3,p4,p5,p6}		0.15

Single-link (MIN) clustering: example

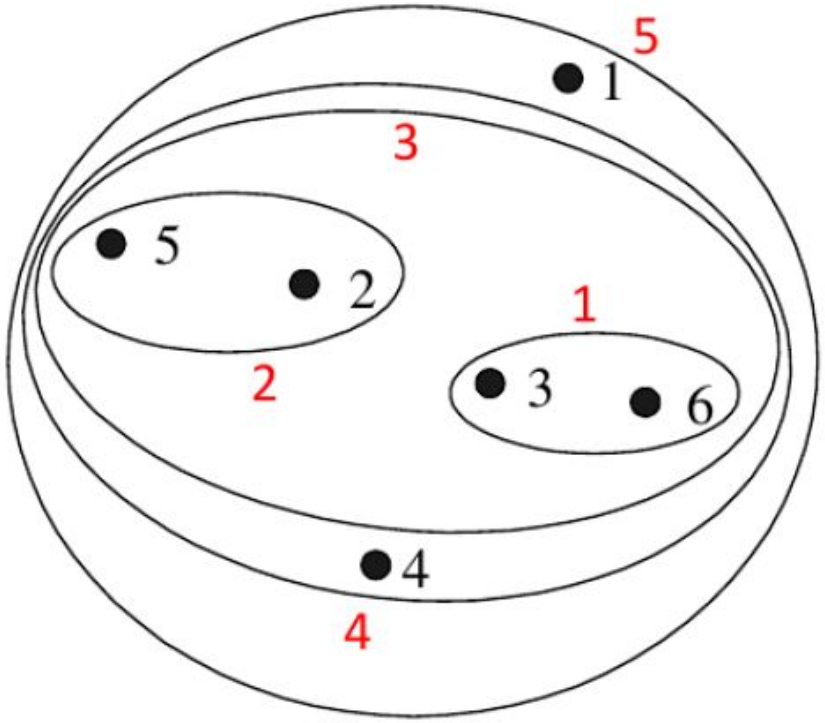


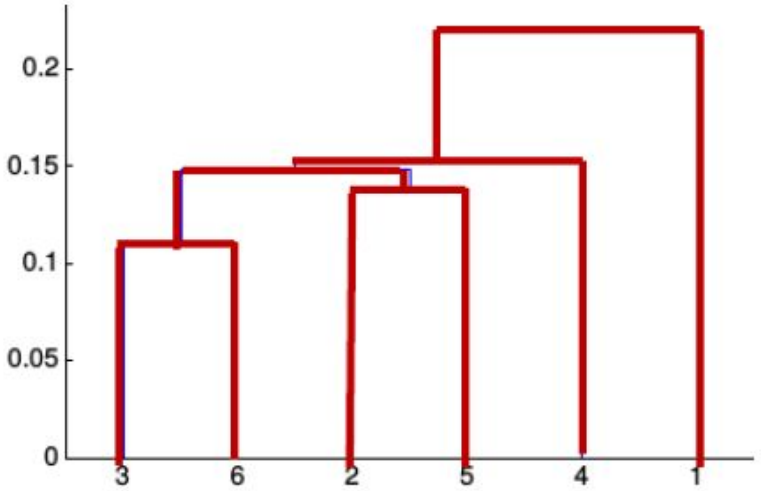


Dist(({3,6,2,5,4}, {1}) = min(dist(3,1), dist(6,1),
dist(2,1), dist(5,1), dist(4,1))
= min(0.22, 0.23, 0.24, 0.34, 0.37)
= 0.22

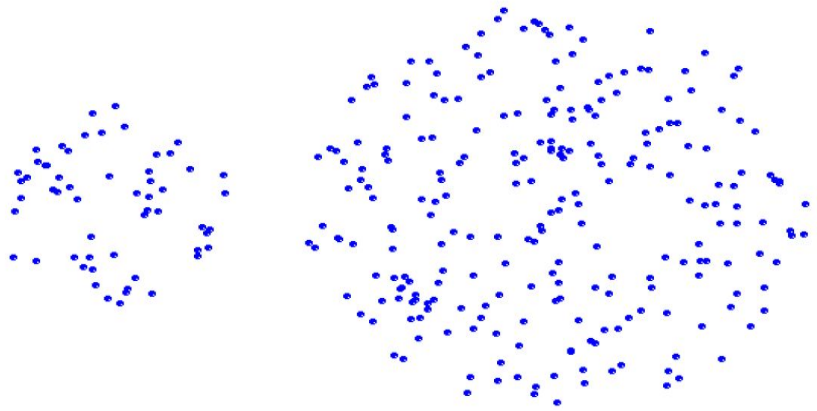
	p1	{p2,p3,p4,p5 p6}
p1	0	0.22
{p2,p3,p4,p5,p6}		0.15

Single-link (MIN) clustering: example

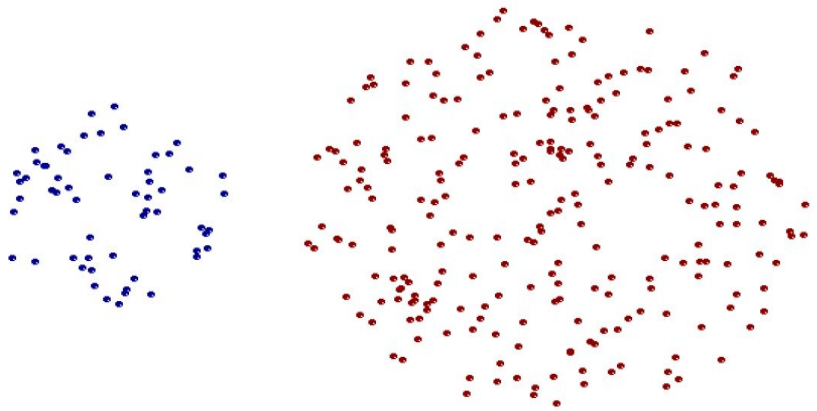




Strengths of single-link clustering

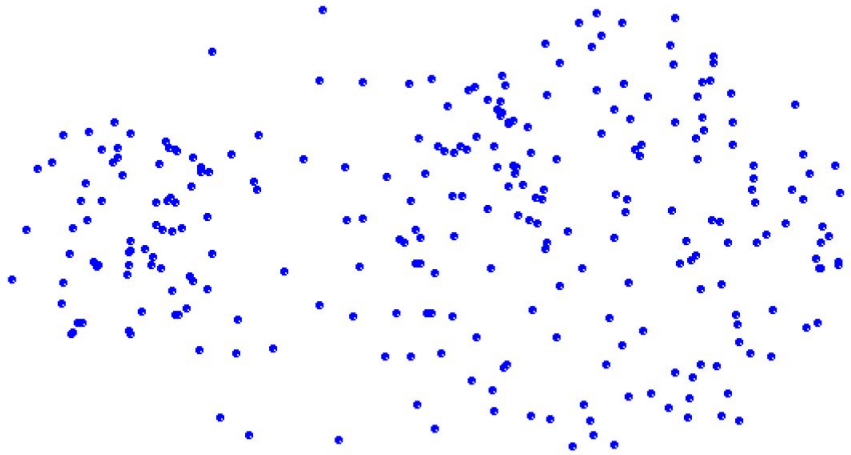


Original Points

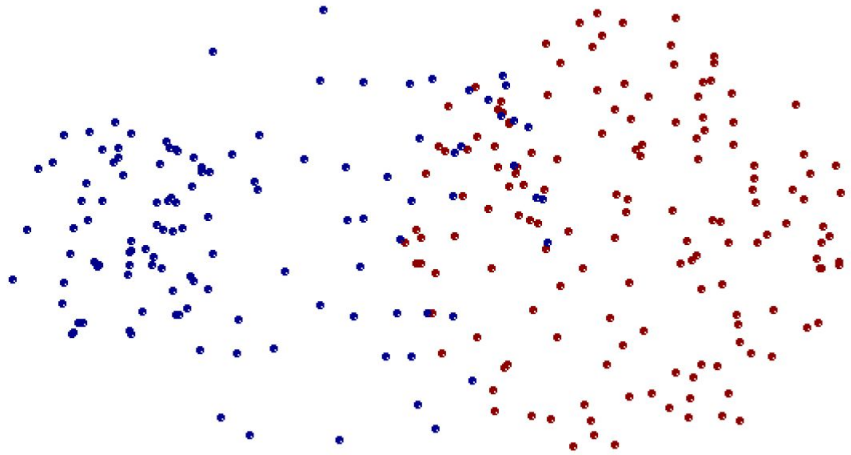


Two Clusters

Limitations of single-link clustering



Original Points



Two Clusters

- Sensitive to noise and outliers
- It produces long, elongated clusters

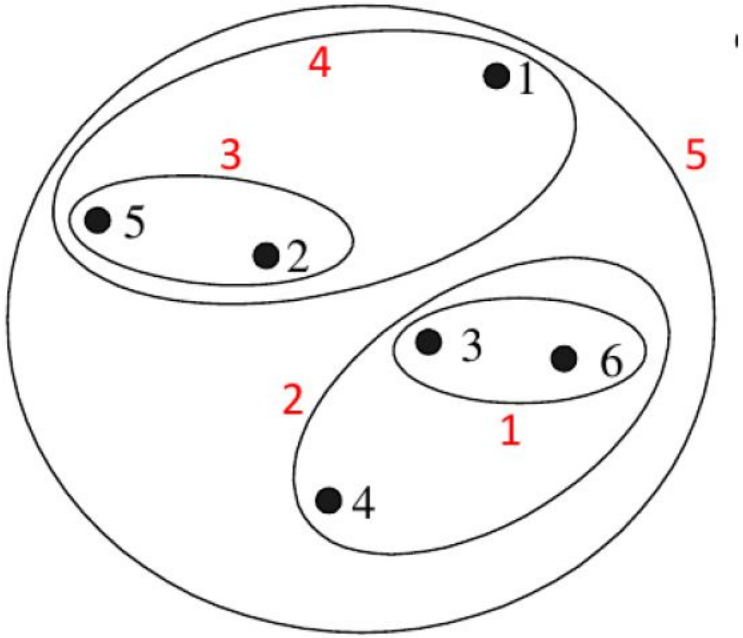
Distance between two clusters

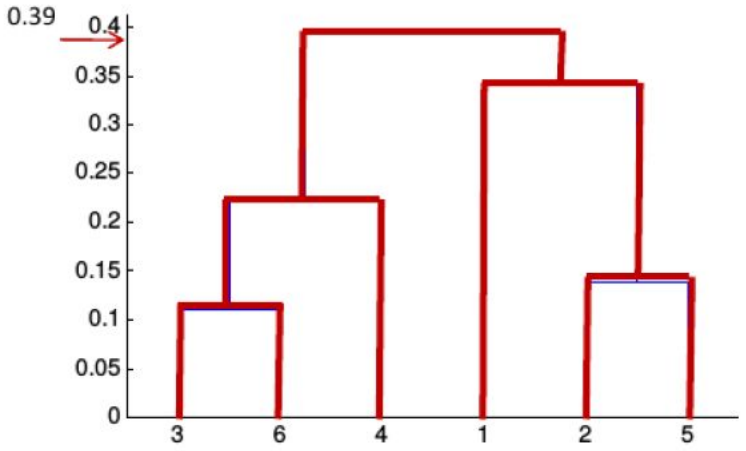
- Complete-link distance between clusters C_i and C_j is the maximum distance between any object in C_i and any object in C_j
- The distance is defined by the two most dissimilar objects

$$D_{cl}(C_i, C_j) = \max_{x,y}\{d(x,y) \mid x \in C_i, y \in C_j\}$$

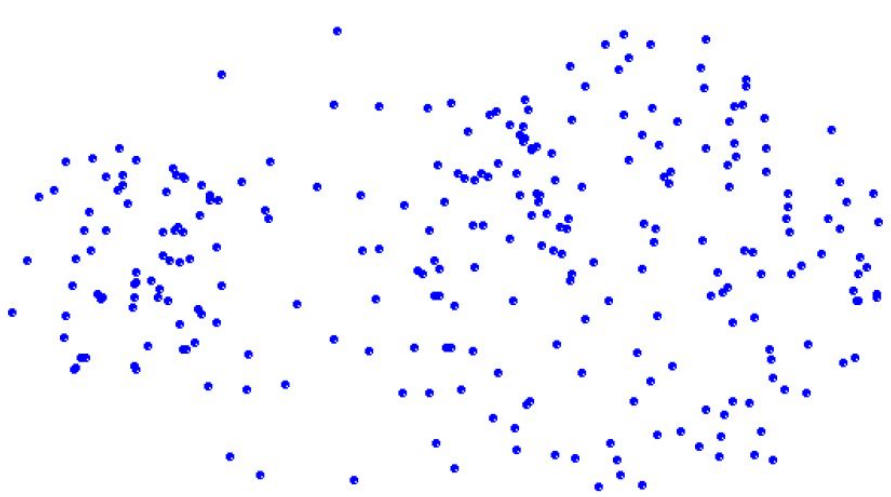
Complete-link clustering: example

- Distance between clusters is determined by the two most distant points in the different clusters

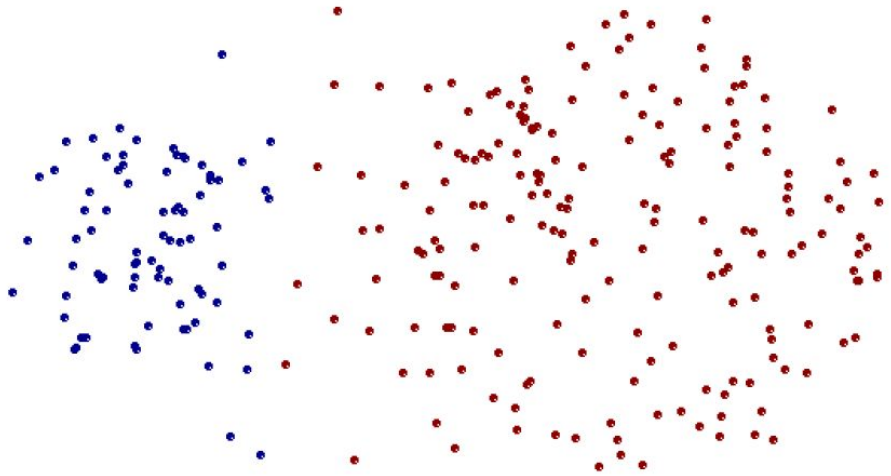




Strengths of complete-link clustering



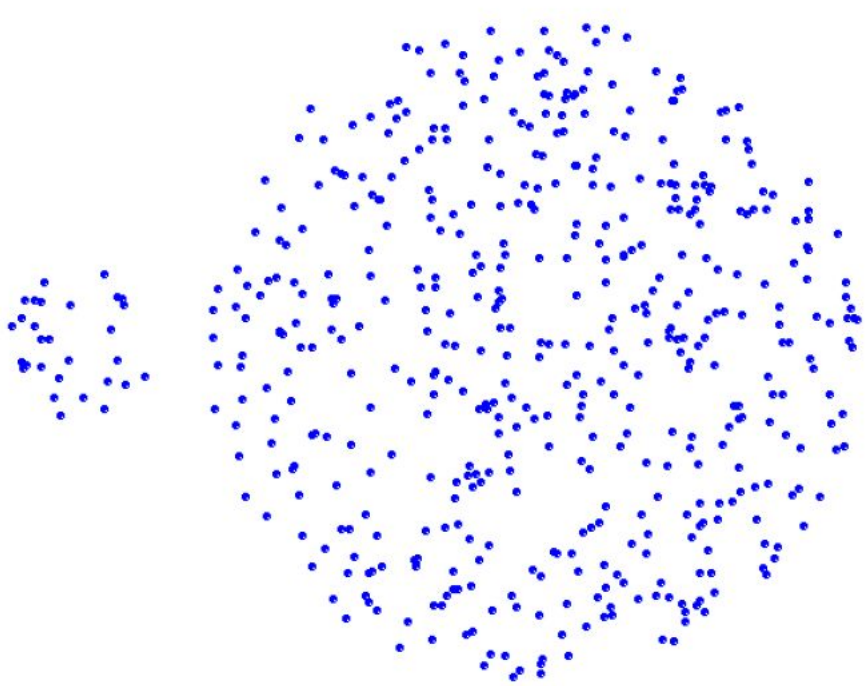
Original Points



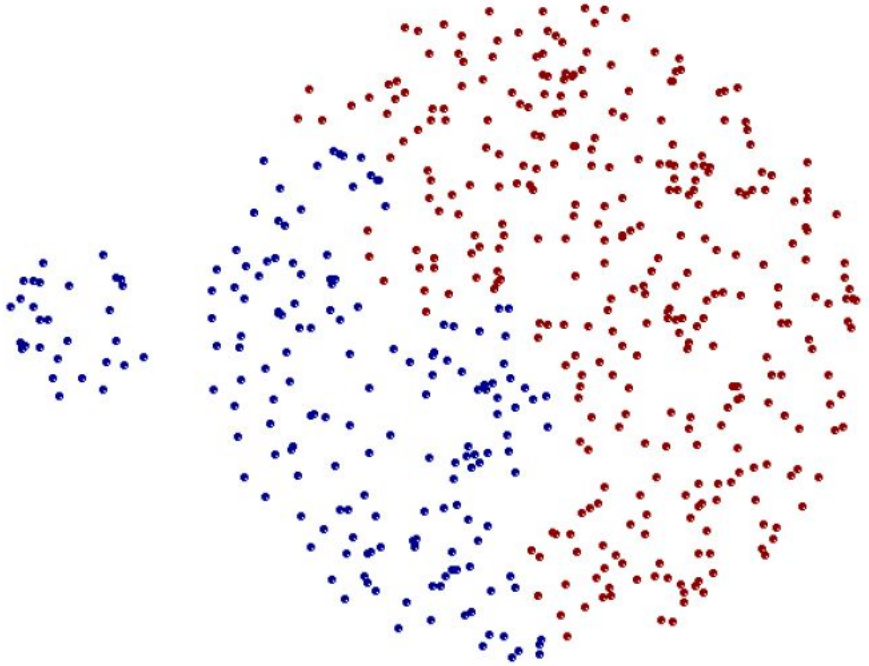
Two Clusters

- More balanced clusters (with equal diameter)
- Less susceptible to noise

Limitations of complete-link clustering



Original Points



Two Clusters

- Tends to break large clusters
- All clusters tend to have the same diameter – small clusters are merged with larger ones

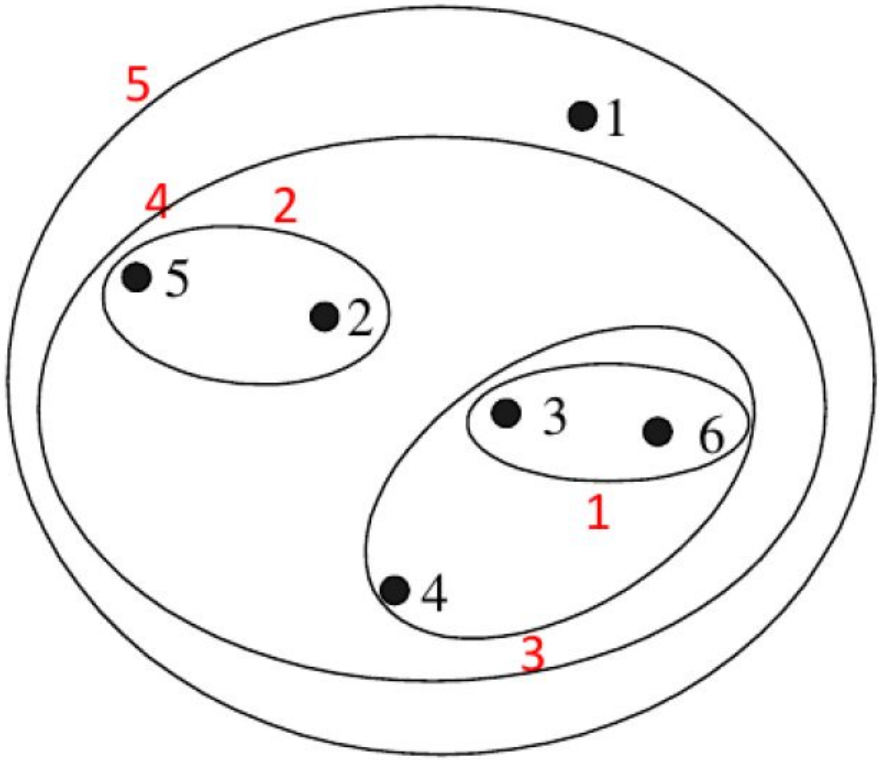
Distance between two clusters

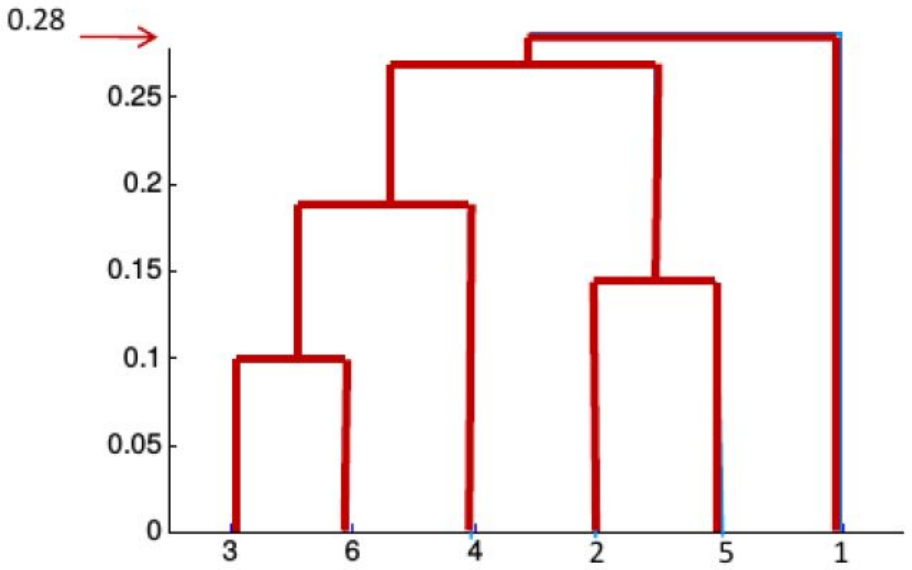
- Group average distance between clusters C_i and C_j is the average distance between any object in C_i and any object in C_j

$$D_{avg}(C_i, C_j) = \frac{1}{|C_i| \times |C_j|} \sum_{x \in C_i,\, y \in C_j} d(x, y)$$

Average-link clustering: example

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





Average-link clustering: discussion

- Compromise between Single and Complete Link
- Strengths
  - Less susceptible to noise and outliers
- Limitations
  - Biased towards globular clusters

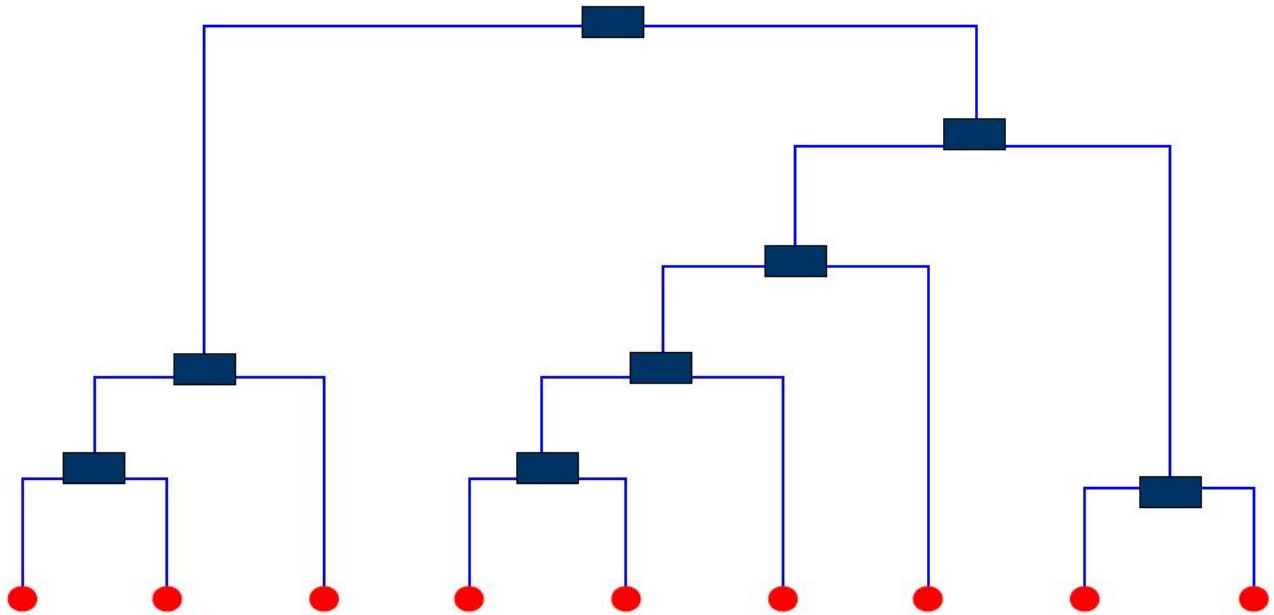
Distance between two clusters

- Centroid distance between clusters C_i and C_j is the distance between the centroid r_i of C_i and the centroid r_j of C_j

$$D_{centroids}(C_i, C_j) = d(r_i, r_j)$$

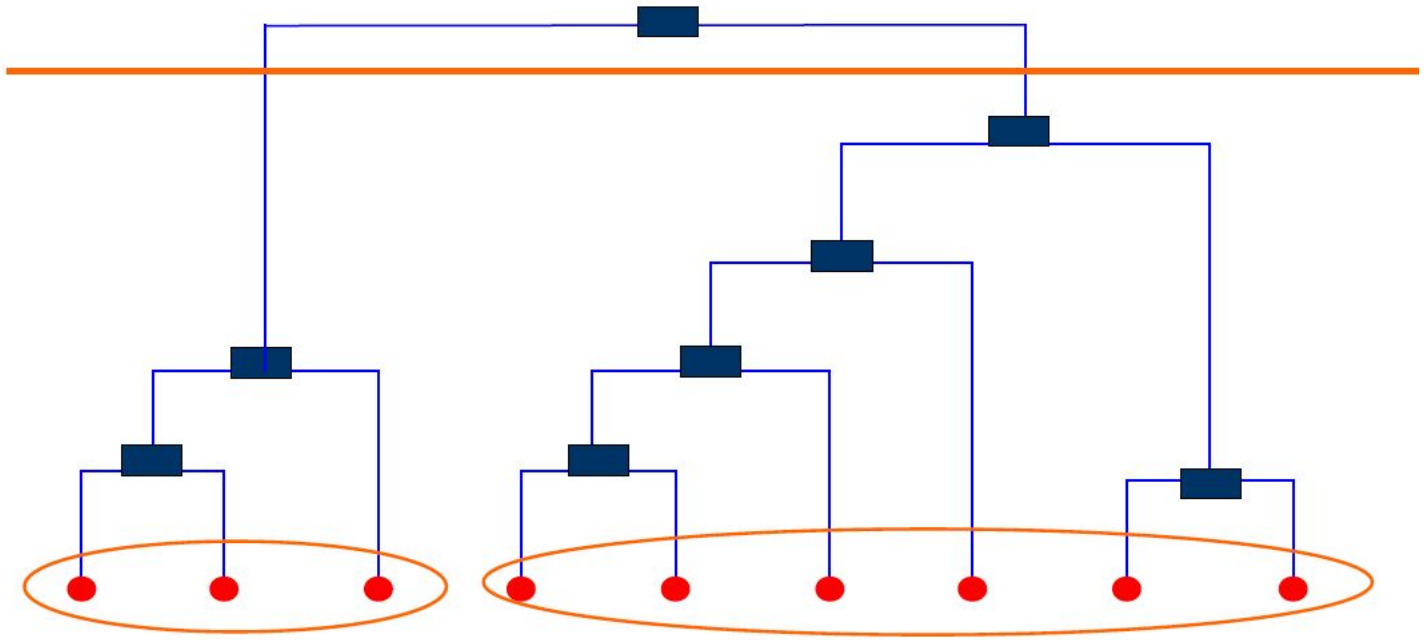
Dendrogram

- A binary tree that shows how clusters are merged/split hierarchically
- Each node on the tree is a cluster; each leaf node is a singleton cluster



Dendrogram

- A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster



Dendrogram

- A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster