Hierarchical Clustering

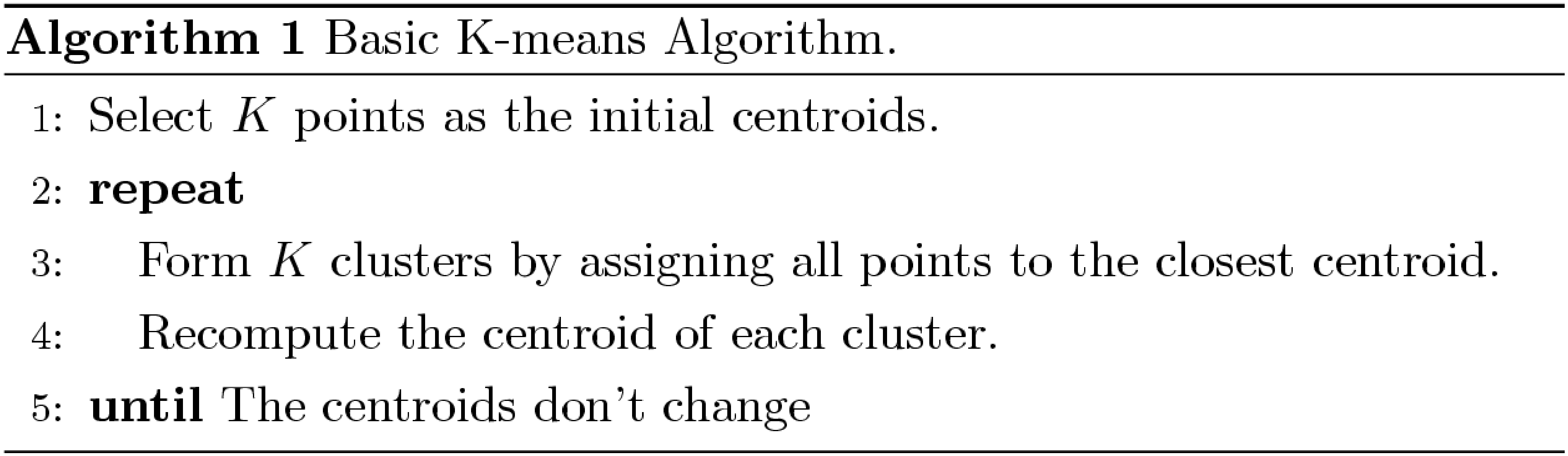
AW

Lecture Overview

1. **Recap**
2. **Modifications and R**
3. **Bisecting K-means + Limitations**
4. **Intro to hierarchical clustering**
5. **Agglomerative Clustering**
6. **Min**

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:



Algorithm 1 Basic K-means Algorithm.

```
Select $K$ points as the initial centroids.
repeat
    Form $K$ clusters by assigning all points to the closest centroid.
    Recompute the centroid of each cluster.
until The centroids don't change
```

K-means Optimization – Solutions

Objective function	Proximity measure	Centroid
min Sum of Absolute Errors (SAE)	Manhattan distance	median
min Sum of Squared Errors (SSE)	Euclidean distance	Mean
min Total Cohesion	Cosine similarity	Mean
Min Sum of Mahalanobis Distances (SMD)	Mahalanobis distance (centered)	Mean

Solutions to Initial Centroids Problem

- Multiple runs
  - Helps, but probability is not on our side
- Sample and use hierarchical clustering to determine initial centroids (we'll see later)
- Select more than k initial centroids and then select among these initial centroids
  - Select most widely separated data points
  - Add post-processing – merge some clusters
- Bisecting K-means (a little later)
  - Not as susceptible to initialization issues

Lecture Overview

1. Recap
2. **Modifications and R**
3. Bisecting K-means + Limitations
4. Intro to hierarchical clustering
5. Agglomerative Clustering
6. Min

Handling Empty Clusters

- Random initial choice K-means algorithm can yield empty clusters
- For single empty cluster there are strategies to rectify, e.g.
  - Choose the point that contributes most to SSE as a centroid for an empty cluster. Continue K-means
  - Choose a point from the cluster with highest SSE that has the highest SE as a centroid for an empty cluster
- If there are several empty clusters, the above can be repeated several times.

Updating Centers Incrementally

- In the basic K-means algorithm, centroids are updated after all points are assigned to a centroid
- An alternative is to update the centroids after each assignment (incremental approach)
  - Each assignment updates zero or two centroids
  - More expensive
  - Introduces an order dependency
  - Never gets an empty cluster
  - Can use “weights” to change the impact

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
  - Can use these steps during the clustering process

K-means in **R**

```
library(cluster)
data(wine, package='rattle')
head(wine) # observe the composition of different wines
wine.stand <- scale(wine[,-1]) #standardize the data = center and divide by st. dev.
wine.fit <- kmeans(wine.stand, 3)
attributes(wine.fit)
wine.fit$centers # cluster centers
wine.fit$cluster # vector of cluster assignments to data points
wine.fit$size  #vector of cluster sizes
wine.fit$tot.withinss # total within-cluster sum of squares
wine.fit$betweenss # between-cluster sum of squares
```

K-means in **R**

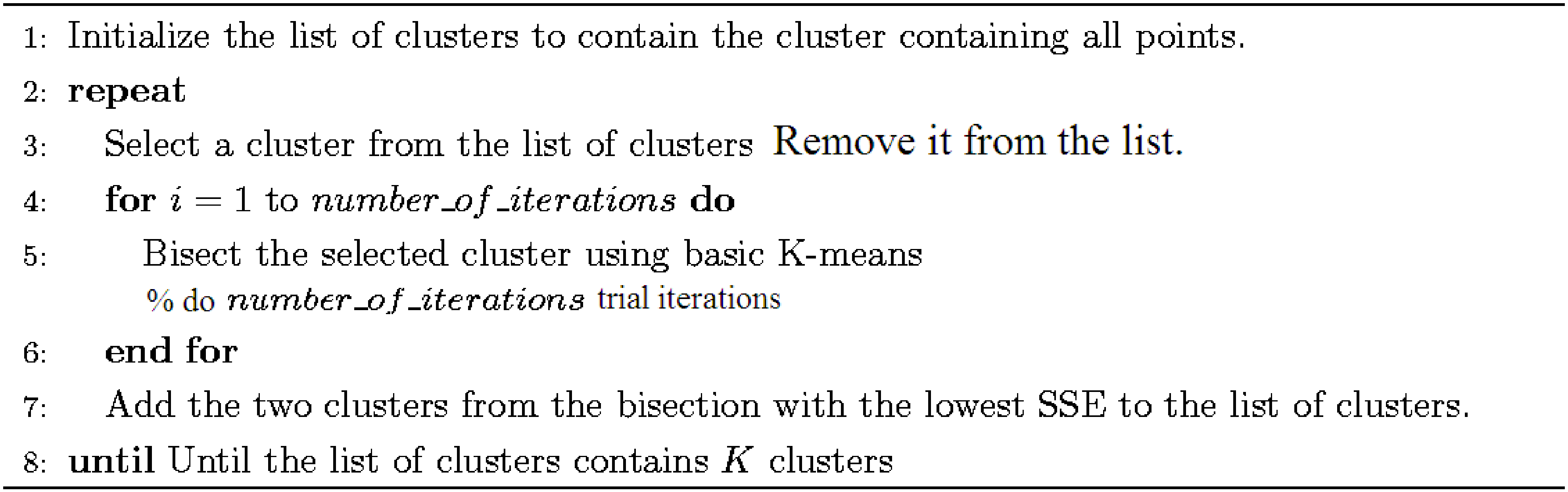
```
clusplot(wine.stand, wine.fit$cluster, main='2D representation of the Cluster solution', color=TRUE, shade=TRUE, labels=2, lines=0)
#plots data points assigned clusters in coord. Principal componet 1 and principal component 2
#lines=0 means line connecting cluster centers is not drawn
#color=True means  the cluster ellipses are colored with respect to their density
#shade=True means cluster eleipses are shaded
#label=2 points ids are on the plot
table(wine[,1],wine.fit$cluster)
means_c<-aggregate(scale(wine[,--1]),by=list(wine$Type),FUN=mean)
#compute multidimensional means by wine type
wine.fit$centers; means_c
# visually compare cluster centers and means of vine types
```

Lecture Overview

1. Recap
2. Modifications and R
3. **Bisecting K-means + Limitations**
4. Intro to hierarchical clustering
5. Agglomerative Clustering
6. Min

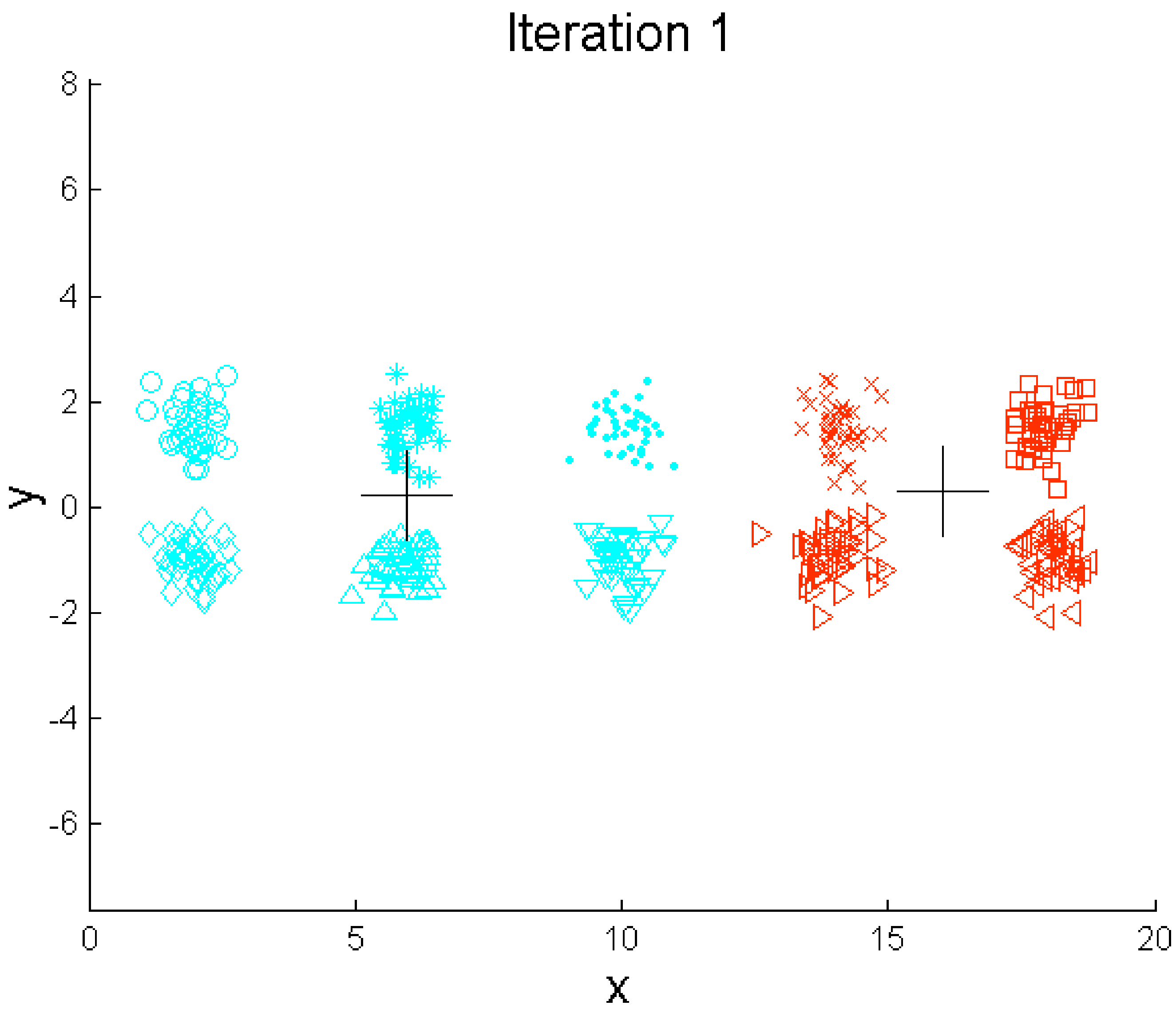
Bisecting K-means Algorithm

- Bisecting K-means algorithm
  - Variant of K-means that can produce a partitional or a hierarchical clustering

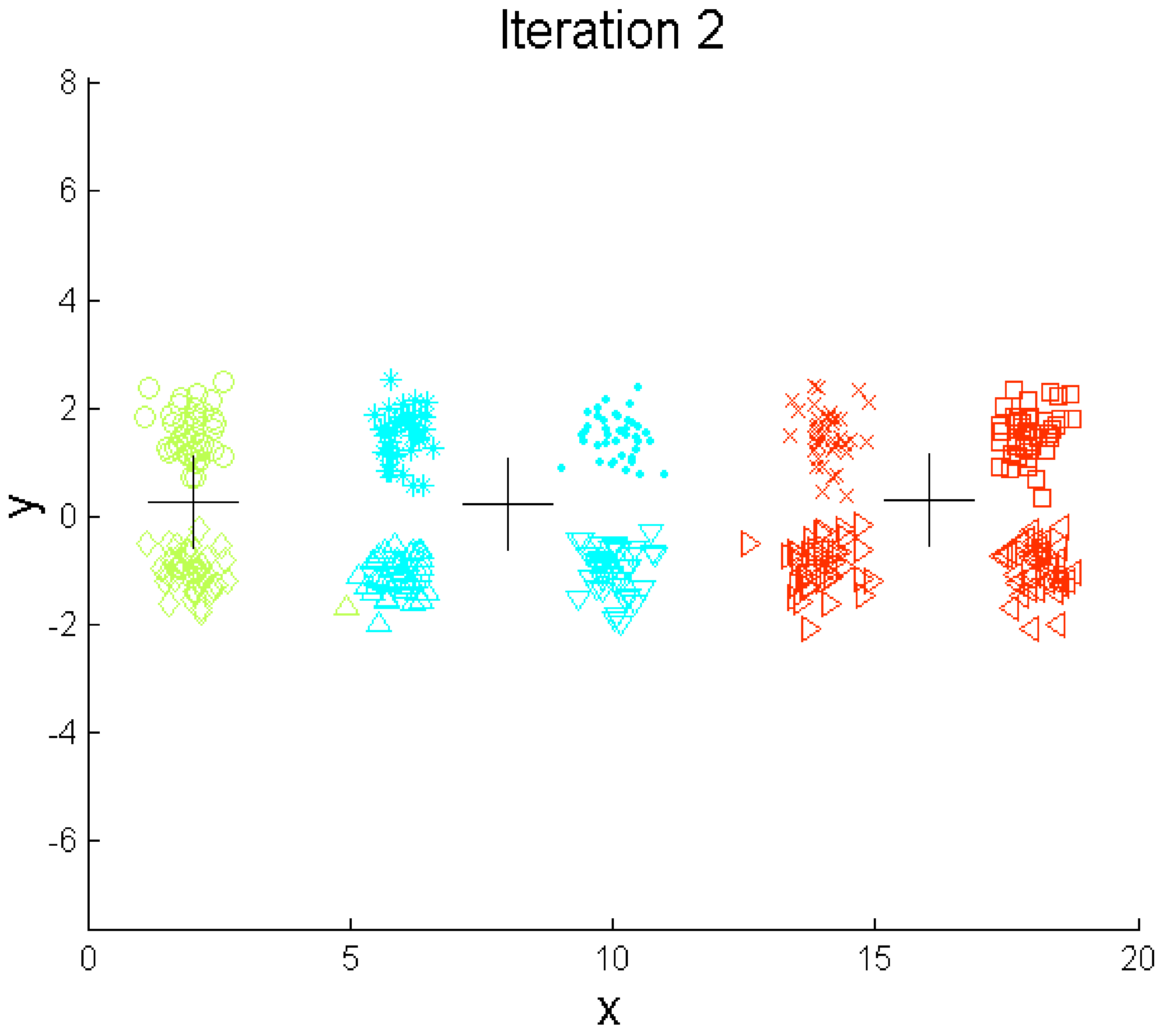


```
Initialize the list of clusters to contain the cluster containing all points.
repeat
    Select a cluster from the list of clusters Remove it from the list.
    for $i = 1$ to $number\_of\_iterations$ do
        Bisect the selected cluster using basic K-means
        % do $number\_of\_iterations$ trial iterations
    end for
    Add the two clusters from the bisection with the lowest SSE to the list of clusters.
until Until the list of clusters contains $K$ clusters
```

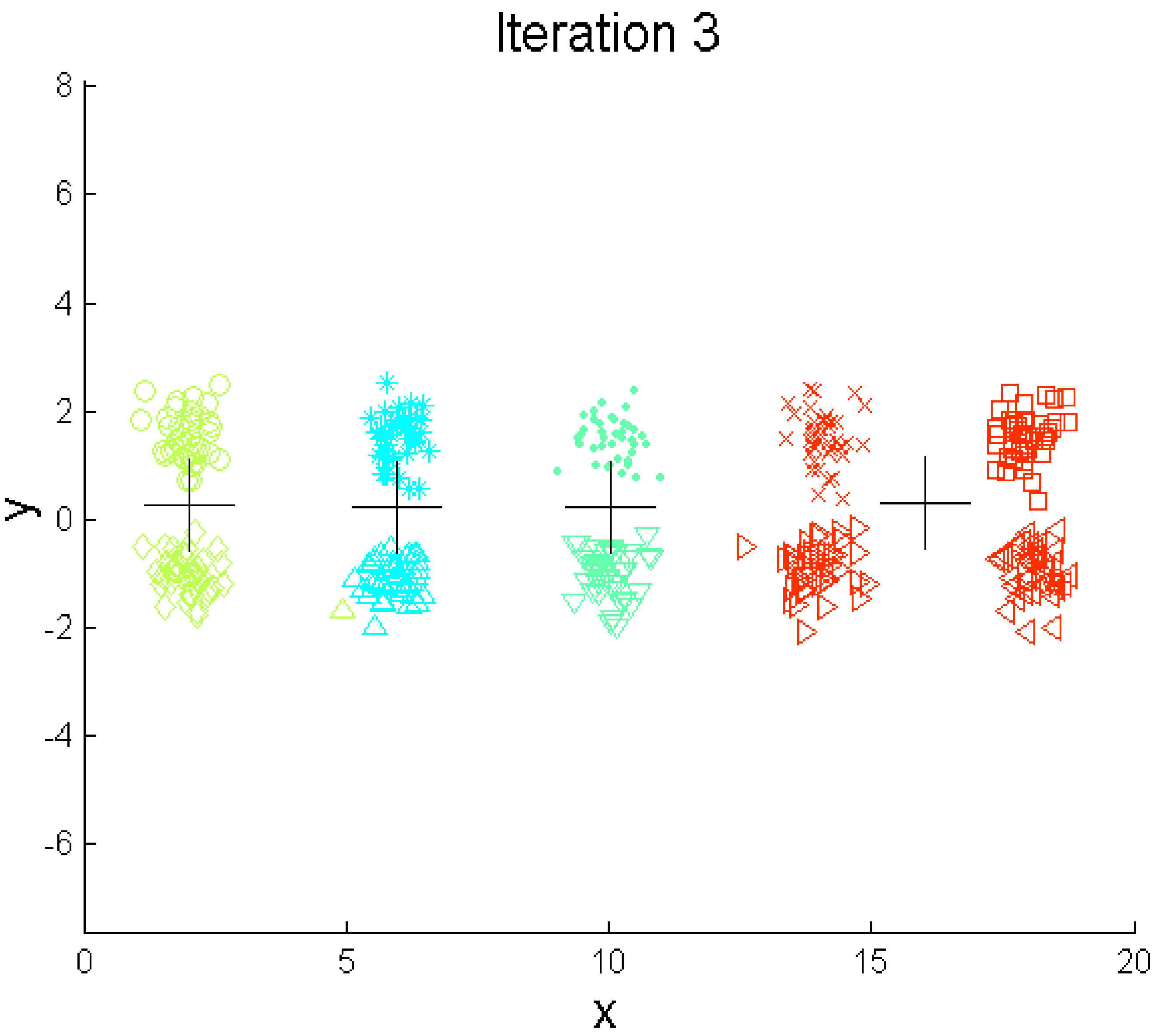
Bisecting K-means Example



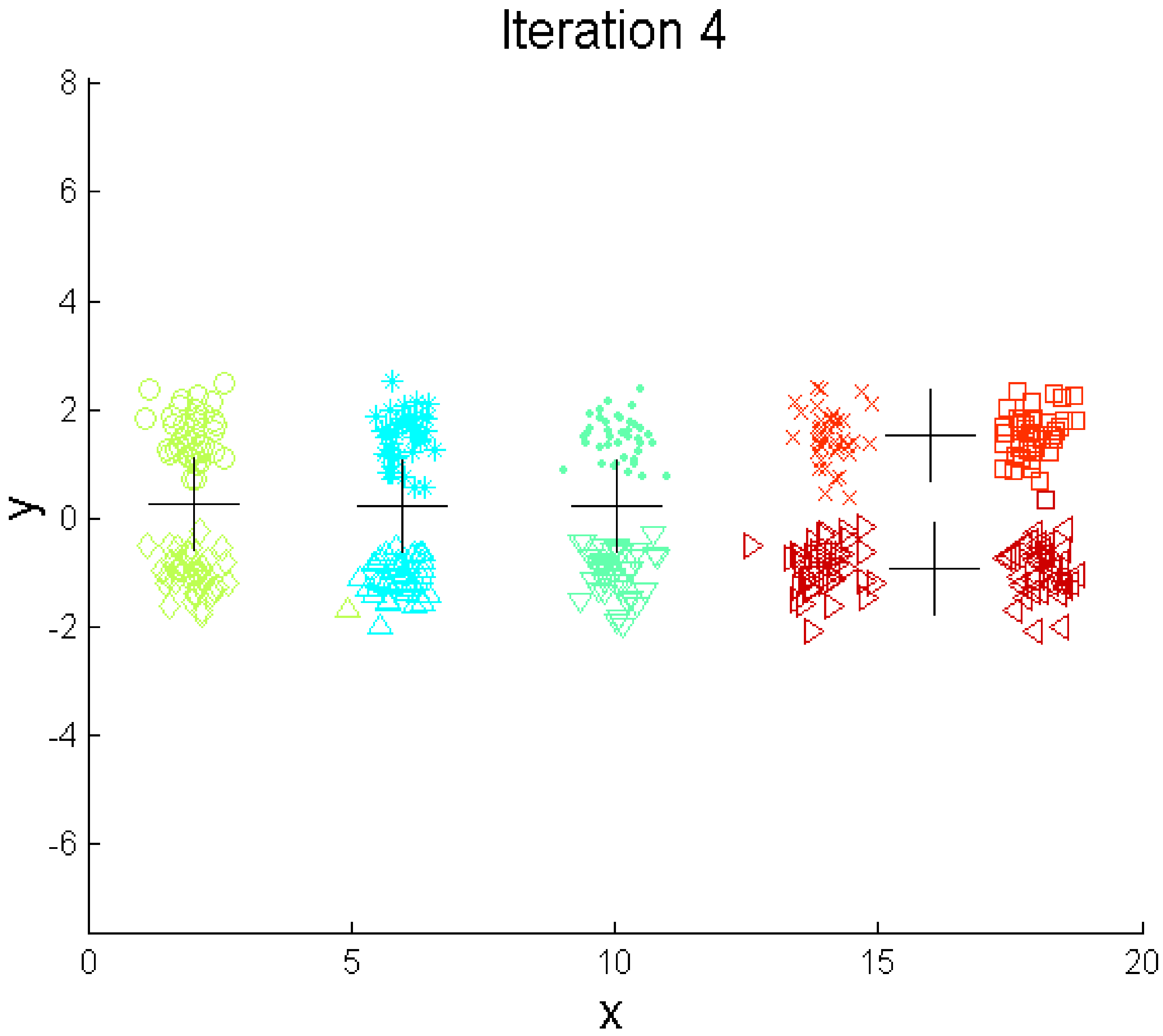
Bisecting K-means Example



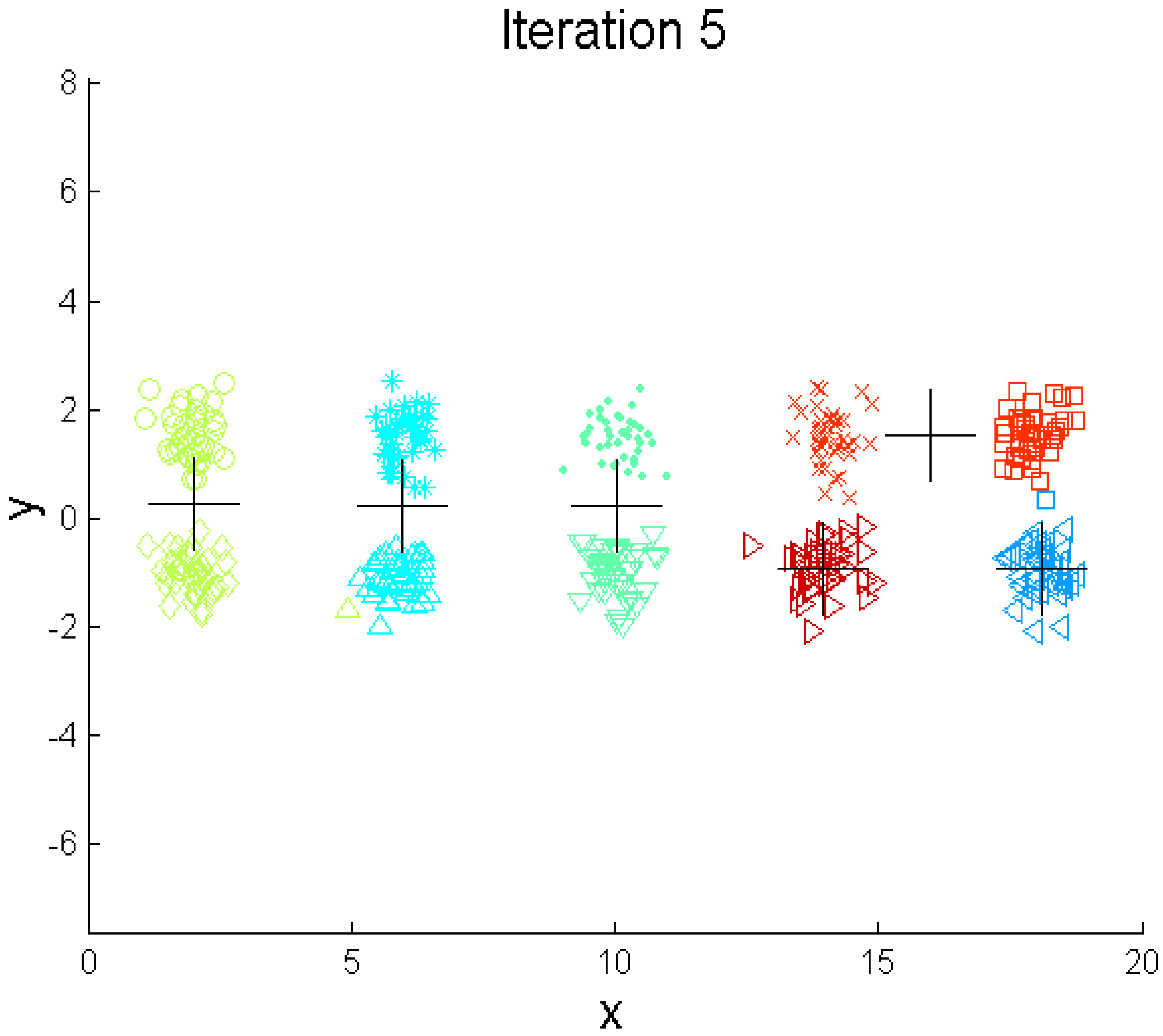
Bisecting K-means Example



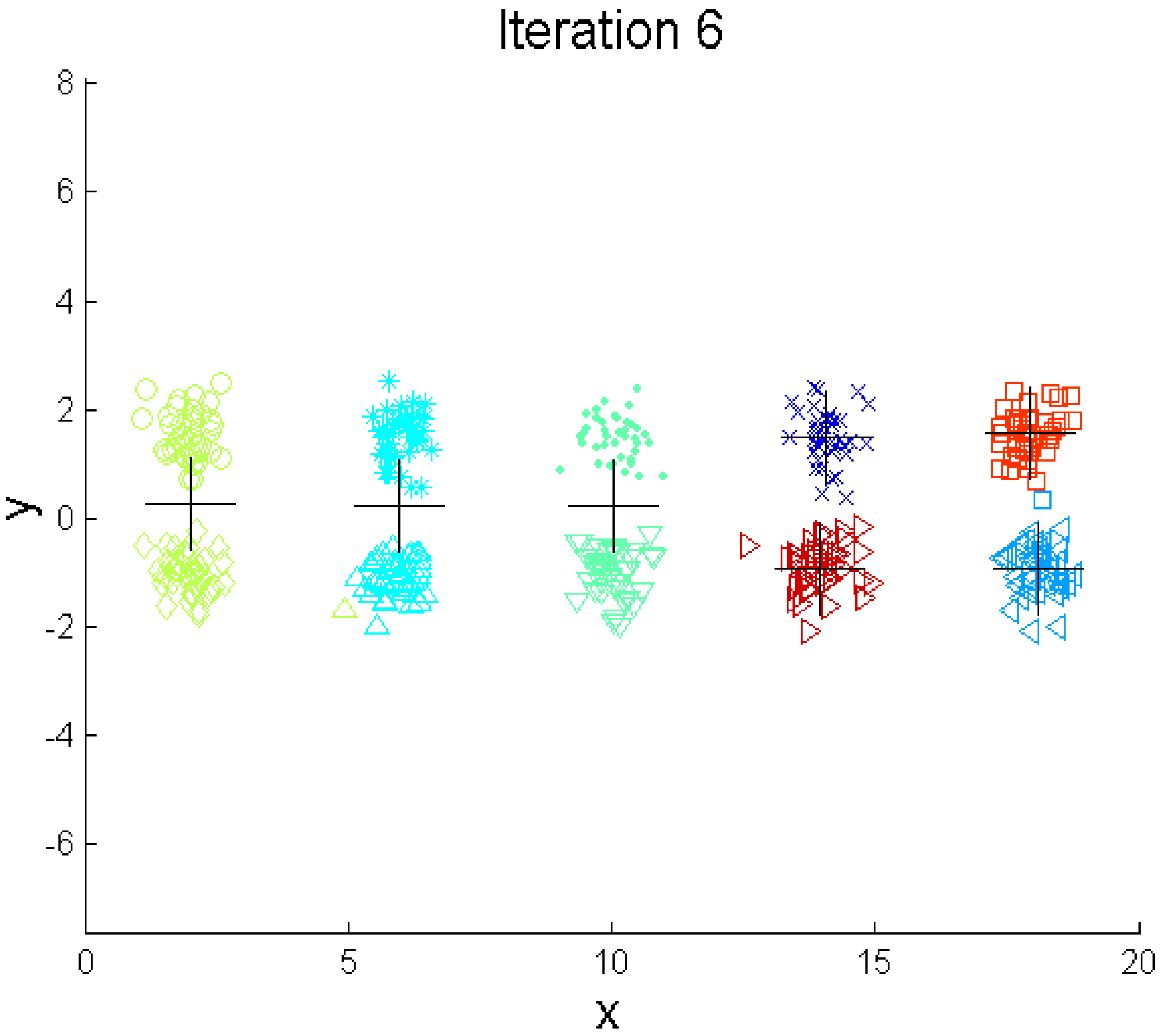
Bisecting K-means Example



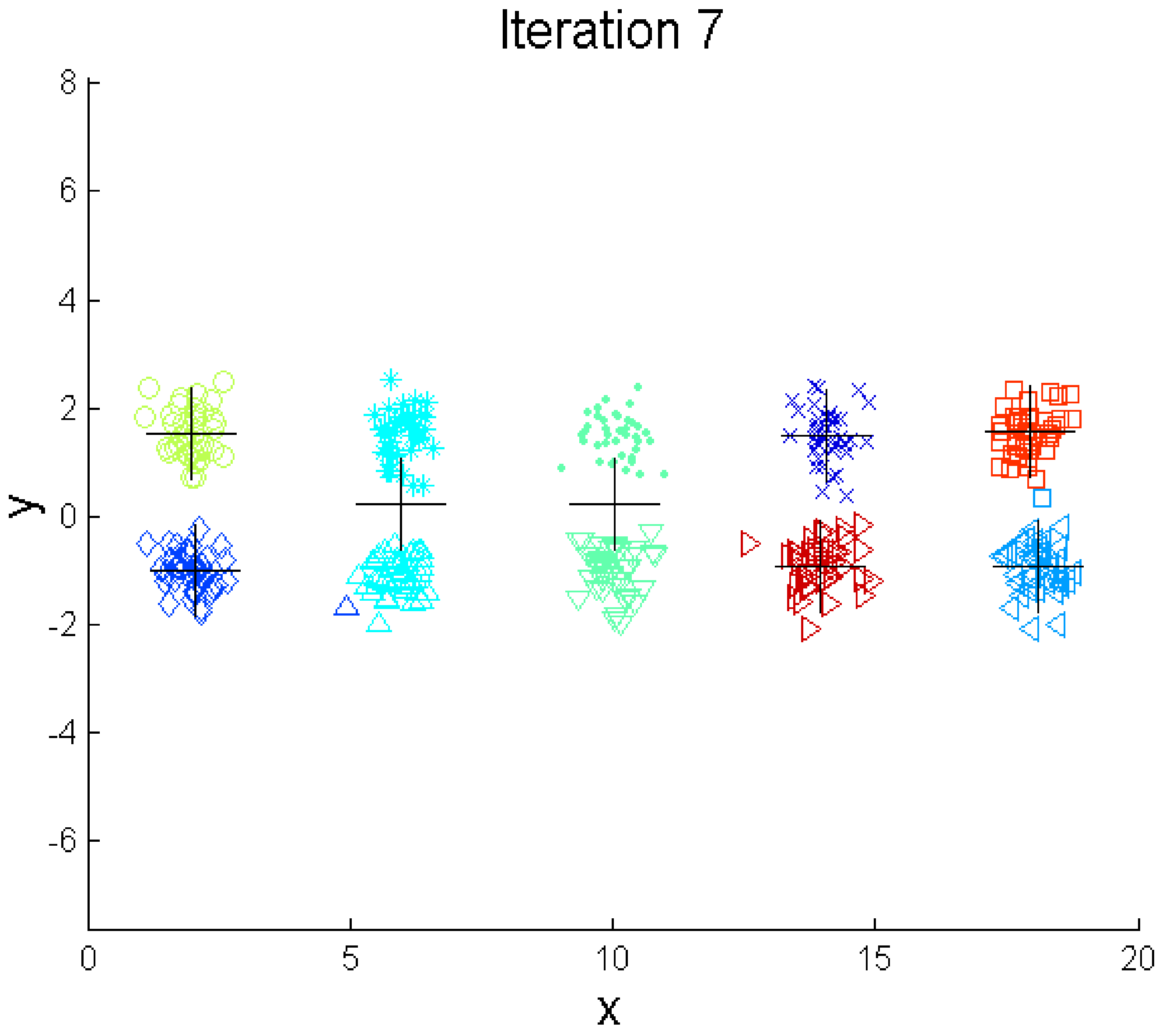
Bisecting K-means Example



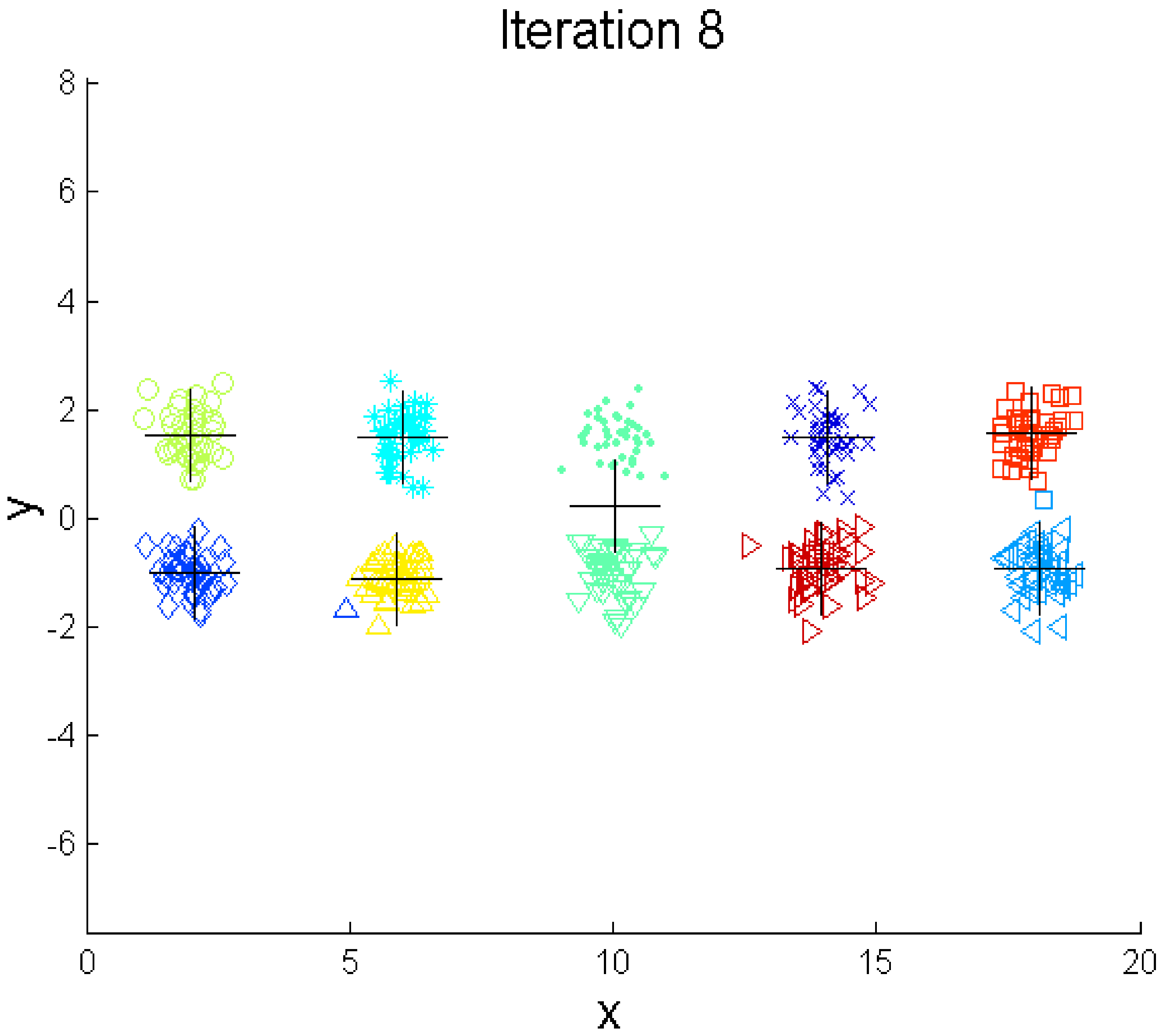
Bisecting K-means Example



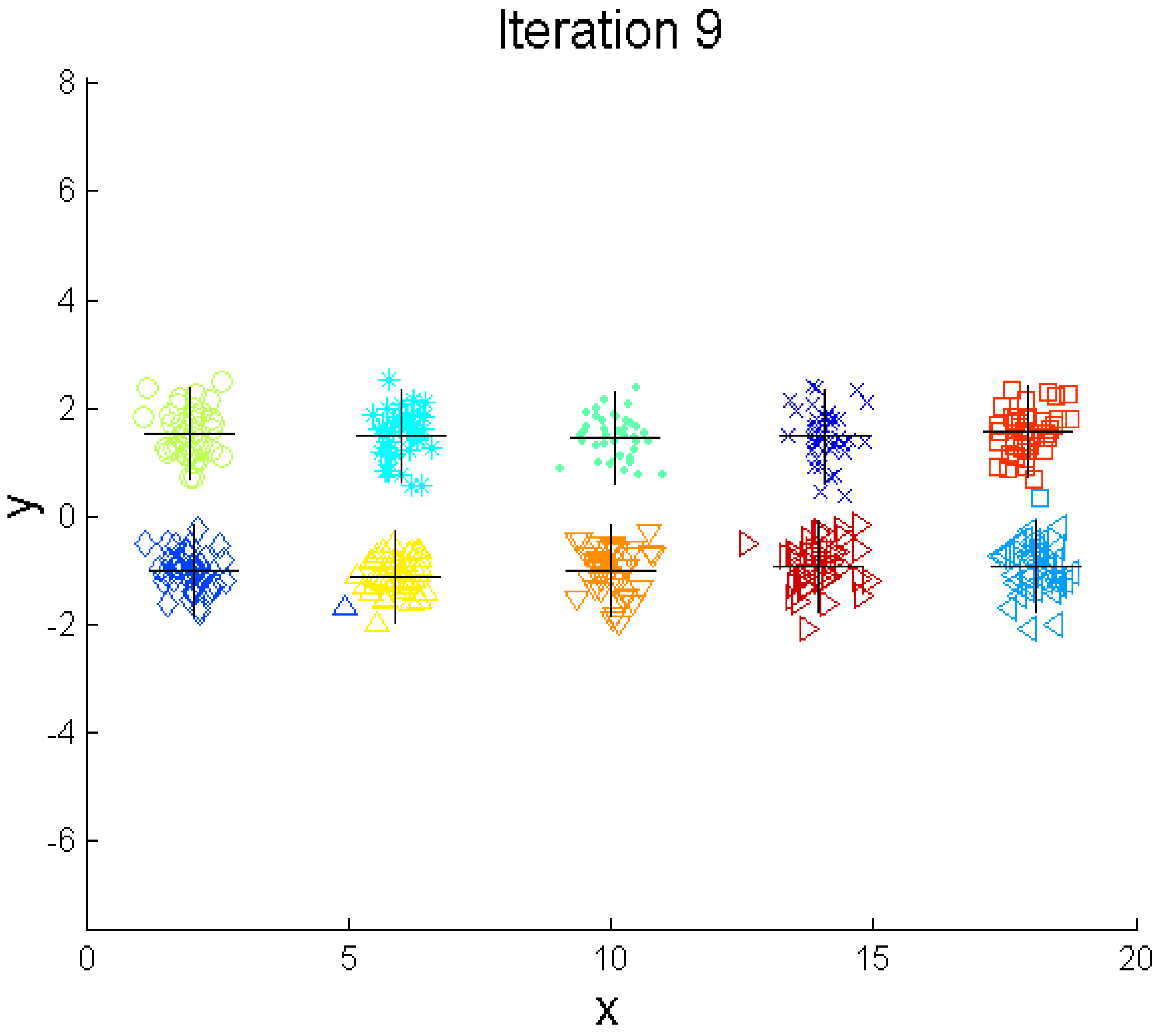
Bisecting K-means Example



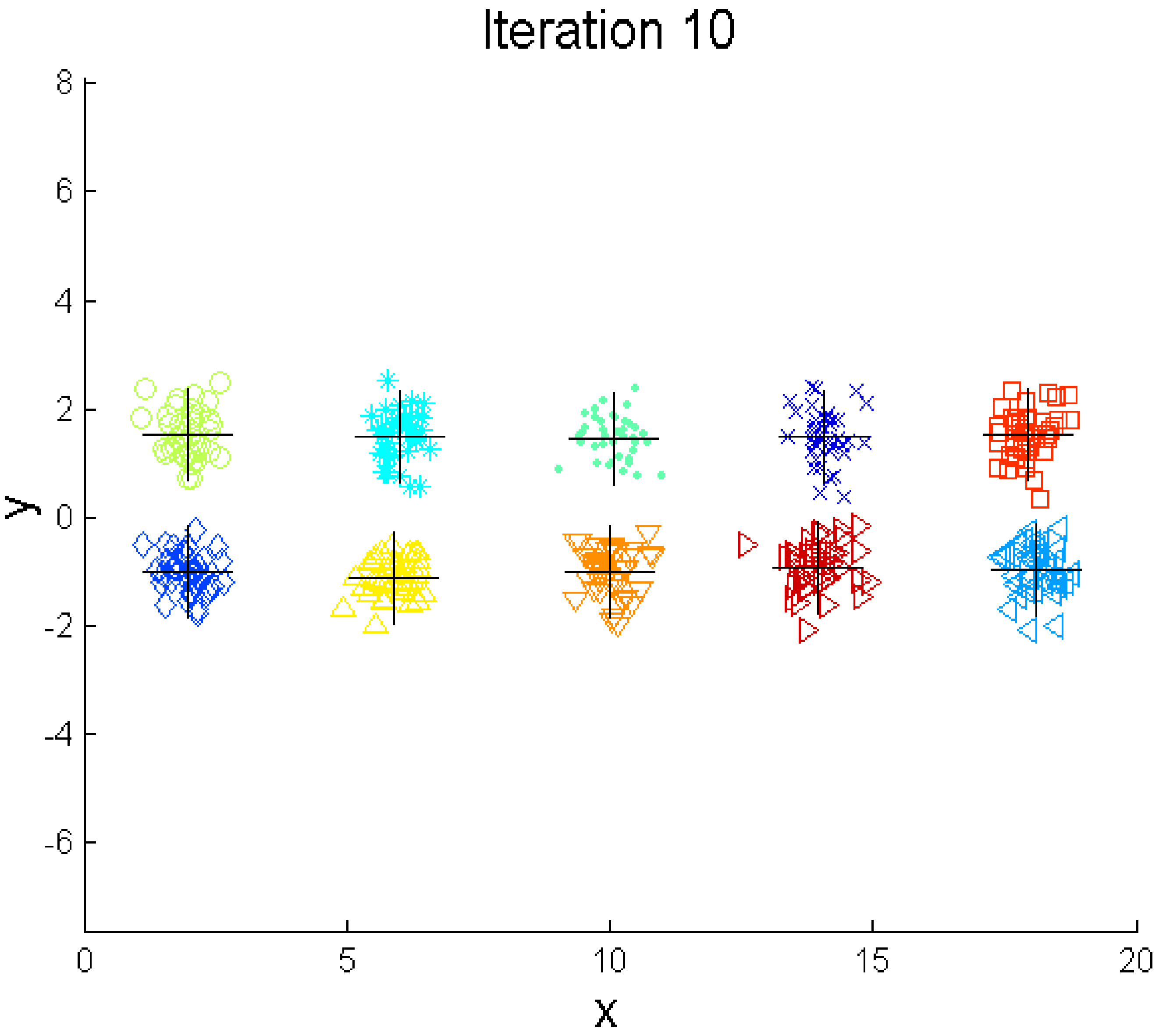
Bisecting K-means Example



Bisecting K-means Example

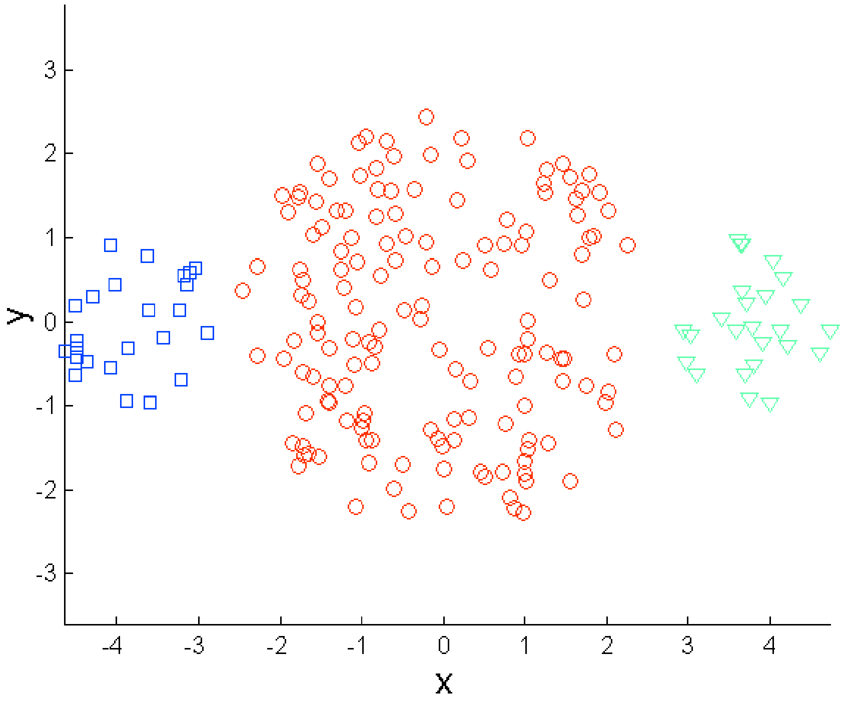


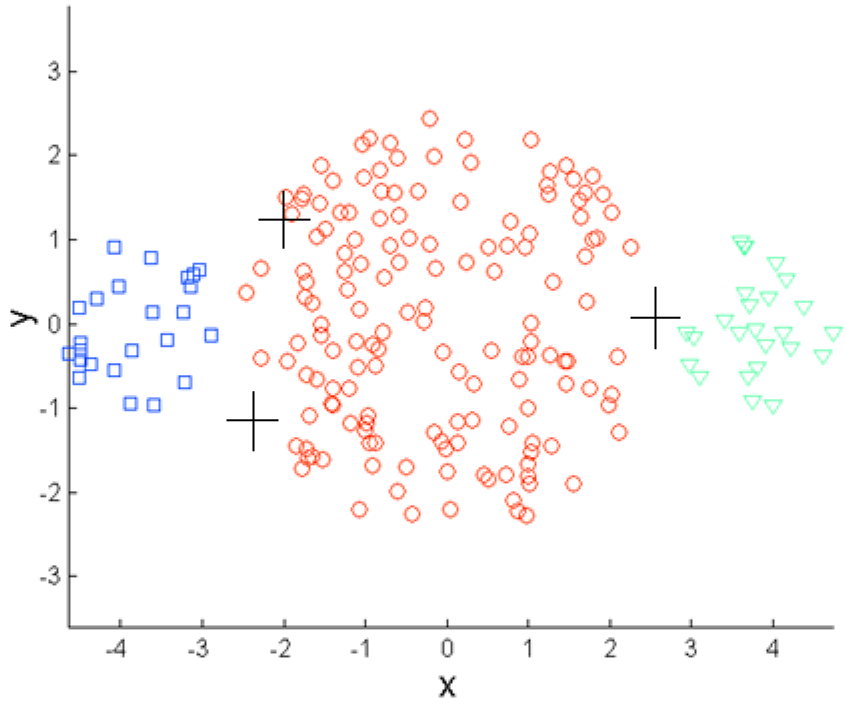
Bisecting K-means Example



Limitations of K-means

- K-means has problems when clusters are of differing
  - **Sizes**



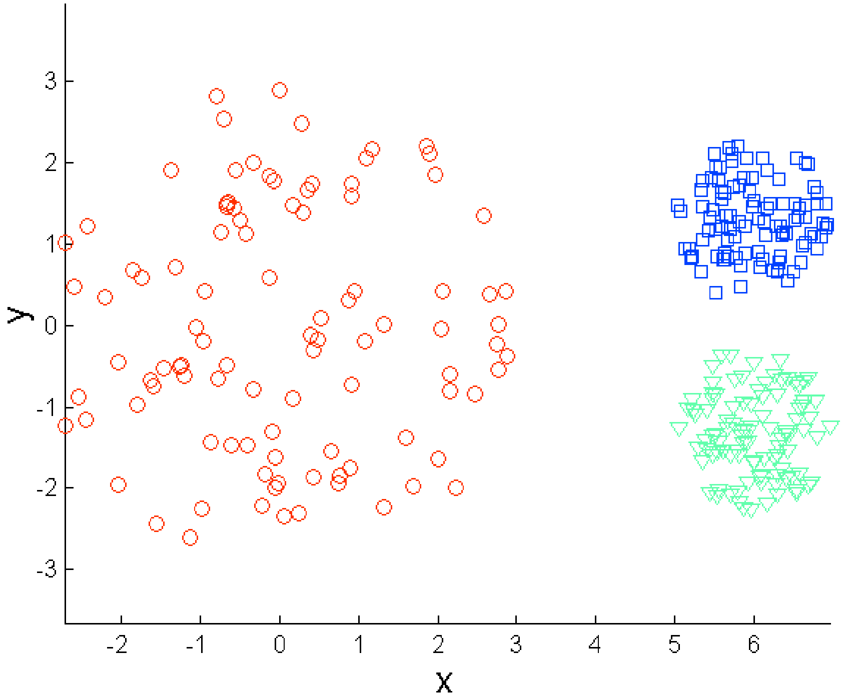


Original Points

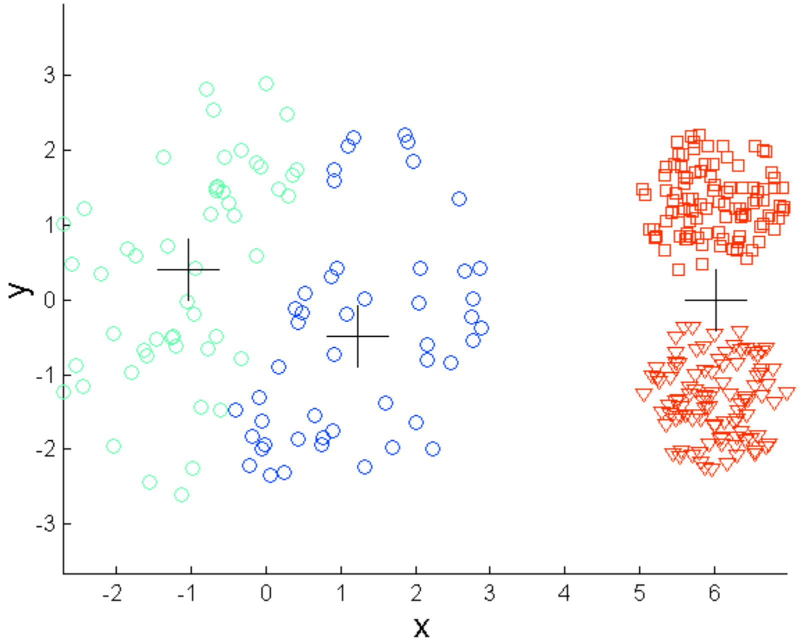
3-means Cluster Centers

Limitations of K-means

- K-means has problems when clusters are of differing
  - Sizes
  - **Densities**



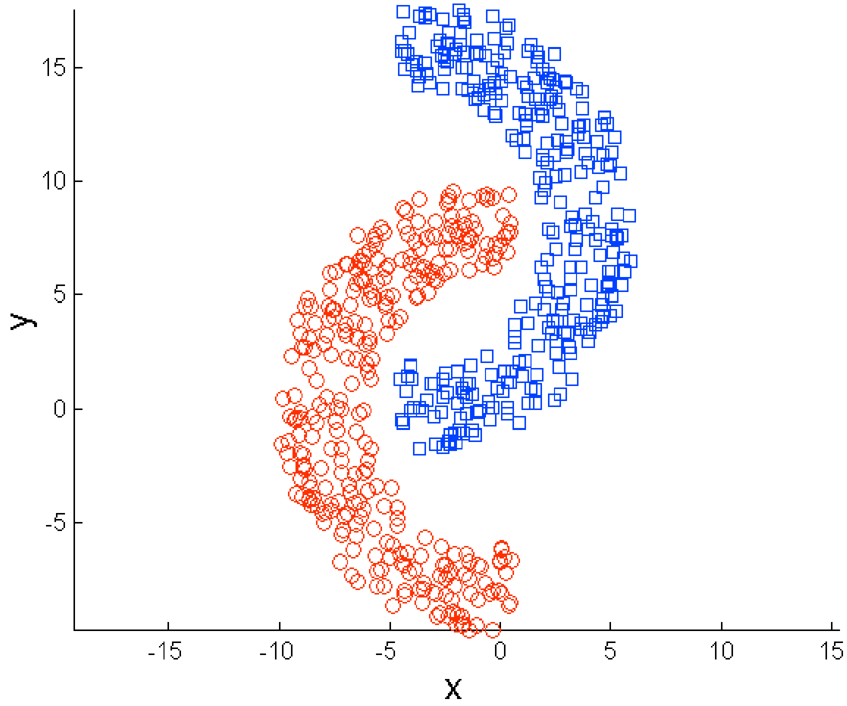
Original Points



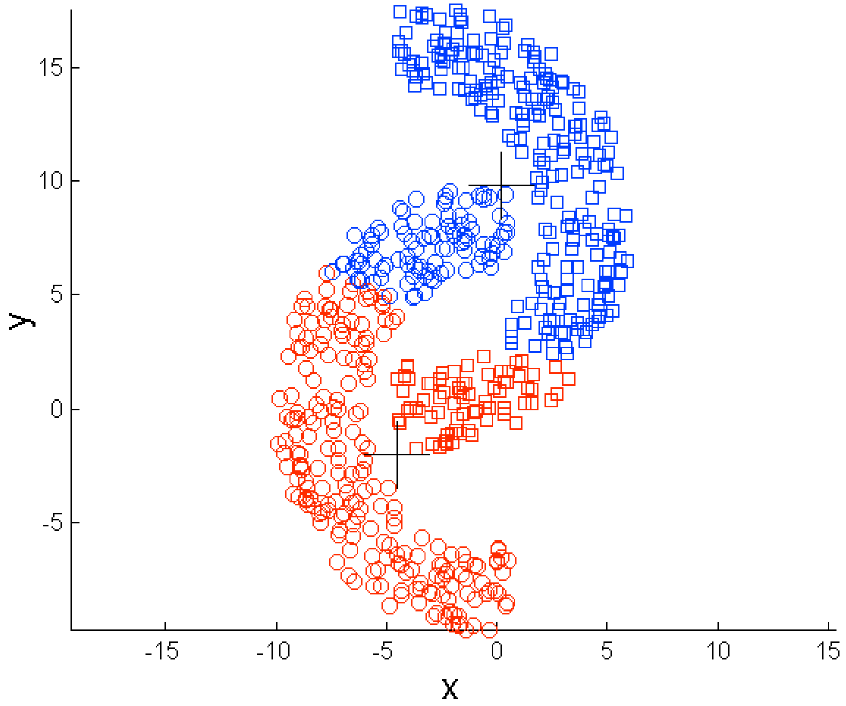
3-means Cluster Centers

Limitations of K-means

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



Original Points

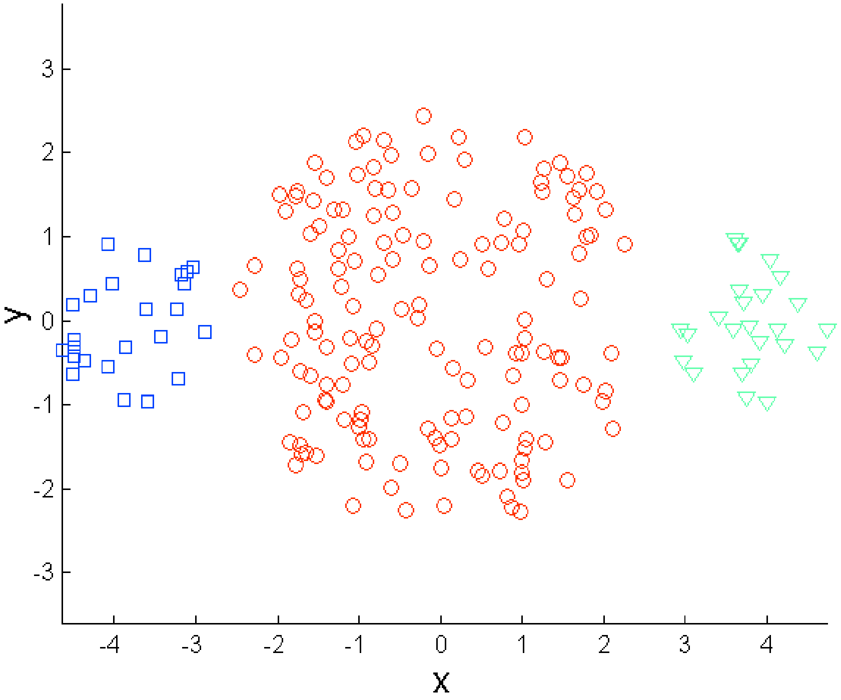


2-means Cluster Centers

Limitations of K-means

- K-means has problems when clusters are of differing
  - Sizes
  - Densities
  - Non-globular shapes
- Possible solution: start with bisecting k means, increasing number of centroids to over $2t > K$, and then join subclasses manually

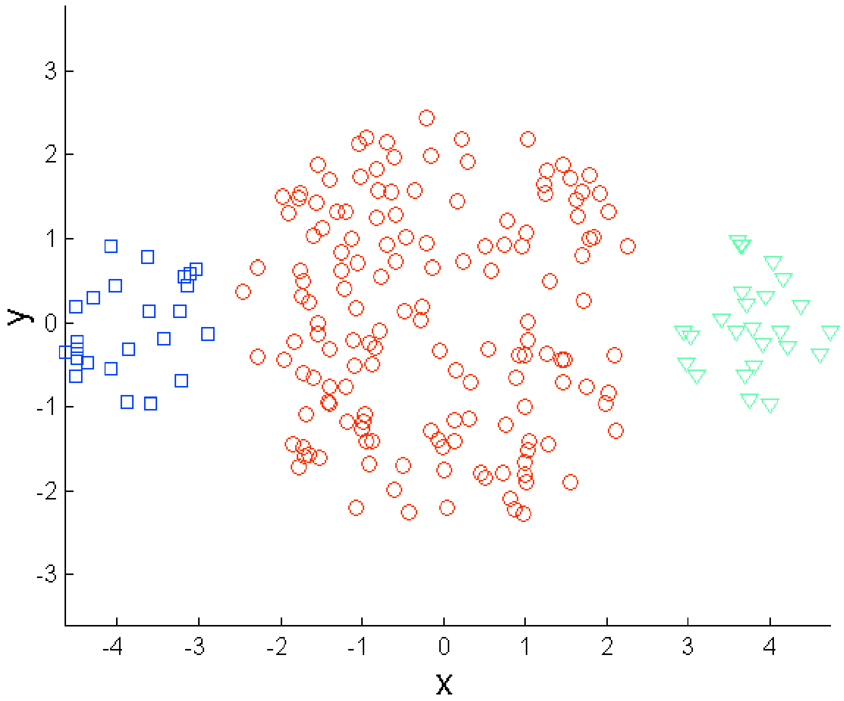
Overcoming Size Difference: Over-clustering

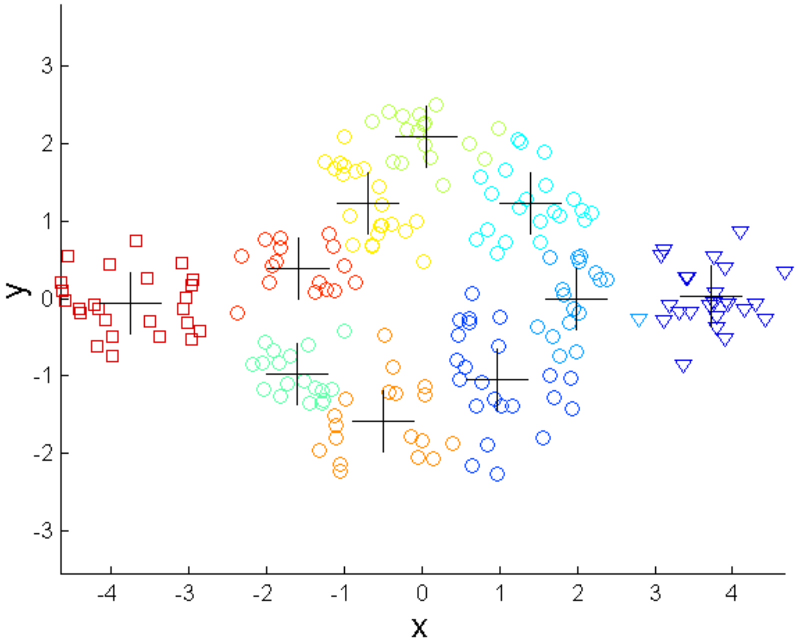


Original Points

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

Overcoming Size Difference: Over-clustering



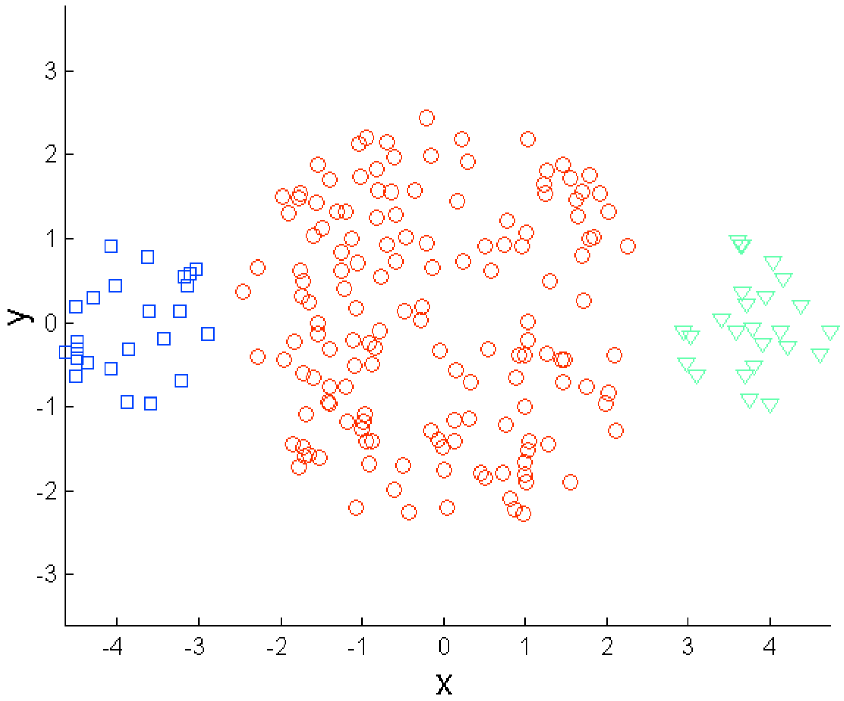


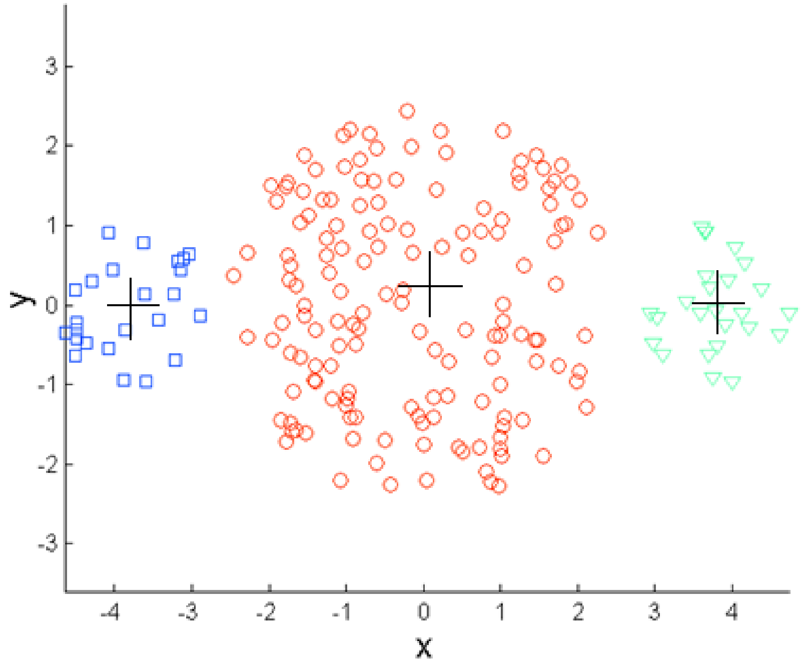
Original Points

10-means Clusters

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

Overcoming Size Difference: Over-clustering



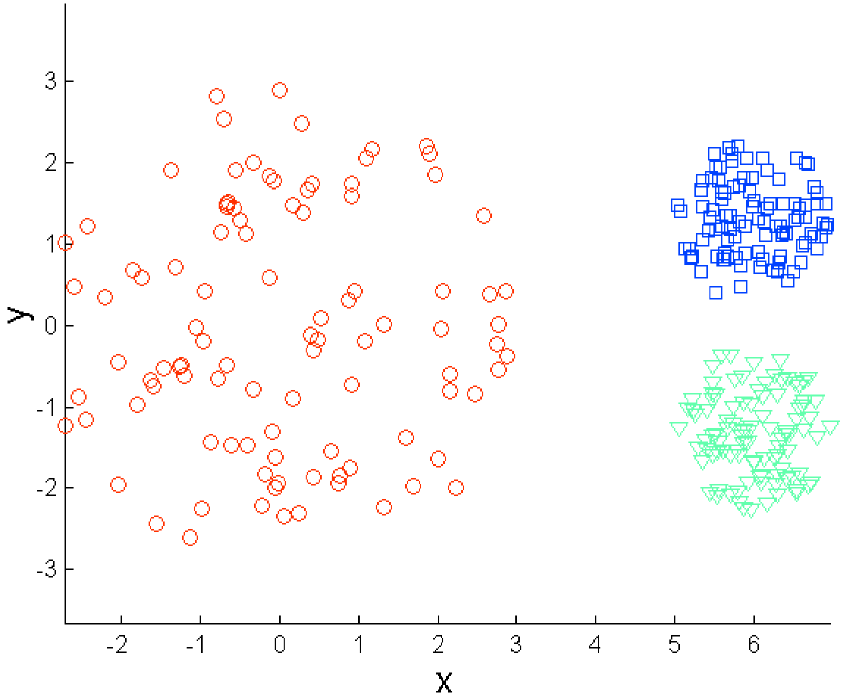


Original Points

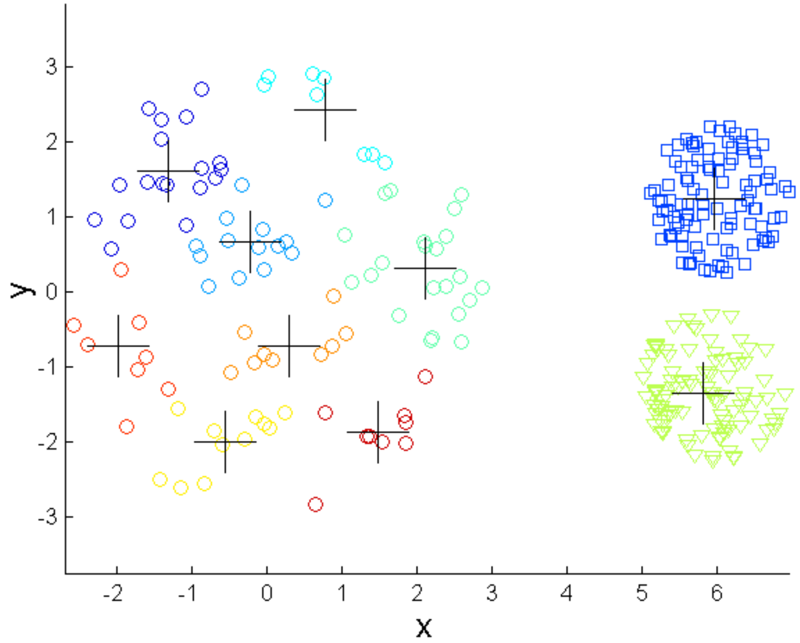
Merged 3-means Clusters

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

Overcoming Density Difference: Over-clustering



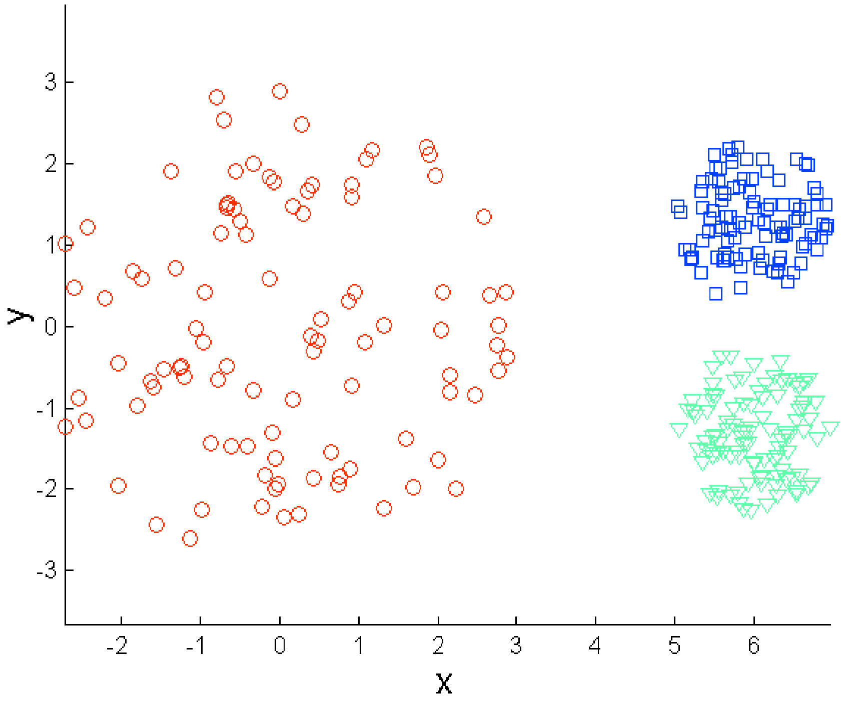
Original Points

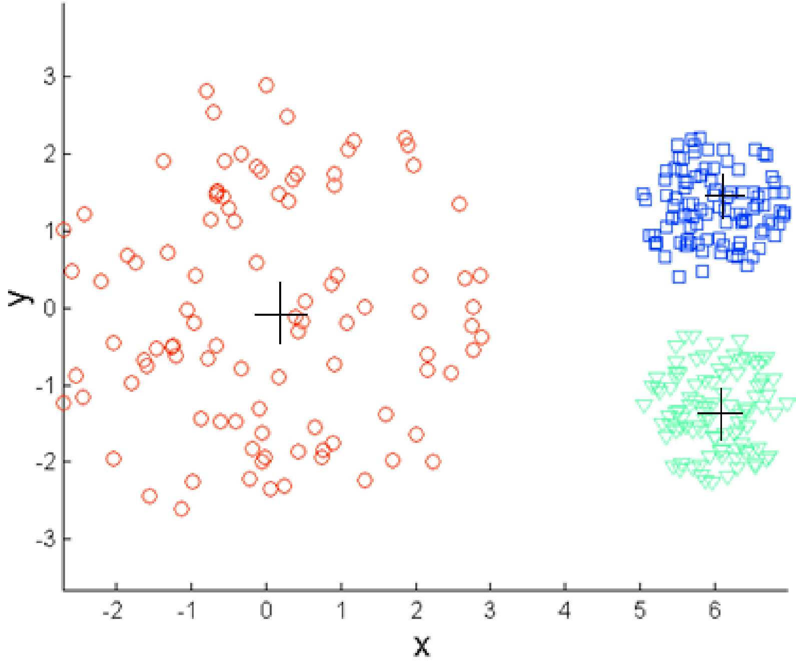


9-means Clusters

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

Overcoming Density Difference: Over-clustering

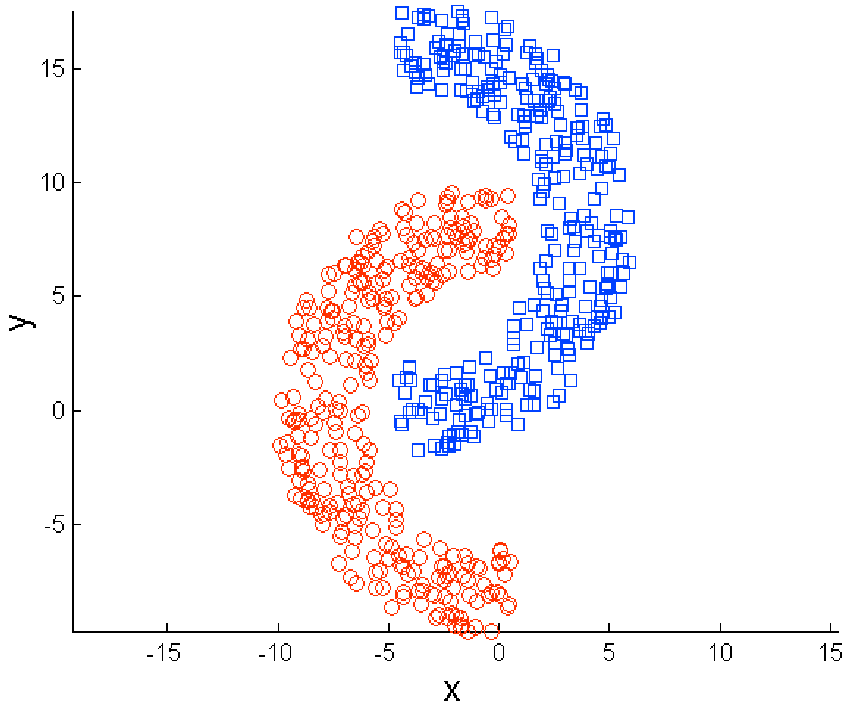


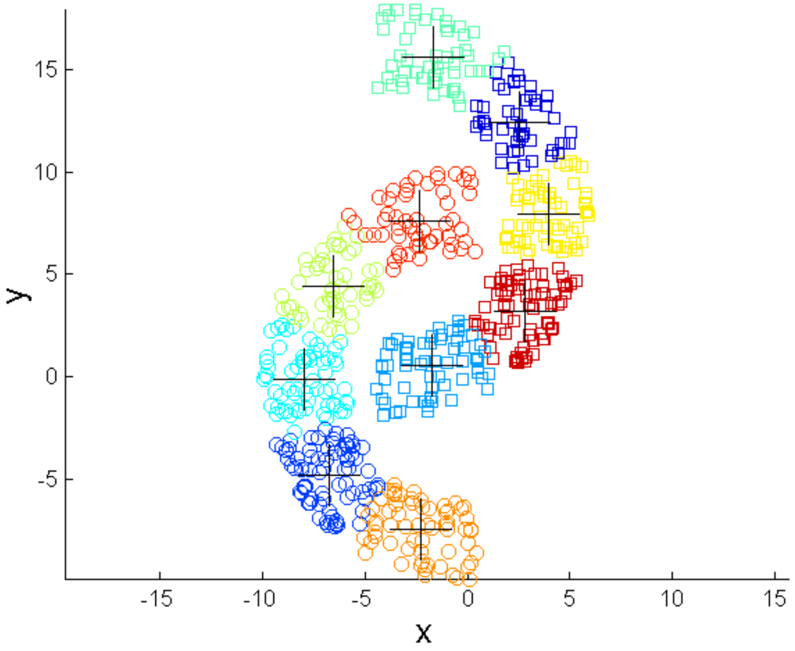


Original Points **Merged 3-means Clusters**

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

Overcoming Shapes: Over-clustering





Original Points

10-means Clusters

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

Lecture Overview

1. Recap
2. Modifications and R
3. Bisecting K-means + Limitations
4. Intro to hierarchical clustering
5. Agglomerative Clustering
6. Min

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
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)

Hierarchical Clustering

- Two main types of hierarchical clustering:
  1. 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
  2. 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

Lecture Overview

1. Recap
2. Modifications and R
3. Bisecting K-means + Limitations
4. Intro to hierarchical clustering
5. Agglomerative Clustering
6. Min

Agglomerative Clustering Algorithm

- More popular hierarchical clustering technique
- Basic algorithm is straightforward:

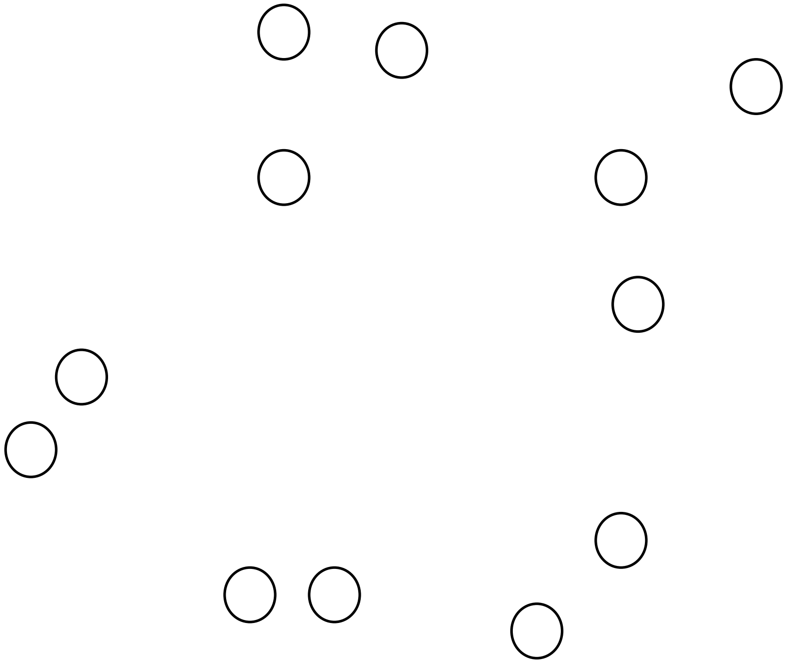
```
1.  Compute the proximity matrix
     Let each data point be a cluster
2.  Repeat
      i.   Merge the two closest clusters
      ii.  Update the proximity matrix
      Until only a single cluster remains
```

Agglomerative Clustering -continued

- Key operation is the computation of the proximity of two clusters
  - Different approaches to defining the distance between clusters distinguish the different algorithms

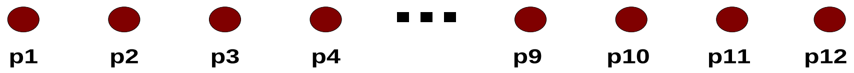
Starting Situation

- Start with clusters of individual points and a proximity matrix computed w.r.t. a predefined measure (distance or similarity)



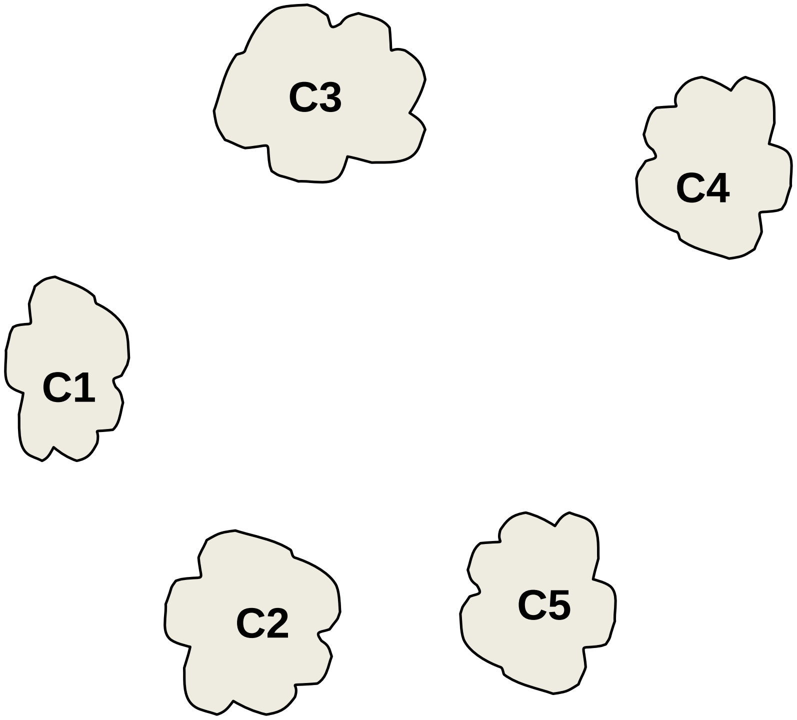
	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix



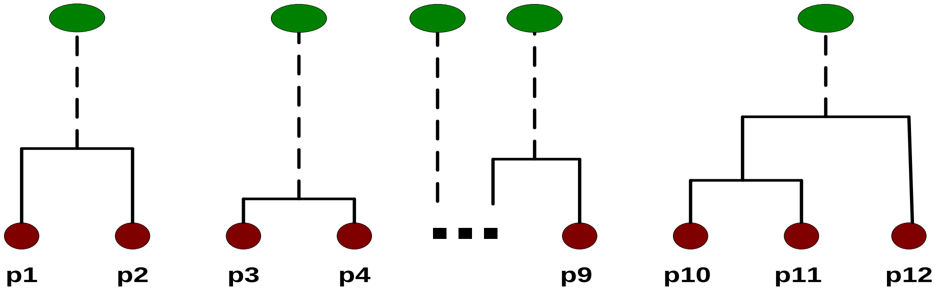
Intermediate Situation

- After some merging steps, we have some clusters



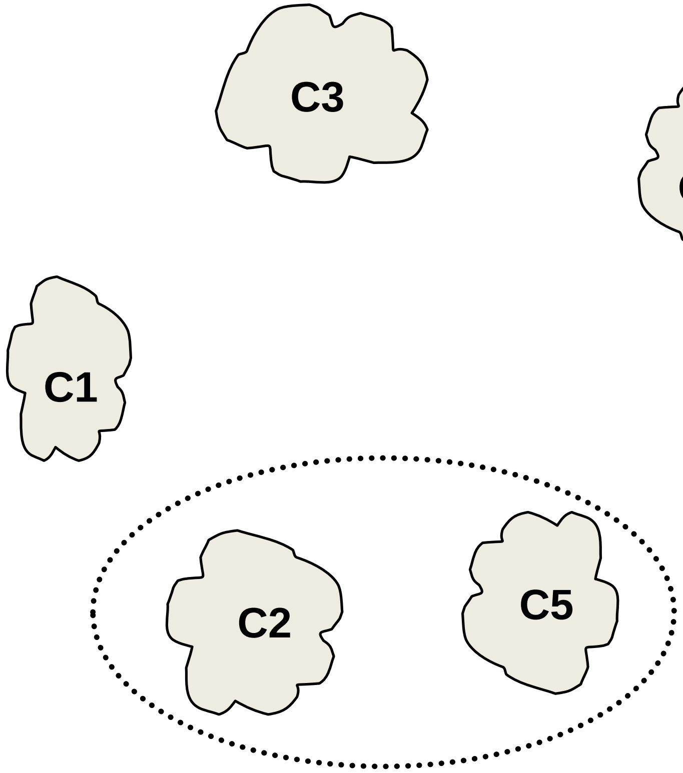
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

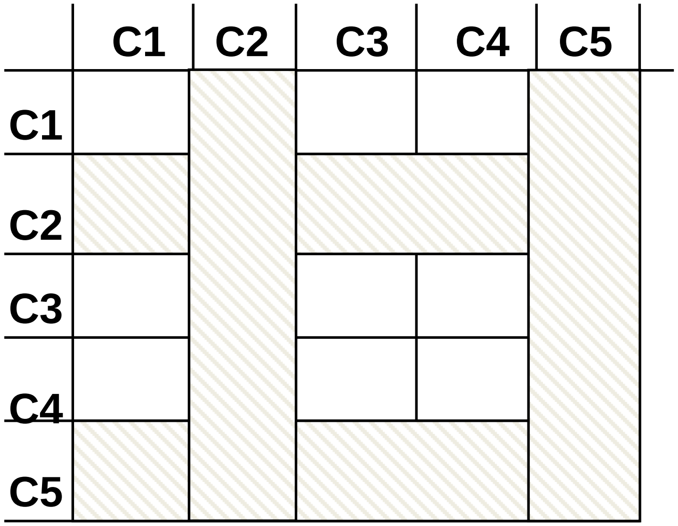
Proximity Matrix



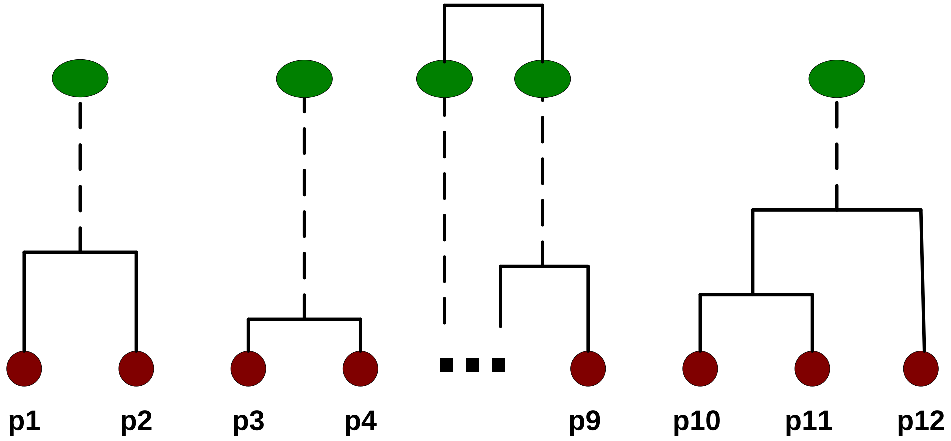
Intermediate Situation

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





Proximity Matrix

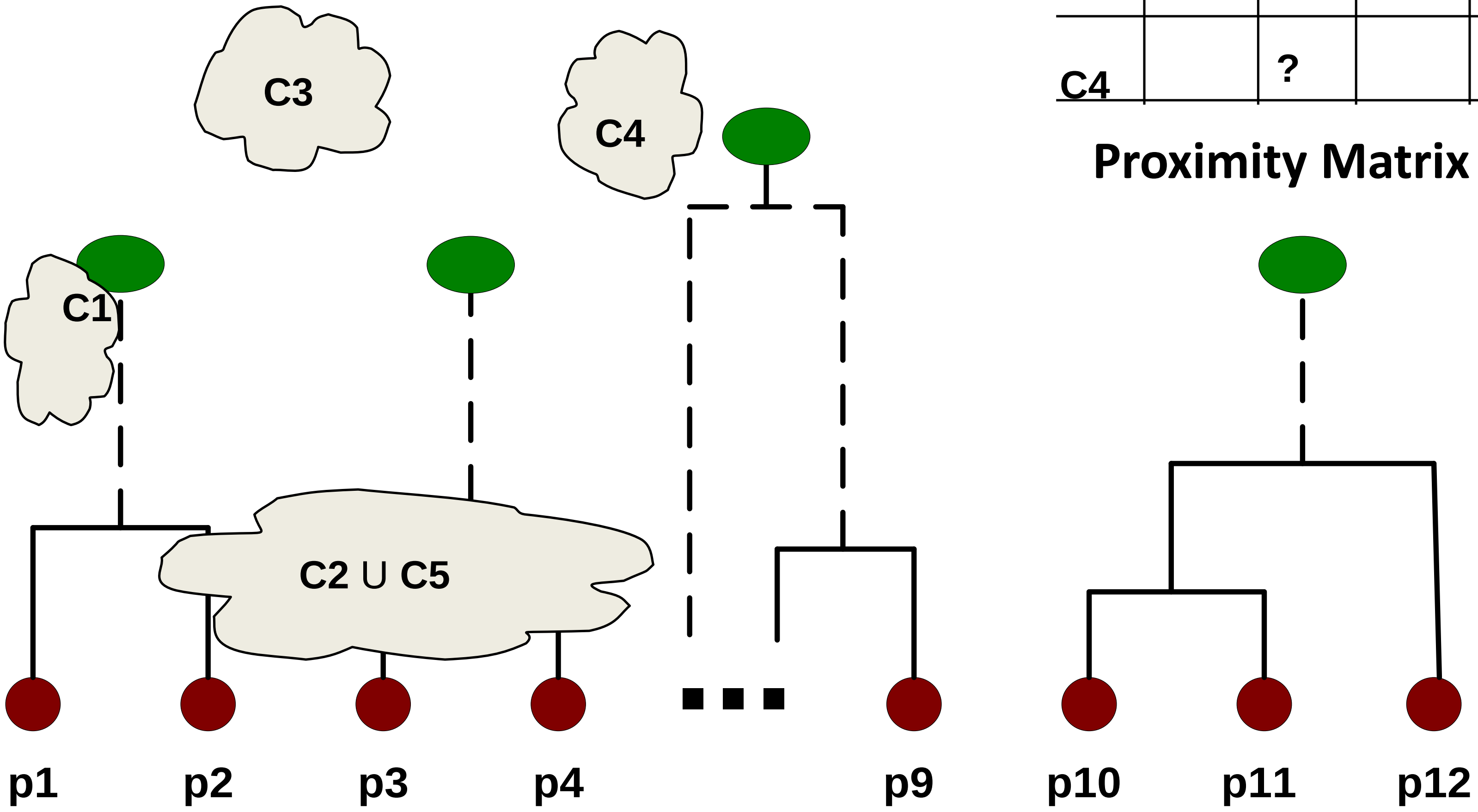


After Merging

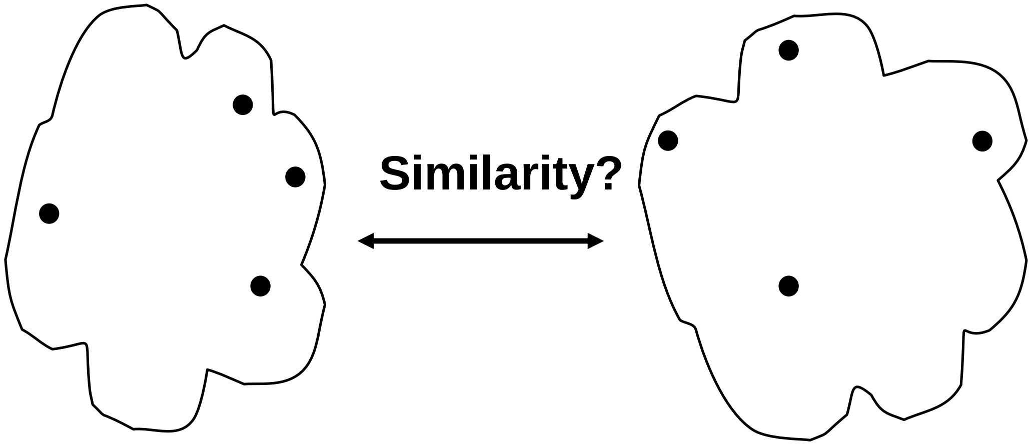
- The question is “How do we update the proximity matrix?”

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

Proximity Matrix



How to Define Inter-Cluster Similarity?



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

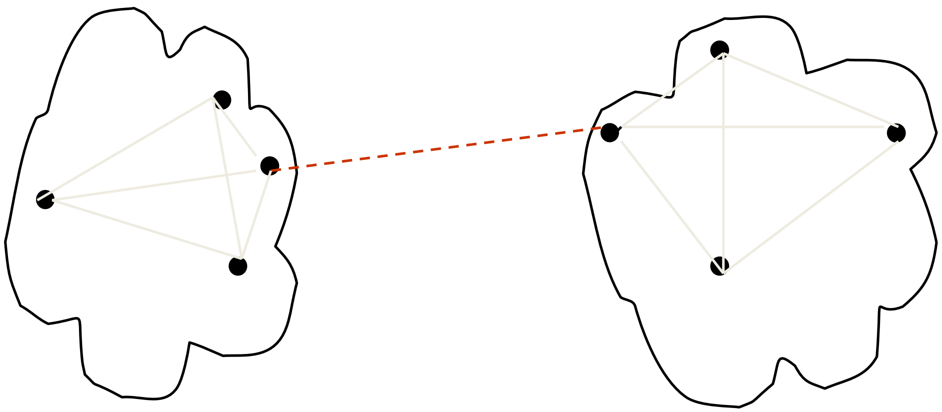
Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward’s Method uses squared error

Lecture Overview

1. Recap
2. Modifications and R
3. Bisecting K-means + Limitations
4. Intro to hierarchical clustering
5. Agglomerative Clustering
6. Min

Cluster Similarity – Min



Find shortest edge (line) between members of different clusters

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

Proximity Matrix

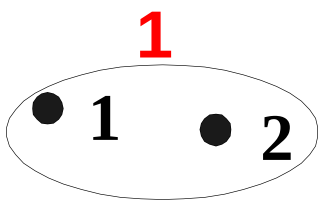
MIN Dendrogram Computation

- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph.

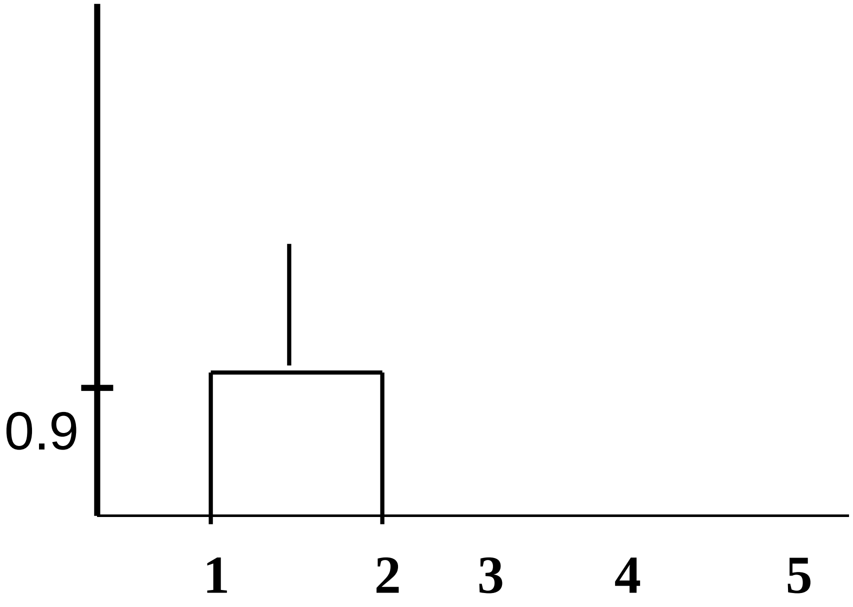
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00

MIN Dendrogram Computation

- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph.

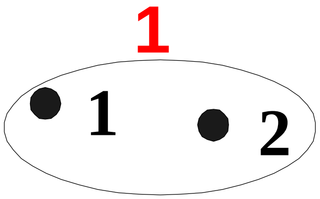


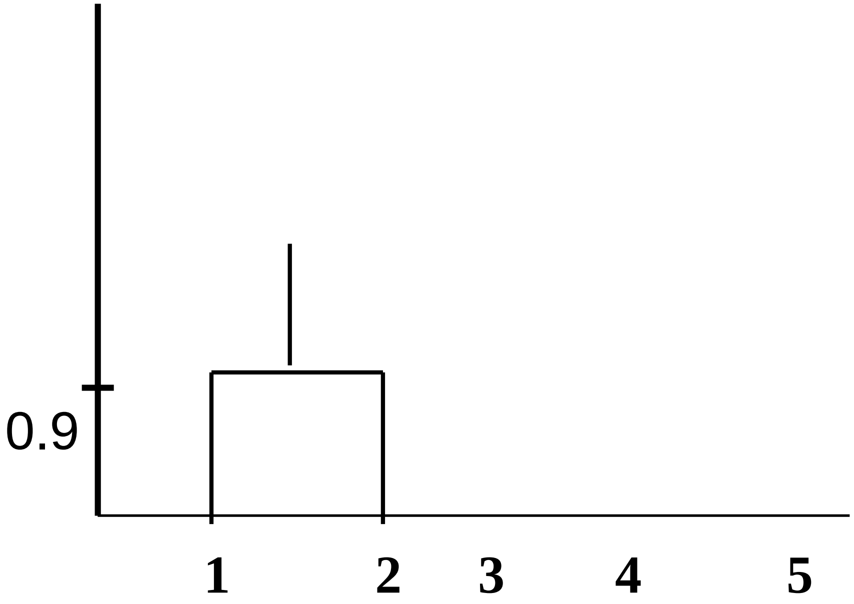
	I1	I2	I3	I4	I5
I1	1.00	0.90	0.10	0.65	0.20
I2	0.90	1.00	0.70	0.60	0.50
I3	0.10	0.70	1.00	0.40	0.30
I4	0.65	0.60	0.40	1.00	0.80
I5	0.20	0.50	0.30	0.80	1.00



MIN Dendrogram Computation

- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph.

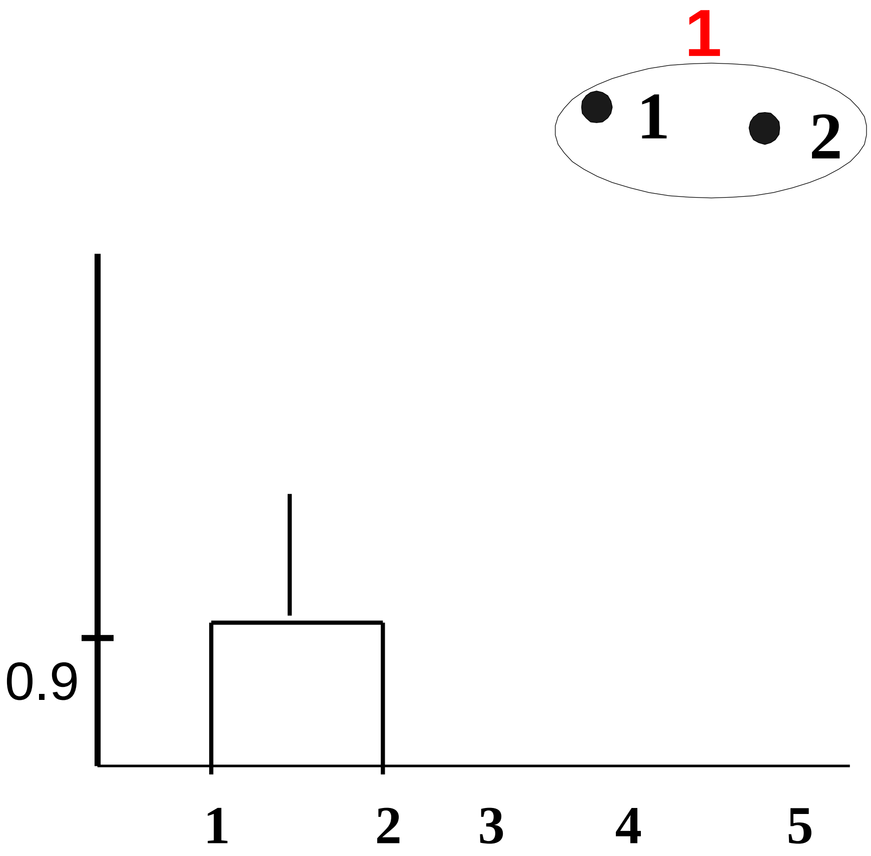




	I1	I2	I3	I4	I5
I1	1.00	1.00	0.70	0.65	0.50
I2	1.00	1.00	0.70	0.65	0.50
I3	0.70	0.70	1.00	0.40	0.30
I4	0.65	0.65	0.40	1.00	0.80
I5	0.50	0.50	0.30	0.80	1.00

MIN Dendrogram Computation

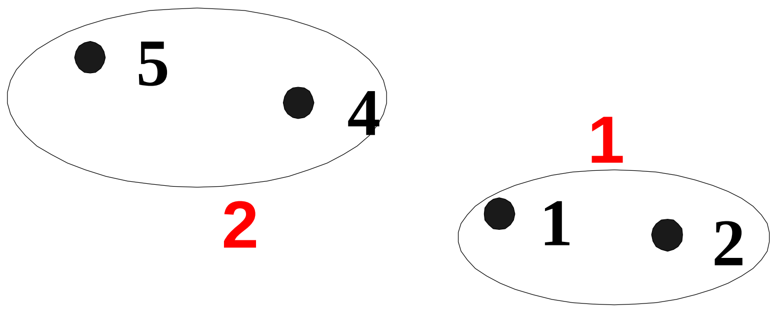
- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph.



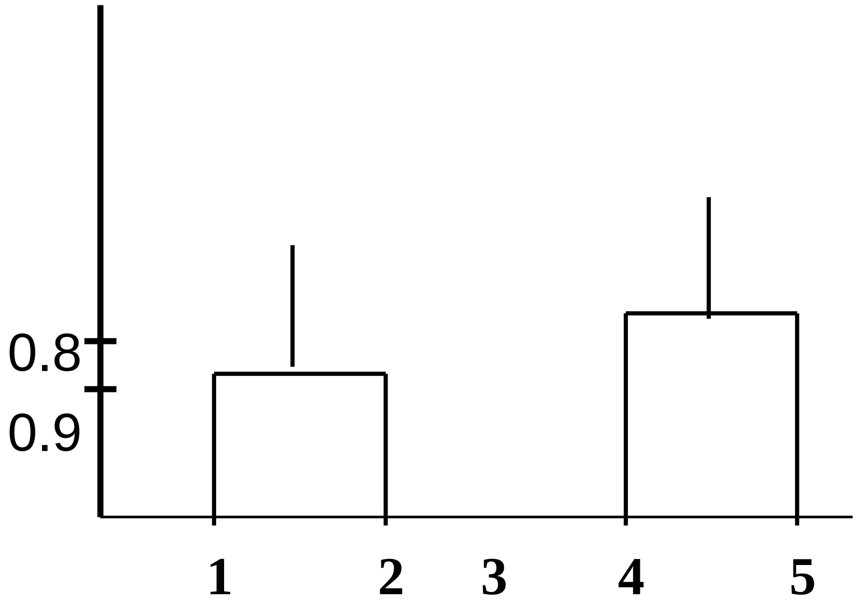
	I1	I2	I3	I4	I5
I1	1.00	1.00	0.70	0.65	0.50
I2	1.00	1.00	0.70	0.65	0.50
I3	0.70	0.70	1.00	0.40	0.30
I4	0.65	0.65	0.40	1.00	0.80
I5	0.50	0.50	0.30	0.80	1.00

MIN Dendrogram Computation

- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph.

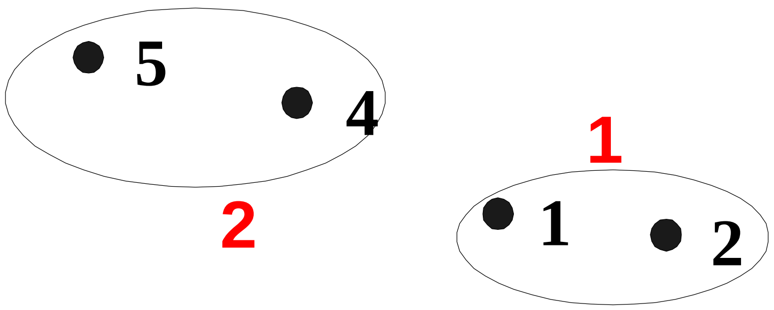


	I1	I2	I3	I4	I5
I1	1.00	1.00	0.70	0.65	0.50
I2	1.00	1.00	0.70	0.65	0.50
I3	0.70	0.70	1.00	0.40	0.30
I4	0.65	0.65	0.40	1.00	0.80
I5	0.50	0.50	0.30	0.80	1.00

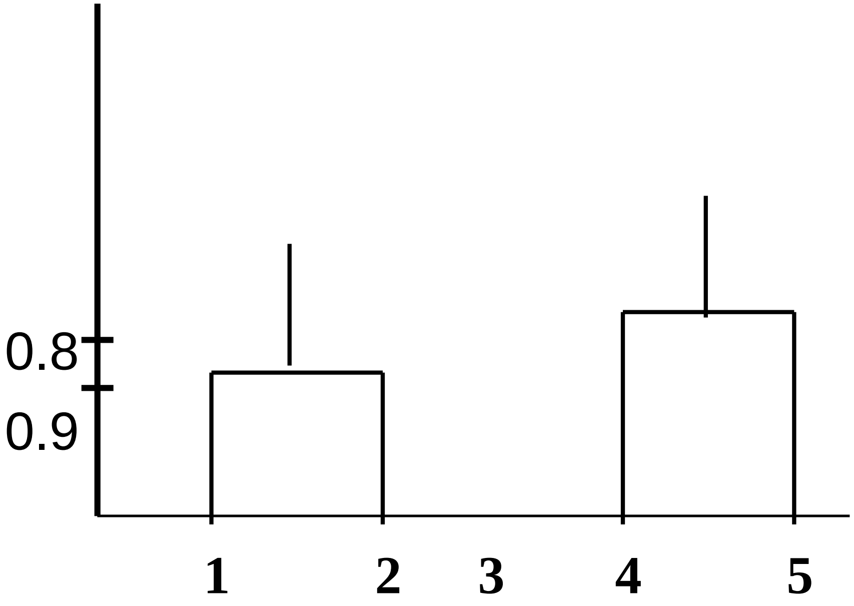


MIN Dendrogram Computation

- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph.

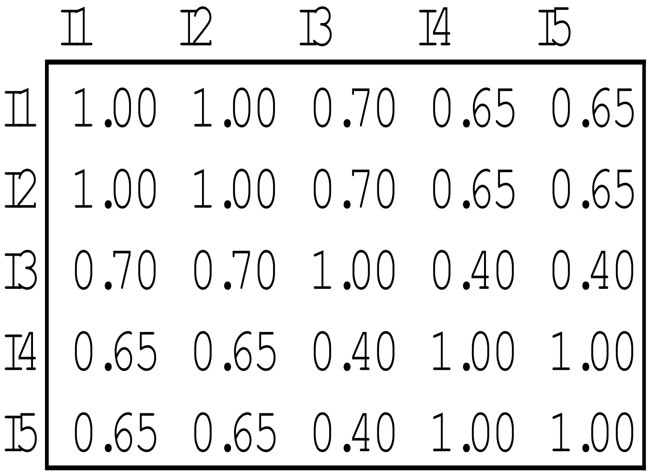


	I1	I2	I3	I4	I5
I1	1.00	1.00	0.70	0.65	0.65
I2	1.00	1.00	0.70	0.65	0.65
I3	0.70	0.70	1.00	0.40	0.40
I4	0.65	0.65	0.40	1.00	1.00
I5	0.65	0.65	0.40	1.00	1.00

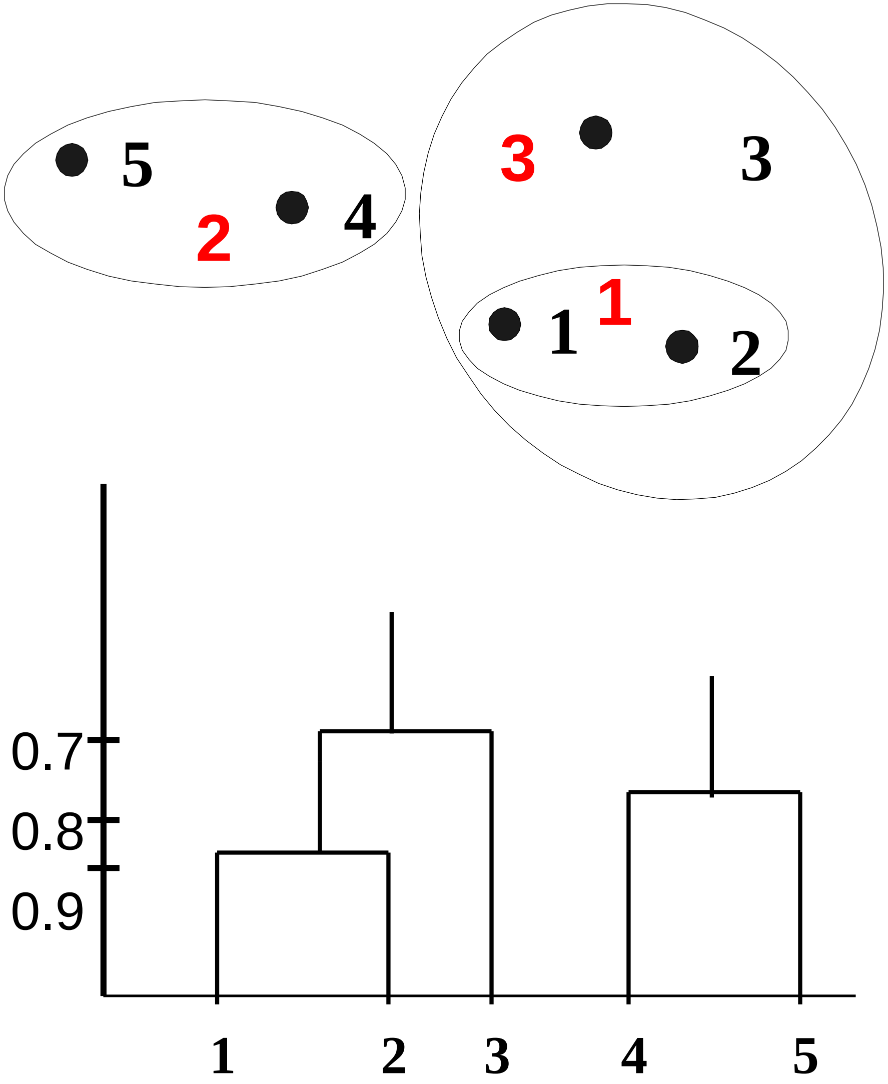


MIN Dendrogram Computation

- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph.

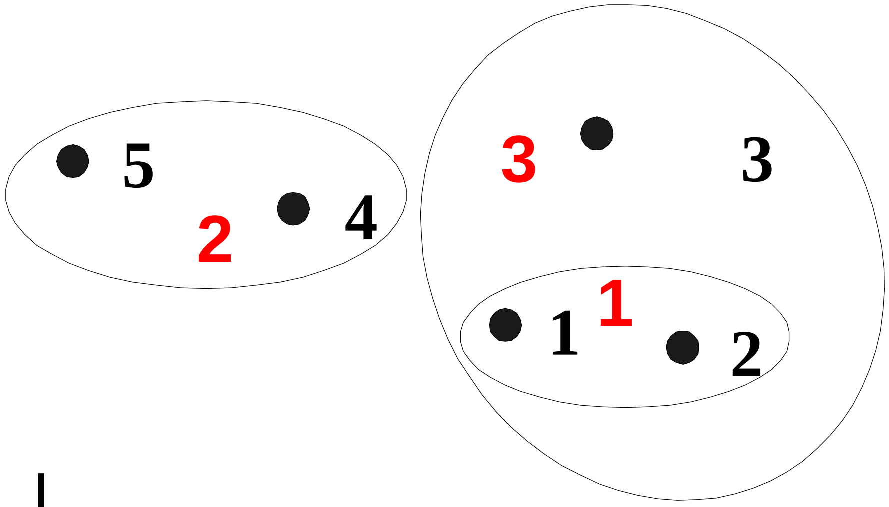


	I1	I2	I3	I4	I5
I1	1.00	1.00	0.70	0.65	0.65
I2	1.00	1.00	0.70	0.65	0.65
I3	0.70	0.70	1.00	0.40	0.40
I4	0.65	0.65	0.40	1.00	1.00
I5	0.65	0.65	0.40	1.00	1.00

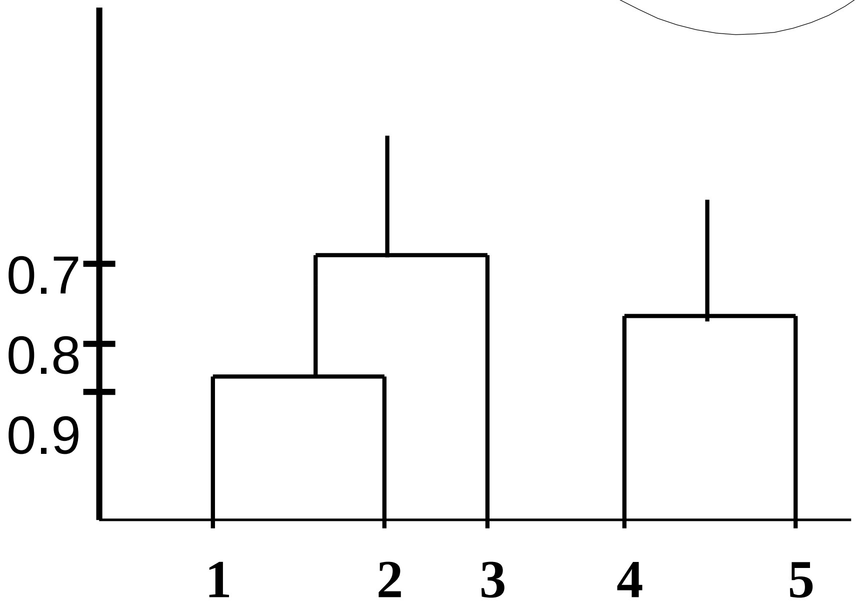


MIN Dendrogram Computation

- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph.



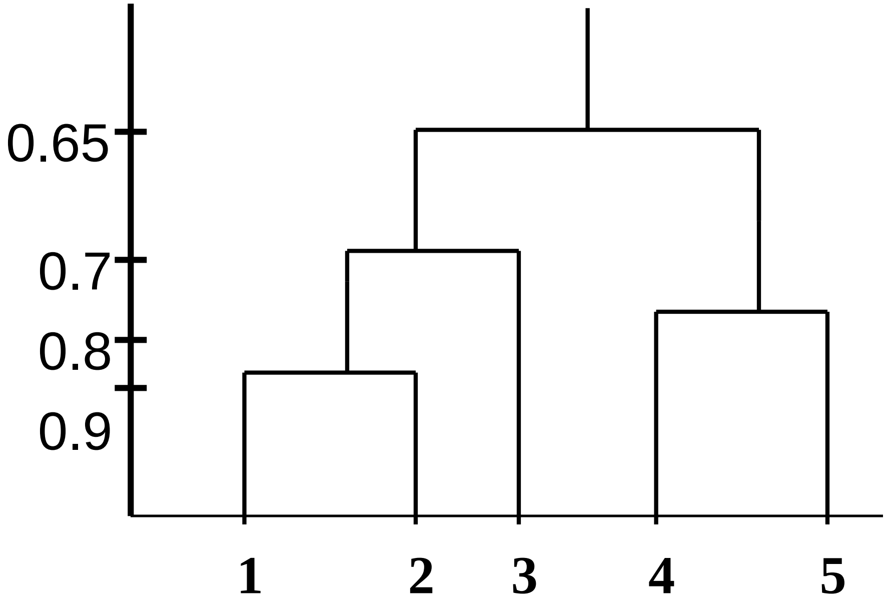
I1	I2	I3	I4	I5
1.00	1.00	1.00	0.65	0.65
1.00	1.00	1.00	0.65	0.65
1.00	1.00	1.00	0.65	0.65
0.65	0.65	0.65	1.00	1.00
0.65	0.65	0.65	1.00	1.00



MIN Dendrogram Computation

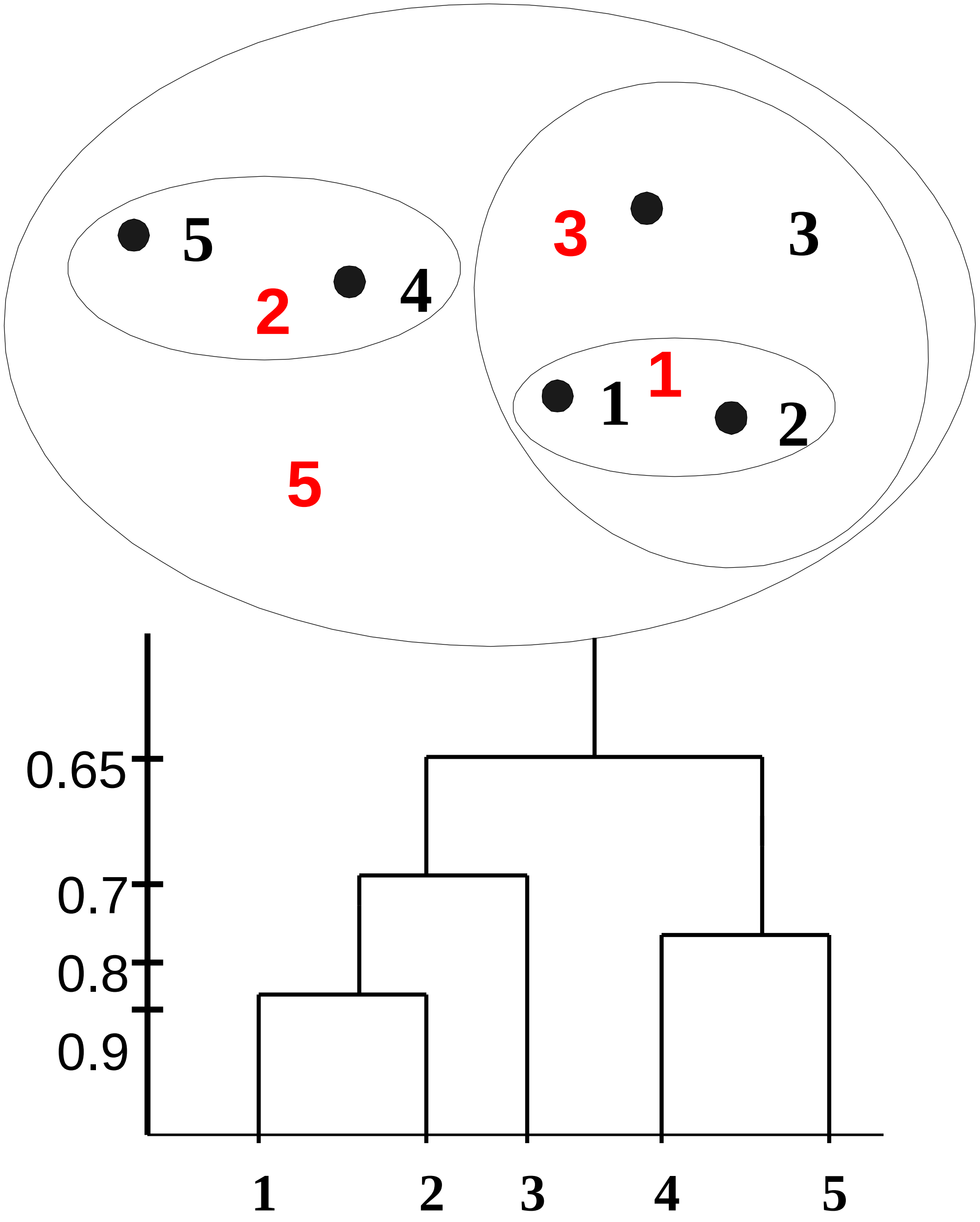
- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph.

I1	I2	I3	I4	I5
1.00	1.00	1.00	0.65	0.65
1.00	1.00	1.00	0.65	0.65
1.00	1.00	1.00	0.65	0.65
0.65	0.65	0.65	1.00	1.00
0.65	0.65	0.65	1.00	1.00



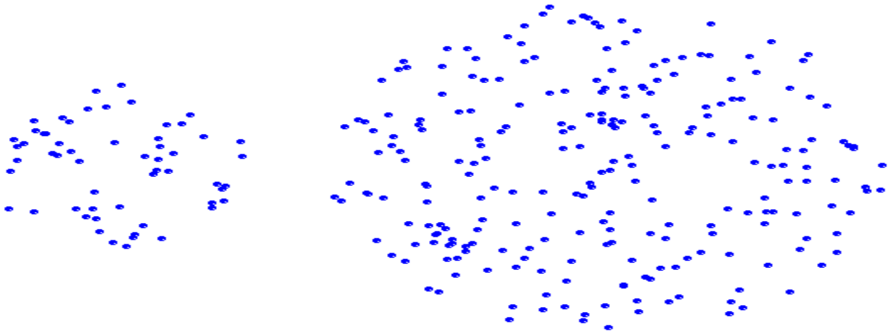
MIN Dendrogram Computation

- Similarity of two clusters is based on the two most similar (closest) points in the different clusters
  - Determined by one pair of points, i.e., by one link in the proximity graph.

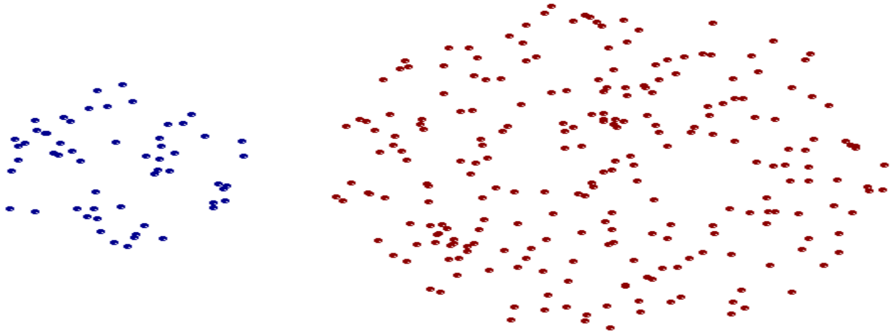


	I1	I2	I3	I4	I5
I1	1.00	1.00	1.00	1.00	1.00
I2	1.00	1.00	1.00	1.00	1.00
I3	1.00	1.00	1.00	1.00	1.00
I4	1.00	1.00	1.00	1.00	1.00
I5	1.00	1.00	1.00	1.00	1.00

Strength of MIN



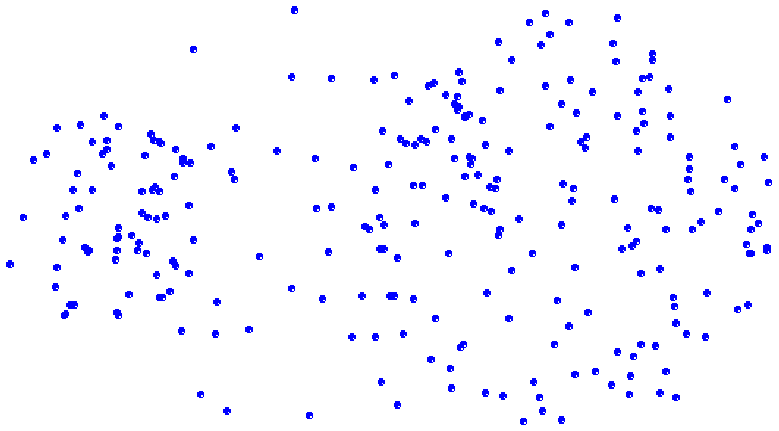
Original Points



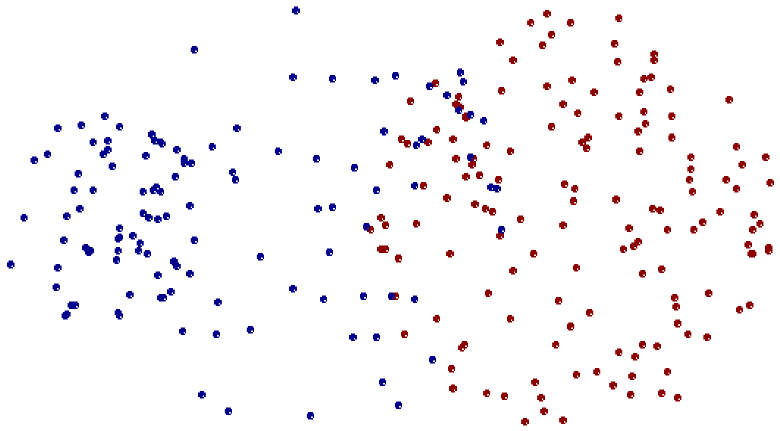
Two Clusters

- Can handle non-elliptical shapes

Limitations of MIN



Original Points



Two Clusters

- Sensitive to noise and outliers

Reading

- TSKK sec’s. 7.2.2, 7.31, 7.3.2